



CACIC 2019

Congreso Argentino de Ciencias de la Computación

ACTAS



XXV Congreso Argentino de Ciencias de la Computación CACIC 2019

Patricia Pesado y Marcelo Arroyo

Compiladores



14 al 18 de octubre de 2019

Facultad de Ciencias Exactas, Físico-Químicas y Naturales

Universidad Nacional de Río Cuarto

ISBN 978-987-688-377-1

e-book

UniRío
editora

XXV Congreso Argentino de Ciencias de la Computación : CACIC 2019 : libro de actas /
Pesado Patricia ... [et al.] ; coordinación general de Pesado Patricia ; Marcelo
Arroyo. - 1a ed. - Río Cuarto : UniRío Editora, 2020.
Libro digital, PDF - (Actas de congresos)

Archivo Digital: descarga y online
ISBN 978-987-688-377-1

1. Computación. 2. Ingeniería Informática. 3. Actas de Congresos. I. Patricia, Pesado II.
Patricia, Pesado, coord. III. Arroyo, Marcelo , coord.
CDD 004

XXV Congreso Argentino de Ciencias de la Computación. CACIC 2019. Libro de actas
Patricia Pesado y Marcelo Arroyo (Coordinadores)

2020 © by *UniRío editora*. Universidad Nacional de Río Cuarto
Ruta Nacional 36 km 601 – (X5804) Río Cuarto – Argentina
Tel: 54 (358) 467 6309 . editorial@rec.unrc.edu.ar . www.unirioeditora.com.ar

Primera Edición: *marzo de 2020* / ISBN 978-987-688-377-1



Este obra está bajo una Licencia Creative Commons Atribución 2.5 Argentina.

http://creativecommons.org/licenses/by/2.5/ar/deed.es_AR



Uni. Tres primeras letras de “Universidad”. Uso popular muy nuestro; la Uni. Universidad del latín “universitas” (personas dedicadas al ocio del saber), se contextualiza para nosotros en nuestro anclaje territorial y en la concepción de conocimientos y saberes construidos y compartidos socialmente. El río. Celeste y Naranja. El agua y la arena de nuestro Río Cuarto en constante confluencia y devenir. La gota. El acento y el impacto visual: agua en un movimiento de vuelo libre de un “nosotros”.
Conocimiento que circula y calma la sed.

Consejo Editorial

Facultad de Agronomía y Veterinaria
Prof. Mercedes Ibañez y Prof. Alicia Carranza
Facultad de Ciencias Económicas
Prof. Ana Vianco
Facultad de Ciencias Exactas, Físico-Químicas
y Naturales
Prof. Sandra Miskoski

Facultad de Ciencias Humanas
Prof. Gabriel Carini
Facultad de Ingeniería
Prof. Marcelo Alcoba
Biblioteca Central Juan Filloy
Bibl. Claudia Rodríguez y Prof. Mónica Torreta
Secretaría Académica
Prof. Ana Vogliotti y Prof. José Di Marco

Equipo Editorial

Secretaria Académica: *Ana Vogliotti*
Director: *José Di Marco*
Equipo: *José Luis Ammann, Maximiliano Brito, Ana Carolina Savino,
Lara Oviedo, Roberto Guardia, Marcela Rapetti y Daniel Ferniot*



CACIC 2019
Congreso Argentino de Ciencias de la Computación

XXV Congreso Argentino de Ciencias de la Computación (CACIC 2019)

Río Cuarto (Córdoba), 14 al 18 de Octubre de 2019

Organizadores

Red de Universidades Nacionales con carreras en Informática (RedUNCI)
Dpto. de Computación – FCEFQyN – Universidad Nacional de Río Cuarto



CACIC 2019

Congreso Argentino de Ciencias de la Computación

Comité Académico

- | | |
|---|--|
| UBA – Cs. Exactas - Garbervetsky, Diego | UN Rio Negro - Vivas, Luis |
| UBA – Ingeniería - Echeverria, Adriana | UN Villa María - Prato, Laura |
| UN La Plata - Pesado, Patricia | UN Lujan - Panessi, Wálter |
| UN Sur - Rueda, Sonia | UN Catamarca – Poliche, Maria Valeria |
| UN San Luis - Piccoli, Fabiana | UN La Rioja – Campazzo, Eduardo |
| UNCPBA - Aciti, Claudio | UN Tres de Febrero - Oliveros, Alejandro |
| UN Comahue - Vaucheret, Claudio | UN Tucumán - Luccioni, Griselda María |
| UN La Matanza - Sposito, Osvaldo | UNAJ - Morales, Martín |
| UN La Pampa - Alfonso, Hugo | UN Chaco Austral - Zachman Patricia |
| UN Lomas de Zamora - Estayno, Marcelo | UN del Oeste - Foti, Antonio |
| UN Tierra del Fuego - Feierherd, Guillermo | UN de Cuyo - García Garino, Carlos |
| UN Salta - Gil, Gustavo | UN de Mar del Plata - Doumecq, Julio Cesar |
| UN Patagonia Austral - Lasso, Marta | U Morón - Padovani Hugo |
| UN San Juan - Rodríguez, Nelson | UAI - De Vincenzi, Marcelo |
| UADER - Vivas, Patricia | U Belgrano - Guerci, Alberto |
| UN Patagonia SJB - Buckle, Carlos | U Kennedy - Panizzi, Marisa |
| UN Entre Ríos - Tugnarelli, Mónica | U Adventista del Plata - Bournissen Juan |
| UN Nordeste - Dapozo, Gladys | UCAECE - Finocchietto, Jorge |
| UN Rosario - Kantor Raul | U Palermo - Álvarez Adriana |
| UN Misiones - Kuna, Horacio | UCA Rosario - Grieco, Sebastián |
| UNNOBA - Russo, Claudia | U Salvador - Zanitti, Marcelo |
| UN Chilecito - Carmona, Fernanda | U Aconcagua - Giménez, Rosa |
| UN Lanús - Azcurra, Diego | U Gastón Dachary - Beyersdorf, Carlos |
| UN Santiago del Estero - Duran, Elena | UCEMA - Guglianone, Ariadna |
| Esc. Sup. Ejército - Arroyo Arzubi, Alejandro | U Austral - Cosentino, Juan Pablo |
| UN Litoral - Loyarte, Horacio | U Atlántida Argentina - Rathmann, Liliana |
| UN Rio IV - Arroyo, Marcelo | UCA La Plata - Bertone, Rodolfo |
| UN Córdoba - Fridlender, Daniel | ITBA - Mon, Alicia |
| UN Jujuy - Herrera Cognetta, Analía | U Champagnat - Pinciroli, Fernando |



CACIC 2019

Congreso Argentino de Ciencias de la Computación

Comité Científico

Abásolo, María José (Argentina)	Estayno, Marcelo (Argentina)	Navarro Martín, Antonio (España)
Aciti, Claudio (Argentina)	Estevez, Elsa (Argentina)	Pardo, Álvaro (Uruguay)
Alfonso, Hugo (Argentina)	Falappa, Marcelo (Argentina)	Pasini, Ariel (Argentina)
Ardenghi, Jorge (Argentina)	Fillottrani, Pablo (Argentina)	Pesado, Patricia (Argentina)
Arroyo, Marcelo (Argentina)	Frati, Emmanuel (Argentina)	Piattini, Mario (España)
Astudillo, Hernán (Chile)	Fridlender, Daniel (Argentina)	Piccoli, María Fabiana (Argentina)
Baldasari, Sandra (España)	García Garino, Carlos (Argentina)	Printista, Marcela (Argentina)
Balladini, Javier (Argentina)	García Villalba, Javier (España)	Puppo, Enrico (Italia)
Barbosa, Luis (Portugal)	Género, Marcela (España)	Olivas Varela, José Ángel (España)
Bertone, Rodolfo (Argentina)	Gomez, Sergio (Argentina)	Ramón, Hugo (Argentina)
Bría, Oscar (Argentina)	Gorga, Gladys (Argentina)	Rexachs, Dolores (España)
Brisaboa, Nieves (España)	Gröller, Eduard (Austria)	Reyes, Nora (Argentina)
Buckle, Carlos (Argentina)	Guerrero, Roberto (Argentina)	Roig Vila, Rosabel (España)
Cañas, Alberto (EE.UU)	Ierache, Jorge (Argentina)	Rossi, Gustavo (Argentina)
Chavez, Edgar (México)	Janowski, Tomasz (Naciones Unidas)	Rosso, Paolo (España)
Coello Coello, Carlos (México)	Kuna, Horacio (Argentina)	Rueda, Sonia (Argentina)
Cuevas, Alfredo Simón (Cuba)	Lanzarini, Laura (Argentina)	Russo, Claudia (Argentina)
Dix, Juerguen (Alemania)	Leguizamón, Guillermo (Argentina)	Salto, Carolina (Argentina)
Doallo, Ramón (España)	López Gil, Fernando (España)	Sanz, Cecilia (Argentina)
Docampo, Domingo (España)	Loui, Ronald Prescott (EEUU)	Simari, Guillermo (Argentina)
Casali, Ana (Argentina)	Luque, Emilio (España)	Steinmetz, Ralf (Alemania)
Castro, Silvia (Argentina)	Madoz, Cristina (Argentina)	Suppi, Remo (España)
Cechich, Alejandra (Argentina)	Malberti, Alejandra (Argentina)	Tarouco, Liane (Brasil)
Cuckierman, Uriel (Argentina)	Malbrán, María (Argentina)	Tirado, Francisco (España)
De Giusti, Armando (Argentina)	Marín, Mauricio (Chile)	Thomas, Pablo (Argentina)
De Giusti, Laura (Argentina)	Mas Sansó, Ramón (España)	Vénere, Marcelo (Argentina)
De Vincenzi, Marcelo (Argentina)	Micolini, Orlando (Argentina)	Velho, Luiz (Brasil)
Deco, Claudia (Argentina)	Mon, Alicia (Argentina)	Vendrell, Eduardo (España)
Depetris, Beatriz (Argentina)	Motz, Regina (Uruguay)	Villagarcía, Horacio (Argentina)
Diaz, Javier (Argentina)	Naiouf, Marcelo (Argentina)	Zanarini, Dante (Argentina)
Dujmovic, Jozo (USA)		



CACIC 2019

Congreso Argentino de Ciencias de la Computación

Comité Organizador

Presidente: Arroyo, Marcelo

Aguirre, Nazareno

Regis, Germán

Daniele, Marcela

Permigiani, Sonia

Castro, Pablo

Bavera, Francisco

Novaira, María Marta

Ponzio, Pablo

Bengolea, Valeria

Degiovanni, Renzo

Arsaute, Ariel

González, Ariel

Uva, Marcelo

Cornejo, César

Politano, Mariano

Kilmurray, Cecilia

Brusatti, Franco

Scilingo, Gastón

Molina, Facundo

Cerdá, Ernesto

Zorzán, Fabio



CACIC 2019
Congreso Argentino de Ciencias de la Computación

Autoridades de la RedUNCI

Coordinadora Titular

Pesado, Patricia (UNLP)

Coordinador Alterno

Estayno, Marcelo (UNLZ)

Junta Directiva

Printista, Marcela (UNSL)

Carmona, Fernanda (UNdeC)

Tugnarelli, Mónica (UNER)

Padovani, Hugo (UM)

Eterovic, Jorge (UNLaM)

Aciti, Claudio (UNCPBA)

Arroyo, Marcelo (UNRC)

Panizzi, Marisa (UK)

Miembro Honorario

De Giusti, Armando (UNLP)

Secretarías

Secretaría Administrativa: Kuna, Horacio (UNaM)

Secretaría Académica: Lasso, Marta (UNPA)

Secretaría de Ciencia y Técnica: Balladini, Javier (UNCOMA)

Secretaría de Asuntos Reglamentarios: De Vincenzi, Marcelo (UAI)

Secretaría de Vinculación Tecnológica y Profesional: Gil, Gustavo (UNSa)

Secretaría de Congresos, Publicaciones y Difusión: Thomas, Pablo (UNLP)



CACIC 2019
Congreso Argentino de Ciencias de la Computación

Índice

XX Workshop Agentes y Sistemas Inteligentes	1
A Simple Differential Evolution Algorithm to Solve the Flexible Job Scheduling Problem	2
Algoritmos de Regresión Lineal aplicados al mantenimiento de un Datacenter	12
An Artificial Immune System for Solving Dynamic Economic Power Dispatch Problems	22
Aplicación de Regresión con Vectores de Soporte en un Sistema Recomendador de Actividades Sociales	32
Automatically Assessing the Need of Additional Citations for Information Quality Verification in Wikipedia Articles	42
DE with Random Vector based Mutatiton for High Dimensional Problems	52
Deep Learning Architecture for Forest Detection in Satellite Data	64
HeartCare: an Agent Oriented Architecture Implemented with Actors	74
Multicore Parallelization of CHC for Optimal Aerogenerator Placement in Wind Farms	85
Optimal Instrumentation: Adjustment and Hybridization of a Simulated Annealing Based Technique	95
Recognizing Handshapes using Small Datasets	105
Sintonización Dinámica del Método Paralelo de Predicción de Incendios Forestales ESSIM-DE	115
XX Workshop Procesamiento Distribuido y Paralelo	125
A Study of Hardware Performance Counters Selection for Cross Architectural GPU Power Modeling	126



Fire Weather Index assessment and visualization	140
Implementación y análisis de rendimiento de un algoritmo paralelo de multiplicación de matrices en 2,5D	150
Modelado de potencia en placas SBC: integración de diferentes generaciones Raspberry Pi	159
SEDAR: Detectando y Recuperando Fallos Transitorios en Aplicaciones de HPC	170
Simulación de altas prestaciones (GPU) para la reproducción del mosquito <i>Aedes aegypti</i> en el cementerio de Santo Tomé, Corrientes	183
Simulación de N Cuerpos Computacionales sobre Intel Xeon Phi KNL	194
XVIII Workshop Tecnología Informática Aplicada en Educación	205
Análisis de la experiencia de utilización del juego serio Desafiate para la autoevaluación de los alumnos	206
AppCase4Blind: una herramienta para asistir a personas no videntes en el proceso de enseñanza-aprendizaje de Grafos	218
Arquitectura basada en microservicios en un sistema de corrección automatizada de exámenes	228
Calidad en uso en plataformas educativas: estudio centrado en la Accesibilidad Web	237
Definición de un estándar para la evaluación de calidad de software educativo	246
Diseño de plantillas para la creación de actividades educativas con Realidad Aumentada en AuthorAR	256
Diseño Instruccional para fortalecer las competencias digitales en ingresantes: el caso de la FaCENA-UNNE	267
Empoderando Personas a Través del Uso de Internet - VIDUC	278



Estrategias didácticas para la enseñanza de la programación imperativa, utilizando el Robot Educativo Programable	288
Herramientas de Autor para Enriquecer Materiales de Lectura. Análisis Comparativo	299
La Realidad Virtual y su impacto en el aprendizaje	314
MoLE: A web authoring tool for building mobile learning experiences	325
Perfiles de tesistas de la disciplina informática y su vinculación con los proyectos de investigación de la UM	335
Procedimiento para instruir en la programación de dispositivos inteligentes	345
Propuesta de automatización de auto-percepción de capacidades visuales de estudiantes universitarios	355
Propuesta de automatización para proyectos de minería de datos educativa	366
Propuesta de Metodología de Gestión de Procesos para la Evaluación de la Calidad en los Entornos Virtuales de Enseñanza y Aprendizaje	376
Simulador en línea para capacitación de individuos hipoacúsicos adultos	386
Un modelo para la incorporación de computadoras livianas extensible en ambientes educativos	397
Una Solución Tecnológica para el uso de TIC en escuelas rurales aisladas	407
Valoración del grado de atención en contextos áulicos con el empleo de interfase cerebro-computadora en el marco de la computación afectiva.	417
XVII Workshop Computación Gráfica, Imágenes y Visualización	427
Plataforma para Repositorios Digitales 3D de Colecciones Biológicas	428
Streamlining the study of the Tierra del Fuego forest through the use of deep learning	438



Visualization to compare karate motion captures	446
XVI Workshop Bases de Datos y Minería de Datos	456
Análisis de Técnicas de Raspado de Datos en la Web - Aplicado al Portal del Estado Nacional Argentino	457
Consultas Métrico Espaciales	467
Data science methodologies selection with hierarchical analytical process and personal construction theory	477
Ensamblado <i>ad hoc</i> de clasificadores para la detección de cáncer de mama usando <i>Scikit-learn</i>	487
Evaluación de la precisión de acierto de un conjunto desbalanceado mediante la combinación de clasificadores	497
Exploratory Analysis of a New Corpus for Political Alignment Identification of Argentinian Journalists	507
Extensión de Proceso de Explotación de Información para la Detección de Asociaciones Espaciales Difusas	517
Genericidad de funciones: El <i>quid</i> para la incorporación de dominios en un Sistema Funcional Inductivo basado en Haskell	527
Grafos y Temporalidad en Bases de Datos. Una aplicación sobre Datos Abiertos Enlazados	537
<i>k</i> -TVT: a flexible and effective method for early depression detection	547
Modelo NoSQL de Propuestas Académicas de Grado	557
Solving All- <i>k</i> -Nearest Neighbor Problem without an Index	567
Towards Measuring the Severity of Depression in Social Media via Text Classification	577



Un estudio comparativo de Bases de Datos Relacionales y Bases de Datos NoSQL	589
Una Mejora en Tiempo del Trie de Sufijos	601
XVI Workshop Ingeniería de Software	611
Análisis del impacto en la performance de una web app utilizando un lenguaje compilado y un lenguaje interpretado	612
Aspect Oriented Behavioral Synthesis	622
Coreografía de dispositivos ubicuos basada en SOA: ¿Cuánto de limitados pueden ser los dispositivos?	632
Detección de objetos espurios en generación automática de entradas	642
Directrices para mejorar la usabilidad de los frameworks web	652
Estudio de la Capacidad de Mantenimiento de las Aplicaciones Móviles Híbridas	662
Evaluación de herramientas libres para la gestión de requerimientos	674
Evaluación de la calidad en patrones de arquitecturas software para sistemas críticos ferroviarios: un enfoque basado en la mantenibilidad	684
Guia para evaluar calidad de datos basada en ISO/IEC 25012	694
Hacia la construcción de drivers eficientes en bounded model checking mediante detección automática de builders	707
Hacia un Sistema I Polimórfico	717
Implementación Básica de Typestates en Rust	727
Inspecciones para Mejorar la Calidad de Modelos en Lenguaje Natural	737



Integración de funcionalidades BPM en Portales Web	747
Plantilla de parametrización de Consultas de Mashup	757
Propuesta de Framework para el Diseño de Interacción en las Redes Sociales de Video	767
PWA para unificar el desarrollo Desktop, Web y Mobile	778
Rapid Releases and Testing Problems at the industry: A survey	787
Reparación de programas aplicando templates: Generando workarounds permanentes a través de SAT	797
Reuso de Funcionalidades en el Subdominio de Patrimonio Cultural	807
Risk determination for the implantation process of software systems	817
Un Análisis preliminar sobre reparación de modelos <i>Alloy</i> utilizando <i>Sketching</i>	826
Una propuesta para clasificación de roles de usuarios en foros de discusión técnicos	836
XIV Workshop Arquitectura, Redes y Sistemas Operativos	846
Estudio experimental de tráfico de video en redes IPv6 multicast IEEE 802.11ac	847
Evaluación de Convergencia del Protocolo OSPF en Redes Definidas por Software	857
Experiencias de Análisis de Consumo Energético en Redes de Sensores	868
Filtering Useless Data at the Source	879
Gestión remota de dispositivos IOT mediante técnicas de mensajería instantánea empleando Bots	889
Implementación de Red de Internet de las Cosas Usando OpenMote y Raspberry Pi	899



Smartcities con LoRaWAN. El Caso de Monitoreo de Condiciones Ambientales de Lagunas en Tiempo Real 910

XI Workshop Innovación en Sistemas de Software 919

A Framework for OBDA: Current State and Perspectives 920

Accesibilidad de edificios por medio de modelado y simulación con aplicaciones 3D 930

An Automated Technique for Analysis of Orthogonal Variability Models based on Anti-patterns Detection using DL reasoning 940

Arquitectura de segmento terreno satelital adaptada para el control de límites de telemetría dinámicos 950

Baboon, Framework Conducido por Red de Petri para Sistemas Reactivos Dirigidos por Eventos 960

El modelo de Programación de Actor aplicado a Edge Computing utilizando Calvin 970

Elicitación de Requerimientos para la Construcción de Modelos Predictivos basados en Sistemas Inteligentes dentro del Ámbito Educativo 980

Evaluación de estrategias públicas de salud digital para el desarrollo de una solución de interoperabilidad local entre Sistemas de Información en Salud 990

Hacia un marco de trabajo para desarrollar sistemas de programación de la producción utilizando servicios en la nube 1001

HMSBeagleVR: Una experiencia inmersiva a bordo del HMS Beagle 1011

Modelo de evaluación de Tecnologías de la Información y la Comunicación para la industria 4.0 1020

Receptor Electrónico de Residuos (RER). Una infraestructura conceptual de recompensas directa de hábitos ciudadanos. 1030



Sistema integrador de dispositivos de la salud con conexión BLE para envío de alarmas	1041
SMARTWATER para la detección de fugas de agua	1051
X Workshop Procesamiento de Señales y Sistemas de Tiempo Real	1061
Cloud/Edge Robotics: Navegación autónoma de Auto-Robot y Cuadrilatero	1062
Development and implementation of a smart greenhouse	1072
Sistema de localización para robots móviles de bajo costo utilizando marcas de referencia artificiales en ambientes de interiores	1081
VIII Workshop Innovación en Educación en Informática	1091
Análisis de una experiencia de la evaluación de la adquisición de la competencia "trabajo en equipo"	1092
Enseñando Python en una propuesta de formación docente en enseñanza de la programación	1100
Experiencias en el uso de TIC y rampas digitales en la enseñanza de informática a alumnos de educación terciaria con discapacidad visual o auditiva	1110
Formación docente en Ciencias de la Computación: experiencias de la primera Especialización para el nivel Primario de la Provincia de Santa Fe	1121
Habilidades de Pensamiento Computacional en docentes de primaria: evaluación usando Bebras	1131
Introducing Agile Methods in Undergraduate Curricula, a Systematic Mapping Study	1139
ProgEnv: Un entorno para la enseñanza de la programación	1149



Revisión Sistemática: Elicitación de requerimientos educativos en serious games	1158
Scrum como Herramienta Metodológica para el Aprendizaje de la Programación	1168
Transición desde programación basada en bloques a basada en texto: una revisión del campo	1180
Una propuesta metodológica para evolucionar hacia un diseño curricular basado en competencias	1190
VIII Workshop seguridad informática	1200
Análisis del anonimato aplicado a criptomonedas	1201
Construcción de un Generador Aleatorio Binario mediante Combinación No Lineal de LFSR	1218
Criptografía Ligera en Internet de las Cosas para la Industria	1228
Detección de canales encubiertos en la Capa de Red	1241
Ensembling to improve infected hosts detection	1251
Forensic Readiness: Guía de buenas prácticas	1261
Marco para la Aplicación de Criptoanálisis	1269
Mejora de la Integridad de Datos en el Modelo OTP-Vote	1278
Modelo de Seguridad IoT	1288
Procedimientos de CiberSeguridad en un Laboratorio de Educación, Desarrollo e Investigación (EDI) para la identificación de vulnerabilidades en su red informática	1297
	1304



II Track de Gobierno Digital y Ciudades Inteligentes

Blockchain y gobierno digital	1305
Desarrollo de Ciudades Inteligentes Sostenibles - Un Caso de Estudio Implementado por el Municipio de Vicente López	1316
Evaluando la Calidad de Uso en aplicaciones web de gobierno electrónico y ciencia ciudadana con estándares ISO/IEC	1325
Framework IoT aplicado a Monitoreo de Precipitaciones de la Provincia del Chaco	1335
Marco de trabajo para la aplicación de línea de producto en el desarrollo de software para gobierno digital	1344
Minería de Emociones y Análisis Visual Aplicado a la Red Social Twitter	1354
Servicio para la gestión inteligente del consumo de energía eléctrica hogareña. Una aplicación móvil para su monitoreo	1364
Taxonomía para aplicaciones móviles de servicios de gobierno en municipios de Argentina	1374



CACIC 2019

Congreso Argentino de Ciencias de la Computación

XX Workshop Agentes y Sistemas Inteligentes

Coordinadores

Leguizamón, Guillermo (UNSL)
Salto, Carolina (UNLPam)
Fridlender, Daniel (UNC)

Comité Científico

Aguirre, Guillermo (Argentina)
Alfonso, Hugo (Argentina)
Alonso-Aleman, Laura (Argentina)
Apolloni, Javier M. (Argentina)
Aragón, Victoria (Argentina)
Bavera, Francisco (Argentina)
Carballido, Jessica (Argentina)
Casali, Ana (Argentina)
Cassano, Valentín (Canadá)
Cecchi, Laura (Argentina)

Cobo, María Laura (Argentina)
Edera, Alejandro (Argentina)
Falappa, Marcelo A. (Argentina)
García, Alejandro (Argentina)
Godoy, Daniela (Argentina)
Gottifredi, Sebastián (Argentina)
Lanzarini, Laura (Argentina)
Maguitman, Ana (Argentina)
Martínez, Diego (Argentina)
Martínez, María Vanina (Argentina)

Mezura-Montes, Efrén (México)
Minetti, Gabriela (Argentina)
Olivera, Ana Carolina (Argentina)
Salto, Carolina (Argentina)
Schiaffino, Silvia (Argentina)
Simari, Gerardo Ignacio (Argentina)
Simari, Guillermo R. (Argentina)
Tamargo, Luciano (Argentina)
Villagra, Andrea (Argentina)
Zunino, Alejandro (Argentina)

Temas

Approximate Reasoning
Bio-Inspired Metaheuristics
Constraints, Satisfiability and Search
Intelligent Data Mining
Intelligent Robotics
Intelligent System Performance Measurements
Intelligent Systems
Knowledge Representation
Machine Learning
Metaheuristics Based on Collective Intelligence
Multi-Agent Systems
Natural Language Processing
Planning and Scheduling
Reasoning and Logic
Semantic Web

A Simple Differential Evolution Algorithm to Solve the Flexible Job Shop Scheduling Problem

Franco Morero, Carlos Bermudez, and Carolina Salto

Facultad de Ingeniería, UNLPam
General Pico, La Pampa, Argentina, CONICET

Abstract. This paper addresses the Flexible Job Shop Scheduling Problem (FJSSP) where the objective is to minimize the makespan. We develop a parallel hybrid Differential Evolution (DE) algorithm to tackle this problem. A random key representation of the FJSSP is adopted, which requires a very simple conversion mechanism to obtain a feasible schedule. This allows the DE algorithm to work on the continuous domain to explore the problem space of the discrete FJSSP. Moreover, a simple local search algorithm is embedded in the DE framework to balance the exploration and exploitation by enhancing the local searching ability. In addition, parallelism of the DE operations is included to improve the efficiency of whole algorithm. Experiments confirm the significant improvement achieved by integrating the propositions introduced in this study. Additional, test results show that our algorithm is competitive when compared with most existing approaches for the FJSSP.

Keywords: flexible job shop scheduling, differential evolution algorithm, parallelism

1 Introduction

A realistic production environment, and with practical applicability, is the Flexible Job Shop Scheduling Problem. Each job has to undergo multiple operations on the various machines. The decision concerns how to sequence the operations on the machines, so that the time needed to complete all the jobs (C_{max}) is minimized. Moreover, an additional decision consists in to assign each operation to the appropriate set of machines. These decisions suggest that the FJSSP is a complex optimization problem (NP-hard problem [1]), consequently, the adoption of metaheuristic [2, 3] has led to better results than classical dispatching or greedy heuristic algorithms [4–6]. Since introduced in 1997 by Storn and Price [7], the Differential Evolution (DE) metaheuristic became very popular among computer scientists and practitioners almost immediately after its original definition.

DE is a stochastic real-parameter global optimizer. It employs simple mutation and crossover operators to generate new candidate solutions, and applies one-to-one competition scheme to greedily determine whether the new candidate or its parent will survive in the next generation. Their successful is due to its simplicity and ease implementation, and reliability and high performance. DE algorithms have been applied to many combinatorial optimization problems

([8–11], among many others), but as far as we are aware, there is few published research work that describes the use of DE to deal with the FJSSP [12].

In this work, a simple DE to solve the FJSSP is design. As DE was originally devised for solving continuous optimization problems, we adopt a real value representation for the FJSSP to make the continuous DE applicable for solving the discrete FJSSP, which implies that algorithm operations should not be modified or adapted to resolve the problem. Another important feature of DE is the little number of parameters to be set, when compared to other evolutionary algorithms. However, the success to find good solutions to a problem depends on discovering the correct values of those parameters [13]. Therefore, we make an analysis in this line to determine the adequate values for these parameters for solving the instances of the FJSSP. Moreover, a simple local search procedure is embedded to the DE to improve their exploration capacities by solving the problem. Finally, parallelism at algorithmic level [2] is incorporated to the design of the DE, with the aim of improving the scalability and reducing the computation time. The experimental methodology we have followed consists of computing the C_{max} values for the different DE proposed to solve the FJSSP and then comparing the obtained results by considering different quality indicators. We find that our simple DE shows a promising behavior to solve the FJSSP.

The paper is organized as follows. In Section 2, we introduce the problem formulation . In Section 3, we present the basic DE algorithm. In Section 4 we explain the adaptations of the DE to solve the FJSSP. In the following section, we introduce the experimental design and in Section 6, we evaluate the results. Some final remarks and future research directions are given in Section 7.

2 The Flexible Job Shop Scheduling Problem

The FJSSP can be formally described as follows. A set $J = \{J_1, J_2, \dots, J_n\}$ of independent jobs and a set $U = \{M_1, M_2, \dots, M_m\}$ of machines are given. A job J_i is broken down by a sequence of $O_{i1}, O_{i2}, \dots, O_{in_i}$ operations to be performed one after the other according to the given sequence. Each operation O_{ij} can be executed on any among a subset $U_{ij} \subseteq U$ of compatible machines. We have partial flexibility if there exists a proper subset $U_{ij} \subset U$, for at least one operation O_{ij} , while we have $U_{ij} = U$ for each operation O_{ij} in the case of total flexibility. The processing time of each operation is machine-dependent. We denote with d_{ijk} the processing time of operation O_{ij} when executed on machine M_k . Pre-emption is not allowed, i.e., each operation must be completed without interruption once started. Furthermore, the machines cannot perform more than one operation at a time. All jobs and machines are available at time 0.

The problem is to assign each operation to an appropriate machine (routing problem), and to sequence the operations on the machines (sequencing problem) in order to minimize the makespan (C_{max}). This measure is the time needed to complete all the jobs, which is defined as $C_{max} = \max_i\{C_i\}$, where C_i is the completion time of job J_i . Table 1 shows an instance of the FJSSP with 3 jobs, 4 machines and 8 operations. The rows and columns correspond to machines and operations, respectively, and the entries of the table are the processing times.

Table 1. Instance Example for the FJSSP

	J_1			J_2			J_3	
	O_{11}	O_{12}	O_{13}	O_{21}	O_{22}	O_{23}	O_{31}	O_{32}
M_1	-	4	9	2	4	9	8	3
M_2	6	8	5	-	6	-	6	5
M_3	5	5	-	1	8	2	-	8
M_4	-	6	7	3	4	2	5	3

Algorithm 1 Differential Evolution Algorithm (DE)

Require: F, Cr, N_p
Ensure: x_{best}
1: initialize(P^0, N_p)
2: $g \leftarrow 0$
3: **while** not meet stop criterion **do**
4: **for** each vector x_i^g from P^g **do**
5: $v_i^g \leftarrow \text{mutate}(x_i^g, P^g, F)$
6: $u_i^g \leftarrow \text{recombine}(x_i^g, v_i^g, Cr)$
7: $x_i^g \leftarrow \text{select}(x_i^g, u_i^g)$
8: add(P^{g+1}, x_i^{g+1})
9: **end for**
10: $g \leftarrow g + 1$
11: **end while**
12: $x_{best} \leftarrow \text{best_solution}(P^g)$

3 Differential Evolution Algorithm: Background

The DE algorithm was proposed by Storn and Price [7] to solve optimization problems with real-valued parameters. DE is a stochastic, population-based optimization method. Despite having a very simple algorithmic structure, DE has demonstrated a high level of performance when solving a wide variety of very complex problems [14]. The optimal or near-optimal solution is obtained by an iterative process which is applied to a set of solutions (population) to achieve a new one. At each step of the process, new solutions arise as a result of perturbations to the current solutions, caused by mutation and recombination operators.

The algorithmic framework of DE is described in Algorithm 1. The first step (Line 1) consists in the initialization of the population P^0 of N_P target vectors of D real values ($x_i = (x_{i,1}, x_{i,2}, \dots, x_{i,D}) \in \mathbb{R}^D (1 \leq i \leq N_P)$). Each component $x_{i,j} \in \mathbb{R} (1 \leq j \leq D)$ represents a variable or a parameter of the optimization problem. Usually, each $x_{i,j}$ is bounded to a value in the range $[li_j, ls_j]$, where $li_j, ls_j \in \mathbb{R}$ are the lower and upper bound, respectively. The N_P target vectors are initialized randomly by applying Equation 1:

$$x_{i,j} = li_j + U(0, 1) \times (ls_j - li_j) \quad (1)$$

where $U(0, 1)$ is a random number with uniform distribution in the range $[0, 1]$.

After the initialization step, an iterative process begins. The mutation operation (Line 5 of Algorithm 1) obtains a donor vector $v_i^g = (v_{i,1}, v_{i,2}, \dots, v_{i,D})$ for each target vector x_i^g from the current population P^g ($0 \leq g \leq max_{gen}$) following Equation 2. To obtain v_i^g , a base vector $x_{r_0}^g$ and other two vectors $x_{r_1}^g$ y $x_{r_2}^g$ are randomly selected from P^g , with r_0, r_1 and r_2 chosen from the set $\{1, 2, \dots, N_P\}$ and all of them are mutually exclusive. The $F \in [0...1)$ factor,

known as scale factor, controls the rate at which the population evolves, in order to avoid their stagnation during the search process. The mutation operator is important to the DE's behaviour because it focuses the search on the most promising areas of the solution space.

$$v_i^g = x_{r0}^g + F \times (x_{r1}^g - x_{r2}^g) \quad (2)$$

The donor vector is modified by the recombination operator (Line 6), with the aim of increasing the population diversity. This operator creates a trial vector u_i^g through mixing components of the donor vector v_i^g and the target vector x_i^g . The most frequently referred crossover operator is the binomial crossover, which is shown in Equation 3:

$$u_{i,j}^g = \begin{cases} v_{i,j}^g & \text{si } r_j < Cr \vee j = j_r \\ x_{i,j}^g & \text{otherwise} \end{cases} \quad (3)$$

where $r_j = U(0, 1)$ is a random value, j_r is also a random value in the set $\{1, 2, \dots, D\}$, and finally Cr is a parameter known as recombination probability, which controls the fraction of parameter values that are copied from the donor.

The last step is the selection operation (Line 7). The trial vector u_i^g competes against the target vector x_i^g regarding their objective values (obtained applying the objective function to each vector). The best vector is selected to be part of the population P^{g+1} of the next generation. Clearly, this competition creates a new population with a performance equal or superior to the current one (Line 8). Consequently, DE is an elitist evolutionary algorithm.

The stopping criteria can be set to a preset maximum number of iterations (max_{gen}) or some other problem-dependent criterion. Whichever the criteria to be set, the choice has a direct influence on the best solution x_{best} obtained by the algorithm (Line 12).

DE performance mainly depends on three parameters: scaling factor of the difference vector (F), crossover control parameter (Cr) and population size (N_P). Some guidelines are available to choose the control parameters [14]. In this work, N_P and F are chosen based on previous knowledge and keep it constant during all runs. On the other hand, a good value for Cr is 0.1 however, to speed up convergence a greater value can be used.

4 Our proposal: Hybrid DE for the FJSSP

In this section, our proposal to solve the FJSSP is detailed. In order to apply the DE algorithm, it is crucial to design a suitable encoding scheme that maps the floating point vectors to the feasible solution for the FJSSP (see Section 4.1). Moreover, our proposal is enhanced by a simple local search (Section 4.2). Finally, a parallel version of our proposal is introduced (Section 4.3).

4.1 Representation

In this work, the DE algorithm still manipulates real-valued vector in order to maintain the simplicity and properties of the DE in their natural configuration. Consequently, the schedule is generated following the random keys encoding scheme [15]. This representation deals with real point vectors, where these

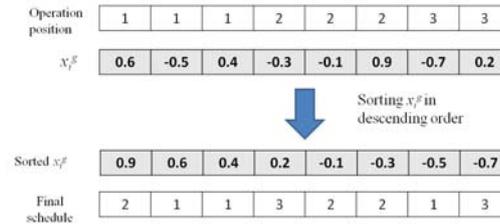


Fig. 1. Example of the decoding process used by the DE to solve the FJSSP.

points are used as sort keys to decode the solution. For an n -job m -machine scheduling problem, each vector's position (a random key) consists of a floating number in $U(-1,1)$ which can be translated to an unique list of ordered operations, after a descending order of the random keys. These steps always obtain a feasible schedule from a real-valued vector. The schedule is a permutation with repetitions [16]. See Figure 1 for an example considering the instance shown in Table 1. Given the vector $x_i^g = [0.6, -0.5, 0.4, -0.3, -0.1, 0.9, -0.7, 0.2]$, it is converted to a schedule $[2, 1, 1, 3, 2, 2, 1, 3]$, which is a permutation of the set of operations that represents a tentative ordering to schedule them, each one being represented by its job number. This valid schedule corresponds to the operation sequence $O_{21}, O_{11}, O_{12}, O_{31}, O_{22}, O_{23}, O_{13}$, and O_{32} .

In order to evaluate x_i^g , the objective value is the makespan (C_{max}). To compute it, each operation O_{ij} in x_i^g is assigned to a feasible machine M_k in U_{ij} with the shortest completion time, and then the load of M_k must be updated. The initial solution is generated by a random procedure (Equation 1), mainly because high performing construction heuristics for the FJSSP are unknown.

4.2 DE and Local Search Method

DE is enhanced with a local search technique, yielding a hybrid DE (HDE) for exploration and exploitation among the solutions to obtain a near optimal solution. In this work, a simple interchange mechanism is implemented in which two positions of the target vector are randomly selected and interchanged. If there is an improvement in the objective function the swap is accepted, otherwise, it is not considered. This local search procedure is applied to the target vectors x_i of the next population (just before Line 11 of Algorithm 1) but not to the trial vector u_i , which is beneficial to avoid both cycling search and getting trapped in a local optimum. Moreover, the frequency of the local search is controlled by the probability p_{LS} . Another important characteristic of this local search procedure is that it does not need a backward conversion because it is applied over the real-valued vector.

4.3 DE and Parallelism

In terms of designing parallel metaheuristics, the DE can be paralleled in different ways [2]. In this work, the aim of the parallelization is not to change the behaviour of the metaheuristic but to speed up the search. For that purposes,

we focus on the parallelization of each iteration of the DE [17]. The population of individuals is decomposed and handled in parallel, using the well-known global parallelization model. A principal process performs the selection operations and the replacement, which are generally sequential. The rest processes (workers) perform the mutation, recombination, and the evaluation of the solutions in parallel. Consequently, this model maintains the sequence of the original algorithm, and hence the behavior of the metaheuristic is not altered.

5 Experimental Design

In this section, we describe the experimental design used in this approach. We have selected a wide range of FJSSP instances used in the literature taking into account their complexity, which is given by the number of jobs and machines, and the wide variation of flexibility in the number of available machines per operation. In this sense, we considered the data set proposed by Brandimarte [18] as a representative one, since the number of jobs ranges from 10 to 20, the number of machines belongs to the set $\{4,15\}$ and the number of operations for each job ranges from 5 to 15, consequently the total number of operations ranges from 55 to 240. The flexibility varies between 1.43 and 4.10.

The parametric configuration considered for the DE's experimentation is the following. The population size, N_P is set to 50. The F factor is equal to 0.9. These values were adopted from previous works. Regarding Cr probability, three different values were considered (0.1, 0.5, and 0.9). For the remaining parameter, P_{LS} , three values are also analysed (0.1, 0.5 and 0.7).

Because of the stochastic nature of the algorithms, we performed 30 independent runs of each test to gather meaningful experimental data and apply statistical confidence metrics to validate our conclusions. As the data do not follow a normal distribution, we used the Kruskal-Wallis (KW) test to assess whether or not there were meaningful differences between the compared algorithms with a confidence level of 99%.

The experimentation is carried out on a cluster of 4 INTEL I7 3770K quad-core processors, 8 GB RAM, and the Slackware Linux with 2.6.27 kernel version. To implement the parallel version of DE, a portable programming interface for shared memory parallel computers such as OpenMP [19] is used.

6 Experimental Results

The first analysis is focused on the effect of using different Cr values in the DE performance, from low to high values (0.1, 0.5 and 0.9). Table 2 shows the best C_{max} values obtained for the DE algorithm using the different Cr values for each instance (columns 3 to 5). Also, the mean C_{max} values together with the mean standard deviation (sd) are presented (columns 6 to 8). Column 2 displays the best known C_{max} value for each instance. Last column shows the results of the KW test, where the symbol “+” indicates significant differences between the algorithms (p -value is inferior to the significance level $\alpha = 0.01$).

Table 2. C_{max} values found by the DE algorithm with different Cr values.

Instance	Opt.	Best C_{max} Values			Mean C_{max} Values \pm sd			KW
		$Cr=0.1$	$Cr=0.5$	$Cr=0.9$	$Cr=0.1$	$Cr=0.5$	$Cr=0.91$	
Mk01	40	40	40	40	40.00 \pm 0.00	40.00 \pm 0.00	40.73 \pm 0.78	-
Mk02	26	27	27	27	27.00 \pm 0.00	27.20 \pm 0.42	27.33 \pm 0.47	+
Mk03	204	204	204	204	204.00 \pm 0.00	204.00 \pm 0.00	204.00 \pm 0.00	-
Mk04	60	60	61	60	61.70 \pm 0.60	65.20 \pm 2.53	64.80 \pm 2.19	+
Mk05	172	174	177	173	175.87 \pm 0.73	181.10 \pm 1.79	174.03 \pm 1.82	+
Mk06	58	66	70	60	67.17 \pm 0.59	71.10 \pm 0.74	63.15 \pm 1.12	+
Mk07	139	144	149	140	144.07 \pm 0.25	150.10 \pm 0.99	142.18 \pm 1.30	+
Mk08	523	523	523	523	523.00 \pm 0.00	523.30 \pm 0.95	523.00 \pm 0.00	-
Mk09	307	321	338	307	325.13 \pm 1.48	343.00 \pm 2.40	310.05 \pm 2.48	+
Mk10	197	237	249	213	241.03 \pm 1.25	252.00 \pm 1.25	218.00 \pm 2.71	+

The DE algorithm with $Cr=0.9$ finds lowest C_{max} values than the rest. Moreover, this configuration reaches the best known C_{max} values in 5 of the 10 instances. Now, analyzing what is happening with the mean C_{max} values, the DE using $Cr=0.1$ presents the lowest values for the four first instances (MK01-04). Furthermore, sd values are equal to zero for these instances, indicating that the algorithm is able to find the optimum value for all the runs. In the remaining instances (MK05-10), the DE with $Cr=0.9$ presents the lowest C_{max} values. Finally, the DE using $Cr=0.5$ shows the worst performance. Given that the p -values of KW test is lower than the level of significance considered, we can state that there are significant differences among the DE with Cr values (except for instances MK01, MK03, and MK08).

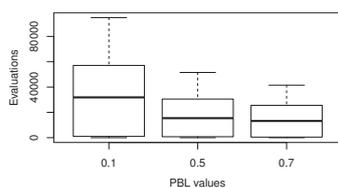
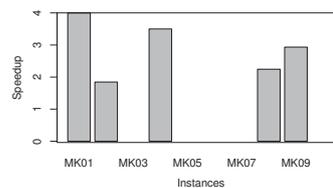
From previous analysis, we can conclude that for instances with relatively few operations, the fraction of parameter values that are copied from the target vector in the recombination operation should be small to allow the algorithm to find the best solutions to the problem. On the other hand, when the complexity of the instances grows up, it is necessary to increase the amount of disturbances in the solutions, and in this way the algorithm could converge to near-optimal solutions. Consequently, we will adopt two different values of Cr for the remaining experimentation: $Cr = 0.1$ for MK01-04 instances and $Cr = 0.9$ for the remaining ones.

Following analysis goes into detail of what happened with the introduction of a local search procedure to the DE algorithm (HDE) to solve the FJSSP. For that purpose, we considered three different p_{LS} values: 0.1, 0.5 and 0.7 (low, medium, and high probability values), i.e. we study how the frequency of the LS application impacts on the DE performance.

Table 3 shows the best and mean C_{max} values obtained for the HDE with the different p_{LS} values. The HDE algorithm applying the local search procedure with high frequency ($p_{LS}=0.7$) obtains low best C_{max} values for the majority of the FJSSP instances. Moreover, this algorithm exhibits the lowest mean C_{max} values for all instances. These indicate that the algorithm is able to find the optimum or the near-optimum values in the majority of the runs. Moreover, the KW test indicates that there are statistical significant differences among the algorithms (p -values are lower than the level of significance). Comparing C_{max}

Table 3. C_{max} values found by the HDE algorithm with different p_{LS} values.

Inst	Opt.	Best C_{max}			Mean C_{max} values \pm sd			KW
		$p_{LS}=0.1$	$p_{LS}=0.5$	$p_{LS}=0.7$	$p_{LS}=0.1$	$p_{LS}=0.5$	$p_{LS}=0.7$	
MK01	40	40	40	40	40.00 \pm 0.00	40.00 \pm 0.00	40.00 \pm 0.00	-
MK02	26	27	26	26	27.00 \pm 0.00	26.80 \pm 0.41	26.77 \pm 0.43	+
MK03	204	204	204	204	204.00 \pm 0.00	204.00 \pm 0.00	204.00 \pm 0.00	-
MK04	60	60	60	60	61.20 \pm 0.66	60.33 \pm 0.48	60.17 \pm 0.38	+
MK05	172	173	173	173	1.73 \pm 0.18	173.00 \pm 0.00	173.00 \pm 0.00	+
MK06	58	62	61	60	63.27 \pm 0.69	62.13 \pm 0.73	61.90 \pm 0.48	+
MK07	139	140	139	140	142.37 \pm 0.96	140.83 \pm 0.79	140.63 \pm 0.61	+
MK08	523	523	523	523	523.00 \pm 0.00	523.00 \pm 0.00	523.00 \pm 0.00	-
MK09	307	307	307	307	309.93 \pm 1.53	307.67 \pm 1.06	307.37 \pm 0.89	+
MK10	197	225	223	219	228.20 \pm 1.97	225.77 \pm 1.36	224.73 \pm 1.66	+

**Fig. 2.** Total number of evaluations for the HDE .**Fig. 3.** Speedup per FJSSP instances .

values from Table 2 and the ones from Table 3, we can observe that the HDE algorithms obtain best

Figure 2 illustrates the distribution of the number of evaluations to find the best C_{max} values for the HDE algorithms with different P_{LS} values. We observe that the HDE algorithm with $P_{LS}=0.7$ needs less number of evaluations than the rest of the algorithms. If we also consider the C_{max} values obtained by each HDE, the one with $P_{LS}=0.7$ is the best approach to solve this problem.

Following analysis is devoted to compare the HDE and its parallel version as described in Section 4.3. The most important measure of a parallel algorithm is the speedup. The speedup is defined as the ratio of the sequential execution time (HDE execution time, in this case) to the parallel execution time. For this analysis, we consider the weak speedup [20]. For that reason and following the best practice by Luque and Alba [3], the stopping criterion is based on the quality of the final solution achieved by the algorithms, which is set to the best known C_{max} for each FJSSP instance (see column Opt of Table 2). Consequently, the speedup values is only reported for the instances for which the HDE algorithm obtains the optimum value.

Figure 3 shows that the use of parallelization is worth while, which allow us to speed up the execution time with respect to the sequential HDE near 3 times in average (the ideal speedup value is 4, the number of available cores per machine).

Finally, to determine the goodness of the metaheuristics considered in this work, we present a comparison of the results from the HDE with several competitive algorithms present in the literature. This allows a comparative assessment

Table 4. Comparison between HDE and population-based Metaheuristics from the literature

	MK01	MK02	MK03	MK04	MK05	MK06	MK07	MK08	MK09	MK10
HDE	40	26	204	60	173	60	140	523	307	219
hGA	40	26	204	62	172	65	140	523	310	214
BEDA	40	26	204	60	172	60	139	523	307	206
IACO	40	26	204	60	173	60	140	523	307	208
HDE	40	26	204	60	172	57	139	523	307	198

of the algorithms for the FJSSP. In this comparison different population-based metaheuristics to solve the FJSSP are considered: i) a hybrid algorithm combining chaos particle swarm optimization with genetic algorithm (hGA) [4], ii) a bi-population based estimation of distribution algorithm (BEDA) [5], iii) an ant colony optimization (IACO) [6], and finally, iv) a hybrid differential evolutionary algorithm [12]. From the comparison, the C_{max} values of HDE are similar with the values of remaining algorithms, for the majority of the ten instances (a comparative table is no included due to lack of space). This observation suggests that the HDE developed in this work is a competitive algorithm to solve the FJSSP. Comparisons regarding computational effort are hard to be carried out because the majority of the works do not report number of evaluations. Consequently, the relative efficiency of referred algorithms are difficult to contrast in order to obtain meaningful comparisons.

7 Conclusions

In this article, we have presented a simple DE algorithm to solve the FJSSP. In this study, the traditional real-parameter global optimizer is considered to maintain the properties of the DE in their natural configuration. The DE is enhanced with a very simple local search procedure, obtaining a hybrid DE (HDE). Moreover, each iteration of the DE is parallelised to speed up the computation. The results indicate that the HDE with a high probability, at which the local search procedure is applied, is able to find the best solutions for the FJSSP. Moreover, when HDE is contrasted with algorithms in the literature, it also becomes a competitive approach. As a consequence, HDE gives good solutions to this NP-hard problem in an efficient and competitive way.

As future research activities, we will plan to extend the study by including another set of instances with high dimensionality. Furthermore, variants of the FJSSP with more constraints will be evaluated considering the approaches developed in this article.

Acknowledgements

This research is supported by Universidad Nacional de La Pampa, and the Incentive Program from MINCyT. The last author acknowledges CONICET.

References

1. M. R. Garey, D. S. Johnson, and R. Sethi, “The complexity of flowshop and jobshop scheduling,” *Math. Oper. Res.*, vol. 1, no. 2, pp. 117–129, May 1976.
2. E.-G. Talbi, *Metaheuristics: From Design to Implementation*. Wiley, 2009.
3. G. Luque and E. Alba, *Parallel Genetic Algorithms: Theory and Real World Applications*. Springer Publishing Company, Incorporated, 2013.
4. J. Tang, G. Zhang, B. Lin, and B. Zhang, “A hybrid algorithm for flexible job-shop scheduling problem,” *Procedia Engineering*, vol. 15, pp. 3678 – 3683, 2011.
5. L. Wang, S. Wang, Y. Xu, G. Zhou, and M. Liu, “A bi-population based estimation of distribution algorithm for the flexible job-shop scheduling problem,” *Computers & Industrial Engineering*, vol. 62, no. 4, pp. 917 – 926, 2012.
6. L. Wang, J. Cai, M. Li, and Z. Liu, “Flexible job shop scheduling problem using an improved ant colony optimization,” *Scientific Programming*, pp. 1–11, 2017.
7. R. Storn and K. Price, “Differential evolution – a simple and efficient heuristic for global optimization over continuous spaces,” *Journal of Global Optimization*, vol. 11, no. 4, pp. 341–359, 1997.
8. B. Teoh, S. Ponnambalam, and G. Kanagaraj, “Differential evolution algorithm with local search for capacitated vehicle routing problem,” *Int. J. Bio-Inspired Comput.*, vol. 7, no. 5, pp. 321–342, 2015.
9. R. Greco and I. Vanzi, “New few parameters differential evolution algorithm with application to structural identification,” *Journal of Traffic and Transportation Engineering (English Edition)*, vol. 6, no. 1, pp. 1 – 14, 2019.
10. P. Hull, M. Tinker, and G. Dozier, “Evolutionary optimization of a geometrically refined truss,” *Structural and Multidisciplinary Opt.*, vol. 31.
11. R. M. K. Rout, “Simultaneous selection of optimal parameters and tolerance of manipulator using evolutionary optimization techniques,” *Structural and Multidisciplinary Optimization*, vol. 40, no. 1-6, pp. 513–528, 2010.
12. Y. Yuan and H. Xu, “Flexible job shop scheduling using hybrid differential evolution algorithms,” *Computers & Industrial Eng.*, vol. 65, no. 2, pp. 246–260, 2013.
13. T. Eltaeib, Tarik, and A. Mahmood, “Differential evolution: A survey and analysis,” *Applied Sciences*, vol. 8, no. 10, 2018.
14. K. Price, R. M. Storn, and J. A. Lampinen, *Differential Evolution: A Practical Approach to Global Optimization (Natural Computing Series)*. Berlin, Heidelberg: Springer-Verlag, 2005.
15. J. C. Bean, “Genetic algorithms and random keys for sequencing and optimization,” *ORSA Journal on Computing*, vol. 6, no. 2, pp. 154–160, 1994.
16. C. Bierwirth, “A generalized permutation approach to job shop scheduling with genetic algorithms,” *Operations-Research-Spektrum*, vol. 17, pp. 87–92, 1995.
17. N. Nedjah, E. Alba, and L. de Macedo Mourelle, *Parallel Evolutionary Computations*. Springer-Verla, 2006.
18. P. Brandimarte, “Routing and scheduling in a flexible job shop by tabu search,” *Annals of Operations Research*, vol. 41, pp. 157–183, 1993.
19. B. Chapman, G. Jost, and R. van der Pas, *Using OpenMP: Portable Shared Memory Parallel Programming*. MIT Press, 2007.
20. E. Alba, *Parallel Metaheuristics: A New Class of Algorithms*. Wiley, 2005.

Algoritmos de Regresión Lineal aplicados al mantenimiento de un Datacenter

Federico Gabriel D'Angiolo¹, Iván Federico Kwist¹ Matias Loiseau¹, David Exequiel Contreras¹, Fernando Asteasuain¹

¹ Universidad Nacional de Avellaneda, Avellaneda, Argentina. Departamento de Tecnología y Administración. Ingeniería en Informática.

fdangiolo@undav.edu.ar, ivankwist@hotmail.com.ar, matiasloiseau@gmail.com, dcontreraspastorini@cev.undav.edu.ar, fasteasuain@undav.edu.ar

Resumen. En el presente Trabajo se describe la aplicación de Algoritmos de Regresión Lineal al estudio del comportamiento climático de un Datacenter. Este análisis permite comprender cómo varían la temperatura y la humedad, con el objetivo de predecir cuál es el grado de ventilación adecuada que debe haber dentro del recinto para mantener operativos a los servidores y sistemas de cómputo que se encuentran.

Palabras claves: Inteligencia Artificial, Regresión Lineal, temperatura, humedad.

1 Introducción

En la actualidad se tiene acceso a una gran cantidad de datos que facilitan la automatización de sistemas y agilizan las tareas que tal vez, hasta hace unos años, resultaban laboriosas. Con el caudal de datos que se puede conseguir, resulta necesario el procesamiento de los mismos con el objetivo de obtener conclusiones sobre un comportamiento en especial o determinar qué tarea debe ser ejecutada para lograr un fin determinado, por eso, es que hoy en día los algoritmos de Inteligencia Artificial (IA), cobran mayor peso. Ejemplo de esto se da en el Trabajo "A Review at Machine Learning Algorithms Targeting Big Data Challenges" [1], donde se describe un análisis comparativo de los algoritmos de Aprendizaje Automático basados en grandes cantidades de datos (Big Data). Dentro de la IA, uno de los campos importantes es el Aprendizaje Automático o Machine Learning el cual, a su vez, se encuentra dividido en tres ramas importantes: Aprendizaje Supervisado, Aprendizaje No Supervisado y Aprendizaje Reforzado. Un ejemplo de lo comentado, se da en el Trabajo "Machine Learning for Engineering" [2], en donde se estudian las técnicas de Aprendizaje Automático para resolver problemas de producción dentro de la automatización. En particular, en el presente Trabajo, se estudia el comportamiento dinámico que tienen las variables como temperatura y humedad dentro de un Datacenter y para esto se recurre entonces a los algoritmos de Aprendizaje

Supervisado, en especial, a la Regresión Lineal Simple y a la Regresión Lineal Multivariable ya que el objetivo es poder analizar la relación de la temperatura y la humedad en distintos puntos del Laboratorio. Este análisis permite estudiar cómo se propaga el flujo de calor dentro del Datacenter para mejorar o aumentar la ventilación dentro del mismo permitiendo una mayor vida útil de los servidores. Como base para el desarrollo de ese trabajo, se toma el caso de “Multiple Linear Regression to Improve Prediction Accuracy in WSN Data Reduction” [3] en donde se comenta la utilización de las técnicas de Regresión Lineal Simple y Múltiple aplicadas a una red de sensores. Resulta interesante este artículo porque no se limita a describir cada una de estas técnicas sino que se detalla cómo la Regresión Múltiple mejora ciertos aspectos de precisión que la Regresión Lineal no llega a abarcar. Por otro lado, se estudió el trabajo “Regresión lineal simple y múltiple: aplicación en la predicción de variables naturales relacionadas con el crecimiento microalgal” [4], en donde se comentan los modelos de Regresión Lineal y Regresión Múltiple para modelar relaciones entre variables como la temperatura, luz, pH y oxígeno disuelto, con el objetivo de analizar cómo influyen dichas variables en el crecimiento de las microalgas. Lo importante del trabajo mencionado, con el que se describe aquí, es cómo adaptar estos modelos matemáticos a las variables sensadas. Por último, para el análisis general de Regresión se estudia el Trabajo: “Caracterización térmica de edificios aplicando el modelo de regresión lineal múltiple” [5], en el cual se presenta una metodología para evaluar del comportamiento térmico de una construcción representativa del centro bonaerense.

Para aplicar estos algoritmos, previamente se realiza un Dataset con valores de temperatura y humedad los cuales fueron obtenidos mediante sensores ubicados dentro del Datacenter. Estos valores se muestrean en el intervalo de meses de septiembre 2018 – marzo 2019 (https://gitlab.com/datasets_gi/dataset) para lograr un conjunto de datos robusto. Con estos, el objetivo es realizar dos estudios: el primero consiste en obtener un modelo lineal que relacione las temperaturas entre dos puntos distintos de la habitación, y el segundo consiste en obtener un modelo lineal entre la temperatura de uno de los servidores con la temperatura que existe en distintos puntos del Datacenter.

Actualmente existen Trabajos relacionados a la toma de datos y análisis de los mismos para automatizar, por ejemplo, una residencia. El objetivo aquí es obtener modelos que permitan estudiar cómo se propagan las variables mencionadas para mantener el correcto funcionamiento de los servidores de forma que su vida útil no se vea disminuida. El Datacenter bajo estudio se encuentra dentro del Laboratorio de Redes de la Carrera de Ingeniería Informática de la Universidad de Avellaneda. Allí se encuentran dos servidores con dos sistemas de refrigeración. En la sección 3 de este Trabajo, se detalla cómo se distribuyen los servidores dentro del recinto.

La distribución de este Trabajo se puede describir de la siguiente forma: en la sección 2 se realiza una introducción al tema de Algoritmos de Regresión, luego en la sección 3 se describe la relación de temperatura entre dos puntos del ambiente con el objetivo de analizar cómo se distribuyen la temperatura y la humedad en el ambiente mediante la obtención de un modelo matemático aproximado y, por último, en la sección 4, se describe la relación de temperatura y de humedad, entre uno de los

servidores con el ambiente, con el objetivo de estudiar cómo se ve afectado dicho servidor por las variables mencionadas. Este último también se enfoca en la obtención de un modelo matemático.

2 Algoritmos de Regresión

Desde el punto de vista matemático, la Regresión tiene dos significados: uno surge de la distribución conjunta de probabilidades de dos variables aleatorias mientras que el otro significado es empírico y nace de la necesidad de ajustar alguna función a un conjunto de datos. [6]

En el caso del presente Trabajo, resulta interesante el segundo punto de vista de este concepto ya que el objetivo es encontrar un modelo lineal que permita predecir la relación entre los distintos valores de temperatura y humedad obtenidos. Previamente a estudiar la Regresión Lineal, resulta conveniente realizar un estudio de linealidad sobre el conjunto de datos para obtener un indicio de que las variables tengan un cierto grado de correspondencia. En base a este primer objetivo, se utiliza el Coeficiente de Correlación para el cual se toma como criterio de diseño que, si el mismo resulta ser un valor mayor o igual 0,7, se pueden ajustar los datos mediante Regresión Lineal, teniendo en cuenta que puede existir cierto error. En caso de que el porcentaje resulte menor a dicho valor, la linealidad no será evidente y se podrá proceder a buscar algún modelo no lineal que mejor ajuste [7].

Para darle un marco matemático al análisis, se comienza definiendo a un Dataset como un conjunto de valores, que se puede describir de la siguiente forma:

$$X = x_1, x_2, \dots, x_n \quad \text{donde } x_n \in \mathbb{R}^n \quad (1)$$

Cada vector de entrada está asociado a un valor real y_i :

$$Y = y_1, y_2, \dots, y_n \quad \text{donde } y_n \in \mathbb{R} \quad (2)$$

Un modelo lineal se basa en la suposición de que es posible aproximar los valores de salida a través de un proceso de regresión basado en la regla:

$$f(x_i) = \alpha_0 + \alpha_1 \cdot x_1 + \alpha_2 \cdot x_2 + \dots + \alpha_n \cdot x_n \quad (3)$$

$$A = \alpha_0, \alpha_1, \alpha_2, \dots, \alpha_n \quad \text{resultan constantes} \quad (4)$$

En la expresión anterior, las constantes se encuentran mediante el Algoritmo de Regresión Lineal. Si la ec (3) resulta tener las constantes α_2 , α_3 , ..., α_m nulas, el modelo resulta ser de Regresión Simple mientras que, si estas constantes no son nulas, el modelo se denomina Regresión Multivariable. Con esto queda definido el modelo de Regresión Lineal para explicar a continuación, cómo se relaciona con el objetivo de este Trabajo [8].

3 Relación de Temperatura entre dos puntos del Datacenter

El primer estudio a realizar consiste en comparar la medición de temperatura de dos puntos distintos del Datacenter. Para esto, se presenta el diagrama de la distribución de los sensores:

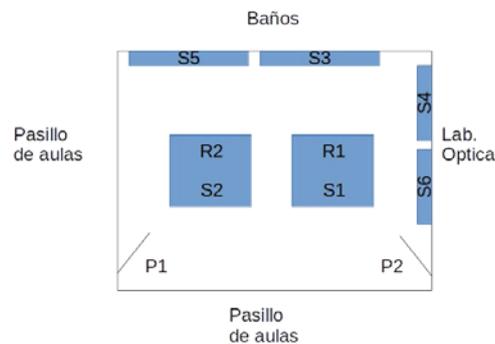


Figura 1. Layout del Datacenter y ubicación de los sensores

Siendo:

R1: Rack 1, donde se encuentra alojado el servidor N.º 1

R2: Rack 2., donde se encuentra alojado el servidor N.º 2

P1: Puerta que da al pasillo.

P2: Puerta que da al Laboratorio de Óptica.

S₁, S₂, S₃, S₄, S₅ y S₆: Sensores de temperatura y humedad, integrados.

Como se puede ver en la Fig.1, el sensor S₄, se encuentra ubicado de forma vertical con el S₆ con lo cual la finalidad de este estudio es encontrar, si existe, una relación lineal en el cambio de temperatura y de humedad de ambos sensores, es decir, si se observa una variación de temperatura (o humedad) en S₄ se debería ver la misma variación en S₆. Para lograr esto, se toma el dataset de los ocho meses comentados y se estudia un modelo de Regresión Lineal Simple.

Previo al estudio de Regresión Lineal, resulta conveniente analizar el coeficiente de correlación con el objetivo de verificar que existe una relación lineal entre los

datos. Al realizar este cómputo, se obtiene el siguiente valor para el coeficiente de correlación:

$$\rho = 0.696$$

El Coeficiente de Correlación toma un valor cercano a 0,7, para lo cual se puede admitir que el modelo mantiene cierta linealidad, es decir que mediante un análisis de Regresión Lineal Simple se obtendría un modelo como el siguiente:

$$T_6 = 10,78 + 0,64 \cdot T_4 \quad (5)$$

El cual, gráficamente quedaría de la siguiente forma:

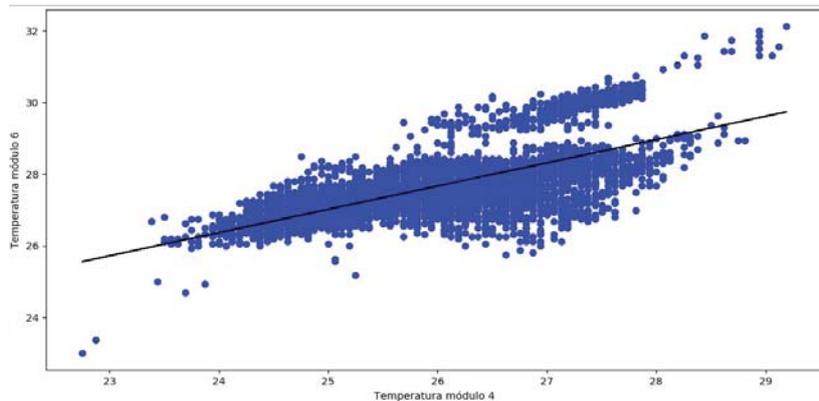


Figura 2. Regresión Lineal de temperatura entre el sensor S_6 y S_4

De la misma forma se realiza el estudio para el análisis de la humedad, obteniendo un coeficiente de correlación de:

$$\rho = 0.91$$

En este caso el Coeficiente de Correlación toma un valor de 0,91, para lo cual se puede admitir que el modelo mantiene una linealidad más estricta en comparación con el caso de la temperatura, es decir que mediante un análisis de Regresión Lineal Simple se obtendría un modelo como el siguiente:

$$H_6 = 7,37 + 0,79 \cdot H_4 \quad (6)$$

El cual, gráficamente quedaría de la siguiente forma:

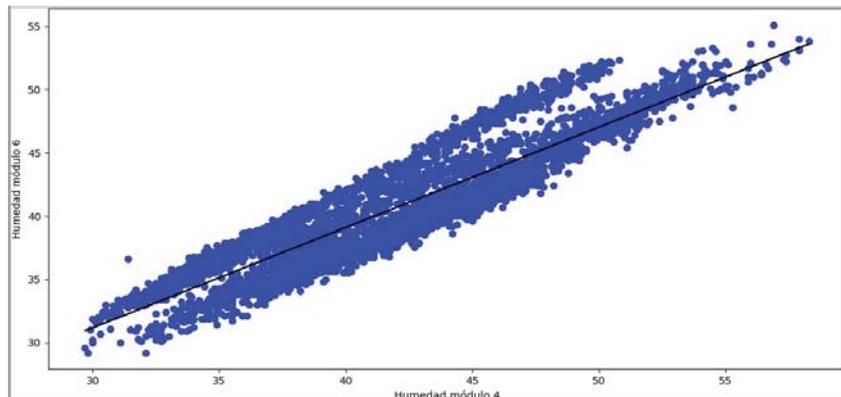


Figura 3. Regresión Lineal de humedad entre el sensor S_6 y S_4

Observando ambos análisis, se puede concluir que la relación de temperatura y humedad para ambos sensores mantiene cierta linealidad sobre todo en el caso de la humedad donde resulta ser más clara. Esto determina entonces que el flujo de calor se traslada linealmente desde S_4 a S_6 . Es importante entonces tener en cuenta que una posible forma de ventilación resultaría estar cerca de S_6 (o en ese mismo lugar), para mantener refrigerado al servidor N.º 1.

4 Relación de temperatura entre el servidor y la del ambiente.

El objetivo de este segundo estudio es encontrar un modelo que permita predecir si la temperatura sobre uno de los servidores se ve afectada por la temperatura existente en los distintos puntos del Datacenter. Para esto, se toma como ejemplo el Servidor N.º1, el cual tiene asociado el sensor S_1 y se propone el siguiente modelo de Regresión Multivariable:

$$T_1 = \alpha_0 + \alpha_2 \cdot T_2 + \alpha_3 \cdot T_3 + \alpha_4 \cdot T_4 + \alpha_5 \cdot T_5 + \alpha_6 \cdot T_6 \quad (7)$$

El modelo propuesto por la ec. (7), supone que la temperatura del sensor S_1 depende linealmente de la temperatura que se encuentra alrededor del servidor la cual se encuentra cuantificada por los sensores S_2, S_3, S_4, S_5 y S_6 . Desde un punto de vista vectorial, se puede decir que T_1 es una combinación lineal de las otras temperaturas, tomadas a estas últimas como vectores. Si algunas de las constantes de la ec.(7),

resulta nula, evidencia que ese vector no tiene influencia sobre T_1 , es decir, la temperatura en ese punto espacial no tiene ponderación suficiente.

Como se mencionó antes, previamente se observará el coeficiente de correlación entre las distintas variables para poder analizar si resulta eficiente aplicar Regresión lineal. Para esto, se tomando los coeficientes de correlación entre T_1 y cada una de las temperaturas mencionadas, se obtiene lo siguiente:

$$\begin{aligned}\rho_{1,2} &= 0.77573 \\ \rho_{1,3} &= 0.45246 \\ \rho_{1,4} &= 0.71656 \\ \rho_{1,5} &= 0.81005 \\ \rho_{1,6} &= 0.78763\end{aligned}$$

Para visualizar estas relaciones, se propone la siguiente imagen:

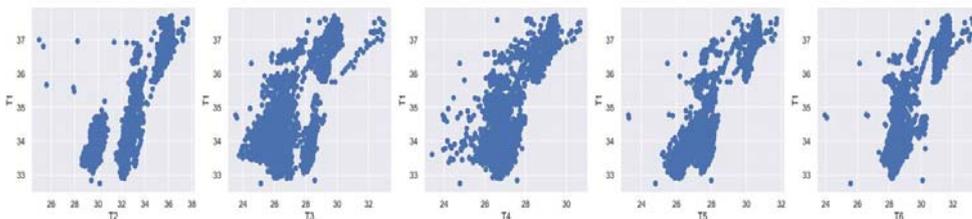


Figura 4. Gráficos de Correlación entre S_1 y los distintos sensores ubicados alrededor del servidor.

Como se puede observar, tanto por los coeficientes de correlación como por los gráficos, la relación no es exactamente lineal en todos los casos, lo cual se puede corroborar con los valores de los coeficientes obtenidos. Es decir, admitiendo que el modelo mantiene cierta linealidad para un coeficiente de $\rho = 0,7$, sería conveniente plantear un modelo de Regresión Lineal Múltiple.

Es importante contemplar el caso de la relación entre el sensor uno y el tres donde el factor de correlación no resulta tener un valor cercano al planteado como criterio.

En base a esta idea, la expresión matemática para el modelo, sería la siguiente:

$$T_1 = 9,49 + 0,18 \cdot T_2 - 0,22 \cdot T_3 + 0,07 \cdot T_4 + 0,53 \cdot T_5 + 0,28 \cdot T_6 \quad (8)$$

Luego para la humedad, se repite el mismo proceso obteniendo los siguientes valores del factor de correlación:

$$\begin{aligned}\rho_{1,2} &= 0.73803 \\ \rho_{1,3} &= 0.91343 \\ \rho_{1,4} &= 0.58855 \\ \rho_{1,5} &= 0.90415\end{aligned}$$

$$\rho_{1,6} = 0.81064$$

Para visualizar estas relaciones, se propone la siguiente imagen:

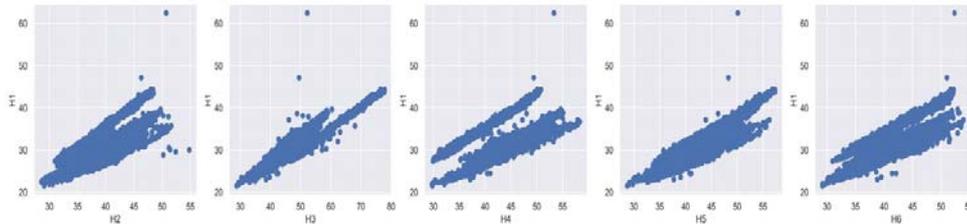


Figura 5. Gráficos de Correlación entre S_1 y los distintos sensores ubicados alrededor del servidor.

Como se puede observar, tanto por los coeficientes de correlación como por los gráficos, la relación de valores exhibe una linealidad más certera en comparación con el caso de la temperatura lo cual permite afirmar que sería coherente plantear un modelo de Regresión Lineal Múltiple.

En base a esta idea, la expresión matemática para el modelo, sería la siguiente:

$$H_1 = 3,16 - 0,01 \cdot H_2 + 0,43 \cdot H_3 + 0,28 \cdot H_4 - 0,26 \cdot H_5 + 0,16 \cdot H_6 \quad (9)$$

Observando los modelos matemáticos de temperatura y humedad, se puede ver que el efecto del sensor números dos que pertenece al servidor dos, contiene constantes que no influyen sobre el servidor uno, es decir, que el servidor dos no impacta de manera importante en cuanto a temperatura y humedad se refiere, sobre el servidor uno.

Para corroborar el efecto de lo observado, se propone tomar el mismo Dataset y obtener un modelo de Regresión Lineal que no contemple el efecto del módulo dos. Con este objetivo, se repiten los mismos pasos y queda la siguiente expresión:

$$T_1 = 16,06 - 0,88 \cdot T_3 - 1,1 \cdot T_4 + 2,61 \cdot T_5 + 0,01 \cdot T_6 \quad (10)$$

El cual comparado con el modelo original, el cual tiene en cuenta el efecto del sensor dos (sobre el servidor dos), se nota que el modelo modifica el peso de cada constante pero el efecto que tiene cada sensor sobre el del servidor uno, es similar al modelo obtenido anteriormente.

Este último análisis lleva a pensar que si se desea obtener un modelo más estricto, se puede tener en cuenta el efecto del módulo dos, sin embargo, se puede trabajar con un modelo menos robusto a los efectos de tener un menor costo computacional a la hora de obtener estas constantes aunque la aproximación no resulta tener la misma fidelidad que el modelo original.

Por último, es importante contemplar en el caso del modelo completo (donde interviene el módulo dos), se puede observar que tanto para la temperatura como para la humedad, los módulos tres y cinco son los que más influyen lo cual lleva a la conclusión de que si se deseara agregar ventilación al Laboratorio, se debería ubicar en lugares cercanos a donde se ubican estos módulos.

5 Conclusión

En base a estos estudios, se puede ver lo importante de analizar el Coeficiente de Correlación previamente al estudio de la Regresión Lineal. En el estudio de temperatura para el caso de Regresión Lineal Simple, el modelo obtuvo un Coeficiente de Correlación inferior comparado con el estudio para la humedad. En base a un criterio de diseño adoptado, se pudo concluir que las condiciones climáticas siguen un modelo lineal aproximado lo cual permite concluir que, para mejorar la ventilación del Datacenter, conviene agregar un sistema de refrigeración cerca del sensor S_6 .

Continuando con el análisis de la Regresión Lineal Múltiple, también se hizo un análisis del Coeficiente de Correlación el cual fue analizado para cada una de las relaciones entre el servidor y los distintos puntos espaciales para observar dónde resulta haber un grado de linealidad mayor. En base a esto, se pudo ver que en el caso de la humedad el modelo resultó tener mayor linealidad en comparación con el caso de la temperatura, es decir, este estudio resultó ser similar al caso de Regresión Lineal Simple en el sentido de que la humedad proporciona un modelo totalmente lineal.

De este último estudio se desprende que, en caso de agregar una ventilación para mejorar el funcionamiento del servidor N.º 1, podría ser donde se sitúa el sensor S_6 , para agregarle mayor énfasis a la climatización del Datacenter.

En el caso de ser más estrictos sobre el Coeficiente de Correlación en el caso del modelo de temperatura, resulta claro entonces que la Regresión Lineal no resulta adecuada para este análisis lo cual lleva a concluir que se puede realizar el estudio de otro tipo de Regresión que mejor se ajuste a estos datos.

Como trabajo a futuro, resulta interesante realizar una generalización de este tema aplicado a distintos escenarios, como por ejemplo, restaurantes, donde resulta importante una ventilación a los efectos de los clientes y el personal de trabajo, también en transporte donde resulta vital mantener determinada ventilación en las distintas épocas del año a los efectos de quienes adquieren estos servicios. Es decir, se podría sensar las variables estudiadas en este trabajo en cada uno de los escenarios mencionados para luego, mediante Regresión Lineal u otro tipo de Regresión, estudiar la generalización y obtener conclusiones. Esta comparación enriquecería la investigación dado que se extendería el campo a ambientes sociales donde la IA pueda tener mayor impacto.

Este Trabajo se realizó en base a los Proyectos de Investigación UNDAVCYT 2014 y PROAPI 2017, desde la Carrera de Ingeniería Informática de la Universidad

Nacional de Avellaneda, bajo el título de “Mantenimiento de parámetros del ambiente del Laboratorio de Redes y Sistemas de Computación mediante protocolos de IoT”.

Referencias

1. Abhinav Rathor., Manasi Gyanchandani.: A Review at Machine Learning Algorithms Targeting Big Data Challenges. ISBN: 978-1-5386-2361-9
2. Jeff Dyck.: Machine Learning for Engineering. Solido Design Automation. 978-1-5090-0602-1
3. Carlos Giovanni Nunes de Carvalho, Danielo Gonçalves Gomes, José Neuman de Souza and Nazim Agoulmine.: Multiple Linear Regression to Improve Prediction Accuracy in WSN Data Reduction. ISBN: 978-1-4577-1792-5
4. Arys Carrasquilla-Batista, Alfonso Chacón-Rodríguez, Kattia NúñezMontero, Olman Gómez-Espinoza, Johnny Valverde, Maritza Guerrero-Barrantes.: Regresión lineal simple y múltiple: aplicación en la predicción de variables naturales relacionadas con el crecimiento microalgal . ISSN: 0379-3962
5. N. Muñoz., B.M. Marino., L.P.Thomas.: CARACTERIZACIÓN TÉRMICA DE EDIFICIOS APLICANDO EL MODELO DE REGRESIÓN LINEAL MÚLTIPLE. ISBN: 978-987-29873-0-5
6. Canavos, G.: Probabilidad y Estadística. Aplicaciones y Métodos.. (1988). ISBN:968-451-856-0
7. Bonaccorso, G.: Machine Learning Algorithms. ISBN 978-1-78588-962-2.
8. Prateek, J.:Artificial Intelligence with Python.. BIRMINGHAM - MUMBAI. First published: January 2017. ISBN 978-1-78646-439-2

An Artificial Immune System for Solving Dynamic Economic Power Dispatch Problems

Victoria S. Aragón¹, Carlos A. Coello Coello² and Mario G. Leguizamón¹

¹ Laboratorio de Investigación y Desarrollo en Inteligencia Computacional* * *
Universidad Nacional de San Luis - Ej. de Los Andes 950
San Luis (5700), ARGENTINA

² CINESTAV-IPN (Evolutionary Computation Group)[†]
Departamento de Computación - Av. IPN No. 2508, Col. San Pedro Zacatenco
México D.F. 07300, MÉXICO

Abstract. In this paper, we propose an artificial immune system called IA_DED, which stands for *Immune Algorithm Dynamic Economic Dispatch*. It is designed for solving the Dynamic Economic Dispatch (DED) problem. Our approach considers the DED problem as a dynamic problem whose constraints change over time. IA_DED considers the activation process that T cells suffer in order to find partial solutions. The proposed approach is validated using several DED problems taken from the specialized literature. Our results are compared with respect to those obtained by other approaches taken from the specialized literature. We also provide some statistical analysis in order to determine the sensitivity of the performance of our proposed approach to its parameters.

Keywords: Artificial immune systems, dynamic economic dispatch problem, metaheuristics

1 Introduction

Electricity generation is the process by which electrical power is generated from other sources of energy. In other words, the generation of electrical energy is done by transforming some other type of energy (chemical combustion, nuclear fission, kinetic energy of flowing water and wind, solar photo-voltaic and geothermal power, among others) into electrical energy. This transformation takes place at a power station by electromechanical generators. It constitutes the first step of the electrical supply system. Then, electrical energy is transmitted and distributed to consumers by means of specialized systems.

The demand for electrical energy from a city, region or country has a variation throughout the day. This variation depends on many factors, such as: types of existing industries in the area and shifts performed on their production, weather (extremes of heat or cold), type of appliances that are most frequently used, type

* * * LIDIC is financed by Universidad Nacional de San Luis. The first and third authors acknowledges support from PROICO 31018.

[†] The second author acknowledges support from CONACyT project no. 1920.

of water heaters installed at homes, the season of the year and the time of day at which the energy demands are considered, among others. The generation of electrical energy should respond to the demand curve; that is, if energy demand is increased, power supply must also increase and vice versa. In the Dynamic Economic Dispatch (DED) problem a sequence of load demands has to be met by minimizing the production cost while some constraints are met.

On the other hand, any time-dependent problem can be considered as a dynamic problem. Such problems can change the objective function, the constraints or both. A change over a constraint exists when the problem conditions change (for instance, how much energy has to produce the system at one point). So, in this paper, the DED problems are considered as dynamic problems whose load demands constraint change over time in a random fashion.

The aim of this study is to assess the performance of the proposed algorithm which is designed to solve the DED problem. The proposed algorithm is able to minimize the production cost as well as the time invested to find it. Considering T load demands, the problem is regarded as a sequence of T problems. But, each problem (at time i): 1) depends on the solution produced for the previous problem (at time $i - 1$) and 2) conditions its successor (at time $i + 1$).

2 Problem Formulation

In the DED problem the main aim is to minimize the total production cost (TC) associated with N dispatch units for a time period:

$$TC = \sum_{t=1}^T \sum_{i=1}^N F_i(P_i^t) \quad (1)$$

where TC is the fuel cost over the whole dispatch period, $\sum_{i=1}^N F_i(P_i^t)$ is the fuel cost for the t^{th} interval, $P^t = (P_1^t, P_2^t, \dots, P_N^t)$ is the power output of each unit at time t , T is the number of intervals in the period, N is the number of generators or units in the system, P_i^t is the power of the i^{th} unit at time t (in MW) and F_i is the fuel cost for the i^{th} unit (in \$/h).

The simplest fuel cost function (i.e., smooth) can be expressed as a single quadratic function: $F_i(P_i^t) = a_i(P_i^t)^2 + b_iP_i^t + c_i$, where a_i , b_i and c_i are the fuel consumption cost coefficients of the i^{th} unit. But, if the valve-point effects are taken into account, the fuel cost function becomes non-smooth and the i^{th} unit is expressed as the sum of a quadratic and a sinusoidal function in the form: $F_i(P_i^t) = a_i(P_i^t)^2 + b_iP_i^t + c_i + |e_i \sin(f_i(P_{min_i} - P_i^t))|$, where e_i and f_i are the fuel cost coefficients of the i^{th} unit with valve-point effects.

The minimization of TC is subject to:

1. Power Balance Constraint: the power generated has to be equal to the power demand required. It is defined as: $\sum_{i=1}^N P_i^t - P_D^t - P_L^t = 0$, where $t = 1, 2, \dots, T$. P_D^t is the power demand at time t , and P_L^t is the transmission power loss at time t (in MW). This value considers the transmission loss due to the geographical distribution of the power stations. In this

paper, to calculate this value we use Kron's formula which represents the losses as a function of the output level of the system generators and it uses some B-matrix loss coefficients. The general form of the loss formula using B-coefficients is: $P_L^t = \sum_{i=1}^N \sum_{j=1}^N P_i^t B_{ij} P_j^t + \sum_{i=1}^N B_{0i} P_i^t + B_{00}$. If transmission power loss is not considered, $P_L^t = 0$.

2. Operating Limit Constraints: units have physical limits regarding the minimum and maximum power they can generate: $P_{min_i} \leq P_i^t \leq P_{max_i}$, where P_{min_i} and P_{max_i} are the minimum and maximum power output of the i^{th} unit in MW, respectively.
3. Ramp Rate Limits: they restrict the operating range of all on-line units. Such limits indicate how quickly the unit's output can be changed: $P_j^t - P_j^{(t-1)} \leq UR_j$ if $P_j^t > P_j^{(t-1)}$ and $P_j^{(t-1)} - P_j^t \leq DR_j$ if $P_j^t < P_j^{(t-1)}$, where $P_j^{(t-1)}$ is the output power of j^{th} unit at a previous hour and UR_j and DR_j are the ramp-up and ramp-down limits of the j^{th} unit in MW, respectively. Due to ramp-rate constraints, equation from item 2. is replaced by: $max(P_{min_j}^t, P_j^{(t-1)} - DR_j) \leq P_j^t$ and $P_j^t \leq min(P_{max_j}^t, P_j^{(t-1)} + UR_j)$ such that

$$\begin{cases} P_{min_j}^t = max(P_{min_j}^t, P_j^{(t-1)} - DR_j) \\ P_{max_j}^t = min(P_{max_j}^t, P_j^{(t-1)} + UR_j) \end{cases} \quad (2)$$

4. Prohibited Operating Zones: they restrict the operation of the units due to steam valve operation conditions or to vibrations in the shaft bearing. Thus, a unit with prohibited operating zones has a discontinuous input-output power generation characteristic which gives rise to additional constraints on the unit operating range. They are: $P_{min_i} \leq P_i^t \leq PZ_{i,1}^L$ or $PZ_{i,k-1}^U \leq P_i^t \leq PZ_{i,k}^L$ or $PZ_{i,n_1}^U \leq P_i^t \leq P_{max_i}$, $k = 2, 3, \dots, n_i$, where n_i is the number of prohibited zones of the i^{th} unit, k is the index of the prohibited operating zones of the i^{th} unit. $PZ_{i,k}^L$ and $PZ_{i,k}^U$ are the lower and upper bounds of the k^{th} prohibited operating zones of unit i .

3 Literature Review

Artificial Intelligence (AI) techniques are appropriate to solve the DED problem because this is a real-world problem with several particular features that make it difficult to solve, since its nonlinear search space is nonsmooth, discontinuous and non-differentiable. In fact, if valve-point effects or prohibited zones are considered, then we are dealing with a nonconvex problem [18].

This section aims to highlight how the DED problem has been tackled using different AI techniques, rather than providing a comprehensive description of each of them. These methods include: evolutionary algorithms [18], differential evolution [7], particle swarm optimization [17], Harmony Search [13] and Artificial Immune Systems [5]. Additional techniques have been reported in [13] and [4]. Other iterative methods are reported in [6] and [12] which minimize T subproblems instead of an NT problem.

4 Our Proposed Algorithm

We propose here an adaptation from a previous algorithm presented in [1]. It is an artificial immune system designed to solve DED problems. It is based on the activation process that T cells suffer. This process is divided in two stages: proliferation and differentiation [11]. The proposed approach is called IA_DED (Immune Algorithm for Dynamic Economic Dispatch problem). It works on a cell population. Each cell is activated in order to find partial feasible solutions. Special receptors present on the cells surface, called T cell receptors (TCR) are used to represent the decision variables of the problem. In this case, each variable is a real value and it represents the output power of a thermal unit, so a *TCR* has N variables, for an N -unit power system.

The algorithm works in the following way (see Algorithm 1). First, the TCRs are randomly initialized within the limits of the units with real values (Step 1) (interval 1). Then, violation rate and objective function value are calculated for each cell (Step 2). Note that only if a cell is feasible, its objective function value is calculated. Next, the following steps are repeated T times (i.e. for each interval)(Step 4 to 23): while a predetermined number of objective function evaluations had not been reached and 5×10^7 iterations had not been performed, the cells are activated according to their feasibility (Step 6). Then, the best solution at time t is recorded (Step 9). The time (interval) is increased (Step 10) and new operational limits are updated according to Eq. (2) (Steps 11-14). Those units whose power outputs fall outside the new operational limits are replaced by random values from the new valid limits (Steps 15-21). Since the power outputs could change, the *TPs* (*TP* is the total power generated by a *TCR*) are updated and the cells are re-evaluated according to the new power demand (Step 23). Finally, (Step 25) the final solution is the union of the solutions found at times 1, 2 to T (*BEST*).

The proliferation process clones N times each cell and the differentiation process changes these clones so that they acquire specialized functional properties. The differentiation process to be applied depends on the feasibility cell.

– Differentiation for feasible cells

Based on a probability P_a , each unit exchanges part of its output power with another unit from the same cell. The idea is to take a value (called d) from a unit (say i) and add it to another unit (say j). The i^{th} and j^{th} units are modified according to: $cell.TCR_i = cell.TCR_i - d$ and $cell.TCR_j = cell.TCR_j + d$, where $d = U(0, P_c * \min(cell.TCR_i - P_{min_i}^t, P_{max_j}^t - cell.TCR_j))$, $U(w_1, w_2)$ refers to a real random number with a uniform distribution in the range (w_1, w_2) and P_c is a change factor ($P_c \in [0, 1]$). The best from among the clones and the original cell passes to the next iteration.

– Differentiation for infeasible cells

The number of decision variables to be changed is determined by a random number $U(1, N)$. Each variable to be changed is chosen in a random way and it is modified according to: $cell.TCR'_i = cell.TCR_i \pm m$,

where $cell.TCR_i$ and $cell.TCR'_i$ are the original and the mutated decision variables, respectively. $m = U(0, 1) * (cell.ECV + cell.ICC)$ (ECV is the equality constraint violation for TCR and ICC is the inequality constraints sum for TCR). In a random way, it is decided if m will be added or subtracted to $cell.TCR_i$. If the procedure cannot find a TCR'_i in the allowable range, then a random number with a uniform distribution is assigned to it ($cell.TCR'_i = U(cell.TCR_i, P_{max_i}^t)$ if m should be added or $cell.TCR'_i = U(P_{min_i}^t, cell.TCR_i)$, otherwise). If the clone is feasible, then the differentiation process stops. Otherwise, the process is applied to the clone instead of the infeasible original cell. This methodology is repeated until N differentiations have been applied or a feasible clone had been reached.

ECV is calculated as at time t , for each cell j , its ECV_j is calculated as $ECV_j = | \sum_{i=1}^N TCR_i^t - P_D^t - TCR_L^t |$, where TCR_i^t , P_D^t and TCR_L^t are the output power for unit i , the load demand and the loss transmission, respectively. This rate indicates how far is the generated power from the demanded power. Thus, if $ECV_j > 0$ then the generated power by cell j is larger than the demanded power and if $ECV < 0$, the power generated by cell j is lower than the required power. ICC is calculated as $\sum_{i=1}^N \sum_{j=1}^{n_i} poz(TCR_i, i, j)$

$$poz(p, i, j) = \begin{cases} \min(p - PZ_{i,j}^L, PZ_{i,j}^U - p) & \text{if } p \in [PZ_{i,j}^L, PZ_{i,j}^U] \\ 0 & \text{otherwise} \end{cases}$$

where n_i is the number of prohibited operating zones and $[PZ_{i,j}^L, PZ_{i,j}^U]$ is the j^{th} prohibited range for the i^{th} unit. A cell is considered as feasible if: 1) $ECV = 0$ for problems without transmission network loss and $0 \leq ECV < \epsilon$ for problems with transmission loss and 2) $ICC = 0$ for problems which consider prohibited operating zones.

5 Numerical Experiments

The proposed algorithm was tested on five 24-h dynamic power systems (T=24). The first example is a 6-unit system. Its data and daily load demands were taken from [16]. The second system has 10 thermal units (10-unit system). The data and daily load demands for this problem were taken from [8]. An extension from this is the 30-unit system. It has the same cost characteristics from the last one. The load pattern is taken as three times the value which is considered in the 10 unit system for a 24hrs time period. The fourth power system has 15 generating units (15-unit system). The data and daily load demands for this problem were taken from [10]. The last test case is a 54-unit system [8]. The detailed data of this system were taken from [14], [15] and [10].

The algorithm was implemented in Java (v. 1.6.0_24) under Linux (UBUNTU 12.04) on a Pentium IV personal Computer while the experiments were performed on an Intel Q9550 Quad Core processor running at 2.83GHz and with 4GB DDR3 1333Mz in RAM. For each problem, 100 independent runs were performed.

Algorithm 1 IA_DED Algorithm

```

1:  $C \leftarrow$  Initialize_Population();
2: Evaluate( $C$ );
3: for  $t \leftarrow 1$  to  $T$  do
4:    $top \leftarrow 0$ ;
5:   while A number of evaluations has not been reached and  $top < 5 * 10^7$  do
6:     Activation_Process( $C$ );
7:      $top ++$ ;
8:   end while
9:    $best_t \leftarrow$  Search_best_at_Population( $C$ );
10:   $t ++$ ;
11:  for  $j \leftarrow 1$  to  $N$  do
12:     $P_{min_j}^t = \max(P_{min_j}, best_{t-1} - DR_j)$ 
13:     $P_{max_j}^t = \min(P_{max_j}, best_{t-1} + UR_j)$ 
14:  end for
15:  for  $i \leftarrow 1$  to  $|C|$  do
16:    for  $j \leftarrow 1$  to  $N$  do
17:      if  $cell_i.TCR_j \notin [P_{min_j}^t, P_{max_j}^t]$  then
18:         $cell_i.TCR_j \leftarrow U(P_{min_j}^t, P_{max_j}^t)$ 
19:      end if
20:    end for
21:  end for
22:  Update_output_power( $C$ );
23:  Evaluate( $C$ );
24: end for
25:  $BEST \leftarrow (best_1, best_2, \dots, best_T)$ ;

```

5.1 Statistical Analysis

The parameters required by IA_DED are: population size (C), maximum number of objective function evaluations, change factor (P_c), differentiation probability (P_a) and tolerance factor (ϵ). This last parameter was set to 0.9 for all the test problems that consider transmission losses. To analyze the effect of C , P_c and P_a on IA_DED's behavior, we tested it with different parameters settings. As part of this process, some preliminary experiments were performed to discard some parameter values. Hence, the selected parameter levels were the following: a) population size (C): 5, 10 and 20 cells, b) probability P_c : 0.1, 0.5 and 0.9 and c) probability P_a : 0.01 and 0.1.

As the results do not follow a normal distribution, we applied the Kruskal-Wallis test, to perform ANOVA and then the Turkey method in order to determine the experimental conditions for which significant differences exist. After the statistical analysis of the results obtained by our proposed, for the five test problems, we can infer the following general conclusions. For the 15-unit system, there are no significant differences when C is fixed and the probabilities vary. However, the median values improve with a small change factor. For the 6-unit system, when C is increased, better results are obtained and they have significant differences. Increasing the change factor from 0.1 to 0.5 and 0.9 improves the results with significant differences. For the 10-unit system, increasing the change factor from 0.1 to 0.5 and 0.9 improves the results with significant differences. When $C = 5$ or $C = 10$, increasing P_c from 0.5 to 0.9, also improves the results. In general, best median values are obtained with the highest probability set for the application of the differentiation operator. For the 30-unit system, increasing

the change factor improves the results with significant differences. Contrary to the previous case, the best median values are obtained with the lowest probability established for the application of the differentiation operator. Considering the 54-unit system, for $C = 5$, increasing the change factor from 0.1 to 0.5 and 0.9 produces better results and they present significant differences. For $C = 10$ or $C = 20$, increasing the probabilities produces better results.

5.2 Comparison of Results and Discussion

Table 1 provides the most relevant features of the problems previously described as well as the maximum number of function evaluations performed by IA_DED.

Table 1. Test Problems Features

Problem	Objective	P_L	POZ	Total Load Demand (MW)	MaxEv	C	P_c	P_a
6-unit system	smooth	Yes	Yes	25954	2000	20	0.5	0.1
10-unit system	non-smooth	No	No	40108	5000	10	0.9	0.1
15-unit system	smooth	Yes	No	60981	30000	20	0.9	0.1
30-unit system	non-smooth	No	No	120324	50000	5	0.9	0.1
54-unit system	non-smooth	No	Yes	111600	40000	5	0.9	0.01

Several methods are selected to be compared with our proposed algorithm, according to the chosen test cases. Our comparison of results is presented in Table 2. It shows the best, mean, worst, standard deviation as well as the running times obtained by the approaches, when available (integer costs are shown but they are not rounded up). For all the test problems, IA_DED found feasible solutions in all the runs performed, considering the parameters settings given in Table 1, except for the 10-unit system where feasible solutions were found in 86% of the runs. The running times are compared in an indirect manner, to give a rough idea of the computational costs of the different algorithms considered in our comparative study.

Analyzing Table 2, for the 6-unit system, IA_DED exceeds by \$104 the cost found by SAMF [3], but our approach obtained this best total fuel cost just in 0.924 seconds while SAMF [3] required 1.965 seconds. For the 10-unit system, IA_DED exceeds by \$1397 the cost found by EBSO [9], but this approach reports a running time of 0.205 minutes, i.e., 12.3 seconds. The other approaches took times measured in minutes to find feasible solutions, whereas our proposed approach took only 2.552 seconds. Considering the 15-unit system IA_DED outperformed all considered approaches. It finds a solution whose total fuel cost is \$759302 in 2.660 seconds. Thus, our proposed approach found the best solution requiring the lowest running time. However, the Brent-Method [6] found an acceptable solution in only 0.53 seconds. For the 30-unit system, IA_DED obtained a best total fuel cost of \$3056592, outperforming all the approaches with respect to which it was compared, except for EBSO [9]. However, EBSO produced a solution which required 0.95 minutes (57 seconds) to obtain a solution which is only 0.08% cheaper than the one produced by IA_DED, but it required 634%

more time than IA_DED. For the 54-unit system, IA_DED outperformed all the other approaches with respect to which it was compared, in terms of the total fuel cost. IA_DED just required 13.169 seconds to find this solution, whose cost is \$1717901. In this case, OCD [12], found a feasible solution which is 3% more expensive than the one produced by IA_DED but it produced it in only 0.132 seconds. It is worth noting that the methods considered in this paper, which subdivide the whole dispatch into T periods such as the Brent Method [6], SAMF [3, 12], and IA_DED, are able to find high-quality solutions in seconds rather than minutes.

6 Conclusions and Future Work

This paper presented an algorithm inspired on the T-Cells of the immune system, IA_DED, which was used to solve dynamic economic dispatch problems. IA_DED is able to handle the different types of constraints that are involved in this type of problem: power balance constraints with and without transmission loss, operating limit constraints, ramp rate limit constraints and prohibited operating zones. Additionally, it can handle both smooth and non-smooth functions.

At the beginning, the search performed by IA_DED is based on a simple differentiation operator which takes an infeasible solution and modifies some of its decision variables by taking into account their constraint violation. Once the algorithm finds a feasible solution, a different differentiation operator is applied. This operator modifies two decision variables at a time, it decreases the power in one unit, and it selects other unit to generate the power that has been taken.

Our proposed approach was validated with five test problems having different features. Comparisons were provided with respect to several approaches that have been reported in the specialized literature. Our proposed approach produced competitive results in all cases, being able to outperform some of the other approaches when running times are considered. The best performance of our proposed algorithm is observed in the largest systems with which it was tested. Besides, best results are obtained when the highest change factor probability is used. As part of our future work, we are interested in testing the algorithm with even larger systems and we intend to incorporate renewable energy resources.

References

1. Victoria S. Aragón and Susana C. Esquivel. Immune algorithm for solving the dynamic economic dispatch problem. In *XXII Congreso Argentino de Ciencias de la Computación*. Universidad Nacional de San Luis, 2016.
2. R. Arul, G. Ravi, and S. Velusami. Chaotic self-adaptive differential harmony search algorithm based dynamic economic dispatch. *International Journal of Electrical Power & Energy System*, 50:85–96, 2013.
3. S. Ganesan and S. Subramani. Dynamic Economic Dispatch Based on Simple Algorithm. *International Journal of Computer and Electrical Engineering*, 3(2):1793–8163, 2011.

Table 2. Comparison of results. The best values are shown in **boldface**. The last column indicates the running time (s \equiv seconds and m \equiv minutes). - denotes that the value was not available.

Problem/ Algorithm	Best(\$)	Mean(\$)	Worst(\$)	Std.	Time
6-unit system					
SAMF [3]	313363	-	-	-	1.965s
Brent Method[6]	313405	-	-	-	0.078s
BPSO-DE [16]	314025	314144	314351	-	21.89s
IA_DED	313467	313497	313534	14.58	0.924s
10-unit system					
EBSO [9]	1017147	1017526	1017891	147.01	0.205m
ICA [8]	1018467	1019291	1021795	-	-
CSADHS [2]	1018681	1018718	1018760	-	2.72m
ICPSO [17]	1019072	1020027	-	-	0.467m
HHS [13]	1019091	-	-	-	12.233m
CDE method3 [7]	1019123	1020870	1023115	-	0.32m
DHS [2]	1019952	1020025	1020107	-	3.34m
AIS [5]	1021980	1023156	1024973	-	19.01m
IA_DED	1018544	1020193	1022064	764.04	2.552s
15-unit system					
SAMF [3]	759406	-	-	-	2.951s
NPAHS [10]	759603	759779	759988	-	250.0s
CSADHS [2]	759689	759766	759845	-	3.36m
SGHS[10]	759897	760118	760343	-	303.3s
Brent Method[6]	760287	-	-	-	0.53s
IA_DED	759302	759542	760125	149.59	2.660s
30-unit system					
HHS [13]	3057313	-	-	-	27.65m
ICPSO [17]	3064497	3071588	-	-	1.03m
CDE method3 [7]	3083930	3090542	-	-	0.67m
EBSO [9]	3054001	3054697	3055944	-	0.95m
IA_DED	3056592	3060513	3064397	1545.83	7.756s
54-unit system					
OCD [12]	1772724	-	-	-	0.132s
ICA [8]	1807081	1809664	1811388	-	-
IA_DED	1717901	1718127	1718411	108.08	13.169s

4. H. Hardiansyah. A modified ABC algorithm for solving non-convex dynamic economic dispatch problems. *International Journal of Electrical and Computer Engineering (IJECE)*, 6(6):2621–2628, December 2016.
5. S. Hemamalini and S. Simon. Dynamic economic dispatch using artificial immune system for units with valve-point effect. *International Journal of Electrical Power & Energy System*, 33:868–874, 2011.
6. M. Sydulu K. Chandram, N. Subrahmany. Brent method for dynamic economic dispatch with transmission losses. *Iranian Journal Electrical and Computer Engineering*, 8(1):16–22, 2009.
7. Youlin Lu, Jianzhong Zhou, Hui Qin, Ying Wang, and Yongchuan Zhang. Chaotic differential evolution methods for dynamic economic dispatch with valve-point effects. *Engineering Applications of Artificial Intelligence*, 24(2):378 – 387, 2011.
8. Behnam Mohammadi-ivatloo, Abbas Rabiee, Alireza Soroudi, and Mehdi Ehsan. Imperialist competitive algorithm for solving non-convex dynamic economic power dispatch. *Energy*, 44(1):228 – 240, 2012. Integration and Energy System Engineering, European Symposium on Computer-Aided Process Engineering 2011.
9. T. Niknam and F. Golestaneh. Enhanced bee swarm optimization algorithm for dynamic economic dispatch. *IEEE Systems Journal*, 7(4):754–762, Dec 2013.
10. Qun Niu, Hongyun Zhang, Kang Li, and George W. Irwin. An efficient harmony search with new pitch adjustment for dynamic economic dispatch. *Energy*, 65:2543, 2014.
11. Leandro Nunes de Castro and Jonathan Timmis. *Artificial Immune Systems: A New Computational Intelligence Approach*. Springer-Verlag, New York, 2002.
12. Abbas Rabiee, Behnam Mohammadi-Ivatloo, and Mohammad Moradi-Dalvand. Fast Dynamic Economic Power Dispatch Problems Solution Via Optimality Condition Decomposition. *IEEE Transactions on Power Systems*, 29(2):982–983, 2014.
13. Pandi V. Ravikumar and Bijaya Ketan Panigrahi. Dynamic economic load dispatch using hybrid swarm intelligence based harmony search algorithm. *Expert Systems with Applications*, 38:8509 – 8514, 2011.
14. M. Shahidehpour. Available from: motor.ece.iit.edu/data/SCUC_118test.xls [accessed 2015].
15. M. Shahidehpour. Available from: motor.ece.iit.edu/data/118_nonsmooth.xls [accessed 2015].
16. K. Vaisakh, P. Praveena, S. Rama Mohana Rao, and Kala Meah. Solving dynamic economic dispatch problem with security constraints using bacterial foraging PSO-DE algorithm. *International Journal of Electrical Power & Energy Systems*, 39(1):56–67, 2012.
17. Wang Ying, Zhou Jianzhong, Qin Hui, and Youlin Lu. Improved chaotic particle swarm optimization algorithm for dynamic economic dispatch problem with valve-point effects. *Energy Convers Manage*, 51:2893 – 900, 2010.
18. Forhad Zaman, Saber M. Elsayed, Tapabrata Ray, and Ruhul A. Sarker. Configuring two-algorithm-based evolutionary approach for solving dynamic economic dispatch problems. *Engineering Applications of Artificial Intelligence*, 53:105–125, 2016.

Aplicación de Regresión con Vectores de Soporte en un Sistema Recomendador de Actividades Sociales

Julieta Ríos¹, Gonzalo Ulla¹ y Agustín Borello Gianni¹

¹Inteligencia Artificial, Departamento de Ingeniería en Sistemas de Información, Universidad Tecnológica Nacional, Facultad Regional Córdoba, Maestro M. López esq. Cruz Roja Argentina, Córdoba, Argentina {julirios299, gonzauilla, agunstinborello21}@gmail.com

Resumen. Este trabajo desarrolla la aplicación de una técnica de aprendizaje automático, denominada Regresión con Vectores de Soporte en un Sistema Recomendador. Este último forma parte de una plataforma online con fines sociales que vincula organizaciones sin fines de lucro con voluntarios y con empresas. Dicha plataforma, denominada Helpo, brinda soporte a la planificación táctica de las actividades sociales gestionadas por organizaciones del tercer sector. Para ello, emplea el algoritmo SVR de la librería Scikit-learn, escrita en Python, en pos de recomendar a las organizaciones cuándo resulta conveniente realizar sus eventos y campañas. El algoritmo mencionado utiliza datos tanto de la organización y actividad a registrar, como del resto de entidades existentes en la plataforma, y recurre a técnicas complementarias como Feature Scaling y Grid Search. De esta forma, el presente trabajo responde a una necesidad social de forma innovadora: maximizando la probabilidad de que actividades sociales tengan éxito al sugerir cuándo ejecutarlas mediante un algoritmo de aprendizaje automático.

Palabras Clave: Regresión con Vectores de Soporte, SVR, Máquina de Soporte Vectorial, SVM, Inteligencia Artificial, Machine Learning, Aprendizaje Automático, actividades sociales, tercer sector.

1 Introducción

La inversión mundial en tecnologías de Inteligencia Artificial (IA) aumentó 59,1% en 2017 con respecto al año anterior, según un estudio de Accenture referido por La Nación [1]. Argentina no es la excepción: por lo menos desde 2014 se evidencian experiencias en el uso de IA que aplican estas tecnologías en instituciones del sector público, en empresas y en organizaciones del tercer sector [2]. Esto demuestra que el creciente interés por esta disciplina no es de naturaleza exclusivamente académica, sino también profesional y social.

A su vez, la cantidad de datos que manipulan actualmente las organizaciones de todo el mundo crece de manera exponencial. Analizar estos datos, caracterizados por su alta complejidad y diversidad, permite obtener información valiosa de ellos que coloca, a quien la posee, en una posición de ventaja competitiva.

Es de esperar que, en función de lo mencionado, la demanda de soluciones de software que apliquen IA crezca. En efecto, analistas predicen que prácticamente todos los nuevos productos y servicios de software implementarán tecnologías de Inteligencia Artificial para 2020 [3].

Precisamente, en este trabajo se describirá un caso de aplicación en el cual se ha implementado una técnica propia de dicha disciplina, basada en aprendizaje automático, como parte del desarrollo de un sistema web y móvil orientado a vincular organizaciones sin fines de lucro con voluntarios y con empresas. Este software, denominado “Helpo”, adopta un modelo de plataforma online multilateral y busca propiciar la realización efectiva de actividades sociales relacionando los tres actores enumerados.

La formulación del problema abordado por el presente trabajo de investigación se establece de la siguiente manera: ¿cómo se puede aplicar una técnica de aprendizaje automático en una plataforma online con fines sociales?

Se espera, de la solución alcanzada por esta investigación, que se pueda responder al problema planteado, aplicando, de forma práctica, un algoritmo de aprendizaje automático. En pos de cumplir con este objetivo, se comenzará desarrollando el fundamento teórico necesario que sustenta todo este trabajo. Luego, se analizará el problema a abordar y se explicará la solución dada al mismo. Por último, se estudiará el comportamiento de esta solución y trabajos relacionados al tema de investigación, concluyendo al respecto.

2 Fundamento Teórico

El aprendizaje automático, conocido en inglés como Machine Learning [4], es una disciplina del ámbito de la Inteligencia Artificial cuya finalidad es desarrollar técnicas para permitir que los sistemas aprendan y resuelvan problemas cotidianos por sí mismos. Aprender, por un lado, implica aumentar el conocimiento y mejorar las capacidades y habilidades de actuación en un entorno, mediante la identificación de patrones de comportamiento en millones de datos. Hacerlo de forma automática, por otra parte, hace referencia a la mejora de los sistemas en forma autónoma a lo largo del tiempo.

En este contexto, un sistema que aprende de forma automática es un artefacto (o conjunto de algoritmos) que, para resolver problemas, toma decisiones basadas en la experiencia acumulada para mejorar su actuación [5].

Para que estos sistemas sean capaces de resolver problemas, deben hacerlo a partir de la selección y adaptación del conocimiento que van adquiriendo [6]. En la fase de selección, el sistema elige las características más relevantes de un objeto y las compara con otras conocidas (si existen) a través de algún método de cotejamiento. En la fase de adaptación, el sistema amolda su modelo de aquel objeto según el resultado del cotejamiento.

Según el tipo de selección y adaptación que un sistema realiza sobre la información disponible es factible identificar diversos paradigmas del aprendizaje automático [7]:

- Aprendizaje supervisado: el algoritmo produce una función que establece una correspondencia entre las entradas y las salidas que se desean del sistema. Los datos

de entrada se encuentran etiquetados, es decir, se conocen los resultados esperados, los cuales se utilizan para corregir las salidas del sistema.

- Aprendizaje no supervisado: en estos algoritmos, el proceso de modelado se lleva a cabo sobre un conjunto de datos sin etiquetar. Al no conocer los resultados deseados, se debe deducir y descubrir la estructura interna presente en los datos de entrada.
- Aprendizaje semi-supervisado: este tipo de algoritmo combina los dos anteriores, generando una función deseada o clasificador a partir de datos tanto etiquetados como no etiquetados.
- Aprendizaje por refuerzo: el algoritmo aprende observando aquello que lo rodea. Su entrada es la retroalimentación que obtiene del mundo exterior en respuesta a sus acciones.

Centrando la teoría en algoritmos de aprendizaje supervisado, se hallan, dentro de dicha clasificación, las Máquinas de Soporte Vectorial [8] (SVM, del inglés Support Vector Machines). Las SVMs buscan seleccionar un hiperplano de separación que equidista de los ejemplos (muestras o vectores de entrada) más cercanos de cada clase para conseguir lo que se denomina margen máximo a cada lado del hiperplano. Al momento de definir este último, sólo se consideran los ejemplos de entrenamiento de cada clase que caen justo en la frontera de dichos márgenes. Esos ejemplos de entrenamiento reciben el nombre de vectores de soporte. Así, un algoritmo de SVM construye un modelo capaz de predecir si un punto nuevo pertenece a una u otra clase.

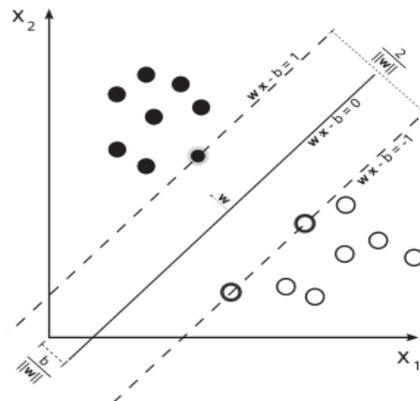


Fig. 1. Ejemplo de dos clases divididas por un hiperplano equidistante.

Las Máquinas de Vectores Soporte adaptadas para resolver problemas de regresión se conocen por el acrónimo SVR [10] (del inglés Support Vector Regression). Así, dado un conjunto de ejemplos de entrenamiento:

$$S = \{(x_1, y_1), \dots, (x_n, y_n)\}$$

en el que se asume que todos los valores y_i de todos los ejemplos de S se pueden ajustar (o cuasi-ajustar) mediante una función lineal, el objetivo de la tarea de regresión

es encontrar los parámetros $w = (w_1, \dots, w_n)$ que permitan definir dicha función lineal, de la forma:

$$f(x) = (w_1x_1 + \dots + w_dx_d) + b$$

En el caso de que los ejemplos no puedan ajustarse por una función lineal, se recurre a la metodología denominada Kernelización [9, 10]. Frente a este concepto, los ejemplos pertenecientes al espacio original de entradas se transforman en un nuevo espacio en el que sí es posible ajustar los ejemplos transformados mediante un regresor lineal. El tipo de transformación dependerá del tipo de función kernel utilizado, pudiendo ser Polinomial-homogénea, Perceptrón, Sigmoidea, Función de Base Radial (RBF), entre otras.

3 Descripción del problema resuelto

Helpo surge como una plataforma online orientada a vincular organizaciones sin fines de lucro con voluntarios y con empresas mediante un sistema web y móvil. Su objetivo es incrementar el flujo de recursos de estas organizaciones, brindando oportunidades para que voluntarios y empresas colaboren en actividades sociales, ya sea donando o participando de eventos y campañas. Así, Helpo da respuesta, de forma innovadora, a una necesidad vigente en la actualidad.

En este contexto, surge la necesidad de implementar algoritmos de Machine Learning que permitan a las organizaciones sin fines de lucro planificar sus actividades. Es decir, conocer en qué mes del año es conveniente organizar un evento o una campaña en base a datos analizados de actividades similares de todas las organizaciones previamente registradas en la plataforma. La sugerencia de organizar una actividad social en determinados meses incrementará la probabilidad de que la misma tenga éxito, lo que significa que la ONG satisfaga un mayor número de necesidades, tanto materiales como de recursos humanos.

4 Explicación de la solución

Para abordar la solución a la necesidad detectada, se hará uso de Scikit-learn [11], una librería de aprendizaje automático de Python, lenguaje de programación utilizado en el desarrollo del sistema Helpo.

El algoritmo SVR [12], presente en la librería, es el empleado en el desarrollo de la presente solución. Dicho algoritmo presenta una implementación del método de aprendizaje automático “Regresión con Vectores de Soporte” y, en base a lo explicado en el fundamento teórico anterior, produce un modelo que depende sólo de un subconjunto de ejemplos de entrenamiento, dado que la función empleada para construir el mismo no se preocupa por los puntos de entrenamiento que se encuentran más allá del margen.

El algoritmo SVR de la librería Scikit-learn toma un conjunto de parámetros de entrada necesarios para producir el modelo adecuadamente, entre los que se encuentran el tipo de kernel empleado -por defecto el kernel RBF o Función de Base Radial [13]-, el coeficiente gamma del kernel seleccionado y el parámetro de penalización C del término del error.

Previo a entrenar los vectores con el algoritmo seleccionado, se pre-procesan los datos haciendo “Feature Scaling” [14] de los mismos, es decir, escalando las características para ubicarlas entre un valor mínimo y máximo dado o para que el valor absoluto máximo de cada característica se ajuste al tamaño de la unidad.

Además, se utiliza una búsqueda de cuadrícula o “Grid Search” [15] para definir adecuadamente los hiper parámetros del algoritmo SVR, previamente mencionados. La búsqueda de cuadrícula genera exhaustivamente candidatos a partir de una cuadrícula de valores de parámetros especificados. Grid Search evalúa todas las combinaciones posibles de valores de parámetros y conserva la mejor combinación para, luego, emplear el algoritmo entrenador con los mejores parámetros.

```

y = pd.DataFrame()
y["pred"] = Mtrain["%Comp"]
training_data = Mtrain.drop(["%Comp"], axis=1)

rmse_error = make_scorer(mean_squared_error, greater_is_better=False)

parameters = {
    'C': [0.8, 0.9, 1],
    'epsilon': [0.04, 0.05, 0.06],
    'gamma': [0.001, 0.003, 0.005, 0.008]
}

svr = GridSearchCV(SVR(), cv=3, param_grid=parameters, scoring=rmse_error)

svr.fit(training_data, y)
print(svr.best_estimator_)

SVR(C=1, cache_size=200, coef0=0.0, degree=3, epsilon=0.06, gamma=0.008,
    kernel='rbf', max_iter=-1, shrinking=True, tol=0.001, verbose=False)

```

Fig. 2. Porción de código en la que se aplica el método “Grid Search”, obteniendo los mejores valores posibles para C, épsilon y gamma, y se entrena el modelo.

A continuación, se entrena el modelo usando dos conjuntos -de entrenamiento y de prueba-. Para el presente trabajo, los conjuntos mencionados surgieron de datos consistentes y reales brindados por tres ONGs relevadas. Estas organizaciones proporcionaron información histórica acerca de sus actividades sociales y, a partir de ellas, el equipo de trabajo extrapoló los datos para generar nueva información, con la finalidad de lograr una base de datos suficiente y completa para entrenar y probar el modelo.

Finalmente, se evalúa el error del modelo con el estadígrafo RMSE [16] (del inglés Root Mean Square Error) que compara los valores predichos con los valores observados reales.

Las características que utilizará el algoritmo para ajustar el modelo son:

- Mes de las actividades previas registradas.
- Porcentaje promedio de completitud de necesidades materiales de las actividades previas registradas de la organización que desea planificar su nueva actividad.
- Porcentaje promedio de completitud de necesidades materiales de actividades previas registradas por otras organizaciones del mismo rubro.
- Porcentaje promedio de completitud de necesidades materiales de actividades previas registradas del mismo rubro que la nueva actividad que se desea planificar.
- Porcentaje promedio de completitud de necesidades de voluntarios de las actividades previas registradas de la organización que desea planificar su nueva actividad.

- Porcentaje promedio de completitud de necesidades de voluntarios de actividades previas registradas por otras organizaciones del mismo rubro.
- Porcentaje promedio de completitud de necesidades de voluntarios de actividades previas registradas del mismo rubro que la nueva actividad que se desea planificar.
- Suscripciones de la organización que desea planificar la nueva actividad.
- Suscripciones de todas las organizaciones del mismo rubro que la organización que desea planificar la nueva actividad.
- Promedio de visitas por actividad de la organización.
- Promedio de visitas por actividad de organizaciones del mismo rubro que la organización que desea planificar la actividad.
- Promedio de visitas por actividad de actividades del mismo rubro que la que se desea planificar.
- N variables booleanas que representan si la necesidad material se va a incluir en la actividad.
- N variables booleanas que representan si la necesidad de voluntario se va a incluir en la actividad.

Dando respuesta a la necesidad del proyecto Helpo, la salida del algoritmo, es decir, la predicción, es el porcentaje de completitud de las colaboraciones y participaciones respecto a las necesidades que se solicitarán en la nueva actividad. Para conseguir la sugerencia del mejor mes, se realizarán doce predicciones (con meses distintos) y aquel mes con el que se obtenga el mayor porcentaje de completitud será el recomendado.

5 Análisis del comportamiento de la solución

En pos de demostrar e ilustrar la solución alcanzada, se plantea a continuación un ejemplo particular.

La organización sin fines de lucro de fantasía “Mundo Feliz” desea planificar cuándo realizar el evento “Maratón solidaria”. Para ello, deberá ingresar en el sistema Helpo los siguientes datos sobre dicha actividad futura:

- Tipo de actividad: en este caso un “Evento”.
- Rubro de la actividad: en este ejemplo, “Deportivo”.
- Ubicación de la actividad: en este caso “Plaza de la Intendencia”.
- Necesidades materiales que solicitar: particularmente para este evento, “Ropa” y “Alimentos”.
- Funciones de voluntarios a cubrir: en el ejemplo, “Policías” y “Fotógrafos”.

Teniendo en cuenta los datos ingresados y demás datos registrados previamente en el sistema tanto de la organización “Mundo Feliz” como de otras organizaciones y sus actividades, Helpo le recomendará a esta organización el mes adecuado para llevar a cabo el evento “Maratón solidaria”. Para ello, se recurrirá al algoritmo SVR de la librería Scikit-learn tal como se explicó anteriormente.

Luego de que la organización sin fines de lucro ingrese los datos de la actividad que desea planificar, el sistema Helpo, en primer lugar, recuperará el modelo entrenado almacenado, previamente, en Amazon Simple Storage Service (Amazon S3) [17].

El modelo en cuestión fue almacenado luego de reiterados ciclos de selección y adaptación. Más precisamente y como se mencionó anteriormente, se utilizó el estadígrafo RMSE para evaluar los errores obtenidos en los conjuntos de entrenamiento y prueba y obtener el modelo que mejor ajustara a la realidad, es decir, el que proporcionara el mínimo error posible. El modelo resultante arrojó un RMSE en datos de entrenamiento igual a 0,0063 y un RMSE en datos de prueba igual a 0,0118.

```

training_error = mean_squared_error(Mtrain['%Comp'],
                                     svr.predict(Mtrain.loc[:, M.columns != '%Comp']))
test_error = mean_squared_error(Mtest['%Comp'],
                                 svr.predict(Mtest.loc[:, Mtest.columns != '%Comp']))
print('RSME en datos de entrenamiento: ' + str(training_error))
print('RSME en datos de prueba: ' + str(test_error))

```

RSME en datos de entrenamiento: 0.006286952394657705
 RSME en datos de prueba: 0.01182923886669542

Fig. 3. Porción de código en la que se visualizan los errores obtenidos en los conjuntos empleados para ajustar el modelo.

Graficando el funcionamiento del modelo, tanto con el conjunto de entrenamiento como con el conjunto de prueba, se observa que el resultado es realmente óptimo. Los errores son relativamente bajos y se visualiza que las predicciones con el conjunto de prueba ajustan de forma muy cercana a la realidad.

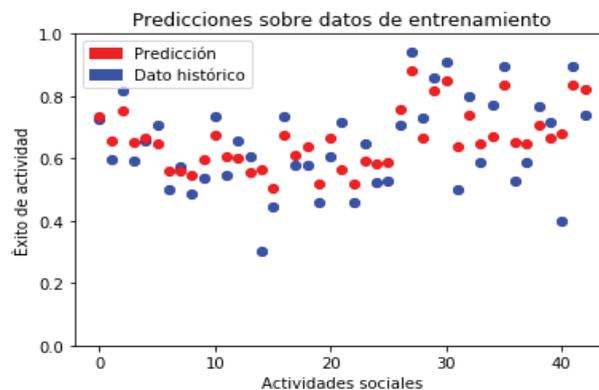


Fig. 4. Gráfico confeccionado con las predicciones sobre el conjunto de entrenamiento.

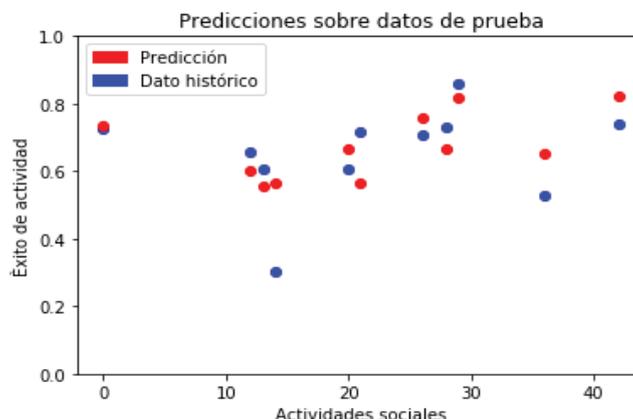


Fig. 5. Gráfico confeccionado con las predicciones sobre el conjunto de prueba.

Retomando el caso planteado, una vez levantado el modelo se procede a evaluar y calcular el porcentaje de completitud por mes de las colaboraciones y participaciones respecto a las necesidades que se solicitan en la actividad “Maratón Solidaria”. Realizando doce predicciones (con distintos meses), se recomienda el mes que obtenga el mayor porcentaje de completitud.

Finalmente, si la organización en cuestión opta por registrar su evento en el mes recomendado por Helpo, esto maximizará la probabilidad de cubrir sus necesidades materiales y de recursos humanos.

6 Trabajos relacionados

En primer lugar, en cuanto al sistema Helpo como producto de software, cabe destacar que no existen, a nivel local, plataformas con características similares. Tomando la totalidad de la República Argentina, es posible identificar soluciones con objetivos compartidos, pero ninguna con foco en Córdoba ni con los mismos actores y procesos de negocio involucrados.

Con respecto a líneas de investigación que aborden temáticas vinculadas a Inteligencia Artificial, Machine Learning y SVR, se han podido relevar trabajos de carácter eminentemente teórico oriundos tanto de Argentina como de la región latinoamericana [8, 9]. Sin embargo, no se han hallado aplicaciones prácticas relacionadas a estas disciplinas.

Ampliando el alcance de búsqueda a un nivel mundial, existen trabajos que denotan aplicaciones de estos conceptos con fines sociales -por ejemplo, para predecir desastres naturales [18]- o con fines de lucro -pronosticar series de tiempo en el rubro financiero [19] o estimar tiempos de viaje [20]-.

Por último, en relación a la utilización de IA y aprendizaje automático en organizaciones del tercer sector, es oportuno señalar que, si bien Argentina es el país de Latinoamérica con las startups de Inteligencia Artificial que más ingresos generan [21],

estos emprendimientos aún no arriban a dicho sector, lo cual representa una clara oportunidad.

7 Conclusión y Trabajos Futuros

Es cierto que, a nivel local, disciplinas y técnicas como Inteligencia Artificial y aprendizaje automático aún se encuentran en una fase de desarrollo incipiente, a pesar de estar en boga últimamente. Aun así, resulta imposible negar el elevado crecimiento que estos campos de las ciencias de la computación enfrentan, inclusive en Argentina.

El caso particular de aplicación detallado en este trabajo articula conocimientos teóricos propios de la disciplina de Inteligencia Artificial con una solución de software orientada a reunir organizaciones sin fines de lucro, voluntarios y empresas. De esta forma, se ha respondido a la pregunta de investigación propuesta inicialmente, detallando cómo aplicar una técnica de aprendizaje automático en una plataforma online con fines sociales. Para ello, se ha empleado el algoritmo SVR de la librería Scikit-learn, escrita en el lenguaje de programación Python.

El valor de negocio que aporta la utilización del algoritmo mencionado radica en recomendar cuándo resulta conveniente registrar una actividad social. Gracias a este trabajo Helpeo ahora responde a una necesidad adicional de sus usuarios: planificación táctica a medida.

Más allá del ámbito académico en el cual fue concebida esta solución, la misma responde a una necesidad social de forma innovadora: aplicando algoritmos de aprendizaje automático para maximizar la eficacia de actividades orientadas al bien común.

Referencias

1. Goldschmidt, O.: Innovación made in Argentina: la inteligencia artificial pide pista en el mercado local, *La Nación*, <https://www.lanacion.com.ar/2114594-innovacion-made-in-argentina-la-inteligencia-artificial-pide-pista-en-el-mercado-local>, último acceso: 1/05/2019.
2. Ingrassia, V.: Avanza el uso de inteligencia artificial en el sector público, privado y ONG argentinas, *Infobae*, <https://www.infobae.com/tendencias/innovacion/2018/04/01/avanza-el-uso-de-inteligencia-artificial-en-el-sector-publico-privado-y-ong-argentinas/>, último acceso: 1/05/2019.
3. Moore, S.: Gartner says AI technologies will be in almost every new software product by 2020, *Gartner*, <https://www.gartner.com/en/newsroom/press-releases/2017-07-18-gartner-says-ai-technologies-will-be-in-almost-every-new-software-product-by-2020>, último acceso: 1/05/2019.
4. González, A.: ¿Qué es Machine Learning?, *CleverData*, <https://cleverdata.io/que-es-machine-learning-big-data/>, último acceso: 1/05/2019.
5. Aggarwal, C.: *Data Classification: Algorithms and Applications*, CRC Press, (2015).
6. Moreno, A., Armengol, E., Béjar, J., Belanche, L., Cortés, U., Gavaldá, R., Gimeno, J.M., López, B., Martín, M., Sánchez, M.: *Aprendizaje Automático*, Edicions UPC, (1994).
7. Russo, C.: *Tratamiento masivo de datos utilizando técnicas de Machine Learning*, WICC, *Entre Ríos* (2016).

8. Betancourt, G.: Las máquinas de soporte vectorial (SVMs), *Scientia et Technica*, Año XI, No 27, (2005).
9. Carmona Suárez, E.: Tutorial sobre Máquinas de Vectores Soporte (SVM), *Dpto. de Inteligencia Artificial, ETS de Ingeniería Informática, UNED*, (2014), [http://www.ia.uned.es/~ejcarmona/publicaciones/\[2013-Carmona\]%20SVM.pdf](http://www.ia.uned.es/~ejcarmona/publicaciones/[2013-Carmona]%20SVM.pdf), último acceso: 3/05/2019.
10. Kernelización: https://es.wikipedia.org/wiki/M%C3%A1quinas_de_vectores_de_soporte#SVR_Regresi%C3%B3n, último acceso: 3/05/2019.
11. Muller, A., Guido, S.: *Introduction to Machine Learning with Python*, O'Reilly, (2016).
12. AlgoritmoSVR: <http://scikit-learn.org/stable/modules/generated/sklearn.svm.SVR.html#sklearn.svm.SVR>, último acceso: 3/05/2019.
13. Kernel RBF: https://en.wikipedia.org/wiki/Radial_basis_function_kernel, último acceso: 3/05/2019.
14. Feature Scaling: <http://scikit-learn.org/stable/modules/preprocessing.html>, último acceso: 3/05/2019.
15. Grid Search: http://scikit-learn.org/stable/modules/grid_search.html, último acceso: 3/05/2019.
16. RMSE: https://en.wikipedia.org/wiki/Root-mean-square_deviation, último acceso: 3/05/2019.
17. Amazon Simple Storage Service: <https://aws.amazon.com/es/s3/>, último acceso: 3/05/2019.
18. Tien Bui, D.: *Spatial prediction of rainfall-induced landslides for the Lao Cai area*, Springer, Berlin Heidelberg (2016).
19. Tay, F., Cao, L.: *Application of support vector machines in financial time series forecasting*, *Department of Mechanical Engineering, National University of Singapore* (2001).
20. Wu, C.: *Travel-time prediction with support vector regression*, *IEEE Transactions on Intelligent Transportation Systems*, Volume: 5, Issue: 4, pp. 276-281, (2004).
21. Argentina, el país con las startups de Inteligencia Artificial más grandes en la región, *Ámbito Biz*, <http://www.ambito.com/935630-argentina-el-pais-con-las-startups-de-inteligencia-artificial-mas-grandes-en-la-region>, último acceso: 3/05/2019.

Automatically Assessing the Need of Additional Citations for Information Quality Verification in Wikipedia Articles

Gerónimo Bazán Pereyra¹, Carolina Cuello¹, Gianfranco Capodici¹, Vanessa Jofré¹, Edgardo Ferretti^{1,2}, and Marcelo Errecalde^{1,2}

¹ Universidad Nacional de San Luis (UNSL), San Luis - Argentina

² Laboratorio de Investigación y Desarrollo en Inteligencia Computacional (UNSL)
e-mails: {ferretti,merreca}@unsl.edu.ar

Abstract. Quality flaws prediction in Wikipedia is an ongoing research trend. In particular, in this work we tackle the problem of automatically assessing the need of including additional citations for contributing to verify the articles' content; the so-called *Refimprove* quality flaw. This information quality flaw, ranks among the five most frequent flaws and represents 12.4% of the flawed articles in the English Wikipedia. Under-bagged decision trees, biased-SVM, and centroid-based balanced SVM –three different state-of-the-art approaches– were evaluated, with the aim of handling the existing imbalances between the number of articles' tagged as flawed content, and the remaining untagged documents that exist in Wikipedia, which can help in the learning stage of the algorithms. Also, a uniformly sampled balanced SVM classifier was evaluated as a baseline. The results showed that under-bagged decision trees with the *min* rule as aggregation method, perform best achieving an F_1 score of 0.96 on the test corpus from the 1st *International Competition on Quality Flaw Prediction in Wikipedia*; a well-known uniform evaluation corpus from this research field. Likewise, biased-SVM also achieved an F_1 score that outperform previously published results.

Keywords: Wikipedia, Information Quality, Quality Flaws Prediction, Refimprove Flaw

1 Introduction

The online encyclopedia Wikipedia is one of the largest and most popular user-generated knowledge sources on the Web. Considering its size and dynamic nature, a comprehensive manual quality assurance of information is infeasible. A widely accepted interpretation of Information Quality (IQ) is the “fitness for use in a practical application” [1], i.e. the assessment of IQ requires the consideration of context and use case. Particularly, in Wikipedia the context is well-defined by the encyclopedic genre, that forms the ground for Wikipedia's IQ ideal, within the so-called *featured article criteria*.³ Having a formal definition of what constitutes a high-quality article, i.e. a featured article (FA), is a key issue; however,

³ http://en.wikipedia.org/wiki/Wikipedia:Featured_article_criteria

as indicated in [2], in 2012 less than 0.1% of the English Wikipedia articles were labeled as featured. At present, this ratio still remains, since there are 5 568 featured articles out of 5 887 173 articles on the English Wikipedia.⁴

Information quality assessment in Wikipedia has become an ever-growing research line in the last years. In the literature, a variety of approaches have been proposed to automatically assess different quality aspects in Wikipedia, such as: (i) featured articles identification [3, 4]; (ii) development of quality measurement metrics [5, 6]; (iii) vandalism detection [7, 8] and (iv) quality flaws detection [9–14], among others. In this paper we will concentrate on the last research trend mentioned above. In particular we will tackle the problem of automatically detecting articles that in spite of having references, they are not enough to verify the content they exhibit. Verifiability of the articles' content is a primary concern and according to the study presented in [10], this information quality flaw, so-called *Refimprove*, ranks among the five most frequent flaws and represents 12.4% of the flawed articles in the English Wikipedia.

Although originally stated as a one-class classification problem [9], the study of this flaw prediction has been carried out by using machine learning algorithms belonging to supervised and semi-supervised learning domains [11–14]. In particular, given the recent results presented for this flaw for the Spanish Wikipedia [14], in the paper at hand we will evaluate the three best performing methods from [14]; namely: centroid-based balanced SVM, biased-SVM and under-bagged decision trees (with different voting rules) in the English version of Wikipedia, since they have not been previously evaluated in this version of the encyclopaedia and they will be compared with the state-of-the-art results based on the uniform evaluation corpus resulting from the 1st *International Competition on Quality Flaw Prediction in Wikipedia* (overviewed in [15]). Besides, we also aim at measuring which method performs best in assessing the problem of the existing imbalances (cf. the breakdown of quality flaw presented in [10]) between the positive samples available (flawed content) and the remaining untagged documents that exist in Wikipedia.

The rest of the article is organized as follows. Section 2 introduces the context of the problem faced in this work. Then, in Sect. 3, we present the formal problem statement and the different prediction approaches evaluated are briefly described. Also, the document model used to represent the articles is discussed. Section 4 reports on the experimental setting carried out and the obtained results. Finally, Sect. 5 offers the conclusions and briefly mentions future work.

2 Related Work

To the best of our knowledge, the first exploratory analysis targeting the existence of IQ flaws in Wikipedia articles was reported in [9]. Besides, the flaw detection task was evaluated as a one-class classification problem presuming that only information about one class, the so-called target class, is available. Then, [2]

⁴ https://en.wikipedia.org/wiki/Wikipedia:Featured_articles

push further the exploratory analysis reported in [9] by presenting the first complete breakdown of Wikipedia IQ flaws for the snapshot from January 15, 2011. Finally, in [10], it is presented a document model composed by 95 features capturing aspects of documents related to their content, structure, edit history, and how they are embedded into Wikipedia's network. According to our literature review, this is the most comprehensive document model built so far based on a features engineering approach; and is the one that will be used in our work.

Based on the works referred above, several studies have followed up this research line. Different classification approaches to tackle quality flaw prediction and IQ assessment in Wikipedia have been proposed (cf. [10–14, 16–18]). The approaches mainly differ in the type of classification algorithm that is applied (e.g., semi-supervised or supervised) and in the underlying quality flaw model (e.g., the number of features, features complexity, and the rationale to quantify flaws). This diversity makes a conceptual comparison of the existing quality flaw prediction approaches difficult. For example in [17] the authors presented a deep learning approach using a recurrent neural network; the quality classification of Wikipedia articles in English, French, and Russian languages was promising without the need of a feature extraction phase. Other machine learning algorithms such as SVM and K-NN are widely used for this task. Such is the case of [16] in which the authors combine the algorithms with a set of features based on the content and structure of the articles. In [18] instead, a quality score function was used to measure the quality of Wikipedia articles written in seven different languages with a precision around 90%; the score is based in the length of the articles, number of references, number of images, headers in first and second level, and the ratio of the length and number of references.

Moreover, the approaches are not directly comparable in terms of their flaw prediction effectiveness that is reported in the individual experimental evaluation studies. This is mainly because the experimental settings differ in the task (e.g., the number of flaws to be detected and their types) and the data set (e.g., the employed Wikipedia snapshot, the applied sampling strategy, and the ratio between flawed and non-flawed articles in the test set). A first attempt to compare the effectiveness of flaw prediction approaches was the 1st *International Competition on Quality Flaw Prediction in Wikipedia*, where the evaluation task was proposed as a one-class classification problem. Nonetheless, a modified version of PU learning (see [12]) achieved the best average F_1 score of 0.815 over all flaws. In the second place, with an average F_1 score of 0.798, [11] tackled the problem as a binary classification problem.

As stated in the introductory section, recently, in [14], in order to automatically detect the Refimprove flaw, a comparative evaluation of alternative state-of-the-art machine learning approaches belonging to different learning paradigms (supervised and semi-supervised) was carried out for the Spanish version of Wikipedia. From among the evaluated methods, there were four that had not been previously evaluated in the literature for this task and three of them; viz. centroid-based balanced SVM, biased-SVM and under-bagged decision trees (with different voting rules) perform best achieving F_1 scores around 0.94.

In the following section, these machine learning approaches are briefly introduced together with the document model used for representing articles.

3 Problem Statement and Flaw Prediction Approaches

We start with a formal definition of the problem faced in this paper, namely the algorithmic prediction of the *Refimprove* quality flaw in Wikipedia (Section 3.1). We then provide the theoretical background of the flaw prediction approaches used in our work (Section 3.2) and finally, we briefly introduce the document model used to represent articles (Section 3.3).

3.1 Problem Statement

Following [10], quality flaw prediction is treated here as a classification problem. Let D be the set of English Wikipedia articles and let f_i be the specific quality flaw that may occur in an article $d \in D$; that is, the *Refimprove* flaw in our case. Let \mathbf{d} be the feature vector representing article d , called document model, and let \mathbf{D} denote the set of document models for D . Hence, for flaw f_i , a specific classifier c_i is learned to decide whether an article d suffers from f_i or not; that is, $c_i : \mathbf{D} \rightarrow \{1, 0\}$. The training of c_i is intricate in the Wikipedia setting. For flaw f_i a set $D_i^+ \subset D$ is available, which contains articles that have been tagged to contain f_i (so-called *labeled* articles). However, no information is available about the remaining articles in $D \setminus D_i^+$ —these articles are either flawless or have not yet been evaluated with respect to f_i (so-called *unlabeled* articles).

In recent studies, c_i is modeled as a one-class classifier, which is trained solely on the set D_i^+ of labeled articles (see e.g. [10]). However, in the Wikipedia setting, the large number of available unlabeled articles may provide additional knowledge that can be used to improve classifiers training. Thus, addressing the problem of exploiting unlabeled articles to improve the performance of c_i lead us to cast the problem as a binary classification task.

3.2 Flaw Prediction Approaches

Despite its theoretical one-class nature, quality flaw prediction has been tackled in prior studies as a binary classification task –which relates to the realm of supervised learning– and the results achieved in practice have been quite competitive [11, 14, 19]. Supervised learning deals with the situation where training examples are available for all classes that can occur at prediction time. In *binary classification*, the classification $c_i(\mathbf{d})$ of an article $d \in D$ with respect to a quality flaw f_i is defined as follows: given a sample $P \subseteq D_i^+$ of articles containing f_i and a sample $N \subseteq (D \setminus D_i^+)$ of articles not containing f_i , decide whether d belongs to P or to N . The binary classification approach tries to learn a class-separating decision boundary to discriminate between P and a particular N . In order to obtain a sound flaw predictor, the choice of N is essential. N should be a representative sample of Wikipedia articles that are flawless regarding f_i .

Centroid-based Balanced SVM The classifier is trained with a balanced set, where the positive class P is uniformly sampled from D_i^+ and N is composed by the resulting $|P|$ centroids of running k -means clustering on $D \setminus D_i^+$.

Biased SVM Since the ratio between the unlabeled data and the positive samples is unbalanced, a more principled approach to solve the problem allows having independent penalty terms for both classes, in opposition to the standard formulation of SVM, where the penalty factor C is applied to elements of both classes in the same way. Hence, we will have a penalty term C_+ for elements belonging to the positive class P and a penalty term C_- for elements belonging to the so-called negative class N (unlabeled data). It is expected that these penalty terms reflect the underlying imbalance proportion of the classes in the dataset.

Under-bagged Decision Trees In this ensemble learning approach, many different decision trees are bagged by under-sampling the majority class, in order to train each decision tree with a balanced dataset. Let us suppose that we split the positive set P in k chunks. We will refer them as P_1, \dots, P_k , respectively. Then, from the unlabeled data N , we under-sample the set by uniformly selecting k subsets N_1, \dots, N_k , such that $|P_i| = |N_i|, \forall i = 1, \dots, k$. Therefore, k different training sets ($T_{i=1, \dots, k}$) can be built by combining P_1 with N_1 , P_2 with N_2 , and so on. When there are no enough positive samples, like in [14], set P can be matched with each subset N_i . In turn, each sampled dataset $T_{i=1, \dots, k}$ is used to train a C4.5 decision tree that will be referred as $C_{i=1, \dots, k}$. Then, for each document j from the test set, the prediction of each classifier $C_{i=1, \dots, k}$ has to be aggregated in a final prediction to decide if article j is found flawed or not.

3.3 Document Model

To model the articles, we used the document model proposed in [10], that is the most comprehensive document model proposed so far for quality flaw prediction in Wikipedia. It comprises 95 article features, including all of the features that have been used in [9, 12] and many of the features that have been used in [11]. Formally, given a set $D = \{d_1, d_2, \dots, d_n\}$ of n articles, each article is represented by 95 features $F = \{f_1, f_2, \dots, f_{95}\}$. A vector representation for each article d_i in D is defined as $d_i = (v_1, v_2, \dots, v_{95})$, where v_j is the value of feature f_j . A feature generally describes some quality indicator associated with an article.

In [10] four such subsets were identified by organizing the features along the dimensions *content*, *structure*, *network* and *edit history*. Content features are computed based on the plain text representation of an article and mainly address aspects like writing style and readability. Structure features rely on an article's wiki markup and are intended to quantify the usage of structural elements like sections, templates, tables, among others. Network features quantify an article's connectivity by means of internal and external links. Edit history features rely on an article's revision history and model article evolution based on the frequency and the timing of edits as well as on the community of editors. In [10], a detailed description for each feature is provided including implementation details. Due to space constraints, these features are not explicitly described in this paper.

4 Experiments and Results

To perform our experiments, we have used the corpus available in the above-mentioned Competition on Quality Flaw Prediction in Wikipedia [15], which has been released as a part of PAN-WQF-12,⁵ a more comprehensive corpus related to the ten most important article flaws in the English Wikipedia, as pointed out in [2]. The training corpus of the competition contains 154116 tagged articles (not equally distributed) for the ten quality flaws, plus additional 50000 untagged articles. The test corpus (19010 articles) contains a balanced number of tagged articles and untagged articles for each of the ten quality flaws, and it is ensured that 10% of the untagged articles are featured articles.

In particular, for the *Refimprove* flaw, there are 23144 tagged articles which were used in our experiments together with the 50000 untagged articles mentioned above. The test set for this particular flaw contains 1998 articles; 999 positive ones, 900 untagged and 99 featured articles —meeting the proportions described above.

4.1 Experimental Setting

For all the SVM classifiers (uniformly-sampled, centroid-based and biased), as usual, their parameters were experimentally derived by a tenfold cross-validated grid-search with different kernels. For the linear kernel, C was set to values in the range $C \in \{2^{-1}, \dots, 2^{11}\}$. For the RBF kernel, in addition to the values evaluated for C , $\gamma \in \{0.125, 0.5, 1, 2\}$. Different configurations of polynomial kernels were also evaluated with $d \in \{2, 3, 4\}$ and $r \in \{0, 1\}$. In particular, for the biased-SVM, the C_+ and C_- mentioned above, in LIBSVM [20] are obtained by multiplying the C value by parameters w_+ and w_- , respectively. Thus, w_- was set to 1 and $w_+ \in \{5, 6, 7, 8\}$ to reflect different penalization values close to the existing imbalances between the classes, whose values are $|P| = 1000$ and $|N| = 8000$. We decided to set up $|P| = 1000$, given that this was the amount of positive samples used in [12, 13], and more importantly in [13] where it was also used the document model proposed in [10] to represent the articles. Likewise, $|N|$ was set up to 8000 articles based on the flaw ratio of 1:8 estimated for the *Refimprove* flaw in [10]. An estimated ratio of 1:8, actually means that every eight articles one of them is expected to contain this flaw.

For the centroid-based SVM, it holds that $|P| = |N| = 1000$. Before selecting it as the “balanced” SVM classifier to be applied to the test set, we performed a statistical study comparing its performance against a classical binary SVM classifier, where N was chosen by randomly sampling from the 50000 untagged articles. For each parameters configuration evaluated in the grid-search, the F_1 scores of each fold were used to gather ten samples for each classifier. In order to analyze whether it is worth selecting the centroid-based approach over the traditional one, all the samples collected for both formulations were statistically compared among each other (One-way ANOVA with Tukey-Kramer Multiple

⁵ The corpus is available at <https://webis.de/data/pan-wqf-12.html>

Comparisons Test) to obtain the best configuration. The results showed that for values of C higher or equal than 2^9 , the existing difference in performance between both formulations was not statistically significant. Hence, both approaches were evaluated on the test set as it can be observed in the first and second rows of Table 2. The little difference in favor of the standard formulation can be observed in the first three columns of the first two rows.

In our implementation of the under-bagged decision trees, we carried out two different experimental settings: with 46 different decision trees and 23 decision trees, respectively. Given that the obtained results for both settings were quite similar, below we describe how the experimental setting was performed for the 23 different decision trees. It is worth mentioning that to perform the experiments with decision trees as well as running k -means to obtain the centroids for the centroid-based balanced SVM, we have used the WEKA Data Mining Software [21]. Moreover, the five ensemble rules presented in Table 1 were programmed in AWK language.

In order to train each decision tree with a balanced dataset, the 23 decision trees were bagged with under-sampling of the untagged documents. Hence, 23 different training sets were built by combining chunks of 1000 articles. From the 23144 positive samples, 23 chunks of 1000 articles were selected. We will refer to them as P_1, \dots, P_{23} , respectively. The remaining 144 articles were discarded. Similarly, from among the 50000 untagged articles, 23 chunks of 1000 articles were randomly selected following a uniform distribution. We will refer to them as N_1, \dots, N_{23} , respectively. The remaining 27000 articles were kept aside. Therefore, 23 different training sets ($T_{i=1, \dots, 23}$) were built by combining P_1 with N_1 , P_2 with N_2 , and so on. That is: $T_1 = P_1 \cup N_1$, $T_2 = P_2 \cup N_2, \dots, T_{23} = P_{23} \cup N_{23}$.⁶

In turn, each sampled dataset $T_{i=1, \dots, 23}$ was used to train a C4.5 decision tree (with default parameters) that will be referred to as $C_{i=1, \dots, 23}$. The performance of each decision tree C_i was evaluated by a tenfold cross-validation. Then, for each document $j = 1, \dots, 1998$ belonging to the test set, the prediction stated by each classifier $C_{i=1, \dots, 23}$ has to be aggregated in a final prediction to decide if article j is found flawed or not. Table 1 presents the five ensemble rules evaluated in our experiments. Whatever the rule used, when it holds that $R_1 \geq R_2$ then the evaluated article is deemed positive; otherwise negative.

4.2 Results

The state-of-the-art F_1 score for the *Refimprove* flaw on the test set of the 1st *International Competition on Quality Flaw Prediction in Wikipedia* is 0.938, which was achieved in [13], by using a variant of PU-learning (cf. [22] for the original version). As we can see in Table 2, the only method that did not achieve this value was the centroid-based balanced SVM. As mentioned above is not possible to fairly compare the performance of this method in a setting so different

⁶ When the 46 decision trees were used, from the remaining 27000 untagged articles, 23000 were uniformly chosen to compose sets N_{24}, \dots, N_{46} which were combined with P_1, \dots, P_{23} , respectively.

Table 1. Strategies and descriptions for ensemble rules as proposed by [23].

Rule	Strategy	Description
Max	$R_1 = \arg \max_{1 \leq i \leq K} P_{i1},$	Use the maximum classification probability of these K classifiers for each class label.
	$R_2 = \arg \max_{1 \leq i \leq K} P_{i2}$	
Min	$R_1 = \arg \min_{1 \leq i \leq K} P_{i1},$	Use the minimum classification probability of these K classifiers for each class label.
	$R_2 = \arg \min_{1 \leq i \leq K} P_{i2}$	
Product	$R_1 = \prod_{i=1}^K P_{i1},$	Use the product of classification probability of these K classifiers for each class label.
	$R_2 = \prod_{i=1}^K P_{i2}$	
Majority vote*	$R_1 = \sum_{i=1}^K f(P_{i1}, P_{i2}),$	For the i^{th} classifier, if $P_{i1} \geq P_{i2}$, class C_1 gets a vote, if $P_{i2} \geq P_{i1}$, class C_2 gets a vote.
	$R_2 = \sum_{i=1}^K f(P_{i2}, P_{i1})$	
Sum	$R_1 = \sum_{i=1}^K P_{i1},$	Use the summation of classification probability of these K classifiers for each class label.
	$R_2 = \sum_{i=1}^K P_{i2}$	

* Function $f(x, y)$ is defined as 1 if $x \geq y$; 0 otherwise.

Table 2. Comparative performance measures.

Algorithm	Validation set			Test set		
	Precision	Recall	F ₁	Precision	Recall	F ₁
Randomly-sampled balanced binary SVM ($C = 2^9$)	0.99	0.97	0.98	0.90	0.98	0.94
Centroid-based balanced binary SVM ($C = 2^9$)	0.98	0.94	0.96	0.90	0.92	0.91
Biased-SVM ($C = 2^3, w_+ = 8, w_- = 1, r = 1, d = 3, \gamma = 0.5$)	0.92	0.97	0.95	0.95	0.94	0.95
Under-bagged DT (Max rule)	–	–	–	0.88	1.00	0.94
Under-bagged DT (Min rule)	–	–	–	0.93	0.99	0.96
Under-bagged DT (Product rule)	–	–	–	0.88	1.00	0.94
Under-bagged DT (Majority vote rule)	–	–	–	0.90	0.99	0.94
Under-bagged DT (Sum rule)	–	–	–	0.90	0.99	0.94

than the one evaluated in [14], where it achieved a good performance for the Spanish Wikipedia; and where based on this evidence, we decided to evaluate it in the English version.

Regarding the under-bagged decision trees, as in [14], the different ensemble rules performed well, being in this case, the *min* rule, the one which obtained the best F_1 score of 0.96 improving the state-of-the-art result by 2.13%. Finally, biased-SVM outperformed the state-of-the-art result by 1.1%. Despite the fact that these improvements may seem small, it is worth considering that the benchmark is high and increasing by 2% the current F_1 score, reduces by approximately 33% the gap to the optimum score. Moreover, our results are directly comparable to the value found in [13], since we used the same data set and document model for representing the articles.

5 Conclusions

In this work, we carried out a comparative study of three state-of-the-art approaches to automatically assess information quality; in particular, to identify the *Refimprove* flaw as a binary classification task. The results obtained showed that the *Refimprove* flaw prediction can be performed with an F_1 score of 0.96, using a document model consisting of 95 features and under-bagged C4.5 decision trees as classification method. This result outperformed the F_1 score of 0.938 achieved in [13], by using a variant of PU-learning. Also, as stated in [22], biased-SVM performed better than PU-learning, but the improvement achieved in our work (1.1%) was not as much as expected according to the evidence provided in the comparative study of [22]. As future work we plan to tackle the remaining flaws evaluated in the 1st *International Competition on Quality Flaw Prediction in Wikipedia*.

Acknowledgments

This work has been partially funded by PROICO P-31816, Universidad Nacional de San Luis, Argentina.

References

1. Wang, R., Strong, D.: Beyond accuracy: what data quality means to data consumers. *Journal of management information systems* **12**(4) (1996) 5–33
2. Anderka, M., Stein, B.: A breakdown of quality flaws in Wikipedia. In: 2nd joint WICOW/AIRWeb workshop on Web quality (WebQuality'12), ACM (2012) 11–18
3. Pohn, L., Ferretti, E., Errecalde, M.: Identifying featured articles in Spanish Wikipedia. In: *Computer Science & Technology Series: XX Argentine Congress of Computer Science - selected papers*. EDULP (2015) 171–182
4. Ferretti, E., Soria, M., Casseignau, S.P., Pohn, L., Urquiza, G., Gómez, S.A., Errecalde, M.: Towards information quality assurance in Spanish Wikipedia. *Journal of Computer Science & Technology* **17**(1) (2017) 29–36

5. Lewoniewski, W., Härting, R.C., Węcel, K., Reichstein, C., Abramowicz, W.: Application of SEO metrics to determine the quality of wikipedia articles and their sources. In: *Information and Software Technologies*, Springer (2018) 139–152
6. Lewoniewski, W.: Measures for quality assessment of articles and infoboxes in multilingual wikipedia. In: *Business Information Systems Workshops*. (2019)
7. Tran, K.N., Christen, P., Sanner, S., Xie, L.: Context-aware detection of sneaky vandalism on wikipedia across multiple languages. In: *19th Pacific-Asia Conference on Knowledge Discovery and Data Mining (PAKDD)*. (2015)
8. Kiesel, J., Potthast, M., Hagen, M., Stein, B.: Spatio-temporal Analysis of Reverted Wikipedia Edits. In: *11 Intl. AAAI Conference on Web and Social Media*. (2017)
9. Anderka, M., Stein, B., Lipka, N.: Towards Automatic Quality Assurance in Wikipedia. In: *20th intl. conference on World Wide Web, ACM* (2011) 5–6
10. Anderka, M.: Analyzing and Predicting Quality Flaws in User-generated Content: The Case of Wikipedia. PhD thesis, Bauhaus-Universität Weimar (June 2013)
11. Ferschke, O., Gurevych, I., Rittberger, M.: FlawFinder: a modular system for predicting quality flaws in Wikipedia. In: *CLEF (Online Working Notes/Labs/Workshop)*. (2012)
12. Ferretti, E., Fusilier, D.H., Guzmán-Cabrera, R., y Gómez, M.M., Errecalde, M., Rosso, P.: On the use of PU learning for quality flaw prediction in wikipedia. In: *CLEF (Online Working Notes/Labs/Workshop)*. (2012)
13. Ferretti, E., Errecalde, M., Anderka, M., Stein, B.: On the use of reliable-negatives selection strategies in the pu learning approach for quality flaws prediction in wikipedia. In: *11th Intl. Workshop on Text-based Information Retrieval*. (2014)
14. Ferretti, E., Cagnina, L., Paiz, V., Donne, S.D., Zacagnini, R., Errecalde, M.: Quality flaw prediction in spanish wikipedia: A case of study with verifiability flaws. *Information Processing & Management* **54**(6) (2018) 1169 – 1181
15. Anderka, M., Stein, B.: Overview of the 1st International Competition on Quality Flaw Prediction in Wikipedia. In Forner, P., Karlgren, J., Womser-Hacker, C., eds.: *Working Notes Papers of the CLEF 2012 Evaluation Labs*. (2012)
16. Dang, Q.V., Ignat, C.L.: Measuring quality of collaboratively edited documents: The case of wikipedia. In: *IEEE 2nd International Conference on Collaboration and Internet Computing (CIC)*, IEEE Computer Society (2016) 266–275
17. Dang, Q.V., Ignat, C.L.: An end-to-end learning solution for assessing the quality of wikipedia articles. In: *13th Intl. Symposium on Open Collaboration*. (2017) 1–10
18. Lewoniewski, W., Węcel, K.: Relative quality assessment of Wikipedia articles in different languages using synthetic measure. In Abramowicz, W., ed.: *Lecture Notes in Business Information Processing*. Volume 303. Springer (2017) 282–292
19. Ferschke, O., Gurevych, I., Rittberger, M.: The impact of topic bias on quality flaw prediction in Wikipedia. In: *51st annual meeting of the association for computational linguistics, ACL* (2013) 721–730
20. Chang, C.C., Lin, C.J.: LIBSVM: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology* **2** (2011) 27:1–27:27
21. Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., Witten, I.H.: The weka data mining software: An update. *SIGKDD Explorations* **11**(1) (2009)
22. Liu, B., Dai, Y., Li, X., Lee, W., Yu, P.: Building text classifiers using positive and unlabeled examples. In: *3rd IEEE international conference on data mining (ICDM'03)*, IEEE Computer Society (2003)
23. Kittler, J., Hatef, M., Duin, R.P.W., Matas, J.: On combining classifiers. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **20**(3) (1998) 226–239

DE with Random Vector based Mutatiton for High Dimensional Problems

Mg. Sebastián Hernández^[1], Dr. Efrén Mezura-Montes^[2], and Dr. Guillermo Leguizamón^[3]

^[1]National University of Southern Patagonia,

^[2]Artificial Intelligence Research Center, University of Veracruz

^[3]National University of San Luis

sebastian.unpa@gmail.com

emezura@uv.mx

legui@unsl.edu.ar

Abstract. Metaheuristic techniques are the current standard for solving optimization problems. Differential Evolution (DE) is one of the most used because all operations are on real floating point numbers and does not require extra coding. However, the performance shown by DE could decay when applied in problems of high dimensionality. In this paper we present RLSDE, a modified version of DE, based on a random vector as a scaling factor for the differential mutation and the application of a local search operator. These modifications constitute an algorithm capable of solving 100D problems using few computational resources. RLSDE is compared against the results obtained with the classic version of DE and ELSDE (Enhanced Local Search Differential Evolution), showing the performance of the proposal.

Keywords: differential evolution, high-dimensional optimization problem, local search

1 Introduction

As technology advances, it is necessary to create and solve increasingly complex mathematical models in order to provide more precise solutions. In turn, these models should involve a large number of variables, which makes the problem of optimization in a problem of high dimensionality (100 or more variables). It should also be considered that they are generally non-linear functions and therefore their resolution is not simple. All these conditions make that optimal search methods require a large computational cost and often fail in their objective due, among other factors, to the exponential growth of the search space and the complexity of the problem. That is why new optimization techniques should be proposed with simple but powerful conditions [1], [4], [6], [13], capable of facing and solving high dimensional problems. In this article a modified and hybrid version of DE is presented. The main modification was made on the differential mutation: the population was classified into three different groups based on their

performance and then this classification was taken into account when choosing the vectors to generate the mutant vector. The other modification that was applied in the generation of the mutant vector is the use of a variable scale factor. Finally, this proposal was hybridized with a random local search engine whose operation is based on the classification of the population used in the differential mutation. The performance of this proposal is evaluated with classical functions of the literature, the results obtained are compared against those presented in [12], obtained by the ELSDE algorithm with the same number of function evaluations. The results obtained by RLSDE show the quality of this proposal. The rest of this paper is organized as follows, the second section describe the basic DE algorithm, the third section presents our version, the RLSDE algorithm. Its performance is analyzed and compared against DE y ELSDE in the fourth section. Finally, the conclusions are given in the fifth section.

2 Differential Evolution

Differential evolution was presented in 1996 by Storn and Price [11]. It is a population metaheuristic whose operation is based on the application of operators to the individuals (vectors) of the population, through a time of evolution:

- **Initialization.** Taking into account that each variable of the search space is confined to a certain region, real values within this range are generated in a random way. At generation zero (initial value), the j th component of the i th individual is defined as:

$$\mathbf{x}_{j,i,0} = \text{rand}_j [0, 1) \cdot (b_{j,U} - b_{j,L}) + b_{j,L},$$

where $[b_{j,L}, b_{j,U}]$ is the definition interval of the j th variable.

- **Differential mutation.** Like most population metaheuristics, DE applies an operator to generate new individuals, disrupting existing ones in the population. This is done by randomly selecting three individuals from the population (targets vectors) and creating the *donor vector* with one of them as *base* and the other two in the form of *scaled difference*:

$$\mathbf{v}_{i,g} = \mathbf{x}_{r_0,g} + F \cdot (\mathbf{x}_{r_1,g} - \mathbf{x}_{r_2,g}),$$

where g represents the evolution time, F is the scale factor applied in the vector difference and $\mathbf{x}_{r_i}, i = 0, 1, 2$ are the randomly selected vectors, and the indices r_1, r_2, r_3 are mutually exclusive. This operator is considered the most important of DE, since it adds population diversity taking into account the characteristics of individuals present in the population.

- **Crossover.** This operator, also called *discrete recombination*, generates the *trial vector* by randomly mixing components of the target and donor vectors, whose indices in the population are the same:

$$u_{j,i,g} = \begin{cases} v_{j,i,g} & \text{if } \text{rand}_j [0, 1) \leq C_r \\ x_{j,i,g} & \text{othercase} \end{cases},$$

where C_r is the crossover rate that will be compared to a random value to decide who will bring the component to the *trial vector*.

- **Selection.** The next step of the algorithm calls for *selection* to determine whether the target or the trial vector survives to the next generation. When comparing the fitness of target vector ($\mathbf{x}_{i,g}$) with the trial vector ($\mathbf{u}_{i,g}$), it is decided whether the target remains at least one more generation in the population or if it is replaced by the trial vector:

$$\mathbf{x}_{i,g+1} = \begin{cases} \mathbf{u}_{i,g} & \text{if } f(\mathbf{u}_{i,g}) < f(\mathbf{x}_{i,g}) \\ \mathbf{x}_{i,g} & \text{otherwise} \end{cases}$$

3 State of the art

DE is a well-known and used population metaheuristic due to its ability to obtain quality solutions. But as it is also affected by the curse of dimensionality, algorithms based on DE are often presented in order to effectively solve high-dimensional problems. In 2014 article [3] was presented. The authors suggest that the strategies used in low dimension (mutation, population size choice, crossover) are not adequate to solve problems in high dimension. In addition, with the change of dimension it is known that the search space increases exponentially but the amount of function evaluations used generally only grows linearly. In 2017 the articles [5] and [2] were presented. In the first of them, the center of gravity of three randomly chosen individuals was used as the base vector for differential mutation, and in the other article the authors used the population covariance matrix because they observe that the traditional approach to do facing the problem of increasing dimension is to increase the evolution time. According to the authors, it is not taken into account that both magnitudes are not modified proportionally and therefore a good performance is not achieved. In 2018 Meselhi et al. present [8], where they propose the use of the Enhanced Differential Grouping (EDG) method, capable of discovering the dependency relationships between variables and then grouping the independent variables in the same subproblem. In 2019, Cai et al. publish cite Cai2019. In this article the authors apply to algorithms of high dimension an algorithm that makes a prediction of the global minimum and from that prediction the direction that the differential mutation should take in search of that optimum is guided.

4 Proposed modifications

DE, like most optimization algorithms, suffers from the *curse of dimensionality* due to the exponential increase of the search space when considering increases in the dimensionality of the problem. Many modifications have been proposed in order to mitigate the effects of increasing the size of the search space [2], [8]. In this case, the proposed modifications are simple and show a significant improvement in performance compared to the efficiency of the original DE version.

4.1 Population partitioning

The population is classified, according to its fitness value, into three groups. For this, the individual with the minimum fitness is $k = 1$ and the individual with the maximum fitness is $k = n$. Then, *A* group contains individuals whose k position on the fitness scale verifies $k < 0.25 \times \text{popsize}$, that is, the group of the best individuals; group *C* contains those where $k > 0.75 \times \text{popsize}$ (the worst individuals) and group *B* includes the remaining individuals of the population. When differential mutation is applied, three individuals are selected in a random manner so that the base vector does not belong to the same group as the two to be used in the difference, in order to favor diversity.

4.2 Vector scale factor

The scale factor F is responsible for *smoothing* the disturbance generated by the vector difference when adding to the base vector in the differential mutation. The lower the value of F , the smaller the size of the steps performed by the mutation and therefore it will take longer to achieve convergence. Larger values of F facilitate exploration, but may cause the algorithm to exceed some optimal values due to its passage width. The classical version of DE uses a fixed scale factor for the vector difference applied in the differential mutation. In this article we propose to use a vector as a scale factor, so that each of the variables of the difference vector is disturbed differently. This new vector is generated from a *center* C and an expansion radius r , where the components are random within the interval $(C - r, C + r)$:

$$\mathbf{F} = [f_1, f_2, \dots, f_D],$$

where the components of the vector are $f_i = C + 2 \cdot r \cdot \text{rand}() - r$

4.3 Local search

It is common to hybridize the algorithms of global optimization with local search algorithms [7], [10], [9]. Local search algorithms intensify the exploration of individuals located around a particular individual, generally of good performance. In our case, local search is applied to the best individual in each generation. The local search operator generates six new individuals for each generation of evolution, from the best individual in the population (\mathbf{x}_{best}), from the average of population (\mathbf{x}_μ), and from the average individual in group *A* (\mathbf{x}_μ^A). It's like that \mathbf{x}_μ^A and \mathbf{x}_μ are obtained respectively as follows:

- The j th component of \mathbf{x}_μ^A is obtained from calculating the average of the j th components of the individuals in group *A*, without taking into account \mathbf{x}_{best} .
- The j th component of \mathbf{x}_μ is obtained from calculating the average of the j th components of all individuals in the population, including \mathbf{x}_{best} .

Then, six new individuals are generated with different characteristics and with different disturbed variables that will be compared, one at a time, against \mathbf{x}_{best} . The only new individual that emerges from disturbing all its components is \mathbf{x}_c , called *centroid individual*, with a very simple mathematical formulation:

$$\mathbf{x}_c = \frac{\mathbf{x}_{best} + \mathbf{x}_\mu^A + \mathbf{x}_\mu}{3}$$

where each individual of the population contributes with its components in a different proportion, which will then be scaled up to a third:

- \mathbf{x}_{best} contributes in full form from the original sum and proportionally to $1/popsiz$ from \mathbf{x}_μ .
- The individuals of group A , contribute with proportion $1/(popsiz_A - 1)$ from the original sum and then with the proportion $1/popsiz$ from \mathbf{x}_μ .
- The individuals of groups B and C , contribute with the proportion $1/popsiz$ from \mathbf{x}_μ .

Then two normalized individuals are generated, according to the infinite norm of vectors:

$$\mathbf{A} \in \mathbb{R}^D, \|\mathbf{A}\|_\infty = \max_{i \in [1, D]} |A_i|,$$

denominated \mathbf{d}_μ^A and \mathbf{d}_μ :

$$\mathbf{d}_\mu^A = \frac{|\mathbf{x}_{best} - \mathbf{x}_\mu^A|}{\|\mathbf{x}_{best} - \mathbf{x}_\mu^A\|_\infty}, \quad \mathbf{d}_\mu = \frac{|\mathbf{x}_{best} - \mathbf{x}_\mu|}{\|\mathbf{x}_{best} - \mathbf{x}_\mu\|_\infty},$$

where its minimum and maximum components can be 0 and 1 respectively. Only fifteen percent of the components of the *distance vectors* will be taken into account, then a permutation is generated on the vector $\mathbf{v} = [1, D]$, in our case $\mathbf{v} = [1, 2, 3, \dots, 100]$. Thus, components $\mathbf{v}(1)$ to $\mathbf{v}(15)$ indicate which \mathbf{d}_μ^A values will be conserved and components $\mathbf{v}(16)$ to $\mathbf{v}(30)$ shows which components of \mathbf{d}_μ will be used. The other components are transformed to zero. With the new versions of \mathbf{d}_μ^A and \mathbf{d}_μ , four new individuals are created in two stages. In the first one, vectors \mathbf{x}_{best} , \mathbf{d}_μ^A and \mathbf{d}_μ are taken:

$$\begin{aligned} \mathbf{x}_{D_1} &= \mathbf{x}_{best} - F_d \cdot (\mathbf{d}_\mu^A * \mathbf{x}_\mu^A) \\ \mathbf{x}_{D_2} &= \mathbf{d}_\mu^A * \mathbf{x}_\mu^A - F_d \cdot \mathbf{x}_{best} \end{aligned}$$

with

$$F_d = \frac{f(\mathbf{x}_{best})}{f(\mathbf{x}_\mu^A)}$$

and in second stage:

$$\begin{aligned} \mathbf{x}_{D_3} &= \mathbf{x}_{best} - F_d \cdot (\mathbf{d}_\mu * \mathbf{x}_\mu) \\ \mathbf{x}_{D_4} &= \mathbf{d}_\mu * \mathbf{x}_\mu - F_d \cdot \mathbf{x}_{best} \end{aligned}$$

with

$$F_d = \frac{f(\mathbf{x}_{best})}{f(\mathbf{x}_\mu)}$$

In both cases the symbol $*$ represents the product component to component between two vectors.

Finally, the last of the individuals is generated: $\mathbf{x}_{best'}$, who exchanges internally five of its variables. This exchange is done taking into account that some of the functions selected for optimization are not separable and this process can help the convergence of those functions.

Example From a hypothetical population are obtained:

$$\begin{aligned} - \mathbf{x}_{best} &= [-0.0654166; -0.0804726; 0.098836; -0.0992805; \dots; -0.0438895], \\ f(\mathbf{x}_{best}) &= 0.0481651791782 \\ - \mathbf{x}_{\mu}^A &= [-0.122108; 0.0187508; -0.0661062; -0.0786949; \dots; -0.124477], \\ f(\mathbf{x}_{\mu}^A) &= 0.0703484791916 \\ - \mathbf{x}_{\mu} &= [-0.0570289; -0.0897206; -0.109161; 0.0178861; \dots; 0.0386951], \\ f(\mathbf{x}_{\mu}) &= 0.093300675794 \end{aligned}$$

Then, the *centroid individual* is created from the three vectors described above:

$$\begin{aligned} - \mathbf{x}_c &= [-0.0815178; -0.0504808; -0.0254772; -0.0533631; \dots; -0.0432238], \\ f(\mathbf{x}_c) &= 0.0317538969857 \end{aligned}$$

Since \mathbf{x}_c has better performance than \mathbf{x}_{best} then the local search is considered successful, \mathbf{x}_{best} is replaced by \mathbf{x}_c although the local search process continues until the six explorations are performed, in case it is possible to further improve \mathbf{x}_{best} .

The permutation vector \mathbf{v} is created randomly, according to this example with $D = 100$, where one of the possible combinations for the first fifteen components could be:

$$[7, 10, 1, 12, 11, \dots, 3, 4],$$

these being the positions of the vector \mathbf{d}_{μ}^A that do not become zero. So \mathbf{d}_{μ}^A , for our example, is a vector with $D = 100$, where only the components 7, 10, 1, 12, 11, \dots , 3, 4 are nonzero. Continuing with hypothetical positions 16 to 30 of the permutation vector \mathbf{v} :

$$[29, 20, 27, 16, 17, \dots, 19, 26],$$

therefore those will be the \mathbf{d}_{μ} positions that will not be transformed into the zero value.

Applying the distance formula and then generating four new disturbed vectors \mathbf{x}_D :

$$\begin{aligned} - \mathbf{x}_{D_1} &= [-0.0915425; -0.0504808; -0.0309095; -0.0573951; \dots; -0.0636804], \\ f(\mathbf{x}_{D_1}) &= 0.036670111539 \\ - \mathbf{x}_{D_2} &= [-0.074396; -0.00829604; -0.037242; -0.0333039; \dots; -0.13158], \\ f(\mathbf{x}_{D_2}) &= 0.0276253117511 \\ - \mathbf{x}_{D_3} &= [-0.0815178; -0.0555942; -0.0387451; -0.0533631; \dots; -0.0386199], \\ f(\mathbf{x}_{D_3}) &= 0.035218104998 \end{aligned}$$

- $\mathbf{x}_{D_4} = [-0.00990799; -0.0482058; -0.112258; -0.00648596; \dots; 0.0326252]$,
 $f(\mathbf{x}_{D_4}) = 0.031244911621$

Since \mathbf{x}_{D_2} has a better performance than \mathbf{x}_{best} , then it replaces it. Then, in the last step five components of \mathbf{x}_{D_2} (\mathbf{x}_{best}) are exchanged:

- $\mathbf{x}_{best} = [-0.074396; -0.00829604; -0.037242; -0.0333039; \dots; -0.13158]$,
 $f(\mathbf{x}_{D_2}) = 0.0276253117511$
- $\mathbf{x}_{best'} = [-0.0333039; -0.00829604; -0.037242; -0.074396; \dots; 0.031471]$,
 $f(\mathbf{x}_{D_2}) = 0.0292658878856$

Therefore, the local search operation was successful and the performance of \mathbf{x}_{best} extracted from the population (\mathbf{x}_c and \mathbf{x}_{D_2}) was improved twice, which returns to continue with the evolution as a new individual.

5 Experiments

An experimental study was conducted, with nine classic functions of the literature, to evaluate the efficiency of the proposed algorithm against the classic DE and ELSDE. The ELSDE algorithm is a modified version of DE where a novel local search operation was presented. This local operation combines both advantage of orthogonal crossover and opposition-based search learning strategy. The authors apply this new local search engine only to an individual in the population, chosen randomly. Finally, concluded that ELSDE is an efficient method for the high-dimensional optimization problems.

5.1 Scalable functions

The set of functions to which RLSDE was applied is the same as that used in [12]. These functions are continuous and have different characteristic. They also have many local extreme points and high optimizing complexity.

1. **Sphere.** Continuous, differentiable, separable and multimodal. Defined in $[-100, 100]^D$:

$$f(\mathbf{x}) = \sum_{i=1}^D x_i^2$$

2. **Rosenbrock.** Continuous, non-separable, multimodal and non-convex. Defined in $[-100, 100]^D$:

$$f(\mathbf{x}) = \sum_{i=1}^D \left[100 (x_{i+1} - x_i^2)^2 + (1 - x_i)^2 \right]$$

3. **Ackley.** Continuous, non-convex and multimodal. Defined in $[-32, 32]^D$:

$$f(\mathbf{x}) = -20 \exp \left(-0.2 \sqrt{\frac{1}{D} \sum_{i=1}^D x_i^2} \right) - \exp \left(\frac{1}{D} \sum_{i=1}^D \cos(2\pi x_i) \right) + 20 + \exp(1)$$

4. **Griewank.** Continuous, unimodal and not convex. Defined in $[-600, 600]^D$:

$$f_1(\mathbf{x}) = 1 + \sum_{i=1}^D \frac{x_i^2}{4000} - \prod_{i=1}^D \cos\left(\frac{x_i}{\sqrt{i}}\right)$$

5. **Rastrigin.** Continuous, separable, multimodal and convex. Defined in $[-5, 5]^D$:

$$f(\mathbf{x}) = 10D + \sum_{i=1}^D [x_i^2 - 10 \cos(2\pi x_i)]$$

6. **Schwefel 2.26.** Continuous, differentiable, separable and multimodal. Defined in $[-500, 500]^D$:

$$f(\mathbf{x}) = -\frac{1}{D} \sum_{i=1}^D x_i \sin(\sqrt{|x_i|})$$

7. **Salomon.** Continuous, non-separable, multimodal and non-convex. Defined in $[-100, 100]^D$:

$$f(\mathbf{x}) = 1 - \cos\left(2\pi \sqrt{\sum_{i=1}^D x_i^2}\right) + 0.1 \sqrt{\sum_{i=1}^D x_i^2}$$

8. **Generalized Penalized Function 1.** Continuous, non-separable and multimodal. Defined in $[-50, 50]^D$:

$$f(\mathbf{x}) = \frac{\pi}{D} \left[10 \sin^2(\pi y_1) + \sum_{i=1}^{D-1} (y_i - 1)^2 [1 + 10 \sin^2(\pi y_{i+1})] + (y_n - 1)^2 \right] + \sum_{i=1}^D u_i$$

with

$$u_i = \begin{cases} 100(x_i - 10)^4 & x_i > 10 \\ 0 & -10 \leq x_i \leq 10 \\ 100(-x_i - 10)^4 & x_i < -10 \end{cases}, \quad y_i = 1 + \frac{x_i + 1}{4}$$

9. **Generalized Penalized Function 2.** Continuous, non-separable and multimodal. Defined in $[-50, 50]^D$:

$$f(\mathbf{x}) = \sum_{i=1}^D u_i + 0.1 \left[\sin^2(3\pi x_1) + (x_D - 1) + \sum_{i=1}^{D-1} (x_i - 1)^2 [1 + \sin^2(3\pi x_{i+1})] \right]$$

with

$$u_i = \begin{cases} 100(x_i - 5)^4 & x_i > 5 \\ 0 & -5 \leq x_i \leq 5 \\ 100(-x_i - 5)^4 & x_i < -5 \end{cases}$$

5.2 Execution parameters

This new proposed algorithm (RLSDE) was executed with the following parameter setting:

- Time of evolution: 3500 generations.
- Population size: 50 individuals.
- Dimension of the search space: $D = 100$.
- Scale factor center: $C = 0.4$.
- Scale factor radius: $r = 0.25$.
- Crossover probability: 0.5.
- Number of executions per experiment of each function: 30.

Considering the six new individuals evaluated in each generation by the local search, the total number of function evaluations per experiment is $3500(50 + 6) = 1.96E5$.

5.3 Results and comparison

The results obtained are shown in the same way that the authors presented the results of ELSDE, showing the average and the standard deviation of each function. One difference to highlight is that in ELSDE they used $1.00E6$ function evaluations for each experiment, whereas RLSDE (the proposed algorithm) used only $1.95E5$ function evaluations.

Table 1 shows the results obtained by DE, ELSDE, and RLSDE for the nine functions of the test suited considered with $D = 100$.

Fun	DE (mean \pm std)	ELSDE (mean \pm std)	RLSDE (mean \pm std)
f_1	4.21E+03 \pm 8.65E+02	2.07E-16 \pm 1.06E-16	1.18E-36 \pm 2.62E-36
f_2	5.50E+01 \pm 8.53E + 00	8.21E+01 \pm 7.95E-01	9.66E+01 \pm 3.10E-01
f_3	2.63E+05 \pm 2.01E+04	1.79E-09 \pm 4.62E-10	3.00E-15 \pm 1.63E-15
f_4	6.20E+01 \pm 4.56E+00	0.00E+00 \pm 0.00E+00	0.00E+00 \pm 0.00E+00
f_5	2.40E+06 \pm 1.38E+06	2.46E-10 \pm 3.50E-10	0.00E+00 \pm 0.00E+00
f_6	4.46E+03 \pm 1.24E+03	1.45E+01 \pm 7.93E+01	1.03E+02 \pm 2.48E+02
f_7	4.19E+00 \pm 1.81E+00	9.99E-02 \pm 1.41E-06	9.99E-02 \pm 2.71E-12
f_8	8.55E+02 \pm 6.89E+01	1.07E-10 \pm 5.22E-11	4.71E-33 \pm 0.00E+00
f_9	8.96E+00 \pm 6.68E-01	3.39E-04 \pm 2.34E-03	1.35E-32 \pm 5.59E-48

Table 1. Comparison of results with dimension $D = 100$

When the mean values obtained by ELSDE and RLSDE are statistically compared through the Wilcoxon Signed-Rank test, the difference between the value of the both algorithms and expected difference μ_0 is not big enough to be statistically significant. Despite this, it is observed that the overall performance of RLSDE is equivalent or better than that of ELSDE, due to the difference between the number of evaluations for both algorithms. Figures 1 a 9 shows the

output of one experiment on each of the functions. It should be noted that the functions f_4 and f_5 do not reach the 3500 generations in their graphic domain since the presented scales are logarithmic and the value obtained is zero. The quality of the solution obtained in F_1 can improve, it is necessary to increase the evolution time or improve the setting of the RLSDE parameters. The functions F_2 , F_6 and F_7 show a rapid convergence and then stagnation, possibly the performance of the local search engine should be improved. Finally, in F_3 , F_4 , F_5 , F_8 and F_9 , the evolution time is well used, achieving quality convergence within the scheduled evolution time. This shows the power of RLSDE in high dimensions.

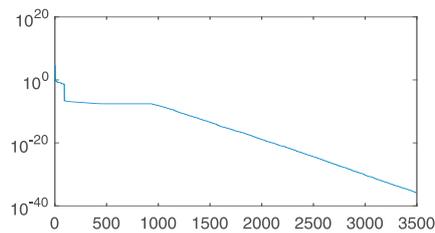


Fig. 1. $f_1 - \mathbf{x}_{best} : 1.07E - 36$

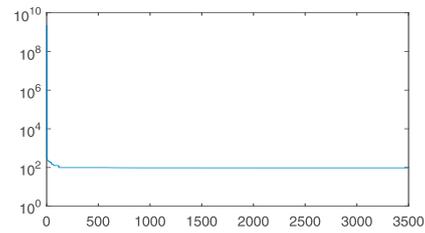


Fig. 2. $f_2 - \mathbf{x}_{best} : 9.59E + 01$

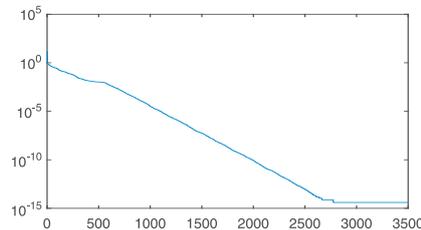


Fig. 3. $f_3 - \mathbf{x}_{best} : 4.44E - 16$

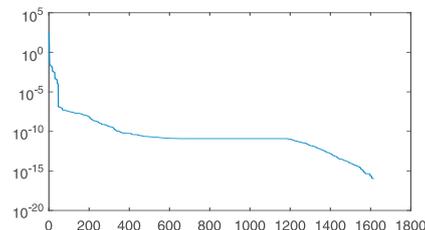


Fig. 4. $f_4 - \mathbf{x}_{best} : 0.00E + 00$

6 Conclusions

RLSDE, a new modified version of DE was presented. The differential mutation operator was modified, classifying the population, in each generation, into three groups according to their fitness. In this way, the base vector was chosen randomly but the vectors used for the difference could not belong to the same group as the base. In this way, the exploration of the search space is favored by

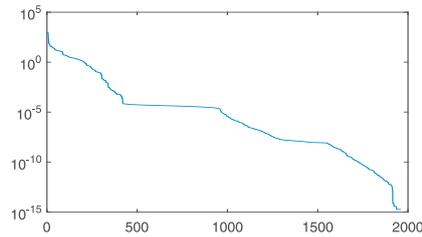


Fig. 5. $f_5 - \mathbf{x}_{best} : 0.00E + 00$

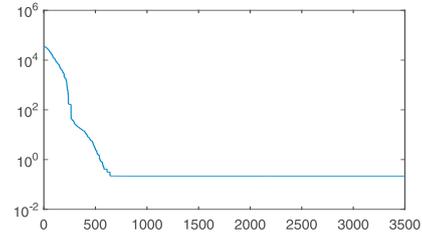


Fig. 6. $f_6 - \mathbf{x}_{best} : 1.27E - 03$

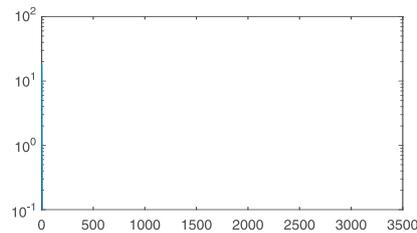


Fig. 7. $f_7 - \mathbf{x}_{best} : 9.98E - 02$

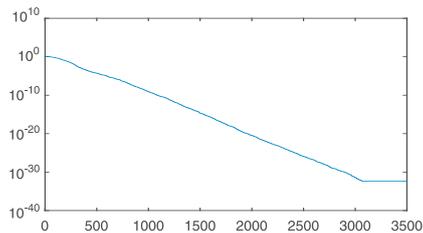


Fig. 8. $f_8 - \mathbf{x}_{best} : 4.71E - 33$

preventing three individuals of similar quality from interacting to generate the trial vector, assuming that these individuals possess very few differences from each other. Another improvement, also within the differential mutation operator, was the application of a random vector as a scale factor. By not applying a uniform value, each of the components was affected to a different extent, so the search space exploration was also favored.

Finally, to favor the exploitation of promising regions, a random local search engine based on distances was applied to the best individual of each generation.

The results obtained are interesting because they are comparable to those obtained by other similar algorithms, but with much less function evaluations.

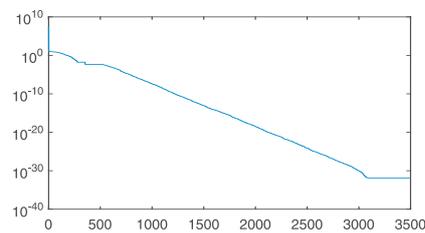


Fig. 9. $f_9 - \mathbf{x}_{best} : 1.34E - 32$

As future work we will try to solve some of the multiple CEC (IEEE) tests and we will try to improve the performance of this version by adjusting the execution parameters to obtain a better performance.

References

1. Cao, B., Zhao, J., Lv, Z., Liu, X., Yang, S., Kang, X., Kang, K.: Distributed parallel particle swarm optimization for multi-objective and many-objective large-scale optimization. *IEEE Access* **5**, 8214–8221 (2017). <https://doi.org/10.1109/access.2017.2702561>
2. Caraffini, F., Neri, F., Iacca, G.: Large scale problems in practice: The effect of dimensionality on the interaction among variables. In: *Applications of Evolutionary Computation*, pp. 636–652. Springer International Publishing (2017). <https://doi.org/10.1007/978-3-319-55849-3-41>
3. Chen, S., Montgomery, J., Bolufé-Röhler, A.: Measuring the curse of dimensionality and its effects on particle swarm optimization and differential evolution. *Applied Intelligence* **42**(3), 514–526 (nov 2014). <https://doi.org/10.1007/s10489-014-0613-2>
4. Deng, C., Dong, X., Yang, Y., Tan, Y., Tan, X.: Differential evolution with novel local search operation for large scale optimization problems. In: *Advances in Swarm and Computational Intelligence*, pp. 317–325. Springer International Publishing (2015). <https://doi.org/10.1007/978-3-319-20466-6>
5. Hiba, H., Mahdavi, S., Rahnamayan, S.: Differential evolution with center-based mutation for large-scale optimization. In: *2017 IEEE Symposium Series on Computational Intelligence (SSCI)*. IEEE (nov 2017). <https://doi.org/10.1109/ssci.2017.8280938>
6. Lin, H.: A differential evolution algorithm based on local search and boundary reflection for global optimization. *2015 11th International Conference on Computational Intelligence and Security (CIS)* pp. 253–257 (2015)
7. Locatelli, M., Maischberger, M., Schoen, F.: Differential evolution methods based on local searches. *Computers & OR* **43**, 169–180 (2014)
8. Meselhi, M.A., Sarker, R.A., Essam, D.L., Elsayed, S.M.: Enhanced differential grouping for large scale optimization. In: *Proceedings of the 10th International Joint Conference on Computational Intelligence*. SCITEPRESS - Science and Technology Publications (2018). <https://doi.org/10.5220/0006938902170224>
9. Michiels, W., Aarts, E., Korst, J.: *Theoretical Aspects of Local Search*. Springer-Verlag GmbH (2006)
10. Noman, N., Iba, H.: Accelerating differential evolution using an adaptive local search. *IEEE Transactions on Evolutionary Computation* **12**, 107–125 (2008)
11. Price, K., Storn, R., Lampinen, J.: *Differential evolution - A practical approach to global optimization*. Springer-Verlag (2005)
12. Xiao-Gang, D., Chang-Shou, D., Zhang, Y., Yu-Cheng, T.: Enhancing local search of differential evolution algorithm for high dimensional optimization problem. In: *2015 34th Chinese Control Conference (CCC)*. IEEE (jul 2015). <https://doi.org/10.1109/chicc.2015.7260973>
13. Zhang, Q., Cheng, H., Ye, Z., Wang, Z.: A competitive swarm optimizer integrated with cauchy and gaussian mutation for large scale optimization. In: *2017 36th Chinese Control Conference (CCC)*. IEEE (jul 2017). <https://doi.org/10.23919/chicc.2017.8028924>

Deep Learning Architecture for Forest Detection in Satellite Data

Gabriel D. Caffaratti^{1,3}, Martin G. Marchetta¹, Raymundo Q. Forradellas¹,
Leonardo D. Euillades^{2,3}, and Pablo A. Euillades^{2,3}

¹ Intelligent Systems Laboratory (LABSIN)

² Special Training and Computer Assisted
Engineering Development Institute (CEDIAC)
School of Engineering - National University of Cuyo

Centro Universitario, Mendoza, Argentina

³ CONICET

{gabriel.caffaratti, martin.marchetta, leonardo.euillades
pablo.euillades}@ingenieria.uncuyo.edu.ar
kike@uncu.edu.ar

Abstract. Deep Learning algorithms have achieved great progress in different applications due to their training capabilities, parameter reduction and increased accuracy. Image processing is a particular area that has received recent attention promoted by the growing processing power and data availability. Remote sensing devices provide image-like data that can be used to characterize Earth's natural or artificial phenomena. Particularly, forest detection is important in many applications like flooding simulations, analysis of forest health or detection of area desertification. The existing techniques for forest detection based on satellite data lack accuracy or still require human expert intervention to correct recognition errors or parameter setup. In this work a Deep Learning architecture for forest detection is presented, that aims at increasing accuracy and reducing expert dependency. A data preprocessing procedure, analysis and dataset composition for robust automatic forest detection is described. The proposed approach was validated with real SRTM and Landsat-8 satellite data.

Keywords: Remote sensing, Forest detection, Deep learning

1 Introduction

The characteristics of the land surface, if adequately processed, can provide fundamental information for many applications like modeling and simulating atmospheric, hydrologic and ecological processes occurring on the surface of the Earth [17]. Recently, some projects like the Shuttle Radar Topography Mission (SRTM) and Landsat 8 provided elevation as well as optical and thermal information with almost full coverage of the Earth's surface. However, SRTM and Landsat 8 remote sensing systems have different drawbacks, compared with other

remote sensing devices, which need to be addressed in order to provide acceptable models. One of these problems is the SRTM radar's inability to penetrate tree canopies, which only allows for Digital Surface Models (DSM), instead of Digital Elevation Models (DEM) [10].

The conversion from DSM to DEM requires the detection of forested areas in order to adjust the elevation in those particular pixels. This has a critical impact in research fields like hydrology, where the simulation of water flow is determined according to the land characteristics. Also, forest detection is required in ecological studies like soil classification, forest health tracking, desertification monitoring, among others [16]. The detection of forested areas is not a trivial task and is usually performed with a combination of vegetation detection techniques and human expert intervention, which is tedious and error-prone.

Machine Learning (ML) is a field that offers different techniques to solve complex problems by learning models. During training, classification algorithms adjust models to recognize different patterns in the data to perform classification over the inputs provided. The input data is analyzed automatically during training based on the expected results of each example in the training set. The learned model is then used for the classification task, rather than having to recognize patterns manually or using custom algorithms for each case, reducing the problem of misinterpretation of the data or the omission of unseen patterns. Therefore, ML techniques have a potential application in forest detection.

Different solutions have been proposed based on ML algorithms to solve forest/no-forest classification problems. However, these techniques usually lack accuracy. In this work, we address this problem by means of a Convolutional Neural Network to generate a forest/no-forest mask. The data used for training are based on radar, optical and thermal information provided by SRTM and Landsat-8, using the JAXA Forest/No Forest (JFNF) mask as ground truth.

This paper is structured as follows. Section 2 presents a literature review of techniques in ML and forest recognition. Section 3 describes the data preprocessing. Section 4 exposes the dataset distribution analysis. Section 5 explains the proposed architecture and the classification algorithm. Experiments performed on a case of study are shown in section 6, followed by conclusions and future work (section 7).

2 Literature

Digital Elevation Models are a common representation in Earth Sciences. In particular, hydrological modeling techniques use DEMs to calculate the terrain slope and aspect to predict the divergence or convergence of the water flow [18]. High resolution DEMs can be obtained from sensing devices such as LiDARs mounted on airplanes or unmanned aerial vehicles (UAV), but this involves a high cost for relatively small area coverage. An alternative is the use of cheaper forms of data retrieval with near to global coverage like sensors mounted on satellites or space shuttles, like the SRTM and the Advanced Spaceborne Thermal Emission and Reflection Radiometer (ASTER) missions, which provided 30 meter resolu-

tion DSMs. Nevertheless, DSMs are different than DEMs as they include tree canopy in their elevation measures [7]. This affects water flow simulations as forested zones can be interpreted as higher ground areas and thus creating flow divergences, while water actually passes under the tree canopy. Because of this, DSMs need to be converted to DEMs performing a reduction of the measured elevation by an average of tree height on forested pixels.

Different efforts have been made to solve the problems described in SRTM and ASTER DEMs [5,9]. However, these solutions require manual adjustments or cannot be applied to different zones. The use of multispectral bands data provided by satellite-based sensors like Landsat-8 or Sentinel help with the detection of vegetation through calculating the NDVI as it is exposed by Ganie et al. [6] and Costa et al. [3]. However, NDVI is not accurate enough to discriminate between trees and other forms of vegetation. In order to avoid or reduce human intervention, accurate automatic detection and classification techniques for forest recognition are required.

Machine Learning algorithms are known for their ability to learn patterns from training samples and constructing models for automatic data classification and prediction. Different techniques have been proposed in the area of forest recognition. A Random Forest-based classification system to determine different species of trees in Australia was proposed by Mellor et al. [11]. Pimple et al. [12] present a similar technique to determine the forest types in the north-east mountains of Thailand. A Multilayer Perceptron (MLP) based technique is proposed by Wendi et al. [15] to recognize forested areas from SRTM and Landsat-8 data. Even though these techniques can improve the SRTM DEMs, they were not tested in wide scale areas, they lack accuracy or still require expert intervention. Deep Learning (DL) techniques have been applied to different image processing tasks with better results. In particular, Convolutional Neural Networks (CNN) is a DL technique widely applied in problems like object recognition and classification, noise detection and stereo matching, among many others [2,13]. This work uses a CNN based system as a binary classifier to create a forest/no-forest (FNF) binary mask, taking SRTM DSM and the different bands of the Landsat-8 satellite as inputs.

3 Data Preprocessing

A dataset was created joining information from the SRTM, Landsat-8 and JFNF projects. Each source of information is accommodated in a cube describing a zone of the Earth. In order to compile a dataset with homogeneous and normalized data, a series of preprocessing steps are required. The data is normalized to the $[-1, 1]$ range, without distorting the difference factor between values, using equation 1.

$$N(\mathbf{v}) = \begin{cases} RNG_m & v < SRC_m \\ RNG_M & v > SRC_M \\ RNG_m + \frac{(v - SRC_m) * (RNG_M - RNG_m)}{SRC_M - SRC_m} & otherwise \end{cases} \quad (1)$$

In the equation, \mathbf{v} is the matrix containing the raster values, v is the value to be normalized, RNG_m and RNG_M are the minimum and maximum values for the range, SRC_m and SRC_M are the minimum and maximum valid values for each input type of the data source (e.g. elevation, reflectance, etc).

The selected information from the SRTM and Landsat-8 missions, and the JFNF project come in different presentations, having distinct projection system, resolution, coverage area, etc. In consequence, each information source requires different standard preprocessing steps in terms of raster manipulation, like Digital Number conversion [14], multiple raster merging, reprojection, pixel size redimension, and raster clipping before performing the data normalization mentioned before.

Once each product has been preprocessed, a final cube is created conformed by a first layer with the SRTM normalized data, and the rest of the layers with the different Landsat-8 bands homogenized and normalized. Since the SRTM raster dimension is 3601x3601 and the Landsat-8 raster bands were clipped to this size, the final dataset cube is a 12x3601x3601 dimension structure representing the description of a $1^\circ \times 1^\circ$ zone of the Earth.

The FNF binary mask is a 3600x3600 matrix which is used as ground truth to train and compare the results of the proposed architecture. The missing row and column to match the SRTM raster size are simply omitted.

4 Data Analysis

After preprocessing the SRTM, Landsat-8 and JFNF data from 12 different zones covering an area from $25^\circ 0' 0''$ S - $53^\circ 0' 0''$ W to $28^\circ 0' 0''$ S - $49^\circ 0' 0''$ W, the training dataset is composed of 89,657,414 No-Forest and 65,862,586 Forest points.

Due to the amount of data available, training with all the points would take a significant amount of time. Therefore, a reduced training set was randomly extracted, whose size was determined by analyzing the distribution of the data. The different layers of the complete dataset were analyzed individually as each one has its own distribution. Smaller samples of different sizes were created randomly and the distribution of each one was compared to the distribution of the full dataset. In order to do that, a histogram of each sample was created, where the number and width of bins (columns in the histogram) were determined with the Freedman & Diaconis [4] rule. The generated histograms resulted in a high concentration of the data points in a relatively short range of values, with a shape similar to a normal distribution.

Even though the width of bins determined was good enough to analyze the data distribution, it also resulted in a large number of bins containing only a few items, as depicted in the left image of figure 1. These bins containing a low number of items represent a problem at the time of comparing a full dataset histogram with a reduced one because they show large relative deviations, which bias the comparison. To overcome this issue, an outlier detection method was used to determine the most representative bins, i.e. those containing a sufficiently large number of items. Following the Iglewicz & Hoaglin [8] work on this topic,

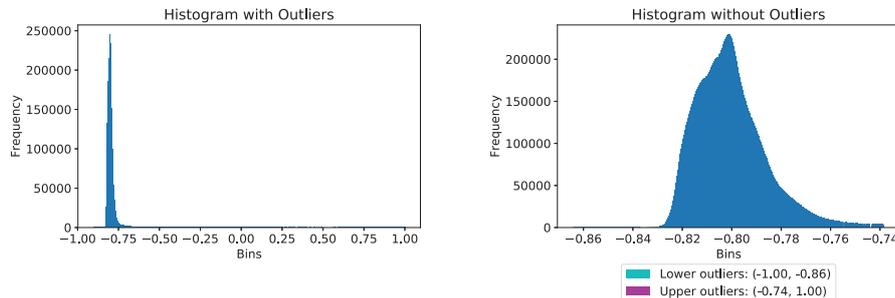


Fig. 1. Forest histograms of the layer 2 using Freedman & Diaconis rule to determine the bins width without outlier removal (*left image*), and the same histogram with outlier removal based on Robust Z-score and Median Absolute Deviation (*right image*).

a robust z-score method with a Median Absolute Deviation is used to determine the importance of each bin in the histogram as shown in equation 2.

$$MAD = median\{|x_i - \tilde{x}|\} \quad M_i = \frac{0.6745(x_i - \tilde{x})}{MAD} \quad (2)$$

M_i is the score of the histogram bin i , \tilde{x} is the median of the sample and 0.6745 is the 0.75th quartile of a standard normal distribution and x_i is the amount of elements in the bin i . Each M_i is then compared with a *threshold* in order to determine the lower and higher values where the bins start being considered as outliers. In our experiments, the suggested *threshold* of 3.5 by Iglewicz & Hoaglin was too restrictive. Instead, we used a *threshold* equal to 4.5 obtaining better results. The resulting histograms after the outlier detection and the determination of the most representative histogram bins range are depicted in the right image of figure 1.

Once the outliers were removed from the histograms, these were used to analyze the reduced samples. The relative deviation of the reduced samples with respect to the full dataset was calculated by comparing the percentage of elements in corresponding bins of the full and reduced datasets, and obtaining the relative absolute difference bin to bin. Once the relative deviations are calculated for each bin, the mean and median error of the histogram for each layer and class was obtained for each reduced dataset. This process was repeated 10 times per reduced sample size that ranged between 5% and 95% obtaining a final average mean and median deviation for each sampling size. Table 1 shows the results for a sample with 30% the size of the full dataset as an example.

The results obtained indicate that a random sample of 30% of the full dataset is the minimum required to ensure a mean and median deviation below 5% and 1% respectively in all the layers of the forest and no-forest classes with respect to the full dataset. The 30% sized dataset has a total of 26,897,225 randomly-selected points for each class, which were then used to train the network.

Table 1. Deviation of 30% sized sample w.r.t. full dataset by class and input layer

Class	Layer type	SRTM	Landsat 8										
		Elev.	B1	B2	B3	B4	B5	B6	B7	B8	B9	B10	B11
No Forest	Mean	1.76	4.58	3.82	2.02	1.71	1.48	0.90	0.84	1.75	3.36	2.17	1.74
	Median	0.65	0.59	0.60	0.55	0.49	0.54	0.54	0.50	0.52	0.73	0.68	0.64
Forest	Mean	1.77	2.54	2.39	2.64	2.44	1.07	1.68	1.65	2.28	2.73	1.52	1.70
	Median	0.58	0.54	0.55	0.60	0.57	0.61	0.54	0.53	0.58	0.68	0.65	0.72

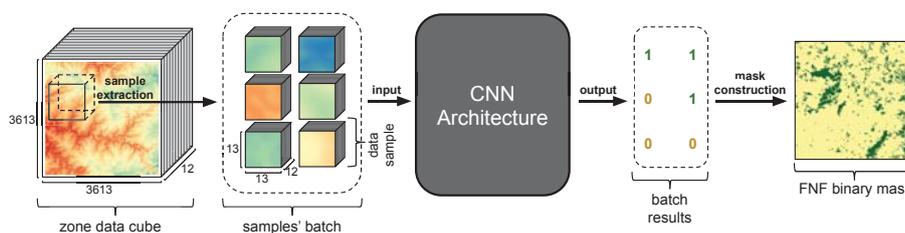
5 System Architecture

The model flow and steps are depicted in figure 2. SRTM rasters at a resolution of 30 meters cover a $1^\circ \times 1^\circ$ zone, which involves a data matrix of 3601×3601 . Additionally, including all the 11 bands of the Landsat-8 satellite clipped at the same size increases the input to a $3601 \times 3601 \times 12$ per zone. Because of the size of the input cube, in order to better use the hardware available the input is split in multiple samples of $13 \times 13 \times 12$ for each different center pixel.

Since the key piece of information is the pixel at the center of the sample, the rest of the surrounding pixels are considered a 6 pixel neighbor area helping to provide contextual information. In order to be able to process border pixels, each matrix of the zone data cube is padded with a 6 pixels border containing zeros, thus generating a $3613 \times 3613 \times 12$ size structure.

In case the model is working in testing mode, all the pixels of the zone input cube are sampled in order and provided in batches to the classification module where the class of each center pixel is predicted. In case the model is set to training mode, only a percentage of the input zone cube is sampled following a random order, but maintaining the same number of samples for each class. The position of each training sample in the zone data cube is also stored in order to retrieve the corresponding ground truth class from the JFNF mask, and then be able to compare the predicted class with the expected class.

With the purpose of providing an optimized classifier, a parameter search was performed over 32 different models, testing alternative architectures with a different number of convolutional layers, feature maps, dense layers, fully con-

**Fig. 2.** Proposed architecture input-output model

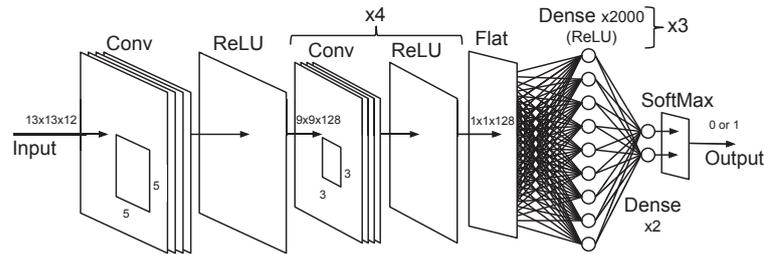


Fig. 3. CNN architecture composed of 5 convolutional layers with 128 feature maps, 3 dense layers with 2000 neurons, and a final SoftMax dense layer with 2 neurons.

nected neurons, kernel sizes at the input layer and alternative optimizers. The architecture described here is the one that presented the best validation accuracy, over a 10 epoch training with a 3 times cross-validation over a 30% reduced dataset.

This architecture is comprised of a series of convolutional layers each one followed by a Rectified Linear Unit (ReLU) activation layer, joining at the end with a fully connected neural network with a SoftMax binary classifier, as depicted in figure 3. Each convolutional layer has a kernel size of 3x3 with no padding and 1x1 stride (except for the first layer which has a 5x5 kernel), generating 128 feature maps. The model predicts the class of a certain pixel by evaluating the twelve different input channels corresponding to the SRTM DSM and Landsat-8 bands along with the neighbor pixels. According to the number of convolutional layers, their kernel size, padding, and size, the minimum input size of 13x13 window is provided creating a final input of 13x13x12. After the five Convolutional/ReLU layers a flattening layer is added in order to generate a vector of features to be fed into a fully connected neural network, which in turn is composed of 4 layers: three layers with 2000 neurons and ReLU activation function, and a last layer with 2 neurons using a SoftMax activation function. The model provides a single value output indicating if the pixel in the center of the input window corresponds to the Forest or No-Forest class.

The classifier is trained using a binary cross-entropy loss function and Stochastic Gradient Descent optimization technique.

The construction module is executed only in testing mode once the classifier module has finished predicting all the pixels' class. It simply reshapes the single vector of results to a matrix of size 3601 x 3601 creating in this way a mask where each position corresponds with a position of the SRTM raster used in the input.

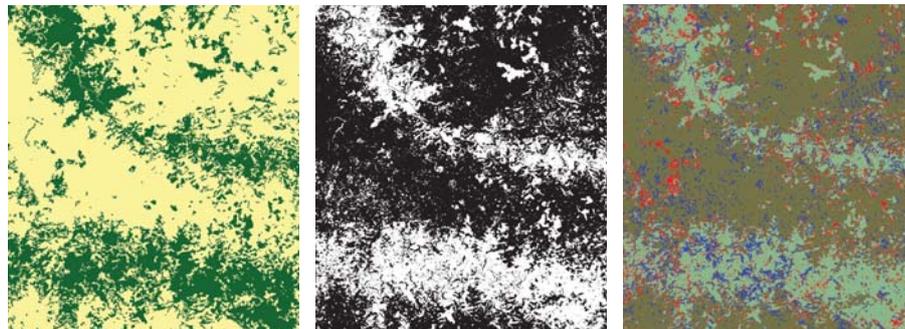


Fig. 4. Zone 13 of the JFNF mask (*left*), our FNF mask (*center*) and comparison (*right*) where correctly predicted points are shown in brown and green (No-Forest and Forest respectively), while errors are shown in blue and red respectively.

6 Case Study

The experiments were performed on an AMD Ryzen 1700 CPU with 32 GB DDR-4 2400 MHz RAM and a NVidia Titan Xp. The source code and the data used in this work are publicly available⁴.

The training samples were created by obtaining the 13x13x12 data cubes at each position of the dataset and were provided in batches of 512 samples to the training algorithm. A k-fold cross-validation training schema was used with $k = 5$. A selection of 45% of the dataset points was randomly selected maintaining the balance between No-Forest and Forest points. Since KFold splits 45% of the points into 5 parts, 36% of the points are selected for training/validation and 9% for test on each of the 5 runs. Then, the training/validation set is split in 85% for training and 15% for validation, generating a 30.6%/5.4% selection from the full dataset respectively and thus, maintaining a higher than 30% dataset selection for training as it was analyzed in the section 4. The network was trained over 30 epochs with a learning rate of 0.01 and default parameters following the recommendations for Stochastic Gradient Descent in [1]. In our tests, other optimizers like Adam produced vanishing gradients so they were discarded.

The classification module obtained an average test accuracy of 91% during training. The results of the KFold cross-validation in terms of average precision, recall and F1-score (with their standard deviations) for the No-Forest class were 91.4% (1.5%), 91.4% (1.5%) and 91.2%(0.4%), while the Forest class obtained 91.4% (0.9%), 91.4% (1.5%) and 91.4% (0.5%) respectively. Then the model was tested against the test zone 13 composed of 8,546,237 No-Forest and 4,413,763 Forest points with an accuracy of 87.36% taking 352 seconds.

The JFNF mask and the one produced by our network for zone 13 are shown and compared in the figure 4. One thing to notice is that in some cases the

⁴ <https://github.com/labsin-uncuyo/py-cnn-geo>

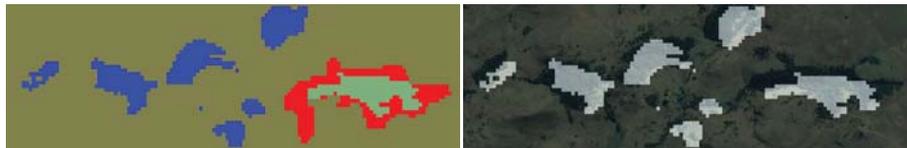


Fig. 5. Small area of the comparison between JFNF mask and our FNF mask showing multiple errors in our results in blue and red (*left image*), but correctly detecting real forested areas omitted by the JFNF mask (*right image*).

network was capable of recognizing real forest points that were not correctly labeled in the JFNF mask (figure 5).

7 Conclusion and Future Work

In this paper, we presented a CNN architecture capable of learning to recognize forested areas from different inputs obtained by remote sensing devices. The model generated a binary mask with an 87.36% accuracy compared with the JAXA FNF mask in less than 6 minutes for $1^\circ \times 1^\circ$ coverage areas. After a detailed analysis of the results, we found that the network is able to recognize forested areas omitted in the ground truth mask, denoting a potential benefit of using our approach even for improving the available data. Also, these misclassified points in the ground truth mask could be introducing noise in the training phase of the network, so we think the results can be further improved by using better data for training.

Future work includes testing additional network models, training and optimization parameters in order to find better classification architectures. Also, we will explore different post-processing techniques for improving the quality of the binary mask and reducing the noise. In addition to this, the proposed solution should be tested in areas with an imbalanced proportion of Forest and No Forest points, or different land characteristics in order to assess the strengths and weaknesses of the model in more difficult scenarios. An analysis of the significance of input layers on the result can help to determine unneeded input data and result in faster and more accurate execution of the network. Finally, other satellite-based data inputs can be explored, like Sentinel mission data, to verify the model's sensitivity to different data sources.

Acknowledgments. We want to gratefully acknowledge the support of NVIDIA Corporation with the donation of the Titan Xp GPU used for this research.

References

1. Bengio, Y.: Practical Recommendations for Gradient-Based Training of Deep Architectures. In: Montavon, G., Orr, G., Müller, K. (eds.) *Neural Networks: Tricks*

- of the Trade. Lecture Notes in Computer Science, vol 7700, pp. 437–478. Springer, Berlin, Heidelberg (2012)
2. Caffaratti, G., Marchetta, M., Forradellas, R.: Stereo matching through squeeze deep neural networks. *Inteligencia Artificial* 22(63), 16–38 (Feb 2019)
 3. Costa, S., Santos, V., Melo, D., Santos, P.: Evaluation of landsat 8 and sentinel-2a data on the correlation between geological mapping and ndvi. In: 2017 First IEEE International Symposium of Geoscience and Remote Sensing (GRSS-CHILE). pp. 1–4 (June 2017)
 4. Freedman, D., Diaconis, P.: On the histogram as a density estimator:L2 theory. *Zeitschrift für Wahrscheinlichkeitstheorie und Verwandte Gebiete* 57(4), 453–476 (1981)
 5. Gallant, J., Read, A., I. Dowling, T.: Removal of tree offsets from srtm and other digital surface models. *ISPRS - International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences XXXIX-B4*, 275–280 (07 2012)
 6. Ganie, M., Nusrath, A.: Determining the vegetation indices (ndvi) from landsat 8 satellite data. *International Journal of Advanced Research* 4, 1459–1463 (09 2016)
 7. Gesch, D., Oimoen, M., Danielson, J., Meyer, D.: Validation of the aster global digital elevation model version 3 over the conterminous united states. *ISPRS - International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences XLI-B4*, 143–148 (06 2016)
 8. Iglewicz, B., Hoaglin, D.C.: *How to Detect and Handle Outliers* (1993)
 9. kuan Liu, J., Liu, D., Alsdorf, D.: Extracting ground-level dem from srtm dem in forest environments based on mathematical morphology. *IEEE Transactions on Geoscience and Remote Sensing* 52(10), 6333–6340 (Oct 2014)
 10. M. Lillesand, T., Kiefer, R., W Chipman, J.: *Remote Sensing and Image Interpretation (Seventh Edition)* (01 2015)
 11. Mellor, A., Haywood, A., Jones, S.D., Wilkes, P.: *Forest classification using random forest with multisource remote sensing and ancillary gis data* (2014)
 12. Pimple, U., Sitthi, A., Simonetti, D., Pungkul, S., Leadprathom, K., Chidthaisong, A.: Topographic correction of landsat tm-5 and landsat oli-8 imagery to improve the performance of forest classification in the mountainous terrain of northeast thailand. *Sustainability* 9 (02 2017)
 13. Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., C. Berg, A., Li, F.F.: Imagenet large scale visual recognition challenge. *International Journal of Computer Vision* 115 (09 2014)
 14. USGS: Landsat 8 (L8) Data Users Handbook (LSDS-1574), <https://www.usgs.gov/media/files/landsat-8-data-users-handbook>
 15. Wendi, D., Liang, S.Y., Sun, Y., Dung Doan, C.: An innovative approach to improve srtm dem using multispectral imagery and artificial neural network. *Journal of Advances in Modeling Earth Systems* 8 (04 2016)
 16. Wilson, J.: Digital terrain modeling. *Geomorphology* 137 (01 2012)
 17. Wilson, J., Gallant, J., Hutchinson, M.: Future directions for terrain analysis, pp. 523–527 (01 2000)
 18. Zhou, Q., Pilesjö, P., Chen, Y.: Estimating surface flow paths on a digital elevation model using a triangular facet network. *Water Resources Research* 47(7) (2011)

HeartCare: an Agent Oriented Architecture Implemented with Actors*

Mariana Falco¹[0002–0959–7435], Leandro Bauret², and Gabriela Robiolo³

¹ LIDTUA/CONICET, Engineering Faculty, Austral University
mfalco@austral.edu.ar

² Engineering Faculty, Austral University
leandro.bauret@ing.austral.edu.ar

³ LIDTUA, Engineering Faculty, Austral University
grobiolo@austral.edu.ar

Abstract. A priority in cardiac patients is the performance associated to a regular follow-up, when they leave the hospital to keep track on their scheduled physical activities. Unfortunately, not all patients begin a rehabilitation program, or if they are part, they do not perform all of the exercises. Consequently, a multi-agent system was designed and developed to enable patients with heart conditions and their doctors to achieve a feasible out-of-hospital follow-up; while monitoring their heart-rate measures and physical workout. The present paper describes how the architecture was built in which Akka and Scala have been used to represent the agents through the actors model. We can conclude that it was a good decision to have used the agent-oriented paradigm for the future scalability of the application (both for the increase in the number of sensors and the possibility of agents with beliefs), but not for the CRUD operations. Future work will include a new development of Heart Care, adding more sensors and newer layers of intelligence.

Keywords: Software agents · Actors · Monitoring of patients.

1 Introduction

Between 2005 and 2015 in Argentina, the number of deaths from heart attacks was increased by 18 percent. It is estimated that between 40,000 and 50,000 strokes occur each year; and of that total, 17,000 people die. The main causes are the delay in the attention or intervention of a doctor in time due to the large number of patients who appear in hospitals [17] and in-hospital mortality. Due to a problem of organization and the lack of adequate technology for more efficient assistance to a larger number of people, the number of deaths due to cardiac disorders is increasing [1].

* The authors thank Tomás Fernández Martínez, Ignacio Berdiñas, Nicolás Velázquez Bustamente, Matías Cicilia, Ignacio Vázquez, and Agustín Masuelli, who collaborated in the development and design of HeartCare. This work is supported by a Research Grant from Austral University.

A major priority in cardiac patients is the performance related to a regular follow-up, while improving the level of medical care [13]. Consequently, an important goal for achieving an improvement with a cardiac patients physical fitness is for them to realize regular physical activities [2, 3], because one of the most important risk factors is physical inactivity [4, 5]. Some patients do not access a cardiac rehabilitation program, do not complete their weekly physical routines, or do not workout at all [6, 7]; for several reasons like lack of transportation, the patient is unaware - he or she does not have a confirmation of the follow-up, lack of support, economic issues, among others.

As technology progresses, its uses entail the creation of various paradigms such as u-healthcare, pervasive healthcare, and others, which allow to move the practice of doctor's office to the patient's daily life [8, 9], through mobile devices, various sensors, smart-glasses, among others. Specifically in cardiac patients, there are various web / mobile applications [10–12] that allow them to control their symptoms, while achieving adherence to their treatment through the management of medications, physiological values, possible symptoms and how to act in front of them, physical exercise routines, and even by reading healthy tips.

There are different studies that approaches this problem from several disciplines, through for example, a system for out-of-hospital follow-up of cardiac patients that belong to one of four risk groups (arterial hypertension, malignant arrhythmias, heart failure, and post-infarction rehabilitation) [13], a multi-agent architecture for heart failure management in a home care environment [18], and an ontology-based framework to support cardiac rehabilitation program [14].

Within Artificial Intelligence, multi-agent systems [21] helps with the development of complex systems, and healthcare is a complex environment. Patient management has seen the most applicability of agents in healthcare [23], although nowadays the whole spectrum is covered. Isern and Moreno [15] define seven types of agent-based applications based on different domains: data-management systems, secure platforms, decision support systems, planning, simulation, care platforms, and monitoring and alarms. The utilization of sensors and wearable devices which monitors the actual state of the patients helps improving the living-at-home systems or ambulatory activities, in order to assist the patients on improving their condition [15].

The present paper tries to answer the following research questions: (RQ1) Is the multi-agent paradigm suited for the application for cardiac rehabilitation?; and (RQ2) What is the most appropriate framework to build the application?. In this context, we have developed an agent-based application called HeartCare that allows the monitoring and medical care of cardiac patients, in order to support ubiquitous contact between chronic patients presenting cardiovascular risk and their doctors or therapists, directly involved in their out-of-hospital follow-up.

The possibility of registering several parameters according to the needs of the patients or the pathology that determines the clinical condition of the patient in ambulatory or domiciliary form is a novelty. It will allow to see possible changes in the evolution of the disease, either improvements with medication or

deterioration in the absence of adequate treatment / diagnosis, and over time to establish predictive values of certain parameters according to the evolution of the disease by conducting studies.

Our main goal here is to present the architecture of the HeartCare application while discussing the decisions made with respect to the agent-oriented paradigm, the technologies, and its implementation. Also, specifying the agents developed, and each of their goals and behaviors. HeartCare is a prototype version of the architecture, and we have implemented the actor's model to represent the agents through Akka [28].

The work is structured as follows: section 2 will describe the application, the medical context addressed, the functionalities implemented and the technologies used. Section 3 will address the definition of the architecture, specifying the agents implemented, and the cooperation between them by means of a sequence diagram. Section 4 will describe the conclusions and future work.

2 HeartCare: Scope and Technologies

2.1 Understanding the scope

The HeartCare application aims to assist in home-based rehabilitation of patients who have suffered cardiovascular accidents, specifically in out-of-the-hospital follow-up. The main objective is to ensure that the recovery can take place in an environment outside hospitals, without the need to attend monthly appointments with a doctor at hospitals with the only purpose of doing a follow-up of the patient's health status and establish the next steps to follow.

Within the medical context and particularly in cardiology, there are two main classifications that place patients on a category based on their limitations during physical activity [11]. The first one, a three phase classification that defines the treatment for patients who have suffered some cardiac affection, and the second one, four classes derived from the NYHA functional classification. In this context, the HeartCare application will seek to accompany patients in phase II (active phase or supervised exercises) and phase III (maintenance phase), and also class I patients that are able to perform physical activity autonomously without affecting their health.

According to the application's goal, it is possible to determine two main users: patient and health staff, where the latter are responsible for monitoring the clinical history, and health condition of the patient. Being HeartCare a mobile and web application, it can be used on computers, mobile devices or tablets. Both patients and specialists require an internet connection in order to use the application. HeartCare provides a mobile patient-oriented application to assist patients in their cardiac recovery.

For the health specialists, HeartCare has a staff-oriented web version to track the health condition and status of their patients. The defined use cases are grouped by functionality with the following criteria:

- **User management:** It allows the patient and the doctor to create a user, and also to login and logout.

- **Measure management:** It allows the patient to add new measurements on the mobile application, and to the doctor as well on the web version.
- **Routine management:** It allows the patient to visualize the set of routines he/she has to do, visualize each exercise, interact with the routine (play, pause, stop), and fill the Borg Scale (explained below).
- **Specialist management:** It allows the doctor to visualize patients and their data, to see measurements registries, and interact with trends graphs.

The patient through his or her smart-phone can visualize the routine defined by its doctor. Each routine contains a set of exercises (like warm-up, stretching, walking, among others) on a specific order, the amount of series or repetitions for each exercise, and the expected Borg Scale - set by the Doctor. Once the patient finishes an exercise (all of the series within each exercise), the application will ask the user to rate the level of perceived exertion according to the Borg Scale.

This scale relatively matches how the patient is feeling while working out, through a number from 6 (none) to 20 (extremely hard). The selection of the numbers (6 to 20) by its creator Dr. Gunnar, was due to a simple way to estimate heart rate by multiplying the number or score selected by 10 which gives an approximate heart rate value for a particular level of activity [24].

Based on the score selected for each exercise, when the level is hard, the patient is having a symptom like dizziness, the heart rate levels are not regular for a person performing physical exercise, the Agent Patient considers that the patient is at risk and so, an alarm to the doctor is sent for him to analyze the severity, and change the plan of exercises or ended due to complications.

2.2 Technologies used

Once the scope of the application was defined, it was necessary to decide which technologies were right for the requirements of the application (web version, mobile version, connection to a heart rate sensor through Bluetooth, monitoring in real time, increased concurrency with more sensors). After analyzing agent-oriented technologies, reading documentations and following some test cases, the team decided that the agent-oriented technologies, most of them described by Agent-Oriented Software Engineering (AOSE), were not a perfect fit for Heart-Care. So, the main difference with most of the developed agent-oriented applications [22] is that we did not choose JADE [27].

Concurrent systems are composed of a large number of computations that run simultaneously and can interact with each other [25]. The execution of computations in concurrent systems can be done in different ways, therefore, several models have emerged that allow the modeling of concurrent systems. One of them is the model of actors, adopted by Scala; in which actors can be seen as the replacement of threads in traditional multi-threaded applications, which are able to achieve interaction with other actors through the concepts of message passing. A key point within this model is that an actor can react to a message received in three different ways: a) creating a finite number of new actors, b)

sending a finite number of messages, and c) controlling its internal state by updating its mutable attributes [30]. HeartCare is expected to have a high number of sensors transmitting at the same time.

Agent-based systems require frameworks with high performance, and concurrent simulation, therefore, it is possible to represent agents through actors. The authors in [26] have implemented an evolutionary multi-agent system (EMAS), while introducing the concept of meeting arenas which allows a more scalable implementation of MAS. A meeting arena is defined by [26] as: Every time an agent wants to perform an action, it chooses an appropriate arena to meet with other similar agents. In their work, the agents and the arenas have been implemented in Akka [28], where the actors execute asynchronously and communicate through message passing.

Based on the above, the second main difference is that we selected an actor-oriented framework (Akka) that allowed us to build the agents by implementing actors and classes. We modeled a multi-agent system with different intelligent agents, with the particular concepts of agency (like flexible behavior, beliefs, desires, and intentions, communication through ACL messages, among others), but as a point of implementation we have coded actors with a layer of intelligence in order to achieve these concepts. One important characteristic is the emergent behavior that arose as a result of the evolution of the patients, evidenced by vital signs and medical data records, which we hope will be a consequence of the analysis of the recorded data.

As such, and in particular for the web version, TypeScript was used as the main language, making use of a web development framework called Angular combined with a web interface framework called UIKit , and a useful library for the creation of graphics named Angular2 Charts. For the mobile version, Javascript was used as the main language, using a native application development framework called React Native. The back-end consists of Scala as the programming language, Akka as a development framework building an agent model with the actor model, Slick as the ORM (Object-Relationship Mapping), and PostgreSQL as a database system. Finally, the sensor used for the implementation of HeartCare is a Polar H10 heart rate sensor [29] that allowed us to control the heart rate with high precision and connect it simultaneously to Bluetooth devices.

3 Describing the architecture of HeartCare

The architecture of the application can be seen in **figure 1**, which is composed of three modules: the mobile client, the web client, and the server consisting of the multi-agent system and the database. The mobile client is responsible for serving the patient's data through a mobile application so that he or she can perform the routines corresponding to his or her recovery. Patient data is collected using the Polar H10 heart rate sensor [29], and sent from the mobile client to the system. On the other hand, the web client provides health specialists with a user interface that allows them to view information on the status and progress of

their patients. Finally, the server is comprised of the multi-agent system that is responsible for making decisions and queries of information to the database.

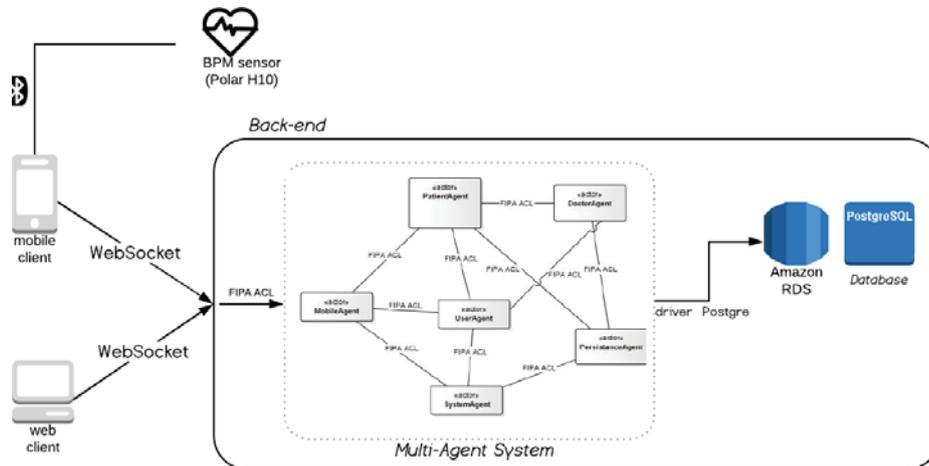


Fig. 1. The HeartCare architecture is displayed by showing each of the modules.

3.1 Specification of the agents

Ferber described that a multi-agent system '(...) contains an environment, objects and agents (the agents being the only ones to act), relations between all the entities, a set of operations that can be performed by the entities and the changes of the universe in time and due to these actions (...) [31]. In our context, the built MAS is composed of the following agents: MobileAgent, PatientAgent, HealthAgent, PersistenceAgent (modelled as a generalization relation with the PersistenceAgent as the superclass, and UserPersistenceAgent, PatientPersistenceAgent, RoutinePersistenceAgent, and MeasurePersistenceAgent as the subclasses), and System Agent.

All of the agents possess a mixed design between the Touring Machine Architecture [19] and the Layered Horizontal Architecture [20]. The layers resemble the Touring architecture but differ in that only one determines the response to input, as in the horizontal layer architecture. The agents are specified as follows:

Persistence agents: These five agents are responsible for persisting the database, and carrying out the CRUD operations of entities. They are the ones which know most about the domain of the problem since they understand all the entities that compose it. They do not have a planning layer since they do not make decisions based on their internal state.

MobileAgent: The MobileAgent contains the reference to the mobile application and is responsible for answering all customer requests and sending the

necessary data to be stored in the agent system. Almost all its behavior is reactive. It has two layers, one reactive and the other planning.

SystemAgent: The SystemAgent is in charge of controlling the creation of the rest of the agents in the system, and it contains the context of the application and the system of existing actors. It is responsible for managing the user authentication of the system, receiving requests for user creation and login. It has a reactivity layer much larger than that of planning, and it does not have a model layer since it does not know about the domain of the problem.

UserAgent: The UserAgent represents the operations on the logged in user. This agent modifies its behavior based on whether the user is a patient or a health agent type. In the case that the user is a patient, then the one who will respond to the messages will be the PatientAgent. Otherwise, if the user is a health agent, he will adopt the behavior of a DoctorAgent. This is because the functionalities and messages available to a patient are not the same as those of the doctor, since they cannot access the same data and have different objectives per se.

DoctorAgent: The structure of the DoctorAgent is similar to that of the PatientAgent. Even though its planning layer is not currently developed, it aims in the future to allow the doctor to assemble the routines based on the objectives of each patient. The main objective of this agent is to achieve a successful cardiac recovery for all his patients following their particular exercise planning and schedule. It is worth mentioning that its structure is equal to the Turing Machine. This agent retrieves the information of the patients of a certain specialist; and is responsible for receiving new observations that the doctor logs into the system for a particular patient.

PatientAgent: The PatientAgent is the one that has as an explicit objective of a proper cardiac rehabilitation of the patient it represents. This is why when starting an exercise, the agent will be responsible for continuously analyzing the patient's condition and the context in which it is located and will make the best possible decision in order to achieve the recovery of the user. This is done through a BDI architecture [16], where the agent retrieves the available data about the user in question, analyzes what were his intentions through the history of execution of patient routines and, in response, executes the best plan that fits the agent's wishes, a plan that is fit in the best way to the wishes of the agent since at all times the PatientAgent will thrive in order to accomplish its objective. In case the agent interprets that the patient is in danger, it will react in such way to prevent and mitigate the risk to which the user is exposed. Otherwise, if the patient is in normal conditions, then the agents behavior of the patient will be in accordance with what was planned to achieve his goal.

3.2 Lessons learned

Regarding the first research question (Is the multi-agent paradigm suited for the application for cardiac rehabilitation?), it is possible to say that the decision to implement the agent-oriented paradigm (AOP) for the HeartCare application was appropriate in the following topics: (a) scalability, in the future new sensors

will be added to improve the intelligence as part of the analysis of the patients condition, and it will also increase the concurrency between them; (b) considering the concurrent use of the system by several users, AOP allows a separate and independent thread of control that represents the state of each user and has modeled as its only desire its specific recovery, seeking to achieve their design objectives based on the data collected about the patient. The biggest disadvantage was observed when implementing the CRUD operations, which require the communication of multiple agents to only perform an update in the database; consequently a simple task is complexed enormously and the process itself ends up being inefficient.

Numerous developments and applications have been made for MAS and although, JADE [27] was found as the most used framework [22], there is a great dispersion in terms of quantity of existing frameworks, the community tends to create new ones instead of unifying existing ones. In this context, and regarding the second research question (What is the most appropriate framework to build the application?), it is feasible to mention that before a rigorous study and analysis, HeartCare was not implemented with any of the existing agent-oriented framework, methodology or programming language. Instead, it was implemented with day-to-day technologies like Akka and Scala. JADE was not selected as the framework, due to its deprecated Android version, the deficiencies of the SOAP protocol, the lack of future maintainability, even though it is open source it does not have a public repository on Git, and it is a verbose framework. One of the main characteristics is that this technological decision allowed us to implement actors while constructing a MAS, by using Akka. We can conclude that it was a good decision to implement the project in Akka since it streamlined the development mainly due to its greater documentation and the large number of people using this framework nowadays.

4 Conclusions

Healthcare systems are complex and highly dynamic, they manage distributed data and resources, and they usually need to achieve interoperability with different information systems. Likewise, the healthcare professionals and staff need a high mobility within the hospital or health institution [23]. As such, several applications have been made to contribute to these topics [22, 15]. In the present paper, a mobile and web application called HeartCare has been presented which main goal is to assist in the home rehabilitation of patients who have suffered cardiovascular accidents. The architecture was built with Akka and Scala have been used to represent the agents through the actors model.

From the realization of the architecture, and as future work, the following proposals arise to improve the current implementation: (a) develop the BDI logic of the HealthAgent so that it can create general plans, and even specific plans taking into account the current state of health and the patient's medical history; (b) improve the planning layer of the PatientAgent so that it is more related to the execution of the patient's routine; (c) add several sensors to represent with a

higher accuracy the state and the environment in which the patient is, improving the system's robustness; (d) train even more the intelligence of the agents for the determination of the routines and a more precise recognition of the danger thresholds; and (e) the use of machine learning in order to identify behavior patterns while monitoring the patients.

Consequently, HeartCare will be able to propose more or less demanding routines, increasing the level of difficulty as he considers it appropriate to achieve his main objective in the most effective way. Also, it is expected that a patient's agent will interact with that of another patient and provide a team routine, selecting a routine in common to meet the objectives of both users.

References

1. Clarin, (2017), En 10 años crecieron 18 las muertes por infartos y apuntan a las demoras en la atención. https://www.clarin.com/sociedad/10-anos-crecieron-18-muertes-infartos-apuntan-demoras-atencion_0_rkN5SyTU-.html Last access: 23/06/2019
2. Fletcher, B., Magyari, P., Prussak, K., and Churilla, J. (2012). Entrenamiento físico en pacientes con insuficiencia cardíaca. *Revista Médica Clínica Las Condes*, 23(6), 757-765.
3. Leon, A. S., Franklin, B. A., Costa, F., Balady, G. J., Berra, K. A., Stewart, K. J., ... and Lauer, M. S. (2005). Cardiac rehabilitation and secondary prevention of coronary heart disease: an American Heart Association scientific statement from the Council on Clinical Cardiology (Subcommittee on Exercise, Cardiac Rehabilitation, and Prevention) and the Council on Nutrition, Physical Activity, and Metabolism (Subcommittee on Physical Activity), in collaboration with the American association of Cardiovascular and Pulmonary Rehabilitation. *Circulation*, 111(3), 369-376.
4. Platter, M., Hofer, M., Hlzl, C., Huber, A., Renn, D., Webb, D., Hfer, S. (2016). Supporting cardiac patient physical activity: a brief health psychological intervention. *Wiener Klinische Wochenschrift*, 128(5-6), 175-181.
5. Doukky, R., Mangla, A., Ibrahim, Z., Poulin, M. F., Avery, E., Collado, F. M., ... and Powell, L. H. (2016). Impact of physical inactivity on mortality in patients with heart failure. *The American journal of cardiology*, 117(7), 1135-1143.
6. Redfern, J., Ellis, E. R., Briffa, T., and Freedman, S. B. (2007). High riskfactor level and low riskfactor knowledge in patients not accessing cardiac rehabilitation after acute coronary syndrome. *Medical Journal of Australia*, 186(1), 21-25.
7. Ferrier, S., Blanchard, C. M., Vallis, M., and Giacomantonio, N. (2011). Behavioural interventions to increase the physical activity of cardiac patients: a review. *European Journal of Cardiovascular Prevention Rehabilitation*, 18(1), 15-32.
8. Abdullah, M. R., Salim, S., and Jamil, M. M. A. (2013, November). Telemedicine system: Wireless healthcare units via RF, GSM, Bluetooth and PDA. In *2013 IEEE International Conference on Control System, Computing and Engineering* (pp. 215-220). IEEE.
9. Bennett, J., Rokas, O., and Chen, L. (2017). Healthcare in the smart home: A study of past, present and future. *Sustainability*, 9(5), 840.
10. Telemadrid, (2015), Desarrollan una 'app' para la rehabilitación cardíaca domiciliaria. <http://www.telemadrid.es/noticias/madrid/Desarrollan-app-rehabilitacion-cardiaca-domiciliaria-0-1659434062-20150223024815.html>. Last access: 23/06/2019.

11. American Heart Association, (2017), HF Path, <http://www.heart.org/en/health-topics/heart-failure/heart-failure-tools-resources/hf-path-heart-failure-self-management-tool>. Last access: 23/06/2019
12. Heart Failure Society of America, (2017), Heart Failure Society of America Mobile Web App: Keeping Track of Your Health Just Got Easier! <https://www.hfsa.org/patient/patient-tools/patient-app/>, Last access: 23/06/2019
13. Salvador, C. H., Carrasco, M. P., De Mingo, M. G., Carrero, A. M., Montes, J. M., Martin, L. S., ... and Monteagudo, J. L. (2005). Airmed-cardio: a GSM and Internet services-based system for out-of-hospital follow-up of cardiac patients. *IEEE Transactions on information technology in Biomedicine*, 9(1), 73-85.
14. Waqiialla, M., and Razzak, M. I. (2016). An Ontology-based framework aiming to support cardiac rehabilitation program. *Procedia Computer Science*, 96, 23-32.
15. Isern, D., and Moreno, A. (2016). A systematic literature review of agents applied in healthcare. *Journal of medical systems*, 40(2), 43.
16. Croatti, A., Montagna, S., Ricci, A., Gamberini, E., Albarello, V., and Agnoletti, V. (2018). BDI personal medical assistant agents: The case of trauma tracking and alerting. *Artificial intelligence in medicine*.
17. Isern, D., Moreno, A., Snchez, D., Hajnal, ., Pedone, G., and Varga, L. Z. (2011). Agent-based execution of personalised home care treatments. *Applied Intelligence*, 34(2), 155-180.
18. Koutkias, V. G., Chouvarda, I., and Maglaveras, N. (2003, September). Multi-agent system architecture for heart failure management in a home care environment. In *Computers in Cardiology, 2003* (pp. 383-386). IEEE.
19. Arango, M., Bahler, L., Bates, P., Cochinala, M., Cohrs, D., Fish, R., ... and Lee, K. C. (1993). The touring machine system. *Communications of the ACM*, 36(1), 69-77.
20. Miller, J. P., Pischel, M., and Thiel, M. (1994, August). Modeling reactive behaviour in vertically layered agent architectures. In *International Workshop on Agent Theories, Architectures, and Languages* (pp. 261-276). Springer, Berlin, Heidelberg.
21. K. Hindriks (2014), The shaping of the agent-oriented mindset. In: *International Workshop on Engineering Multi-Agent Systems*, pp. 1-14.
22. Miller, J.P., and Fischer, K.(2014), Application Impact of Multi-Agent Systems and Technologies: A Survey. In: *Agent-Oriented Software Engineering: Reflections on Architectures, Methodologies, Languages, and Frameworks*. pp. 27-53. Springer, Berlin, Heidelberg.
23. Annicchiarico, R., Garcia, U. C., and Urdiales, C. (Eds.). (2008). *Agent technology and e-Health*. Springer Science Business Media.
24. Borg G.A. (1982) Psychophysical bases of perceived exertion. *Medicine and Science in Sports and Exercise*. 14:377-381.
25. Roscoe, A. W. (2010). *Understanding concurrent systems*. Springer Science Business Media.
26. Krzywicki, D., Byrski, A., and Kisiel-Dorohinicki, M. (2014). Computing agents for decision support systems. *Future Generation Computer Systems*, 37, 390-400.
27. JAVA Agent DEvelopment Framework, <http://jade.tilab.com/>". Last access: 23/06/2019
28. Akka, <https://akka.io/>, Last access: 23/06/2019
29. Polar H10 Heart Rate Sensor, <https://www.polar.com/en/products/accessories>, Last access: 23/06/2019.
30. Bharti, R. *HIVE: An Agent Based Modeling Framework*, Thesis - San Jose State University, 2016.

31. Ferber, J., and Weiss, G. (1999). Multi-agent systems: an introduction to distributed artificial intelligence (Vol. 1). Reading: Addison-Wesley.

Multicore Parallelization of CHC for Optimal Aerogenerator Placement in Wind Farms

Martin Bilbao¹, Guillermo Leguizamon²

¹ Universidad Nacional de la Patagonia Austral, Caleta Olivia, Argentina
mbilbao@uaco.unpa.edu.ar

² Universidad Nacional de San Luis, Argentina
legui@unsl.edu.ar

Abstract. In this paper, we study a parallelization of CHC algorithm (Crossover elitism population, Half uniform crossover combination, Cataclysm mutation) to solve the problem of placement of wind turbines in a wind farm. We also analyze the solutions obtained when we use both, the sequential and parallel version for the CHC algorithm. In this case we study the behavior of parallel metaheuristics using an island model to distribute the algorithm in different cores and compare this proposal with the sequential version to analyse the number of evaluation to find the best configuration, output power extracted, plant coefficient, evaluations needed, memory consumption, and execution time for different number of core and different problem sizes.

Keywords: Wind Energy, Weibull Distribution, Wind Power, Evolutionary Computation, Metaheuristics

1 Introduction

Nowadays, the importance of environmental care is one of the most addressed issues in society and the main agenda of governments. For that we are being increasingly using renewable energy with a special care on avoiding the emission of gases to the atmosphere. Today a kind of energy that is moving around the world is the wind energy, which basically transforms the wind energy into electrical energy. It is designed for large-scale wind farms. A wind farm is a set of wind turbines distributed so that maximize wind energy contained in the wind and turns it into usable electricity. The capital interest is to produce a maximum of energy at the same time as reducing the total cost of the wind farm and choose its position is a strategic decision to minimize the *wake effect* [1] in order to maximize the produced energy. The goal in this paper is obtain a better configuration of the wind farm using parallel metaheuristics and analyze different scenarios to understand the behaviour of our algorithm.

Over the past years other kind of algorithms were used to solve similar problems, Simulated Annealing and Distributed Genetic Algorithms [2][3]. In a previous work we used CHC and GPSO considering constant North wind [4], and CHC and Simulated Annealing considering the real wind distribution and flat terrain [5]. Now, we design a parallel algorithm to solve two big scenarios using constant wind and regular terrain. The objective of this study is introducing different versions of a parallel algorithm to solve more complex and realistic instances. We analyze the best farm configuration found, fitness value, produced power, efficiency, performance of the algorithms in terms of their running time and number of evaluations needed to obtain the best solution, speed-up and parallel efficiency.

The rest of the article is structured as follows: Section 2 explains the wake model, power model, and cost model used. Section 3 will detail the algorithm used. Section 4 describes the algorithms for wind farm design, objective function and representation of wind turbine locations. In Section 5 we will detail the analysis of sequential algorithm and its results. In section 6 we describe the parallel design of the algorithms and technologies used for this problem. In section 7 we will explain the results obtained for the parallel design and finally Section 8 summarizes the conclusions and future work.

2 Wind Energy Optimization Problem

In this section we describe the mentioned inter-turbine wake effect model, the power model, and the cost model for our further mathematical manipulations. These are the basic components to deal with a realistic farm design, and they are combined together into an objective for the needed guidance of the function algorithms in their quest for an optimal farm configuration.

Design a wind farm efficiently is a very important work to generate the maximum energy possible, it depends of the relative distribution of the wind turbines for obtaining an optimal geometry of the wind farm. Wind turbines receive lower wind speed and less energy captures if they are located behind others. This effect called *the wake effect*[1] can be reduced by optimizing the geometry of the wind farm. The environment could has irregularities as lakes, mountains, forbidden places. Therefore, we need to design a wind farm with that conditions, so in this work we pursue the following goals, maximize the power output extracted and minimize the cost of installation.

2.1 Wake Effect Model

The used model in this work is similar to the wake decay model developed by Katic's[6]. Depending on the farm geometry the wind turbines that are upwind of other wind turbines result in lower wind speeds that the downwind turbines, as show in Fig. 1. The *velocity deficit*[2] measures this effect as shown in equation 1

$$dV = U_0 - U_x = U_0 \cdot \frac{1 - \sqrt{1 - C_t}}{\left(\frac{1+2kX}{D}\right)^2}, \quad (1)$$

where U_0 is the initial free stream velocity, U_x is the velocity in the wake at a distance X downstream of the upwind turbine (eq. 2), C_t is the thrust coefficient of the turbine, D is the diameter of the upwind turbine, and k is the wake decay constant. This model assumes that the kinetic energy deficit of interacting wakes is equal to the sum of the energy deficits of the individual wakes. Thus, the velocity deficit at the intersection of several wakes is:

$$U_x = U_0 \left[1 - \sqrt{\sum_{i=1}^N \left(1 - \frac{U_i}{U_0}\right)^2} \right]. \quad (2)$$

2.2 Power Model

The previous wake model directly defines the power model, that is to be maximized. The power curve for the wind turbine under consideration is a Gamesa G47 [7], whose power model (in KW) follows here:

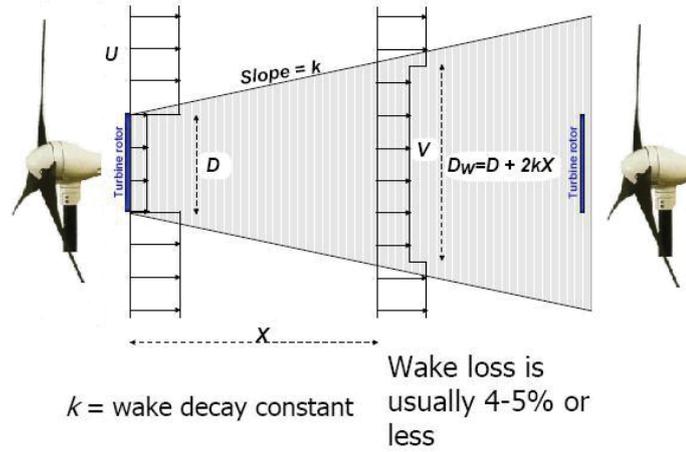


Fig. 1. Schematic of wake model

$$P_i = \begin{cases} 0 & \text{for } U_x < 4m/s, \\ \rho \times A \times U_x^3 \times C_p & \text{for } 4m/s \leq U_x < 12.5m/s, \\ 700 \times C_p & \text{for } 12.5m/s \leq U_x \leq 25m/s, \\ 0 & \text{for } 25m/s < U_x, \end{cases} \quad (3)$$

where U_x is defined in eq.2, ρ is the density of the environment ($1.23kg/m^3$), A is the swept rotor area and C_p is the power coefficient of the wind turbine (0.40 in this case).

Thus, the total power generation for all the turbines in the wind farm is:

$$P_{tot} = \sum_{i=1}^N P_i, \quad (4)$$

where N is the total number of turbines. For that, the power curve of this wind turbine is showed in Fig. 2

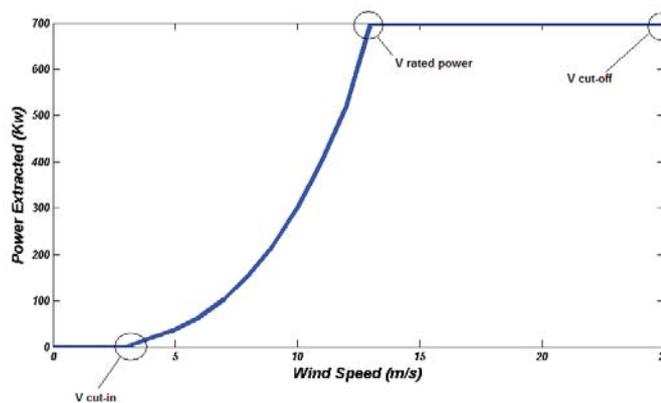


Fig. 2. Power curve of wind turbine

where velocity cut in is when the wind turbine starts to generate power, velocity rated power is when the wind turbine generate the maximum power, and velocity cut out is when the wind turbine stop to generate power.

2.3 Cost Model

In our case only the number of wind turbines is necessary for calculating and hopefully minimizing the total cost. The total cost per year for the entire wind farm assuming a predefined and constant number of wind turbines can be expressed as follows:

$$cost_{tot} = cost_{gy} \cdot N(2/3 + 1/3 \cdot e^{-0.00174N^2}), \quad (5)$$

where $cost_{gy}$ represents cost per wind turbine per year, in this case it value is € 400.000. We not consider the cost of installation, cost of foundation and cost of maintenance.

3 Metaheuristic Algorithms

Metaheuristics [14] are guided search techniques used in the optimization, where potential solutions are improved by incorporating an objective function. These techniques do not guarantee to find the optimal solution to the problem, but guarantee to find a good solution in reasonable computational times. For that in NP-hard problems the metaheuristics are a good option to find acceptable solutions very quickly.

3.1 CHC

In this section we describe the basic principles of the CHC [13] using in this work to resolve the problem. CHC is a sort of genetic algorithm and that have provided in the past good solutions in problems like RND (Radio Network Design) that share some points in common to our work[8]. The CHC algorithm was designed to work with populations coded as binary strings. CHC is a type of genetic algorithm that does not use mutation to produce new solutions; insteads it uses a mechanism called *HUX* crossover. The selection of individuals to complete the next generation is under only an elitist approach between parents and children. The R best solutions are retained and will be present in the next generation. When stagnation in the population is detected, a cataclysmic method of restarting is used. The population tends to be homogeneous due to the absence of mutation and the elitist approach because there is no diversity; in order to solve this problem CHC implements a mechanism called *incest prevention*. The parents are selected randomly, but crossover takes place only if the individuals are not too close between them (Hamming distance) exceeds a certain threshold called *the threshold of incest*. As the population evolves, fewer individuals have the condition of not incest; in this case it is necessary to reduce the threshold. Every time that no change appears in the population (after one iteration) the threshold reduces in one unit.

The mechanism of crossover HUX also preserves diversity. This crossover copies in the two offspring all bits matched in both parents, and then copies half bits different in each offspring, such the Hamming distance between children and between children and parents is high. Once that the threshold of incest is 0, if q iterations pass without any new solution has entered the population, it means that the population has converged and the algorithm has stagnated, thus requiring a restart. All individuals except the best are modified by a mutation by bit inversion with very high probability (in our case is 50%).

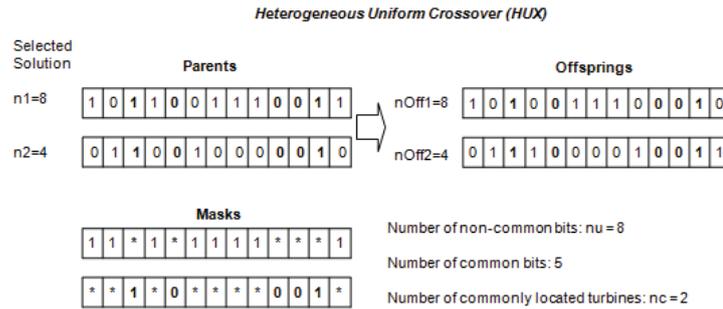


Fig. 3. Crossover HUX for CHC algorithm

Figure 3 shows an example of crossover HUX. It generates a mask with the common bits from the parents and non-common bits are assigned randomly to each child taking into account that each one must take half of the bits not common. The respective pseudocode of the CHC algorithm is shown in Algorithm 1.

4 Wind Farm Design

In this section we show one of the most important things in this kind of algorithms, the representation of the potential solutions, each problem has a representation that best fits, we consider the binary representation to represent into the wind farm each solution because is very simple to match the real problem with our problem. Also we define the fitness function to be maximized in this case.

4.1 Representations of Wind Turbine Locations

A wind farm is logically divided into many small square like cells. Each cell in the wind farm grid can have two possible states: it contains a wind turbine (represented by 1) or it does not contain a wind turbine (represented by 0). A 20×20 and 30×30 grid is used here as the ground platform to place the wind turbine. A binary string with 400 in the first case and 900

Algorithm 1 CHC

```

1:  $t \leftarrow 0$ ; /* evaluation */
2:  $initialize(Pa, Distance)$  /*Initialize the population and the distances */
3: while not stop criterion( $t, Pa$ ) do
4:    $Parents \leftarrow selected(Pa)$ ; /* Selected parents */
5:    $Offspring \leftarrow HUX(Parents)$  /* Crossover HUX */
6:    $evaluate(Pa, Offspring)$  /*evaluate Offspring*/
7:    $Pa \leftarrow elitism(Offspring, Pa)$ 
8:   if  $Pa$  no change then
9:      $distance \leftarrow distance - 1$ ;
10:    if  $distance == 0$  then
11:       $reset(Pa)$ 
12:       $initialize(distance)$ 
13:    end if
14:  end if
15:   $t \leftarrow t + 1$  /* One more generation */
16: end while
17: Return: best solution found.

```

bits in the second case represents the presence or absence of the wind turbine in the wind farm. There are 2^{400} and 2^{900} candidate solutions respectively. The width at each cell, in the center of which a turbine would be placed, is equal to five rotor diameters, $5D$ (or 235 m) giving thus, the resulting dimension is $100D \times 100D$ in the first case and $150D \times 150D$. The $5D$ square grid size also satisfies the rule of thumb spacing requirements in the vertical and horizontal directions. The wind turbine considered in this paper has the next properties: rotor diameter 47 m, trust coefficient 0.88 and wake decay constant 0.11.

4.2 Objective Function

The objective function that we are maximizing is the annual profit got from the wind farm, defined as follows[9]:

$$profit = \left[st - \left(\frac{cost_{tot}}{P_{tot}} \right) \right] P_{tot}, \quad (6)$$

where st represents the estimated selling price for a KWh of electrical energy on the market, in this case its value is 0.1, P_{tot} represents the total expected energy output (kWh) of the wind farm per year, and $cost_{tot}$ is given by [eq. 5].

5 Analysis Sequential Algorithms for Wind Farm Design

We study the scalability in two cases, dimension of problem 20×20 , and 30×30 , and we show graphics of evaluations to need for each size, evolution of fitness, different configurations of the wind farm, graphics of scalability and result tables with the average fitness, standard deviation, average power output, average efficiency, average time execution and number of wind turbines. The only case that we study is the case of uniform wind direction with a wind speed of 12 m/s; the wind direction comes from the north of the wind farm. The only change in wind speed for this case would occur in the wake of the wind turbines. The stop criteria for each algorithm is 5,000,000 evaluations. We used Mallba library[10] to design the algorithms and we have executed this problem in CPU Multi-Core 2xQuad-Core 2.00 GHZ. Number of individuals in population was 128, probability of Crossover was 0,8 and 0.5 the probability of reinitialize the population.

5.1 Case a: Sequential CHC applied for Wind Farm of dimension 20×20

For the first case with dimension 20×20 we obtained the next values, fitness: $e1.8466e + 7$, standard deviation: $1.13e + 3$, power output: 49,913.15 kw, efficiency: 0,80 and 80 numbers of turbines. For the performance of the algorithm we analyze the number of the evaluations needed, the time execution and memory used during the execution of sequential CHC, this values was 4,836,777 evaluations, 103.09 seconds of execution time and 416 MB of RAM memory used. The configuration of obtained layout for the wind farm is as follows:

5.2 Case b: Sequential CHC applied for Wind Farm of dimension 30×30

For the second case with CHC algorithm we obtained the next values fitness: $e3.4857e + 7$, standard deviation: $1.81e + 3$, power output: 94,596.12 kw, efficiency: 0,74 and 192 numbers of

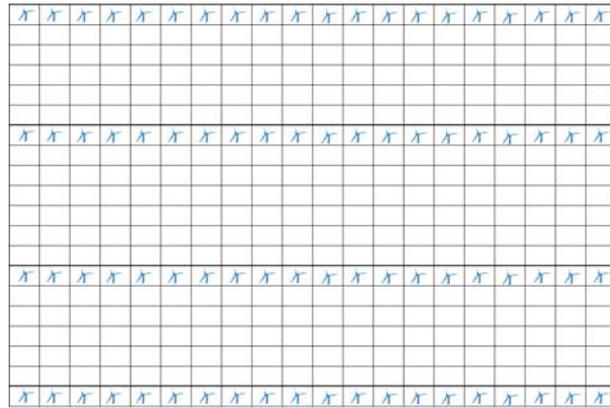


Fig. 4. Scenario 20 x 20

turbines. For the performance of the algorithm we analyze the number of the evaluations needed, the time execution and memory used during the execution of sequential CHC, this values was 4,394,530 evaluations, 256.28 seconds of execution time and 521 MB of RAM memory used. The configuration of wind farm obtained with CHC is as follows:

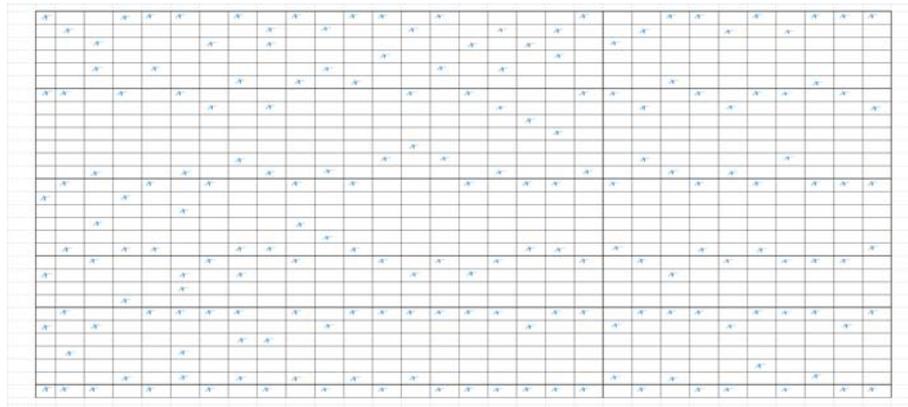


Fig. 5. Scenario 30 x 30 with Sequential CHC

6 Parallel CHC algorithm

In section 3 we explained the generic CHC, now we introduce a new version of CHC, in this case we will divide the population in several small populations called *islands* using a distributed island model[12]. In this model exists several islands that will be executed in different cores. Each population will communicate with a operator called migration. With this operator we send solutions to other population or island to introduce diversity into the population. Each island execute one CHC algorithm with a population size equivalent to p/n where p is the total population and n is the number of island in the model.

For this problem we use the distributed island model with ring connected and we need to define a new operator for work with different island, that operator is called *migration operator*

which is responsible for managing the exchange of solutions between islands, for that we define a new operator with this features.

6.1 Analysis of Parallel Algorithms for Wind Farm Design

Now we solve the problem of wind farm design with parallel CHC, in this case we use the model island introduced in the previous section and a multicore processor with eight cores (and eight islands). We will analyze four cases for each instance (20×20 and 30×30), the first case we use only one core with eight island, the second case we use two cores with eight island (four island for each core), the third case we use four cores (two island for each core), and the last case we use eight cores (each island in each core). We study the effort of the algorithm studying the evaluations needed to find a solution, the time execution, the average fitness, standard deviation, average power output, average efficiency, and number of wind turbines. The stop criteria for each algorithm is 5,000,000 evaluations. We used Mallba library[10] to design the algorithms and we have executed this problem in CPU Multi-Core $2 \times$ Quad-Core 2.00 GHZ. For migration operation we use the following values: a-synchronized operation mode, period: 10 evaluations and 5 individual of rate.

6.2 Case a: Dimension 20×20 Parallel CHC

For the first case with Parallel CHC algorithm we obtained the values in table 1.

Table 1. Dimension 20×20 for Parallel CHC (average values)

Cores	Fitness(€)	Standard Deviation	Power Output(kw)	Efficiency(%)	Number of Turbines
1	1.8804e+7	98,537.68	50,357.02	0.75	80
2	1.8808e+7	61,096.24	50,575.46	0.75	80
4	1.8813e+7	93,383.71	50,590.40	0.75	80
8	1.8819e+7	99,246.99	50,635.86	0.75	80

We analyze the number of the evaluation needed, the time execution and memory used during the execution of parallel CHC, this values are shown in the table 2.

Table 2. Performance of Parallel CHC (average values)

Cores	Eval. needed	Time Execution (Sec)	Memory used (MB)
1	766,459.97	95.47	339
2	501,392.23	44.65	301
4	438,985.16	23.78	272
8	219,577.60	11.73	208

Is important to note that the performance has improved in all parameters except in the number of turbines respect at the sequential version of CHC. The configuration of wind farm obtained with CHC is as follows:

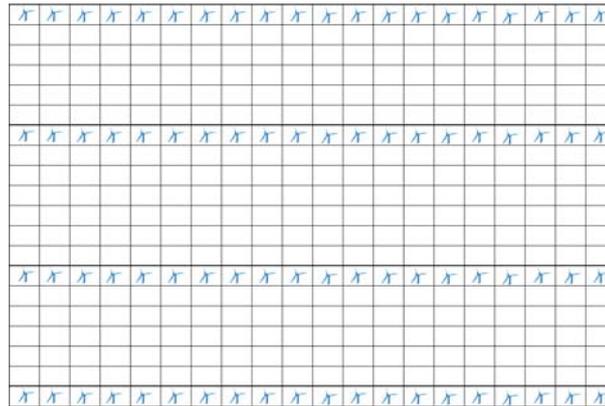


Fig. 6. Scenario 20 x 20 with Parallel CHC

6.3 Case b: Dimension 30 × 30 Parallel CHC

For the second case with Parallel CHC algorithm we obtained the next values in table 3. We

Table 3. Dimension 30 × 30 for Parallel CHC (average values)

Cores	Fitness(€)	Standart Deviation	Power Output(kw)	Efficiency(%)	Number of Turbines
1	3.4917e+7	153,384.50	94,930.20	0.75	180
2	3.4954e+7	160,900.72	94,933.22	0.75	180
4	3.4968e+7	133,226.84	94,938.51	0.75	180
8	3.5011e+7	144,202.54	94,948.51	0.76	180

analyze the number of the evaluation needed, the time execution and memory used during the execution of parallel CHC, this values are shown in the table 4.

Table 4. Performance of Parallel CHC (average values)

Cores	Evaluation needed	Time Execution (Sec)	Memory used (MB)
1	945,856.97	181.91	461
2	758,258.27	88.61	433
4	637,945.16	43.30	402
8	579,610.77	22.65	395

Is important to note that the performance has improved in all parameters except in the number of turbines respect at the sequential version of CHC. The configuration of wind farm obtained with CHC is shown in Fig 7:

7 Conclusions and Future Work

Test result of those cases study with constant wind speed and direction demonstrate that the performance of our Parallel CHC is optimal for designing wind farms. We have obtained very

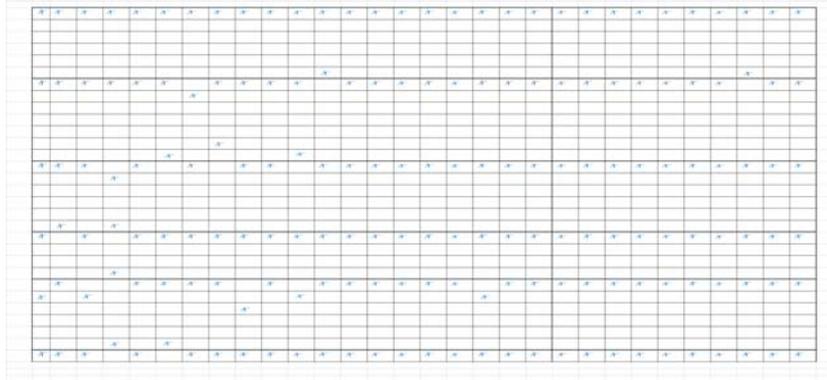


Fig. 7. Scenario 30 x 30 with Parallel CHC

good configurations for this problem with parallel CHC and improve our sequential version of CHC. The speed-up obtained is super linear in some cases and good speed-up in the other cases, so we conclude that this version of parallel CHC is a good alternative to design wind farm. As future work we can add new farm models, including actual factors, such as terrain effect and a aesthetic impact, more winds scenarios, real data etc. Also, we intend to use new parallel algorithms to still enhance the performance of the present study.

References

1. J.F Manwell, J.G. McGowan, and A.L.Rogers, *Wind Energy Explained-Theory, Design and Application*, 1st ed. , Reprint with correction, Jhon Wiley & Sons Ltd. ,2003, p.384 & p.44.
2. M. Bilbao, E. Alba, “Simulated Annealing for Optimization of Wind Farm Annual Profits” - *2nd International Symposium on Logistics and Industrial Informatics Austria*, 2009.
3. H.S.Huang, “Distributed Genetic Algorithm for Optimization of Wind Farm Annual Profits”. *Intelligent Systems Applications to Power Systems*, 2007. ISAP 2007. International Conference on Volume , Issue , 5-8 Nov. 2007 Page(s):1 - 6.
4. M.Bilbao, E. Alba, “GA and PSO Applied to Wind Energy Optimization” . *CACIC 09*, Jujuy, Argentina. 2009.
5. M. Bilbao, E.Alba, “CHC and SA Applied to Wind Energy Optimization Using Real Data”. *CEC 10*, Barcelona, 2010.
6. I. Katic, J. Hojstrup and N. O. Jensen, “A Simple Model for Cluster Efficiency”, *European Wind Energy Association Conference and Exhibition*, Rome-Italy, pp. 407-410, 7-9 October 1986.
7. “www.gamesa.es”
8. E. Alba, G. Molina, F. Chicano, “Optimal Placement of Antennae using Metaheuristics” . *Numerical Methods and Applications (NM&A-2006)*, Borovents, Bulgaria. 2006.
9. U. A. Ozturk and B. A. Norman, “Heuristic methods for wind energy conversion system positioning”, *Electric Power Systems Research*, vol.70, pp. 179-185, 2004.
10. Mallba Project: *Software Library in Metaheuristic* (1999 – 2002), University of Malaga, University of La Laguna, University Polytechnic of Catalonia. <http://neo.lcc.uma.es/mallba/easy-mallba/index.html>.
11. Multicore Processor Architecture: “www.intel.com”
12. E.Alba, “Parallel Metaheuristics, A New Class of Algorithms”, Wiley-Interscience (September 8, 2005).
13. Larry J.Eshelman, “The CHC Adaptive Search Algorithm: How to Have Safe Search When Engaging in Nontraditional Genetic Recombination”, *Foundations of Genetic Algorithms* pp. 265–283 (1991).
14. A Eiben and J Smith, “Introduction to Evolutionary Computing”, Springer, pp.XII–287 (2015).
15. J.Serrano González and M.Burgos Payán and J.M Riquelme Santos and F. González-Longatt, “A review and recent developments in the optimal wind-turbine micro-siting problem”, *Renewable and Sustainable Energy Reviews*, pp.133–144 (2014).
16. R.Barthelmie and L.Jensen, “Evaluation of wind farm efficiency and wind turbine wakes at the nysted offshore wind farm”,pp.240–243, *Wind Energy* (2010).

Optimal Instrumentation: Adjustment and Hybridization of a Simulated Annealing Based Technique

José Hernández¹, Carolina Salto^{2,3}, Gabriela Minetti², Mercedes Carnero¹,
Carlos Bermudez², Mabel Sánchez⁴

¹ Grupo de Optimización - Facultad de Ingeniería – Universidad Nacional de Río Cuarto,
Río Cuarto, Argentina
{jhl, mcarnero}@ing.unrc.edu.ar

² Laboratorio de Investigación en Sistemas Inteligentes (LISI) - Facultad de Ingeniería -
Universidad Nacional de La Pampa– General Pico, Argentina - ³ CONICET
{saltoc, minettig}@ing.unlpam.edu.ar

⁴ Departamento de Ingeniería Química, Universidad Nacional del Sur (UNS) and Planta
Piloto de Ingeniería Química - PLAPIQUI (UNS-CONICET), (8000) Bahía Blanca, Argentina
msanchez@plapiqui.uns.edu.ar

Abstract. This work presents an analysis on the behavior of a particular metaheuristic technique inspired by Simulated Annealing, SA, for the resolution of a problem of sensor network design in process plants, SNDP. This design problem is formulated as a combinatorial optimization problem and includes the selection and determination of the number of process variables that must be measured to achieve the specific state of knowledge about the plant. Different parametric configurations for the proposed heuristic are analyzed and discussed, as well as their performance in plants of increasing size and complexity.

Keywords: sensor network, combinatorial optimization, metaheuristics.

1 Introduction

In a chemical plant, having a knowledge of the process state at any time is absolutely crucial with impact on aspects such as economic, safety or control. In other words, the monitoring of the plant must be guaranteed. The information is collected by sensors distributed throughout the plant, responsible for measuring and transmitting the values of magnitudes such as temperature, humidity, pressure, etc. The set of devices used in the measurement is called the sensor network (SN, Sensor Network). The SN design is systematically made by formulating an optimization problem called Sensor Network Design Problem (SNDP), which is a discrete optimization problem.

In general, the number of variables that are involved in these problems for a real work scenario is quite large and the formulation can be more or less complex depending on the performance criteria and the restrictions set imposed on it.

Therefore, to use an efficient optimization tool that allows to solve the problem for different complexity and size formulations is an important topic.

There are exact methods that can find an exact solution to the problem by providing a global optimum. For real plant problems with a high number of variables, these strategies need huge amounts of resources and cannot offer a solution in acceptable times. Heuristics, on the other hand, can provide good solutions, from an industrial point of view, they take less time to compute and, for the last decades, they have been paid special attention to solve difficult engineering problems.

SNDP was formulated by Bagajewicz [1], and tackled by Nguyen and Bagajewicz [2] using a new tree search method that exploits certain cost properties of the different nodes in the tree to efficiently prune non optimal nodes using a breadth-first/level traversal tree search method to obtain the global optimum. Other approaches modeled the problem combining the integer and non-linear programming and solved it by means of depth-first or breadth-first tree searches. The main disadvantage of all these methods is that they are too time consuming. In order to mitigate this disadvantage, different metaheuristic methods were proposed to solve mono or multi-objective problem instances, such as genetic algorithms [3], swarm intelligence [4], among others population-based metaheuristics. Furthermore, hybridized metaheuristics have been reported to solve this problem, Carnero et. al. [5] proposed the PBIL_SOTS technique, which combines estimation of distribution algorithm with a tabu search improved by using an oscillation strategy.

The Simulated Annealing (SA), is a simple, general purpose Monte-Carlo method, which was developed for combinatorial optimization [6]. In contrast with the population-based metaheuristics aforementioned, it may be classified into the trajectory-based group and it has proved to be an efficient method to solve many hard combinatorial optimization problems [7]. However, heuristics are, in most cases of general purpose. Its adaptation for to the resolution of a particular problem implies, among other aspects, to make an adjustment of its parameters that can obtain the best performance of the proposed technique.

The goal of this work is to solve the Optimal SNDP applying a SA hybridized with heuristic methods, focusing in the adjustment (or tuning) of the algorithmic control parameters to reach an equilibrium between the solution quality and time consuming. A comparison with other methods and numerical results of tests on several instances are given, and the effectiveness of the proposed method is analyzed.

The rest of this article is organized as follows. In Section 2 the SNDP is described. Section 3 introduces and explains the approach proposed in this work. Section 4 refers to the experimental analysis and the methodology used. Finally, the main conclusions and future lines of research are drawn in Section 5.

2 Sensor Network Design Problem

The SNDP is summarized as a problem of finding the minimum cost network that satisfies precision and estimability constraints. Formally, a SNDP solution has to satisfy these constraints for a set of key variable estimates, as stated by Eq.(1), where \mathbf{q} is an n -dimensional vector of binary variables such that $q_i = 1$ if variable i is

measured, and $q_i = 0$ otherwise, \mathbf{c}^T is the cost vector; $\hat{\sigma}_k$ is the estimate standard deviation of the k -th variable contained in S_σ after a data reconciliation procedure is applied, and E_l stands for the degree of estimability of the l -th variable included in S_E . Furthermore, S_σ and S_E are the set of key process variables with requirements in precision and ability to be estimated, respectively.

$$\begin{aligned}
 & \min && \mathbf{c}^T \mathbf{q} \\
 & \text{s.t.} && \\
 & && \hat{\sigma}_k(\mathbf{q}) \leq \sigma_k^*(\mathbf{q}) \quad \forall k \in S_\sigma \\
 & && E_l(\mathbf{q}) \geq 1 \quad \forall l \in S_E \\
 & && \mathbf{q} \in \{0,1\}^n
 \end{aligned} \tag{1}$$

In this formulation, measurements are subject to non-correlated random errors; there is only one potential measuring device for each variable, and there are no restrictions for the localization of instruments. The feasibility of the constraints can be checked by executing a variable classification and data reconciliation procedure.

3 Hybrid Simulated Annealing Algorithm

In this section, a description of SA and its variants is introduced. After that, the Hybrid Simulated Annealing, HSA, to optimize the cost instrumentation in chemical plants is explained and how the SA variants are adapted the HSA.

3.1 Simulated Annealing Algorithm

Simulated annealing is a well-studied trajectory-based metaheuristic used to address discrete and, to a lesser extent, continuous optimization problems. The SA algorithm simulates the energy changes in a system subjected to a cooling process until it converges to an equilibrium state (steady frozen state), where the physical material states correspond to problem solutions, the energy of a state to cost of a solution, and the temperature to a control parameter.

At the beginning (with a high temperature), SA accepts solutions with high cost values under a certain probability in order to explore the search space and to escape from local optima. During the annealing process this probability decreases according to temperature cooling; intensifying the search and reducing the exploration in order to exploit a restricted area of a search space.

Simulated annealing evolves by a sequence of transitions between states and these transitions are generated by transition probabilities. Consequently, SA can be mathematically modeled by Markov chains, where a sequence of chains is generated by a transition probability, which is calculated involving the current temperature.

Most of the search components of SA are fixed in function of the problem to be solved. Consequently, the search space, cost (evaluation) function, perturbation operator, and local search are directly related to the problem. The main search

components, which are variable during the process, are the initial temperature, the temperature through their annealing schedules, and the Markov chain length [7].

Therefore, one of the most important issues in SA is the choice of the right initial temperature, which must not be too high to conduct a random search for a period of time but high enough to allow moves to almost all neighborhood states. The classical and intuitive method consists in computing a temperature such that the acceptance ratio is approximately equal to a given value [6]. Given a T_s seed temperature, the initial temperature is computed by the procedure shown in Algorithm 1. The output, T_0 , is determined such that, when applying the Boltzmann criterion, worse solutions are accepted with a high probability value. T_0 achieve this, the algorithm starts from a T_s that is increased until the aforementioned acceptance is reached.

The scheme to control the annealing or cooling process is also crucial, so that the system cools gradually from a higher temperature, ultimately freezing to a global minimum state. Many attempts have been made to derive or suggest good schedules [7]. The most known cooling process in the literature are proportional, exponential [6], and logarithmic schemes [8]. Furthermore, a random schedule is considered [9].

In the proportional one, the temperature is updated using Eq. (2), where α is a constant close to, but smaller than, 1 and calculated as Eq. (3) shows. This scheme is the most popular cooling function, since the temperature decay is not too slow neither too fast allowing to achieve an equilibrium between exploitation and exploration.

$$T_{k+1} = \alpha T_k \quad (2) \quad \alpha = k/(k+1) \quad (3)$$

The exponential cooling scheme produces the temperature decay by applying Eq.(4), where the constant $\alpha^k < 1$ is calculated as presented in Eq. (5). This schedule quickly cools the temperature reducing the required time and iterations to converge to a good solution. In big and complex problems, this becomes in a disadvantage, given that the equilibrium between the exploitation and exploration is broken.

$$T_{k+1} = \alpha^k T_k \quad (4) \quad \alpha = e^k/e^{k+1} \quad (5)$$

Algorithm 1: Pseudocode of algorithm for setting initial temperature T_0

Function *init_temp*(T_s)

initialize $T_0 = T_s$;

while (acceptability rate is not reached)

 Increment T_0 ;

 generate a solution \mathbf{q}^1 ;

 evaluate \mathbf{q}^1 in H^1

for $i=0$ to test

 generate a new solution \mathbf{q}^2 from \mathbf{q}^1 applying Swap mutation under P_{swap} ;

 evaluate \mathbf{q}^2 in H^2 ;

 Apply Boltzmann criterion and count solutions that was

 accepted;

end

end

return T^0 ;

The logarithmic schedule modifies the temperature, as shown in Eq. (6), the chain converges to a global and minimal energy value, where the constant C is computed as Eq. (7) indicates. This schedule is too slow to be applied in practice but has the property of the convergence proof to a global optimum [10].

$$T_{k+1} = CT_k \quad (6) \qquad C = \log(k)/\log(k+1) \quad (7)$$

Finally, the random schedule combines the three previous cooling schemes in only one schedule process. At each iteration, this scheme randomly selects one of these schemes in order to reduce the temperature. In this way, the advantages of these three schemes are aggregated and their disadvantages mitigated.

The Markov chain length (MCL) is the number of required transitions (moves) to reach the equilibrium state at each temperature. This number can be either static or adaptive. At the first case, it is calculated before the search starts. For instance, a given proportion of the neighborhood of the current solution is explored. Another static approach assumes that each temperature T_k is held constant for a sufficient and fixed number of iterations.

For the adaptive way, the Markov chain length depends on the characteristics of the search. For instance, [11] consider that to reach the equilibrium state at each temperature is not necessary, i.e., the cooling schedule is applied as soon as an improving candidate (neighbor) solution is generated. In this way, the computational effort can be drastically reduced without compromising the solution quality. Ali et al. [12] propose another adaptive approach, which uses both the worst and the best solutions found in the Markov chain (inner loop) to compute the next MCL.

3.2 Hybrid Simulated Annealing Algorithm for Optimal Cost Instrumentation in Chemical Plants

In Hernández [13], an adapted and hybridized SA algorithm to solve the SNDP in chemical plants was proposed. SA works as main heuristic with a subordinated ad-hoc local search, inspired in tabu search with strategic oscillation technique, SOTS, giving rise to the Hybrid Simulated Annealing (HSA_SOTS) algorithm. The hybridization in HSA_SOTS is applied in two levels: in the first one to generate an initial solution, and in the second level to improve the solution during the annealing process.

The perturbation scheme of the current solution is carried out through a certain swapping number of measured variables to unmeasured ones and vice versa in order to generate a candidate solution \mathbf{q}_2 from \mathbf{q}_0 . This swap mutation is applied over each variable with a certain probability (called P_{swap}). Furthermore, the temperature is updated using the geometric criterion [14].

In this work, we study an important algorithm design issue that involves the main search components and are variable during the process i.e., the temperature and the MCL. In order to study the impact of different initial temperatures in the performance of the HSA_SOTS, we use T_s values belonging to $\{1,900\}$ from small to large seeds. In this way, we test very dissimilar seeds allowing a different number of HSA_SOTS's main loop iterations. When the cooling scheme is studied, we propose four different HSA approaches: HSAProp that adopts the proportional annealing schedule, HSAExp uses the exponential cooling scheme, HSALog employs the

logarithmic schedule, and HSARand that applies a random cooling scheme. Finally, we consider three different ways to compute the MCL. At the first, MCL is statically calculated and each t is held constant for 30 iterations (number commonly used in the literature), named MCLs. The other two ways to implement the adaptive techniques proposed by Cardoso et al. [11] and Ali et al. [12] identified as MCLa1 and MCLa2, respectively.

4 Experimental Design

In order to evaluate HSA_SOTS performance, a test set of 5 design problems were considered which comprise processes of different complexity and size and whose operation can be represented by both linear and non-linear models. The sizes of the considered instances range from 28 to 82 decision variables. The interested readers can gain access to the file containing information about the case studies from https://www.ing.unrc.edu.ar/archivos/sndp_cases.doc. The complexity of the set of constraints imposed on all case studies can be found in [5]. The stop condition of the HSA_SOTS's variants is to reach 1250 iterations of the mean loop.

The computational environment used in this work to carry out the experimentation consists of computers with Processor Intel Core i5 CPU 4440 @ 3.10 GHz, 4GB RAM, using MatLab R2011b. Because of the stochastic nature of the algorithms, 30 independent runs of each instance were performed to gather meaningful experimental data and statistical confidence metrics were applied to validate the results and conclusions. As a result, a total of 7200 executions were carried out.

4.1 Analysis of Results

In this section, we summarize and analyze the results of using the HSA_SOTS's variants proposed in this work on the all problem instances. First, we analyze the behavior of the variants HSAProp, HSAExp, HSALog, and HSARand with T_s values belonging to $\{1,900\}$ and the three ways to compute the Markov chain length. Table 1 presents the best and median cost values found by these 24 variants for all instances, besides the percentage of hits (%hits). In this table the best results are boldfacing. Figure 1 shows the average execution total time for each algorithmic variant and case study.

From the analysis of the result quality, an important separation of the case studies is observed (see Table 1). For the first four cases, the all algorithmic variants find the best known solution in each execution. However when the case study 5 is solved, different behaviors between the proposed algorithms is detected. These differences are statistically corroborated using the Kruskal-Wallis (KW) test with a confidence level, $\alpha=0.01$.

Consequently, the results for the fifth case study deserves a detailed analysis. If the T_s parameter is considered, the highest percentages of hits are reached for $T_s=900$, and the optimal solution is found by 11 of 12 proposed approaches. This selection for T_s is statistically supported by the median values because they are equal to the optimal in 8 of 12 opportunities against 4 for $T_s =1$. Analyzing the three MCL options, a

significant improvement in the solution quality is observed if the adaptive variants are applied. In other words, only the application of MCLa1 or MCLa2 in HSA_SOTS warrants to find the optimum. Finally, the results are assessed considering the cooling scheme. None direct relation between the kind of cooling process and the result quality is observed, but the behavior of these schedules is highly dependent of the combination of the T_s value and MCL option. For example, when the parametric configuration $T_s=1$ and MCLs is applied the best results are found by HSAExp, but if $T_s=900$ the best option is HSAProp.

From the computational effort point of view, the HSA_SOTS's variants that implement MCLs minimize this effort, obtaining the best known solution for the first four case studies in every run. For the case study 5, the MCLs application allows to reach a relatively high percentage of hits with $T_s=900$ and the proportional cooling scheme (HSAProp) is used. Furthermore, a 100% of hits is achieved by the most expensive HSA_SOTS variants (MCLa1 and MCLa2), computationally speaking. However, the MCLa1 application is significantly less expensive than the MCLa2 one.

Table 1. Best and median cost values found by these 24 variants for the all cases and its percentage of hits.

Case Study	Algorithms		$T_s=1$			$T_s=900$		
	MCL	Cooling Scheme	Min.	Median	% hits	Min.	Median	% hits
1	All	All	1106.46	1106.46	100.00	1106.46	1106.46	100.00
2	All	All	735.00	735.00	100.00	735.00	735.00	100.00
3	All	All	2928.00	2928.00	100.00	2928.00	2928.00	100.00
4	All	All	1154.34	1154.34	100.00	1154.34	1154.34	100.00
5	MCLs	HSAProp	50845.37	54974.18	0,00	50845.16	50845.16	53.84
		HSAExp	50845.16	54974.18	7.6	50845.16	54974.18	9.09
		HSALog	50846.39	54974.18	0,00	50845.16	50846.18	36.36
		HSARand	50845.37	52909.78	0,00	50845.37	54974.18	0,00
	MCLa1	HSAProp	50845.16	54973.16	40,00	50845.16	50845.16	100,00
		HSAExp	50845.16	52909.67	40,00	50845.16	52909.16	50,00
		HSALog	50845.16	50845.67	50,00	50845.16	50845.16	100,00
		HSARand	50845.16	50845.16	62.5	50845.16	50845.16	62.5
	MCLa2	HSAProp	50845.16	52909.16	50,00	50845.16	50845.16	100,00
		HSAExp	50845.16	50845.16	62.5	50845.16	50845.16	62.5
		HSALog	50845.16	50845.16	62.5	50845.16	50845.16	100,00
		HSARand	50845.16	50845.16	71.4	50845.16	50845.16	100,00

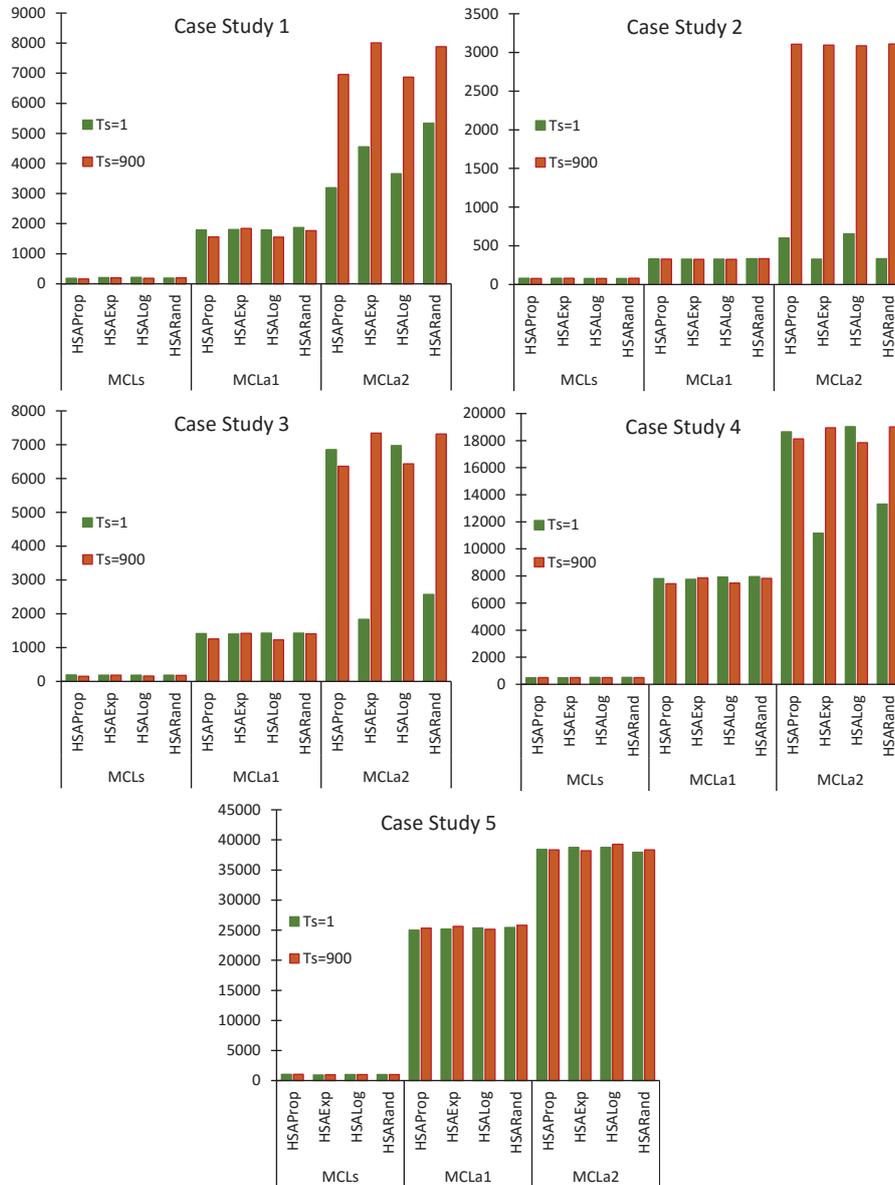


Fig. 1. Total time (in seconds) spent by each algorithmic variant for all case studies.

Summarizing, for the case studies 1, 2, 3, and 4, the HSA_SOTS's variants that implement MCLs always obtain the best solution with the minimum effort, showing statistically similar behaviors (KW test with $\alpha=0.01$). But if the complexity of the case to solve grows (case study 5), a trade-off between quality and time must be

Table 2. Comparison of the best HSA_SOTS' variant and PBIL_SOTS, considering best solution statistics for each case study.

Case study	Best solution		Mean best solution \pm SE		KW
	PBIL-SOTS	Best HSA_SOTS' variant	PBIL-SOTS	Best HSA_SOTS' variant	
1	1106.50	1106.50*	1106.50 \pm 0.00	1106.46 \pm 0.00	=
2	735.00	735.00	735.00 \pm 0.00	735.00 \pm 0.00	=
3	2928.00	2928.00	2929.20 \pm 0.69	2928.00 \pm 0.00	=
4	1154.34	1154.34	1154.34 \pm 0.00	1154.34 \pm 0.00	=
5	50845.16	50845.16	50886.63 \pm 41.29	50845.16 \pm 0.00	\neq

achieved. In this sense, the best algorithmic approaches are HSAProp and HSALog with $T_s=900$ and the application of MCLa1 to calculate the Markov chain length.

4.2 Comparison of HSA_SOTS Variants and the Literature Approaches

In this section, we compare the behavior of the best algorithmic variants of HSA_SOTS versus other well-known algorithm found in the literature for solving these SNDP case studies. In this sense PBIL_SOTS, introduced by Carnero et al. [5] has recently reported results for these cases.

To compare this state-of-the-art algorithm versus the best HSA_SOTS's variant, in Table 2 the best solution achieved by each of them for the 5 case studies, and their respective mean best solution and standard deviation are shown. Furthermore, a KW test is carried out to corroborate the similarities or differences between them. In general, we can observe that the algorithm from literature behaves similarly to HSA_SOTS in the least complex case studies, while statistically different behaviors are observed for the fifth case study. In this sense, two advantages in favor of HSA_SOTS are observed due to the average best solution is equal to the best known one, and the standard deviation indicates that it is found in every execution.

5 Conclusions

This work presents a SA hybridized with heuristic methods (HSA_SOTS), which is focused on the algorithmic control parameter tuning to solve the Optimal SNDP. As a consequence, 12 algorithmic variants arisen for each initial temperature considered ($T_s \in \{1,900\}$). These variants apply four different cooling schemes and tree ways to calculate the Markov Chain Length. Five SNDP case studies of a growing complexity were used to test and analyze our proposals.

From this analysis arises that the HSA_SOTS's variants, which implement MCLs, always obtain the best solution with the minimum effort for the case studies 1, 2, 3, and 4. Instead, for the most complex case (5), a trade-off between quality and time is achieved when HSA_SOTS uses a $T_s=900$, MCLa1, and applies the proportional

cooling scheme. Furthermore, these proposals are competitive with the literature approaches.

In future works, other SNDP formulations will be tackled improving the SA main heuristic by introducing different specific local search mechanisms.

6 References

1. Bagajewicz M. Design and Retrofit of Sensors Networks in Process Plants. *AIChE Journal*, 3,2300–2306 (1997).
2. Nguyen D. Q., Bagajewicz M.; New Efficient Breath-first/level Traversal Methodology for the Design and Upgrade of Sensor Networks, *AIChE Journal*, 57(5), 1302-1309 (2011)
3. Gerkens C., Heyen G., Use of Parallel Computers in Rational Design of Redundant Sensor Networks, *Comput. Chem. Eng.*, 29, 1379-1387 (2005).
4. He Y.J., Ma Z.F, Optimal Design of Linear Sensor Networks for Process Plants: A Multi-Objective Ant Colony Optimization Approach, *Chemometrics and Intelligent Laboratory Systems*, 135, 37-47 (2014).
5. Carnero M., Hernández J.L., Sánchez M., Optimal Sensor Location in Chemical Plants using the Estimation of Distribution Algorithms, *Industrial & Engineering Chemistry Research*, vol. 57(36), 12149–12164 (2018).
6. Kirkpatrick S., Gelatt C., Vecchi M.P., Optimization by Simulated Annealing, *Science*,220(4598),671-680 (1983).
7. Talbi E. G., 2009, *Metaheuristics: From Design to Implementation*, Wiley.
8. Hajek B., "Cooling Schedules for Optimal Annealing," *Mathematics of Operations Research*, vol. 13, no. 2, pp. 311-329 (1988).
9. Bermudez C., Salto C., Minetti G. Solving the Multi-Period Water Distribution Network Design Problem with a Hybrid Simulated Annealing. In: Pesado P., Aciti C. (eds) *Computer Science – CACIC 2018*. CACIC 2018. Communications in Computer and Information Science, vol 995. Springer, Cham (2019).
10. Geman S., D. Geman, "Stochastic Relaxation, Gibbs Distributions, and the Bayesian Restoration of Images," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 6, no. 6, pp. 721-741, (1984).
11. Cardoso M. F., Salcedo R. L., de Azevedo S. F., Nonequilibrium Simulated Annealing: A Faster Approach to Combinatorial Minimization, *Ind. Eng. Chem. Res.*, vol. 33, pp. 1908-1918, (1994).
12. Ali M., Trn A., Viitanen S., "A Direct Search Variant of the Simulated Annealing Algorithm for Optimization Involving Continuous Variables," *Computers Operations Research*, vol. 29, no. 1, pp. 87 -102, (2002).
13. Hernandez J., Salto C., Minetti G., Carnero M., Sanchez M.C., Hybrid Simulated Annealing for Optimal Cost Instrumentation in Chemical Plants, *Chemical Engineering Transactions*, 74, 709-714 (2019).
14. Du Ke-Lin, Swamy M. N. S. (Ed), *Search and Optimization by Metaheuristics, Techniques and Algorithms Inspired by Nature*, Springer International Publishing, Switzerland (2016).

Recognizing Handshapes using Small Datasets

Ulises Jeremias Cornejo Fandos^{1*}, Gaston Gustavo Rios^{1,3*},
Franco Ronchetti¹, Facundo Quiroga¹, Waldo Hasperué^{1,2}, and Laura
Lanzarini¹

¹ Instituto de Investigación en Informática III-LIDI, Facultad de Informática,
Universidad Nacional de La Plata

² Investigador Asociado - Comisión de Investigaciones Científicas (CIC)

³ Becario de entrenamiento - Comisión de Investigaciones Científicas (CIC)
{ucornejo,grios}@lidi.info.unlp.edu.ar

Abstract. Advances in convolutional neural networks have made possible significant improvements in the state-of-the-art in image classification. However, their success on a particular field rests on the possibility of obtaining labeled data to train networks. Handshape recognition from images, an important subtask of both gesture and sign language recognition, suffers from such a lack of data. Furthermore, hands are highly deformable objects and therefore handshape classification models require larger datasets.

We analyze both state of the art models for image classification, as well as data augmentation schemes and specific models to tackle problems with small datasets. In particular, we perform experiments with Wide-DenseNet, a state of the art convolutional architecture and Prototypical Networks, a state of the art few-shot learning meta model. In both cases, we also quantify the impact of data augmentation on accuracy.

Our results show that on small and simple data sets such as CIARP, all models and variations of achieve perfect accuracy, and therefore the utility of the data is highly doubtful, despite its having 6000 samples. On the other hand, in small but complex datasets such as LSA16 (800 samples), specialized methods such as Prototypical Networks do have an advantage over other methods. On RWTH, another complex and small dataset with close to 4000 samples, a traditional and state-of-the-art method such as Wide-DenseNet surpasses all other models. Also, data augmentation consistently increases accuracy for Wide-DenseNet, but not for Prototypical Networks.

Keywords: sign language, hand shape recognition, convolutional neural networks, densenet, prototypical networks, small datasets

1 Introduction

Sign Language Recognition is a field in the intersection of computer vision and language translation that seeks to create systems capable of translating videos of people speaking in sign language into text.

* equal contribution

In recent years, new advances in machine learning using models such as convolutional and recurrent neural networks have improved our ability to tackle complex recognition problems such as speech recognition, image classification or object detection[5]. These advances are fueled by a combination of improvements in three areas; better datasets, better models, and more compute power. While the last two are mostly independent of a particular field, the availability of quality datasets for a given field limits the application of these new advances. For example, common image classification datasets such as MNIST, CIFAR10, CIFAR100 and ImageNet contain thousands of examples per class [4].

The process of recognizing a sign language consists of several steps, ranging from image preprocessing, body part detection, facial expression recognition, handshape recognition, language modeling and language translation. Of these steps, handshape recognition plays the most crucial role in the interpretation of signs[10,15]. However, sign language recognition cannot currently take full advantage of state-of-the-art models, since the availability of labeled, quality data for training models is very limited [10]. Lack of data also impairs the development of accurate handshape recognition models [10].

In particular, Convolutional Neural Networks (CNN), a type of neural network that takes advantage of convolutional layers to learn arbitrary convolutional filters, have proven very effective at image classification [5], including the classification of handshapes in images [14]. However, in most applications, convolutional neural networks are trained using thousands of images per class. In handshape recognition tasks, the datasets are considerably smaller and of lower quality, and therefore the performance of the models suffers accordingly [10,18,14].

In this work we propose to evaluate and compare new methods devoted to deal with small datasets in order to improve the current state-of-the-art in hand shape recognition for sign language.

Our approach consists of comparing different techniques for improving model performance in these conditions: data augmentation and prototypical networks for few shot learning and semi supervised learning. Our data augmentation scheme consists of basic augmentation operators such as rotations, translations and crops. We compare this with new technique approach.

In the following subsection we summarize previous efforts on training CNN on handshape datasets. Section 2 describes the datasets and models we employed in our experiments, which are detailed along with results in Section 3, and Section 4 contains the conclusion of our work.

1.1 Related Work

Recent years have seen the rise in the use of deep learning models for sign language recognition, specifically the use of convolutional neural networks to extract image features or directly classified hand images. [10] trained a CNN to recognize handshapes from the RWTH handshape dataset, which contains 3200 labeled samples and 50 different classes. The model was based on a pre-trained network with a VGG architecture, and employed a semi-supervised scheme to take advantage of approximately one million weakly labeled images, achieving

an accuracy of 85.50%. This constitutes the first attempt at adapting a model to overcome the low availability of labeled images for training. [16] employed a radon transform as a feature for an ad hoc classifier that employed clustering as a quantization step and K nearest neighbors for the final classification. They tested the model on the LSA16 dataset, which contains only 800 examples, obtaining an accuracy of 92.3%. [14] evaluated several CNNs on the LSA16 and RWTH datasets, including both vanilla and pre-trained models. The use of pre-trained models helps to alleviate the lack of labeled data, since pretraining the convolutional filters establishes a prior that a further classifier can exploit for handshape recognition. This work is the second and last instance we found where a specific strategy was employed to alleviate the lack of data. Their best models of an accuracy of 95.92% for LSA16 and 82.88% for RWTH. [12] trained a simple neural network to classify a new dataset they created, which contains 6000 examples and 10 classes, reaching an accuracy of 99.20%. [18] train a CNN on a custom dataset with 36 classes, 8 subjects and 57000 sample images. However, the samples correspond to video sequences and therefore are highly correlated; while there are approximately 2000 images per class, there are only eight image sequences, one for each subject. Each of this image sequences contains approximately 250 images which are highly correlated, and therefore it is best to consider the dataset as having eight image sequences per class. They obtained an accuracy of 94.17%, [2] trained a simple CNN with only 6 layers using the ASL Finger spelling dataset, obtaining an accuracy of 80.34%. The dataset consists of 60000 images of 25 different classes, but they were captured as videos so they are also highly correlated as in the previous case. [3] employed the Jochen Triesch Database (JTD), which contains only 10 classes and 72 samples per class, as well as the NAO Camera Hand Posture Database, which contains 4 classes and 400 examples per class. They trained a simple CNN with a multichannel image containing the results of the Sobel operator as input, obtaining an F-score of 94% and 98% in each dataset perspective. [1] trained a deep CNN on the Hand Gesture Dataset LPD, which contains 3250 images of only 6 classes, obtaining an accuracy of 99.73%.

This brief review confirms our previous statement that while CNN are being consistently applied to handshape recognition tasks, most of these datasets are small and ad hoc, that is, recorded specifically for the purpose of testing a single model and not developed with the intent of providing a benchmark and complete training set for handshape recognition models. It is also worth noticing that some datasets are so small that it is very easy to obtain near-perfect performance with simple models. Also, many datasets are not readily available, given that the authors have not publish the data and do not provide any means of obtaining it. We note that the RWTH and LSA16 are both publicly available and current models have been shown to achieve less than perfect accuracy for them. While the dataset in [12] has been easily solved, it is interesting because it targets general handshapes instead of those specific to sign language. We will call this dataset CIARP.

2 Datasets and Models

We selected three datasets, LSA16 [16], RWTH-PHOENIX-Weather (RWTH) [11] and CIARP [12], because they contain images whose setting varies greatly, have been evaluated already, and possess different quantities of examples or distributions of samples per class.

We employed two different classification models to analyze their ability to learn from these small handshake datasets; Prototypical Networks [17] and DenseNet [8]. Prototypical Networks is a model that was designed explicitly to deal with small sample sizes. On the other hand, DenseNet is currently the state of the art in image classification with convolutional neural networks, and while it has not been explicitly designed for small datasets, it has shown exceptional performance in many different tasks.

We also experimented with data augmentation to analyze its capacity to compensate for the lack of data.

In the following subsections we describe in more detail the selected datasets and models.

2.1 Datasets

LSA16 [16] contains images of 16 handshapes of the Argentinian Sign Language (LSA), each performed 5 times by 10 different subjects, for a total of 800 images of size 32x32. The subjects wore colored hand gloves and dark clothes on a white background. The dataset is balanced, with 50 images per class. There is only one hand in each image which are centered and isolated from the background.

RWTH [11] is composed of a selection of hand images of size 132x92 cropped from videos of the sign language interpreters at the German public tv-station PHOENIX. There are a total of 45 different hand signs. The interpreters wore dark clothes in front of an artificial grey background. Many images possess significant movement blur, others contain both hands of the interpreter and hands are not always perfectly centered.

The dataset is highly imbalanced with some classes having just 1 sample while others have as many as 529 samples. We removed classes that had less than 20 samples following [14], to guarantee a minimum amount of images per class for the networks to learn.

CIARP [12] contains 6000 images of size 38x38 acquired by a single color camera. The images were manually labeled and correspond to 10 classes of hand gestures. The hands are centered and were segmented from the background, which was replaced by black pixels. The small size of the images and low amount of classes give this dataset lower complexity compared to LSA16 and RWTH. The classes in the data set correspond to handshapes not based on sign language, but are similar enough that the comparison remains valid.

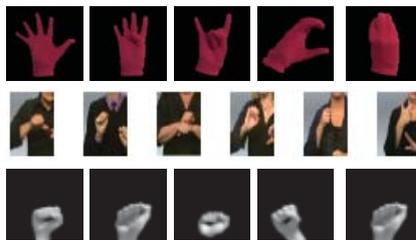


Fig. 1: Sample images from the LSA16 (first row), RWTH-PHOENIX-Weather (second row) and CIARP (third row) datasets.

2.2 Models

Prototypical Networks for Small Datasets Prototypical Networks [17] is a meta-learning model for the problem of few-shot classification, where a classifier must generalize to new classes not seen in the training set, given only a small number of examples of each new class. The ability of an algorithm to perform few-shot learning is typically measured by its performance on n -shot, k -way classification tasks. First a model is given a query sample belonging to a new, previously unseen class. Then, it's also given a support set, S , consisting of n examples, each from k different unseen classes. Finally, the algorithm then has to determine which of the support set classes the query samples belong to. Schemes for few shot classification tasks like Prototypical Networks can also be of use for training small datasets where all classes are known.

Prototypical Networks applies a compelling inductive bias in the form of class prototypes to achieve impressive few-shot performance. The key assumption is made is that there exists an embedding in which samples from each class cluster around a single prototypical representation which is simply the mean of the individual samples. This idea streamlines n -shot classification in the case of $n > 1$ as classification is simply performed by taking the label of the closest class prototype.

DenseNet We selected DenseNet as it is the current state of the art model in many domains and can handle small datasets with low error rate[13].

DenseNet [8] works by concatenating the feature-maps of a convolutional block to the feature-maps of all the previous convolutional blocks and using this value as input for the next convolutional block. This way each convolutional block receives all the collective knowledge of the previous layers maintaining the global state of the network which can be accessed.

Convolutional networks construct informative features by fusion both spatial and channel-wise information within local receptive fields at each layer. Squeeze and excitation blocks (SE block) [7] focus on the channel-wise information used in the convolutional layers. SE blocks improve the quality of representations produced by the network by modeling the interdependency between channels to

perform feature recalibration. SE blocks can be included in any model that uses convolutional layers to improve its performance at low computational cost. We added SE blocks to our DenseNet model to improve its performance.

Data Augmentation Image data augmentation is a set of techniques that aim at artificially augmenting the amount of data that can be obtained from the images in the dataset. These techniques modify the images in the dataset with a set of predefined operations to create new images that can be used to train a model. In this manner, we can compensate for the lack of variability in a small dataset[4].

3 Experiments

We performed classification experiments on LSA16, RWTH and CIARP hand-shape datasets. For each experiment, we split the dataset in training and test sets, with the latter taking 25% of the samples. The split was stratified, maintaining the proportion of samples of each class in both sets.

We applied normalization feature-wise subtracting the mean and dividing by the standard deviation of each feature. For data augmentation we used horizontal flipping, a 10 and 30 degree rotation and a resampling of the images creating new versions of them with a different size reducing each image by 10% and 20% in width and height. We found that a 10 degree rotation gave better results because a rotation of 30 degrees showed to be too high for the nature of the datasets.

We made multiple experiments with Prototypical Networks and DenseNet to find out which hyperparameter configuration was the best for each dataset: with and without data augmentation. We describe the hyper parameters for each model/dataset combination.

3.1 Prototypical Network

As mentioned in section 2.2, we can use Prototypical Networks' ability to work with small datasets even if all samples are labeled.

Therefor we experimented with Prototypical Networks using an embedding architecture composed of four convolutional blocks. Each block comprises a {64, 128}-filter 3×3 convolution, batch normalization layer, a ReLU nonlinearity and a 2×2 max-pooling layer.

We used the same encoder for embedding both support and query points. All of our models were trained with the ADAM[9] optimizer. We used an initial learning rate of 10^{-3} and cut the learning rate in half every 2000 episodes.

We trained prototypical networks using Euclidean distance in the 1-shot and 5-shot scenarios with training episodes containing 16, 20 and 10 classes (for LSA16, RWTH and CIARP respectively) and 5 query points per class. We found it advantageous to match the same value of n for train and test scenarios, and to use a higher value of k (more classes) per training episode. We computed

classification accuracy for our models by averaging over 1000 randomly generated episodes from the test set.

In the experiments performed with RWTH we used the same four-block embedding architecture by adding an eight-block architecture with the same layer composition with the idea of analyzing the need to increase the size of the network given the difficulty of the dataset. The difference in the results obtained in 1-shot and 5-shot scenarios for this dataset was very large. We found that 5-shot scenarios gave better results. Using this discovery we only used 5-shot learning in the remaining experiments.

The best configurations for all datasets is the 5-shot scenario with equals n for train and test scenarios by using more than or equal to 5 classes per training episode. Better results were obtained when the number of classes approaches the total amount of classes in the dataset except on CIARP where the best results were obtained when the number of classes per training episode is 5. In addition, the best configurations of the embedding architecture is a 64-filter for all datasets.

3.2 Wide-DenseNet

We employed a variation on DenseNet called Wide-DenseNet which follows the strategy used by wide residual networks.[6].

We employed a Wide-DenseNet including SE blocks after each dense and transition block. We performed a grid search of hyperparameters to find the model with the best accuracy, averaged over all datasets. We tried growth rate values of 32, 64 and 128 and depth of dense layers of no more than [6,12,24,16], where each number represents the number of dense blocks.

We trained the models using a batch size of 16, an initial learning rate of 10^{-3} with categorical cross entropy optimizer and 400 epochs with a maximum patience of 25. The resulting model used a growth rate of 64 and two dense blocks with 6 and 12 layers respectively, for all datasets.

3.3 Results

In table 1, we can observe that all models have a lower accuracy on the RWTH dataset, which is expected since it has more classes, unsegmented hand images and class imbalances. Prototypical Networks have similar accuracy for LSA16 and CIARP datasets beating the rest of the models, also expected since both datasets have very few examples. For LSA16 they achieve better accuracy than VGG16 and DenseNet; and for CIARP they achieve similar or better accuracy than LeNet CNN and DenseNet. The accuracy of DenseNet on the RWTH is slightly bigger than for other models. Our hypothesis was that Prototypical Networks obtained low accuracy because the images of the hands were unsegmented. It should be noted that the use of data augmentation did not bring significant improvements in the accuracy obtained in LSA16 and CIARP.

Another fact to consider is that better results were obtained with those parameters that reduced the size of the architectures.

<i>Method</i>	<i>LSA16</i>	<i>RWTH</i>	<i>CIARP</i>
LeNet [12]	-	-	99.20
Inception (fine-tuning) [10]	-	85.50	
VGG16 [14]	95.92	82.88	
Inception+SVM (pre-trained) [14]	93.67	78.12	-
DenseNet	98.07	91.10	99.93
DenseNet ++	98.90	94.00	99.99
Prototypical Networks	99.15	79.93	99.98
Prototypical Networks ++	99.26	80.85	100.00

Table 1: Accuracy of various convolutional neural network based models on three datasets: LSA16, RWTH and CIARP. Models with "++" used data augmentation as described in this section.

In figure 2, we can observe the accuracy of Prototypical Networks and DenseNet models trained by varying sample sizes. We performed experiments using the same embedding architectures and configurations described in this section varying the training sample sizes with percentages of 44%, 67% and 85% and a fixed test size of 25%. From the obtained results, we can see that the performance of the DenseNet models increases as more training examples are provided. From figure 2(b) we can see that the DenseNet model trained using data augmentation obtains better results than the one trained without. On the other hand, Prototypical Networks models do not show a significant increase in performance as the percentage of samples increases. In figures 2(b) and 2(c) we can observe how the use of data augmentation, on RWTH and CIARP datasets respectively, results in Prototypical Networks models with great accuracy improvement compared to the results obtained on LSA16, figure 2(a)), where the increase of performance from the use of data augmentation is minimal.

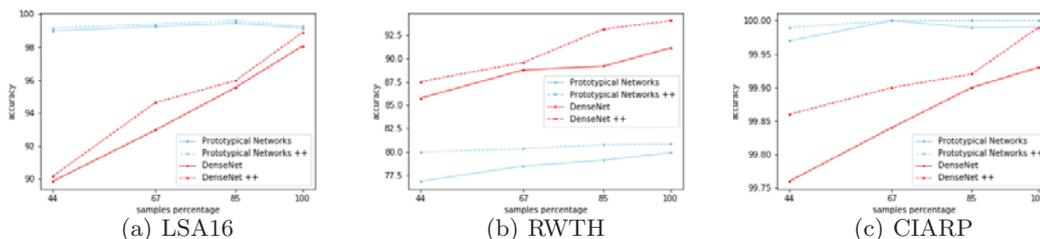


Fig. 2: Accuracy of Prototypical Networks and DenseNet models trained by varying sample sizes on three datasets: LSA16, RWTH and CIARP. Each plot represents a different dataset where the x-axis is the percentage of samples used and the y-axis is the accuracy obtained. Models with "++" used data augmentation as described in this section.

4 Conclusion

We have performed experiments to evaluate the mean accuracy of Prototypical Networks and Wide-DenseNet on three handshape recognition datasets with and without data augmentation techniques. For all datasets we found models that showed a performance on par with or better than the state of the art. All models achieve near-perfect accuracy on CIARP. This shows that the dataset is too simple as a benchmark for handshape recognition. While it has more samples on the other datasets (6000), the samples are too homogeneous and do not have enough variation to generalize results to real-world application. Prototypical Networks provide a new state-of-the-art accuracy on the LSA16 dataset, surpassing all other known methods. Wide-DenseNets also improve upon the state of the art, and come close to prototypical networks. However, we can observe that the performance gap between the two datasets decreases sharply when the sample size increases. We have also obtained a new state-of-the-art on the RWTH dataset with Wide-DenseNet, while Prototypical Networks also improved upon all previous results; this shows that newer convolutional architectures can work better with less data, but there's still room for improvements using specialized models.

In future work, we will focus on comparing with other datasets to better understand the relationship between models and dataset complexities for handshape recognition. We also see the need to compare with pre-trained models, which are another way to alleviate the lack of data in a certain domain, as well as methods that can take advantage of unlabeled data. Finally, we will investigate the possibility of merging data sets from different sign languages to augment the sample size, as well as identify the types of data augmentation that lead to an improvement in state-of-the-art models.

References

1. Alani, A.A., Cosma, G., Taherkhani, A., McGinnity, T.M.: Hand gesture recognition using an adapted convolutional neural network with data augmentation. 2018 4th International Conference on Information Management (ICIM) pp. 5–12 (2018)
2. Ameen, S., Vadera, S.: A convolutional neural network to classify american sign language fingerspelling from depth and colour images. *Expert Systems* 34(3), e12197 (February 2017), <http://usir.salford.ac.uk/id/eprint/41255/>
3. Barros, P., Magg, S., Weber, C., Wermter, S.: A multichannel convolutional neural network for hand posture recognition. pp. 403–410 (09 2014)
4. Cubuk, E.D., Zoph, B., Mane, D., Vasudevan, V., Le, Q.V.: Autoaugment: Learning augmentation strategies from data. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. pp. 113–123 (2019)
5. Goodfellow, I., Bengio, Y., Courville, A.: *Deep Learning*. The MIT Press (2016)
6. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR) pp. 770–778 (2015)
7. Hu, J., Shen, L., Sun, G.: Squeeze-and-excitation networks. 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition pp. 7132–7141 (2017)

8. Huang, G., Liu, Z., van der Maaten, L., Weinberger, K.Q.: Densely connected convolutional networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (2017)
9. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization (2014), <http://arxiv.org/abs/1412.6980>, cite arxiv:1412.6980Comment: Published as a conference paper at the 3rd International Conference for Learning Representations, San Diego, 2015
10. Koller, O., Ney, H., Bowden, R.: Deep hand: How to train a cnn on 1 million hand images when your data is continuous and weakly labelled. In: IEEE Conference on Computer Vision and Pattern Recognition. pp. 3793–3802. Las Vegas, NV, USA (Jun 2016)
11. Koller, O., Ney, H., Bowden, R.: Deep hand: How to train a cnn on 1 million hand images when your data is continuous and weakly labelled. In: IEEE Conference on Computer Vision and Pattern Recognition. pp. 3793–3802. Las Vegas, NV, USA (Jun 2016)
12. Núñez Fernández, D., Kwolek, B.: Hand posture recognition using convolutional neural network. In: Mendoza, M., Velastín, S. (eds.) Progress in Pattern Recognition, Image Analysis, Computer Vision, and Applications. pp. 441–449. Springer International Publishing, Cham (2018)
13. Pham, H., Guan, M., Zoph, B., Le, Q., Dean, J.: Efficient neural architecture search via parameters sharing. In: Dy, J., Krause, A. (eds.) Proceedings of the 35th International Conference on Machine Learning. Proceedings of Machine Learning Research, vol. 80, pp. 4095–4104. PMLR, Stockholmsmässan, Stockholm Sweden (10–15 Jul 2018), <http://proceedings.mlr.press/v80/pham18a.html>
14. Quiroga, F., Antonio, R., Ronchetti, F., Lanzarini, L.C., Rosete, A.: A study of convolutional architectures for handshape recognition applied to sign language. In: XXIII Congreso Argentino de Ciencias de la Computación (La Plata, 2017). (2017)
15. Ronchetti, F., Quiroga, F., Estrebou, C., Lanzarini, L., Rosete, A.: Sign language recognition without frame-sequencing constraints: A proof of concept on the argentinian sign language. In: Ibero-American Conference on Artificial Intelligence. pp. 338–349. Springer (2016)
16. Ronchetti, F., Quiroga, F., Lanzarini, L., Estrebou, C.: Handshape recognition for argentinian sign language using probsom. Journal of Computer Science and Technology 16(1), 1–5 (2016)
17. Snell, J., Swersky, K., Zemel, R.S.: Prototypical networks for few-shot learning. CoRR abs/1703.05175 (2017), <http://arxiv.org/abs/1703.05175>
18. Tang, A., Lu, K., Wang, Y., Huang, J., Li, H.: A real-time hand posture recognition system using deep neural networks. ACM Transactions on Intelligent Systems and Technology (TIST) 6(2), 21 (2015)

Sintonización Dinámica del Método Paralelo de Predicción de Incendios Forestales ESSIM-DE

María Laura Tardivo^{1,3,*}, Paola Caymes Scutari^{1,2}, Germán Bianchini¹ y Miguel Méndez Garabetti¹

¹ Laboratorio de Investigación en Cómputo Paralelo/Distribuido (LICPaD), Dpto. de Ingeniería en Sistemas de Información - UTN-FRM - Mendoza - Argentina.

² Consejo Nacional de Investigaciones Científicas y Técnicas (CONICET)

³ Departamento de Computación - UNRC- Córdoba - Argentina.

Resumen La calidad de las soluciones obtenidas por los métodos de predicción está condicionada por la falta de precisión o incertidumbre presente en los parámetros de entrada que alimentan los modelos, por lo cual resulta de interés el desarrollo y perfeccionamiento de métodos que permitan tratar la incertidumbre, a fin de obtener predicciones confiables. ESSIM-DE es un método general de predicción y reducción de incertidumbre basado en Análisis Estadístico, la metaheurística Evolución Diferencial y Cómputo de Alto Rendimiento, y ha sido aplicado a la predicción de la línea de fuego en incendios forestales. En este trabajo se presenta una estrategia de sintonización aplicada de forma automática y dinámica al parámetro que regula la cantidad de generaciones del proceso evolutivo, a fin de mejorar su rendimiento y evitar estancamiento y/o convergencia prematura. Se describen las métricas monitorizadas, los puntos y acciones de sintonización en el modelo distribuido y los resultados obtenidos con diferentes casos de quemas controladas.

Keywords: Sintonización Dinámica, Evolución Diferencial, Predicción de Incendios, Cómputo de Alto Desempeño

1. Introducción

Los incendios forestales constituyen año a año una gran amenaza en diferentes regiones del mundo, sobre todo en época estival, donde las temperaturas elevadas y la sequía prolongada propician el ambiente ideal para el desarrollo de estos fenómenos. Debido a las graves consecuencias que producen, contar con herramientas que permitan predecir su comportamiento resulta de importancia para colaborar en los planes de lucha contra el fuego y en la prevención de incendios.

La predicción de un incendio forestal consiste en determinar cuál será la propagación del fuego sobre el terreno en un instante de tiempo futuro. Generalmente, los métodos de predicción implementan modelos que describen el

* Este trabajo ha sido financiado mediante los proyectos SIUTIME0004736TC y SIUTNME0004819 de la UTN-FRM.

comportamiento del fuego, y utilizan como dato de entrada un grupo de variables representando aquellos factores que condicionan la propagación. Entre ellas encontramos la velocidad y dirección del viento, la pendiente del terreno, el tipo de material combustible, la humedad de dicho material, etc. Lamentablemente, no es posible contar con los valores exactos para estos factores, debido a la imposibilidad de dotar todo el terreno forestal con instrumentos de medición y, sobre todo, debido a que algunos cambian dinámicamente durante el desarrollo del incendio. Esta falta de conocimiento respecto de los valores de los parámetros se denomina incertidumbre, y podría provocar consecuencias drásticas si las salidas de los modelos obtienen soluciones alejadas de la realidad.

En los últimos años se han desarrollado, estudiado y mejorado una serie de métodos pertenecientes a la clase de los denominados Métodos Conducidos por Datos con Múltiples Soluciones Solapadas (*Data Driven Methods with Multiple Overlapped Solutions*, DDM-MOS [1]), los cuales obtienen predicciones de la línea de fuego en base a la agregación de múltiples soluciones, y que se enfocan en reducir este impacto negativo que provoca la incertidumbre. En particular, hemos desarrollado el método ESSIM-DE (por las siglas en inglés para “Sistema Estadístico Evolutivo con Modelo de Islas y Evolución Diferencial”), el cual arroja una predicción en base a tres pilares fundamentales: Análisis Estadístico, para obtener patrones de comportamiento del fuego, la metaheurística Evolución Diferencial (*Differential Evolution*, DE) [3], para orientar la búsqueda entre un experimento factorial de simulaciones, y Cómputo Paralelo [6], para acelerar el proceso de predicción y potenciar el tamaño de la muestra.

Con el fin de mejorar el rendimiento de ESSIM-DE, hemos definido e incorporado al método una estrategia dinámica y automática de sintonización, la cual es presentada en este trabajo. Las estrategias de sintonización automáticas y dinámicas [5] permiten calibrar, mejorar, ajustar, o modificar algún aspecto crítico, cuello de botella o factor limitante de mejor desempeño de la aplicación, y consisten en incorporar las etapas de instrumentación, monitorización, análisis y sintonización, durante la ejecución del programa. En la etapa de instrumentación se incluyen directivas en el código fuente para registrar ciertas métricas de interés. Dichas métricas en general corresponden a algún modelo del comportamiento de la aplicación, y definen conocimiento específico acerca de la misma, el cual puede responder a modelos matemáticos de rendimiento, lógica difusa, heurísticas definidas *ad hoc*, entre otras. En la etapa de monitorización se registran las métricas, para su posterior análisis y sintonización. En la etapa de análisis se procesan los valores registrados y, en base este conocimiento específico, se determinan las acciones de sintonización necesarias para mejorar el rendimiento de la aplicación. Posteriormente, en la etapa de sintonización se aplican las acciones definidas tendientes a mejorar los aspectos críticos. Todas estas etapas se realizan en tiempo de ejecución.

En este trabajo presentamos un modelo de rendimiento que permite sintonizar dinámica y automáticamente una de las etapas claves de ESSIM-DE, la Etapa de Optimización llevada a cabo por el algoritmo Evolución Diferencial. Esta metaheurística es un optimizador poblacional (basado en múltiples solu-

ciones) y consiste en un proceso iterativo en el cual se evoluciona una población de individuos, cada uno de los cuales representa una solución al problema. Un posible problema de rendimiento asociado a Evolución Diferencial lo constituye el estancamiento y/o la convergencia prematura hacia un óptimo local [4,8]. En este trabajo se establece un criterio mediante el cual es posible sintonizar la cantidad de generaciones por las cuales han de evolucionar las poblaciones a fin de detectar de manera anticipada una tendencia al estancamiento y/o a la convergencia prematura. El modelo de rendimiento utiliza información estadística de la tendencia de la dispersión de la población, monitorizada de manera distribuida. El objetivo es evitar que estos dos fenómenos afecten de manera negativa el rendimiento del método, tanto en la calidad de las predicciones como en el tiempo de respuesta, y que la sintonización sea realizada independientemente del caso de quema considerado y de las características particulares de la ejecución.

2. Predicción de Incendios con ESSIM-DE

Para realizar la predicción del frente de fuego, en ESSIM-DE el desarrollo total del incendio es dividido en diferentes instantes de tiempo discretos, denominados pasos de simulación. Arquitectónicamente, ESSIM-DE posee una doble jerarquía de procesos que se organizan en islas paralelas y colaboran mediante migración [6], bajo la supervisión de un proceso *Monitor*.

En la Figura 1 pueden apreciarse las cuatro etapas que involucra el método, de las cuales nos centraremos en la Etapa de Optimización (**EO**), la cual se lleva a cabo entre los procesos *workers* (**W**) y el *Master* (**M**) de cada isla. Esta etapa permite evolucionar una población de individuos en base al algoritmo Evolución Diferencial. Cada individuo representa una combinación de valores para las variables que determinan el avance del fuego (la velocidad de viento, dirección y pendiente del terreno, tipo de material combustible, humedad del material combustible vivo, etc). El proceso *Master* inicializa la población (**pop**), aplica los operadores de mutación y cruzamiento de DE para generar nuevos individuos candidatos, y distribuye los individuos entre los *workers*. Estos utilizan el estado actual del incendio y junto con un individuo para realizar la simulación y posteriormente evalúan la aptitud de la predicción obtenida, ponderando la precisión de la simulación mediante una comparación entre las celdas quemadas en el incendio real y las celdas alcanzadas por el fuego en el mapa obtenido. Por lo tanto, la aptitud representa el porcentaje de coincidencia entre ambos mapas. Posteriormente se lleva a cabo las subsiguientes etapas: Estadística (**EE**), de Calibración (**EC**) y de Predicción (**EP**), que recopilan la información y los resultados obtenidos por todas las islas en la etapa **EO**, y en base a ello permiten arrojar la predicción global del estado del incendio. Una primera mejora al método ESSIM-DE fue la incorporación de un operador **OR**, el cual se aplica una vez que se obtiene la predicción para el paso de simulación considerado. Este operador genera una nueva población (nuevo espacio de búsqueda). Para mayor detalle en el funcionamiento de las etapas de ESSIM-DE es posible consultar [7].

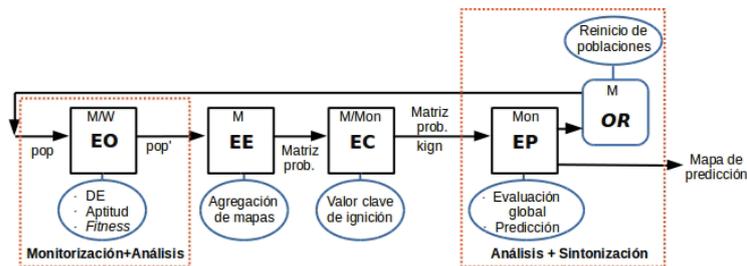


Figura 1. Esquema general de ESSIM-DE. **EO**: Etapa de Optimización, **EE**: Etapa Estadística, **EC**: Etapa de Calibración, **EP**: Etapa de Predicción, **OR**: Operador de Reinicio poblacional, **DE**: Evolución Diferencial, **M**: *Master*, **W**: *Worker*, **Mon**: *Monitor*. Línea punteada: proceso de sintonización-ESSIM-DE(ldr)

3. Sintonización Dinámica: modelo de rendimiento

El proceso *Master* de cada isla es el que determina cuándo la población finaliza su evolución para continuar con las siguientes etapas del proceso de predicción. En ESSIM-DE la condición de fin del ciclo evolutivo de la **EO**, determinada por alcanzar cierta cota máxima o límite de iteraciones, tiene una doble influencia en el proceso de optimización, puesto que limita la cantidad de generaciones evolutivas por las cuales evolucionará cada población, y en consecuencia, determina el tiempo de ejecución del proceso evolutivo. Por lo tanto, se propuso definir el proceso de sintonización dinámica [2] aplicado al límite de iteraciones evolutivas.

En la definición del proceso de sintonización hemos considerado dos posibles problemas asociados a la evolución llevada a cabo por DE: la convergencia prematura y el estancamiento. La convergencia prematura es la situación en la cual la población converge a un óptimo local, debido a la pérdida de diversidad; por su parte, el estancamiento es la situación en la cual el optimizador no es capaz de generar nuevas soluciones mejores que sus antecesoras, aún cuando la población presente cierta diversidad [4]. El problema del estancamiento de la población depende de los movimientos efectivos del optimizador DE. Cuando se genera un nuevo individuo se produce un movimiento en el espacio de búsqueda. Este movimiento se considera efectivo si el nuevo individuo generado posee mejor valor de aptitud respecto de su antecesor. De todos los posibles movimientos que se realizan en la población, algunos son efectivos, mientras que otros no lo son, y por lo tanto estos últimos implican un esfuerzo computacional en vano. Para abordar el tratamiento de estos problemas se propuso cuantificar dos métricas diferentes para la población:

- Movimientos efectivos (métrica ME): cuantifica la cantidad de individuos que luego de un ciclo evolutivo han sido mejorados (mejor valor de aptitud que el de su predecesor).
- Diversidad poblacional (métrica IQR): cuantifica la dispersión de la población (variabilidad de la distribución de la población). Se computa en base al Rango Intercuartil de los valores de aptitud de los individuos.

Estas métricas fueron monitorizadas y registradas en análisis experimental, utilizando diferentes casos de quema controladas. En una primera aproximación, se utilizaron valores fijos como cota de ambas métricas. Sin embargo, fue posible notar que la forma de las gráficas de la métrica IQR a lo largo de las iteraciones es variable de un caso a otro. Esto se debe a que la distribución de los valores de aptitud está influenciada por múltiples factores (velocidad de convergencia del método, factor de mutación, probabilidad de cruzamiento, tasa de migración, dimensiones del mapa, entre otros). Por tal motivo, el modelo de rendimiento definido plantea utilizar la información de la dispersión de la población en la sucesión de valores de IQR, obtenidos a lo largo de las generaciones evolutivas. El fin es detectar si la población se encuentra en un estado con tendencia a estancamiento y/o convergencia prematura, haciendo uso de la información reciente acerca de las distribuciones, de acuerdo al caso considerado.

Para lograrlo, se propuso registrar el mínimo valor de IQR obtenido, actualizar dicho valor a lo largo de las iteraciones, y compararlo respecto del valor IQR obtenido en la iteración corriente. En la Figura 1 se ejemplifica la idea. El eje x representa la generación evolutiva (iteración) y el eje y el valor de IQR registrado. En la gráfica se señala el mínimo IQR registrado (Min_IQR) y el IQR de la iteración corriente ($Curr_IQR$). Si la diferencia entre ambos, denominada $Delta$, es muy pequeña, entonces la población ha logrado una distribución muy similar en cuanto a valores de aptitud, en alguna de las iteraciones anteriores. Siendo $Delta = (Min_IQR - Curr_IQR)$, el criterio de sintonización queda definido por la expresión (1), en la cual se verifica que la tendencia de IQR sea decreciente ($Delta \geq 0$) y $Delta$ acotado por un cierto valor pequeño $umbral_DELTA$. Por su parte, también se considera el valor de la métrica ME , requerida para analizar si el optimizador aún cuenta con movimientos efectivos a realizar sobre el espacio de búsqueda, acotada a un cierto valor $umbral_ME$.

En ESSIM-DE, la expresión (1) es computada en cada iteración a lo largo de la **EO**, de acuerdo con la distribución actual de la población, para cada mapa de quema y condiciones particulares de la ejecución. La propuesta pretende mejorar los tiempos de respuesta respecto del método sin sintonización, puesto que la nueva condición utilizada como criterio de corte es una propiedad específica de cada población, lo que permite evitar aquellos ciclos innecesarios. El proceso de sintonización se incluye en la Figura 1 con recuadros en líneas de puntos rojos. Los valores de aptitud son registrados en la Etapa de Monitorización y las métricas son computadas en la Etapa de Análisis del *Master*. Al finalizar cada iteración, cada *Master* envía las métricas de su isla hacia el proceso *Monitor* (**Mon**), quien realiza una agregación de los valores recibidos en su Etapa de Análisis, y determina si existe alguna isla con tendencia a estancamiento y/o convergencia prematura en la **ES**. Si se evalúa como verdadera la expresión (1) con los valores computados de alguna de las islas, el *Monitor* decide frenar la evolución de todas las islas, realizar la predicción agregando la información de todas ellas, e iniciar con el siguiente paso de predicción.

$$Delta \leq umbral_DELTA \wedge Delta \geq 0 \wedge ME \leq umbral_ME \quad (1)$$

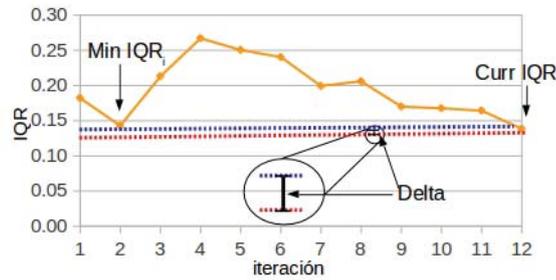


Figura 2. Ejemplo de cómputo del valor *Delta*.

Cuadro 1. Casos de estudio: dimensión, pendiente y tiempo de inicio, fin e incremento.

Caso	Ancho (m)	Alto (m)	Pend. (grados)	T. ini. (min)	Incr. (min)	T. fin (min)
A	89	109	21	2	2	14
B	60	90	6	2	2	10
C	89	91	21	2.5	2.5	12.5

4. Resultados Obtenidos

La propuesta fue validada en base a la experimentación con tres casos de quemas controladas pertenecientes al proyecto SPREAD [9] (ver Tabla 1). Cada experimento fue efectuado con ESSIM-DE sin el proceso de sintonización, y ESSIM-DE(ldr), sigla con la que denominamos al método con el proceso de sintonización **dinámica** del **límite** de iteraciones y con el operador de **reinicio** de poblaciones. Los resultados de un total de 10 ejecuciones con diferentes semillas fueron promediados. El modelo de islas fue configurado con 5 islas, de 7 *workers* cada una. El proceso de migración involucra el 20% de los individuos de la población, y se realiza en cada iteración. El tamaño de cada población se definió como 200 individuos. Para ESSIM-DE y ESSIM-DE(ldr) se utilizó la misma configuración de los parámetros evolutivos: probabilidad de cruzamiento 0.3, factor de mutación 0.9, cruzamiento binomial. El valor utilizado como *umbral_ME* fue establecido al 20%. El valor establecido para el parámetro *umbral_DELTA* fue de 10^{-3} . La evaluación de la calidad de la predicción se cuantifica en base a un función de *fitness*, que responde al índice Jaccard [7], donde se realiza la división entre A y B, siendo A el conjunto de celdas en el mapa real sin el subconjunto de celdas quemadas antes de iniciar el proceso predictivo, y B representa el conjunto de celdas en el mapa resultado de la **EP**, sin el subconjunto de celdas quemadas antes de iniciar la predicción (celdas que se eliminan para evitar resultados sesgados). Por lo tanto, el valor de *fitness* puede ser considerado el porcentaje de coincidencia entre el mapa obtenido de la simulación y el mapa real, y estará comprendido en el rango [0,1]: un valor igual a 1 representa una predicción perfecta, y un valor igual a cero indica el máximo error.

Los resultados obtenidos son presentados en las figuras 3, 4 y 5. En cada figura se incluyen dos gráficos. El gráfico (a) representa los promedios de *fitness* para cada paso de predicción, obtenidos por ESSIM-DE(ldr), y comparados respecto de la versión sin sintonización ESSIM-DE. El gráfico (b) muestra la distribución de los valores de *fitness* obtenidos para cada paso de predicción y método, lo que permite analizar la dispersión de los resultados y la robustez del método en términos de la obtención de soluciones diferentes bajo distintas ejecuciones. La Tabla 2 muestra los valores promedio de tiempo de ejecución obtenidos.

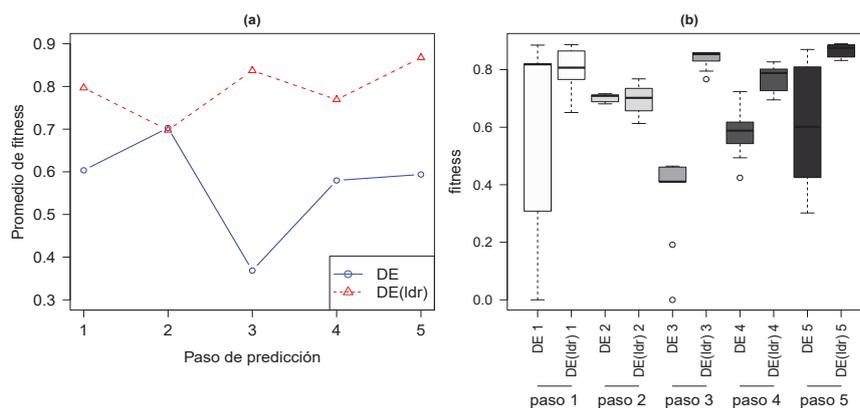


Figura 3. Caso A. (a) Promedio de *fitness* con ESSIM-DE y ESSIM-DE(ldr). (b) Distribución de los valores de *fitness* obtenidos en cada paso de predicción.

La Figura 3 muestra los resultados obtenidos para el caso de quema A de la Tabla 1. El incendio se inicia en el minuto 2 y tiene una duración de 14 minutos, constituyendo un paso de calibración y cinco pasos de predicción (calibración: minuto 2 al 4, pasos de predicción: primer paso minuto 4 al 6, segundo minuto 6 al 8, tercero minuto 8 al 10, cuarto minuto 10 al 12, quinto minuto 12 al 14). En general se puede observar del gráfico (a) que ESSIM-DE obtiene baja calidad de las predicciones, sobre todo en el paso de predicción 3. Observando la figura (b), en el paso 1 y en el paso 5, se puede observar amplia variabilidad de los resultados obtenidos con ESSIM-DE, mientras que para ESSIM-DE(ldr) se observa menor distribución en todos los pasos de predicción, lo que indica una robustez en dichos pasos en términos de la obtención de soluciones bajo diferentes ejecuciones. En general, ESSIM-DE(ldr) obtiene mejor rendimiento, con promedio de *fitness* superiores a 0.7 en todos los pasos de predicción, y con una reducción del tiempo de ejecución del 38% aproximadamente. Esta reducción de tiempo está asociada a la capacidad de ESSIM-DE(ldr) de detectar anticipadamente la tendencia al estancamiento y/o convergencia prematura, evitando ciclos innecesarios, y por ende, logrando menor tiempo de respuesta. Las reducciones de tiempo de ejecución son muy relevantes en el contexto de métodos de predicción,

permitiendo la toma de decisiones con anticipación al avance del incendio. Es importante remarcar que la reducción del tiempo no sólo depende de la velocidad con la que se detecta el estancamiento y la convergencia prematura, sino que además depende de las características del caso de quema y del tamaño del mapa considerado, lo cual influye en el comportamiento global del sistema en sus diferentes etapas.

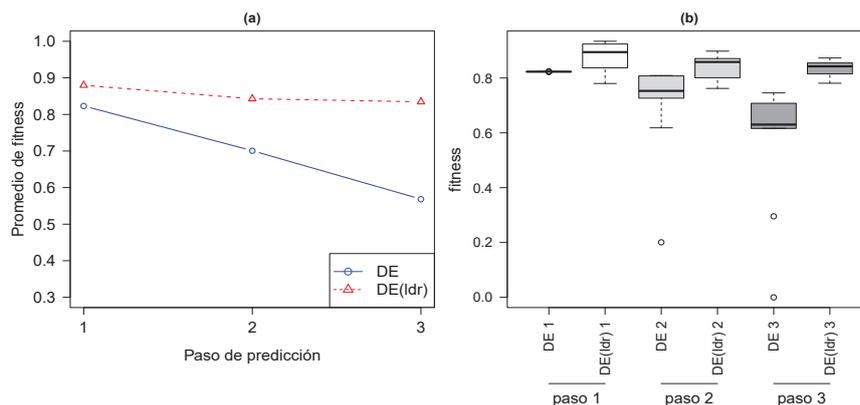


Figura 4. Caso B. (a) Promedio de *fitness* con ESSIM-DE y ESSIM-DE(ldr). (b) Distribución de los valores de *fitness* obtenidos en cada paso de predicción.

La Figura 4 muestra los resultados obtenidos para el caso de quema B de la Tabla 1. El incendio consta de tres pasos de predicción (minutos 4-6, 6-8, 8-10). Se puede observar del gráfico (a) que ESSIM-DE posee una tendencia decreciente en la calidad de las predicciones, con un bajo porcentaje de coincidencia en el tercer paso de predicción, menor al 60%. Si bien en el primer paso de predicción se puede observar buena calidad, superior a 0.8 y con baja distribución de los resultados obtenidos (ver gráfico (b), paso 1 para ESSIM-DE), el método con sintonización ESSIM-DE(ldr) logra mejorar todos los pasos de predicción, con promedios de *fitness* superiores a 0.85 en todos los pasos. En este caso, ESSIM-DE(ldr) obtiene una ganancia en tiempo de ejecución, con una importante reducción del 73%, aproximadamente.

La Figura 5 muestra los resultados obtenidos para el caso de quema C de la Tabla 1. El incendio consta de tres pasos de predicción. Se puede observar del gráfico (a) que ESSIM-DE(ldr) obtiene mejor promedio de *fitness* en el primer y tercer paso de predicción, mientras que ESSIM-DE obtiene una calidad cercana a 0.9 en el segundo paso de predicción. Es importante destacar en este experimento que ESSIM-DE(ldr) mejora significativamente la calidad de la predicción obtenida en el tercer paso de predicción respecto de ESSIM-DE, logrando un porcentaje de coincidencia con el incendio real cercano a al 80% de coincidencia (*fitness* cercano a 0.8). Además, se puede observar a partir de los

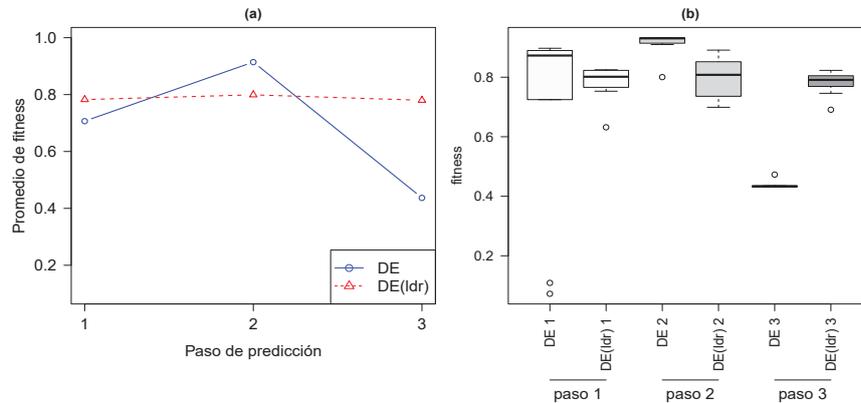


Figura 5. Caso C. (a) Promedio de *fitness* con ESSIM-DE y ESSIM-DE(ldr). (b) Distribución de los valores de *fitness* obtenidos en cada paso de predicción.

resultados de la Tabla 2 que el método con sintonización dinámica logra reducir los tiempos de ejecución en un 63 % respecto del método sin sintonización, aproximadamente.

Cuadro 2. Tiempo de ejecución promedio en segundos

Caso	ESSIM-DE	ESSIM-DE(ldr)	Reducción de tiempo
A	3540	1292	63.5 %
B	1696	446	73.7 %
C	2332	1438	38.3 %

En general, considerando el total de los casos, se mejoró la calidad de las predicciones obtenidas respecto del método sin sintonización. Con respecto a los tiempos de respuesta, los resultados han arrojado una importante reducción del tiempo de ejecución respecto del método sin sintonización, entre un 38 % y un 73 % menos. Ambas mejoras están asociadas a la detección temprana del estancamiento y la convergencia prematura, evitando ciclos innecesarios al optimizador Evolución Diferencial, y obteniendo mejores individuos que aportan al global de la solución. Es importante destacar que el uso de un modelo matemático/estadístico como base de conocimiento experto en el proceso de sintonización dinámica reduce el tiempo insumido en la toma de decisiones, a diferencia de otras estrategias con mayor costo computacional, como búsqueda, procesos iterativos, lógica difusa, redes neuronales, aproximaciones, entre otras. Esto se debe a que el análisis y las decisiones de sintonización se basan sólo en la evaluación de expresiones matemáticas. En el contexto de métodos de predicción como

ESSIM-DE estas reducciones de tiempo resultan fundamentales para obtener predicciones a más corto plazo.

5. Conclusiones

En este trabajo se presentó una propuesta para sintonizar dinámicamente el método ESSIM-DE, un método general de reducción de incertidumbre, con el fin de mejorar el rendimiento tanto en calidad de las predicciones obtenidas como en el tiempo de respuesta. El modelo de rendimiento definido utiliza la información de la dispersión de la población en la sucesión de valores obtenidos a lo largo de las generaciones evolutivas, con el fin de detectar tendencia a estancamiento y/o convergencia prematura de la población. Es computado de manera distribuida, de acuerdo con la distribución corriente de la población, para cada mapa de quema y condiciones particulares de la ejecución. Los resultados obtenidos han demostrado que la propuesta mejora tanto la calidad como los tiempos de respuesta respecto del método sin sintonización, puesto que la nueva condición utilizada como criterio de corte es una propiedad específica de cada población, lo que permite evitar aquellos ciclos innecesarios por haberse alcanzado cierto nivel de convergencia.

Como trabajo a futuro se propone sintonizar los valores de umbral de las métricas definidas en el modelo de rendimiento, y contemplar otros parámetros del método que puedan ser potencialmente sintonizables, como el tamaño de la población o los parámetros relativos al modelo paralelo/distribuido.

Referencias

1. Bianchini, G. *et al*: Wildland fire growth prediction method based on Multiple Overlapping Solution. *J. Computational Science*, vol 1, nro 4, pp- 229-237 (2010)
2. Caymes-Scutari, P., Bianchini, G., Sikora, A., Margalef, T.: Environment for Automatic Development and Tuning of Parallel Applications. In: *Int. Conf. on High Performance Computing & Simulation (HPCS)*, pp. 743-750, Innsbruck (2016)
3. Storn, R., Price, K. Lampinen, J.: *Differential Evolution: A Practical Approach to Global Optimization*. Springer, New York (2005)
4. Lampinen, J., Zelinka, I.: On Stagnation Of The Differential Evolution Algorithm. *Journal of Computer Physics*, pp.76-83 (2000)
5. Naono, K. and Teranishi, K. and Cavazos, J. and Suda, R.: *Software Automatic Tuning: From Concepts to State-of-the-Art Results*. Springer, New York (2010)
6. Mattson, T., *et al*: *Patterns for Parallel Programming*, Addison-Wesley (2004)
7. Tardivo, M.L., Caymes-Scutari, P., Méndez-Garabetti, M., Bianchini, G: Optimization for an Uncertainty Reduction Method Applied to Forest Fires Spread Prediction. *Computer Science - CACIC 2017*, pp. 13-23, Springer (2018)
8. Yang, M., Li, C., Cai, Z., Guan, J.: Differential Evolution With Auto-Enhanced Population Diversity. *IEEE Transactions on Cybernetics*, vol.45, pp. 302–315 (2015)
9. D. X. VIEGAS, *Project Spread Forest Fire Spread Prevention and Mitigation.*, (2004), <https://cordis.europa.eu/project/rcn/60354/factsheet/fr>, accedido el 2-9-2019.



CACIC 2019

Congreso Argentino de Ciencias de la Computación

XX Workshop Procesamiento Distribuido y Paralelo

Coordinadores

Printista, Marcela (UNSL)
De Giusti, Laura (UNLP)
García Garino, Carlos (UNCu)

Comité Científico

Ardenghi, Jorge (Argentina)
Balladini, Javier (Argentina)
Bertogna, Leandro (Argentina)
Bianchini, Germán (Argentina)
Cancela, Héctor (Uruguay)
Caymes Scutari, Paola (Argentina)
Chichizola, Franco (Argentina)
De Giusti, Armando (Argentina)
De Giusti, Laura (Argentina)
Denham, Mónica (Argentina)
Doallo, Ramón (España)

Fabero, Juan C. (España)
García Garino, Carlos (Argentina)
Gil Costa, Verónica (Argentina)
Iacono, Lucas (Argentina)
Laguna, Carlos Alfonso (España)
Luque, Emilio (España)
Marín, Mauricio (Chile)
Martín, María J. (España)
Naiouf, Marcelo (Argentina)
Olcoz Herrero, Katzalin (España)
Pacini, Elina (Argentina)

Pardo, Xoan (España)
Piccoli, Fabiana (España)
Piñuel, Luis (España)
Printista, Marcela (Argentina)
Rexachs, Dolores (España)
Rodríguez de Souza, Josemar (Brasil)
Rodríguez León, Casiano (España)
Suppi, Remo (España)
Tinetti, Fernando (Argentina)
Tirado, Francisco (España)
Tourinho, Juan (España)

Temas

Algoritmos Paralelos
Métricas de Performance y Eficiencia en Paralelismo
Escalabilidad
Optimización de algoritmos paralelos
Modelos de Arquitecturas paralelas
Computación de Altas Prestaciones
Predicción de Performance en Algoritmos Paralelos
Balance de Carga en Sistemas Distribuidos/Paralelos
Mapping, scheduling y virtualización en arquitecturas paralelas
Arquitecturas multiprocesador orientadas a Paralelismo
Software de base para multicores, simétricos y asimétricos
Arquitecturas distribuidas como Procesadores Paralelos
Evaluación de performance en sistemas paralelos homogéneos y heterogéneos
Cluster, Grid y Cloud Computing
Middleware para arquitecturas de procesamiento paralelo
Fallas de hardware y software en Arquitecturas Paralelas. Tolerancia a Fallas
Aplicaciones de Sistemas Distribuidos/Paralelos
Simulación de Alta Performance

A Study of Hardware Performance Counters Selection for Cross Architectural GPU Power Modeling

Martín Pi Puig¹, Laura de Giusti^{1,2}, Marcelo Naiouf¹, and Armando De Giusti^{1,3}

¹ Instituto de Investigación en Informática LIDI, CEA-CIC, Facultad de Informática, Universidad Nacional de La Plata, La Plata, Argentina.

² Comisión de Investigaciones Científicas de la Provincia de Buenos Aires (CIC), La Plata, Argentina.

³ Consejo Nacional de Investigaciones Científicas y Técnicas (CONICET), CABA, Argentina.

{mpipuig, ldgiusti, mnaiouf, degiusti}@lidi.info.unlp.edu.ar

Abstract. In the exascale race where huge corporations are spending billions of dollars on designing highly efficient heterogeneous supercomputers, the real need to reduce power envelopes forces current technologies to face crucial challenges as well as it demands the scientific community to evaluate and optimize the performance-power ratio. While energy consumption continues to climb up, the viability of these massive systems becomes a growing concern. In this context, the relevance of specific power-related research works turns into a priority. So we here develop an exhaustive step-by-step process for selecting a comprehensive set of hardware performance counters to serve as an input in an eventual GPU cross-architectural power consumption model. Our experiments show a high power-performance correlation between shared GPU events. Also, we present a set of events that delivers exclusive performance information in order to predict accurately GPU power fluctuations.

Keywords: GPU · Architectures · Performance Counters · Power Consumption · Correlation · Prediction.

1 Introduction

Currently, High Performance Computing (HPC) systems are power-bounded, and this trend is expected to continue for the upcoming future. Then, an exascale computing system (order 10^{18} FLOPS) estimated to arrive in 2021, is expected to work under a rigorous 20-40 megawatts power scope [1][2][3]. To meet this performance (a thousandfold increase over the first petascale computer) in an acceptable power cost will require to exploit improvements and innovations in hardware and software design. The major hurdles for the scientific community include excessive energy consumption, higher fault rate, and porting applications from petascale to exascale computing [4][5].

So as to improve energy efficiency the HPC community has adopted at an impressive rate the use of accelerator-based supercomputers. Furthermore, the NVIDIA Graphics Processing Units (GPUs) have completely outperformed its main competitor, the Intel Xeon Phi, becoming the principal general purpose accelerator for any supercomputer system. Current general purpose computing on GPUs (GPGPU) applications can be found in the HPC area [6] but there are plenty of emerging applications in other domains. Examples are deep learning [7], autonomous vehicles [8], image processing [9][10], encryption [11][12], medical algorithms [13][14], among others.

Together with their performance capabilities, GPU-based accelerators are also popular because of their remarkable energy efficiency [15][16], as portrayed in The Green500 list [17]. The June 2019 ranking shows that 8 of 10 top (performance-per-watt) computer systems incorporate GPU accelerators.

As the usage of GPUs has become dominant, it is increasingly critical to investigate the power consumption profile, to identify eventual power bottlenecks and to find mechanisms to maximize their energy efficiency when running real applications.

Since not all the current GPUs that constitute world's largest HPC centers are equipped with power sensors that can directly assess the power dissipation, the device power is either inspected from a isolated power meter or derived from the examined current and voltage. Nevertheless, variations in power consumption happen so fast that an external measurement reading cannot provide an in time accurate sample. Also such approaches do not represent a practical solution especially on large-scale distributed systems being the fact that they involve additional hardware devices.

In contrast, the design and use of software approaches to predict power consumption is still in progress. Albeit some prior attempts have been made with some constrains like targeting a particular application or architecture, they cannot contribute to the understanding of GPU power consumption considering the lack of analysis in the metrics they can gather.

So an alternative is to directly measure processor events that are causing energy and temperature changes, and to analyze metrics correlation with power consumption. The best proxies for this approach are the hardware performance counters found in all modern computer architectures. Considering a very large number of counters, the task of selecting events that are most representative of the full system power profile is becoming a challenge. Furthermore, the complexity naturally increases when different types of GPU architecture implementations are considered. The matter is further problematic by the fact that each one has a finite number of registers that can be simultaneously recorded reducing the prediction accuracy.

Generally, selected counters and power prediction models vary from one CPU/GPU architecture to the next due to differences in the accessibility and availability of hardware counters and the fact that different subset of counters can reduce the power prediction error in a given microarchitecture.

In this work, we investigate and analyze hardware performance counters on two different GPU architectures to test its correlation with power consumption. Furthermore, we are seeking for a common subset of performance events in order to build a statistical cross-architecture power prediction model.

The rest of this paper is organized as follows. In Section 2 we review some of the related work. Section 3 provides some background information about performance counters, power measurement, the studied GPU architectures and a brief description of the instrumented benchmarks. Section 4 describes the practical methodology while Section 5 covers our experimentation analysis. Finally, Section 6 includes some conclusions of this work.

2 Related Work

Similar previous works have widely focused on using hardware counters to analyze performance and power in CPU architectures. Isci et al. [18] have developed a software tool that provides real time power measurements for Intel Pentium 4 processors using hardware counters. Lim et al. [19] have constructed a power estimation model for an Intel Core i7 system based on hardware performance counters and a statistical regression method. Bellosa [20] has exploited active hardware units information (from performance events) to establish a thread-specific energy accounting. On the GPU side, Song et al. [21] have combined hardware performance counter data with machine learning and advanced analytics to model power-performance efficiency on GPUs. They briefly described the event collection and selection processes.

3 Background

3.1 Performance Counters

Hardware performance counters are a set of special-purpose registers built into today architectures to read and store the counts of low-level hardware activities within the device. They provide high-speed access to a considerable amount of performance information related to the circuit functional units, multiple caches, main and special memories, register banks, etc.

However, the types, meanings and number of hardware counters that we can access heavily depends on the architecture being used due to the aggressive growth of technology processes. Apart from architectural restrictions, the NVIDIA compute capability also limit the number of events we can collect from a particular GPU version. As a result, our Fermi-based accelerator with compute capability 2.0 contains 4 hardware registers and 74 countable events while our Pascal one (capability 6.1) presents 8 registers and 70 events.

So for a given NVIDIA GPU, we can collect the full range of events through three different approaches. The low-level one implies to use CUPTI (CUDA Performance Tools Interface) [22], a dynamic library that enables the creation of profiling and tracing tools. It is generally included in the CUDA Toolkit [23].

Whereas it supports all available platforms, the library needs modifications in the source code.

On the other hand, we can employ the PAPI-CUDA component to quantify GPU performance events [24]. It is a hardware performance counter measurement technology based on CUPTI. The library is distributed with the latest releases of PAPI and it also needs a source code instrumentation.

Finally and on a high level, the NVPROF tool enables the collection and description of profiling data from the command-line [25]. It does not require to modify the application source code and it also includes a Visual Profiler to automatically generate a timeline of application’s CPU and GPU activity including events and metrics.

In this work, we make use of NVPROF to gather performance event samples during a given kernel execution.

3.2 Power Measurement

We apply a software measurement approach, where GPU power consumption is monitored using the NVIDIA Management Library (NVML) [26].

NVML is a C-based API for supervising and managing diverse NVIDIA GPUs features like the ability to set/unset ECC (Error Correction Code), or to monitor memory usage, temperature, utilization rates, and more. Moreover, this library provides the ability to query power consumption at runtime through the built-in power sensor.

Therefore, we coded a power measurement tool that queries the GPU sensor via NVML and stores data logs to disk. The sampling frequency is fixed at 62.5Hz (16 ms) [27][28].

3.3 GPU Architectures

This work focuses on Fermi and Pascal NVIDIA GPU architectures. Among these different hardware generations, we examine the Fermi-based Tesla C2075 GPU and the Pascal-based GTX Titan Xp GPU exploring performance relations. The old-fashioned Fermi architecture correspond to the initial GPU generations (after Tesla) while Pascal represents a recent graphics hardware. Table 1 details the specifications of the studied devices.

Table 1. Hardware specifications.

Feature	Tesla C2075	GTX Titan Xp
Architecture	Fermi	Pascal
Process Size (nm)	40	16
Transistor Count (M)	3000	11800
TDP (Watts)	225	250
CUDA Cores	448	3584
Memory Size (GB)	6	12
Memory Bandwidth (GB/s)	144	547

These GPUs are designed for massive parallel processing employing the SPMD (Single Program Multiple Data) paradigm. They consist of a large number of streaming multiprocessors (SMs), each one containing multiple small hardware units called streaming processors (SPs) for classical processing, load/store (LD/ST) units, special function units (SFUs) to handle particular calculations and a set of high-speed registers for thread local storage. Each SP has one or multiple fully pipelined integer arithmetic logic units and floating point units. The GPUs also include a L1 cache memory for each SM and an unified L2 cache memory shared by all the SMs. More than that, these devices have an on-chip cached texture memory for 2D spatial locality (image and video applications). They also incorporate an off-chip DRAM global memory supporting a fixed GB data capacity. Then, a PCI-Express bus is used to connect the host system to one or multiple GPUs for bidirectional data transferring.

In the software side, a kernel is executed in parallel by an array of threads. These threads are arranged as a grid of thread blocks in which different kernels can have a diverse grid/block configuration. The grid of blocks and the thread blocks are organized in a three-dimensional hierarchy. So, when a CUDA program invokes a kernel grid, the blocks are enumerated and distributed to SMs with available execution capacity. Threads in a thread block execute concurrently on one SM, and multiple thread blocks can execute concurrently on one SM. As thread blocks finishes, new blocks are launched on the vacated SMs. Finally, the basic scheduling unit of GPU execution is the warp. It is a group of threads that are executed simultaneously by an SM. The mapping between warps and thread blocks can affect the kernel performance.

3.4 Benchmarks

To explain our experimental methodology, we tested a wide variety of GPU applications collecting at the same time power and performance counters data. We chose workloads that exploits different computational, communication and synchronization patterns representing a spectrum of application models.

The benchmarks include Rodinia, Polybench and CUDA SDK programs [29][30][31]. In particular, Rodinia is a set of benchmark applications designed for heterogeneous computing environments. To provide a high-level abstraction of common computing, memory access and communication patterns, each application is classified according to the Berkeley's dwarf taxonomy [32]. Polybench is a selection of CUDA codes to test different NVIDIA GPUs. In addition, we also include samples from the CUDA SDK that are incorporated in the CUDA Toolkit.

These benchmarks include programs representing a high number of real world applications like Data Mining, Image and Video Processing, Linear Algebra, Finance, Simulation, Physics, among others.

4 Experimental Methodology

In this section, we explain the proposed experimental methodology.

Firstly, we modified all the involved applications to include a profiling CPU thread using the Pthreads library, that periodically collects live power data from built-in sensors via NVML. Then, the only communication between the power measuring thread and computational threads is a flag variable. Each kernel execution is repeated multiple times for an accurate analysis. The problem sizes for the different applications were chosen to ensure enough power data points. Finally, power readings are stored in a log file for further processing.

Later, we run the compiling process for all the application benchmarks on both GPU architectures. Furthermore, in order to gather a record of performance events the on-chip GPU monitoring counters are periodically sampled. This counters acquiring process occur in a new application execution using NVPROF, which stores a csv file with hardware counter information.

Lastly, we combine hardware counter data with power readings using Python for statistical analysis. Figure 1 resumes the mentioned experimental methodology.

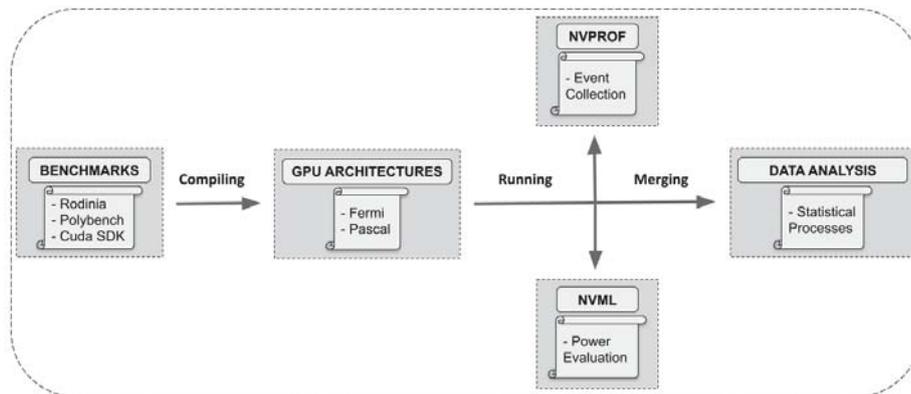


Fig. 1. Experimental framework.

5 Evaluation and Analysis

5.1 Performance Event Support

Since the evaluated GPUs correspond to very distinctive architectures, we got different performance events depending on the accelerator hardware. Figure 2 details the supported events in the examined generations.



Fig. 2. Supported events.

Only 50% of the collected events are shared by both architectures. As one and the other are very distant on the NVIDIA generation roadmap, their hardware features sorely differ.

Apart from the increase in the number of single precision and double precision cores, number of load/store units, number of special function units, size of register file and memory caches, the Pascal architecture integrates other radical technological improvements with respect to Fermi.

The Titan Xp includes half precision floating point operations for high performance Deep Learning. Also, it supports unified memory, a single memory address space accessible from any processor in a system. This technology allows applications to allocate data that can be read or written from code running on either CPUs or GPUs. While Fermi GPUs featured a configurable shared memory and L1 cache that could split the allocation of memory between L1 and shared memory functions depending on workload, beginning with Maxwell, the cache hierarchy was changed. A Pascal SM has its own dedicated pool of shared memory and an L1 cache that can also serve as a texture cache depending on workload.

These factors lead to considerable differences in available performance attributes across different GPU architectures.

5.2 Performance Event Selection

In this section we would like to restrict our exploration to a small subset of performance events that would have strong impact on GPU power consumption. Since we are searching for an online power prediction model, we limit ourselves to the minimum number of hardware registers that are available on both microarchitectures. In this case, the Tesla GPU has 4 performance event slots so it delimits the number of events that can be simultaneously gathered in real time for the whole application.

The steps involved in our experimentation are specified below:

1. GPU performance events and power samples are collected for each benchmark by running the CUDA applications multiple times.
2. All performance events are normalized based on the total number of elapsed clock cycles per multiprocessor yielding a performance event rate for each counter on every application.
3. Events showing a zero value across all applications are removed in a pre-processing task.
4. Pearson, Spearman and Kendall Tau correlations are computed between performance rates and power consumption for all the benchmarking applications.
5. A filter is created to eliminate those events that are not common among all correlation methods.
6. Two thresholds are established, one for positive correlation and the other for the negative one. These border values are chosen so as to filter a high number of performance events within step 5.
7. Event rates showing an overall correlation below the threshold are discarded.

8. From the remaining set of strong power-performance correlation, the mutual correlation is analyzed. This process identifies redundant information.

9. Select as many events as the architecture support. Start by picking an unique event rate for each hardware functionality.

So, we then attempt to find the correlation between selected performance event rates and GPU power consumption employing different statistical approaches: Pearson, Spearman and Kendall Tau methods. While Pearson correlation depicts linear relationships and Spearman represent monotonic behaviors, the Kendall Tau mechanism usually gives deep statistical properties.

Figure 3 shows the power-performance correlation plots for every method on each accelerator. This plot depicts the reduction from 45 to 16 high correlated event rates.

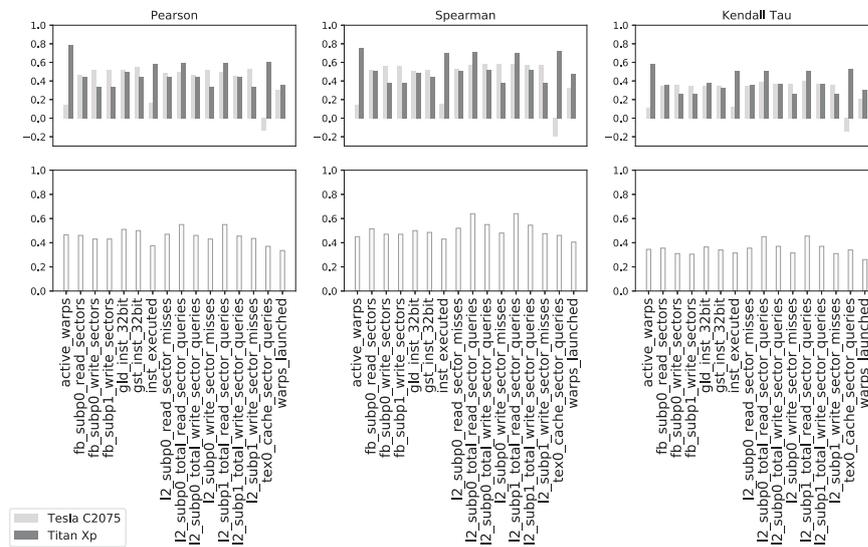


Fig. 3. Pearson, Spearman and Kendall Tau methods for high correlated events.

The first charts row exposes the different statistical correlations for every GPU. On the other hand, second row shows the absolute mean correlation involving the two GPU for each performance rate. This chart gives an insight into high performance-power correlated events through different NVIDIA GPU architectures. It can be seen that these strong correlated events cover the whole accelerator hardware functionality: execution system (*active_warps*, *inst_executed*, *warps_launched*), memory system (*fb_subp0_read_sectors*, *fb_subp0_write_sectors*, *fb_subp1_write_sectors*, *gld_inst_32bit*, *gst_inst_32bit*) and caches system (*l2_subp0_read_sector_misses*, *l2_subp0_total_read_sector_queries*, *l2_subp0_total_write_sector_queries*, *l2_subp0_write_sector_misses*, *l2_subp1_total_read_sector_queries*, *l2_subp1_total_write_sector_queries*, *l2_subp1_write_sector_misses*, *tex0_cache_sector_queries*).

In order to identify redundant information, Figure 4 considers the events in concern, illustrating the two Pearson correlation heatmaps across the selected performance-power correlated events. In this case we use only the Pearson approach because it is more susceptible to outliers, and thus more robust. We also include power data to analyze the strongest correlated event rates.

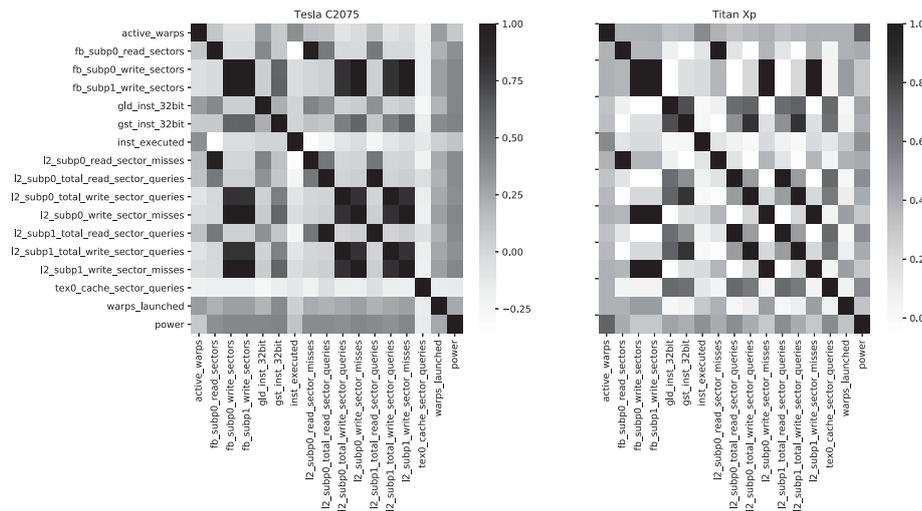


Fig. 4. Correlation heatmap for each GPU.

Strong correlations are expressed by foggy shadows and weak ones are represented by brighter shades. The diagonal terms show that each variable always perfectly correlates with itself. The other components are the Pearson's correlation coefficient between two performance event rates.

Last row of the lower triangular matrix (or last column of the upper one) exposes a high correlation of the selected events with power consumption across different GPU architectures.

As we aim to find the minimum set of hardware performance rates that highly correlate with power, the selection procedure consists on picking an unique event rate for each hardware functionality in the accelerator.

We chose *inst_executed*, the total number of executed instructions per warp, because it has a low correlation coefficient with memory and cache events and it has high correlation with events in the same functional unit (*active_warps* and *warps_launched*) in both GPUs. These considerations allow this event rate to provide non-redundant information to the prediction model.

Furthermore, we pick *gst_inst_32bit*, the total number of global store instructions that are executed by all the threads across all blocks. The event rate has strong correlation with cache performance counters across GPU generations but

it is not restricted to a DRAM memory subpartition since it counts every store instruction in each parallel thread block.

Lastly, we select *l2_subp0_read_sector_misses*, the number of read requests sent to DRAM from slice 0 of L2 cache, due to it has no correlation with *inst_executed* and a weak one with *gst_inst_32bit*. L2 misses/Global transactions correlation exists because of the intrinsic relationship in memory hierarchy in all nowadays computer systems. As the selected event for memory subsystem considers write transactions, we want to complement it by including read operations. L2 cache misses addresses this situation computing DRAM reads from cache failures.

Unlike [21] we found a considerable contribution from L2 performance counters to power consumption (Pearson's coefficients from 0.35 to 0.59).

To sum up *inst_executed*, *gst_inst_32bit* and *l2_subp0_read_sector_misses* with an *elapsed_cycles_sm* normalization are the selected hardware performance counters for an eventual GPU cross-architectural power prediction scheme.

In a last phase, we focus on determining the influence of each selected event rate in an prior power estimation model. Table 2 depicts statistical information from a preliminary Ordinary Least Squares regression model.

Table 2. OLS Linear Regression statistical properties.

Event Rate	Coefficient	P-Value	T-Value
<i>inst_executed_per_cycle</i>	20.153	7.068e-16	9.451
<i>gst_inst_32bit_per_cycle</i>	17.461	5.977e-14	8.605
<i>l2_subp0_read_sector_misses_per_cycle</i>	409.706	5.039e-11	7.289

P-values (probability values) and regression coefficients work together to explain which model relationships are statistically significant and the nature of those links. The coefficients describe the mathematical connections between each event rate and power consumption. Moreover, t-values gives an insight into the coefficient standard error (in power units). Furthermore, the smaller the p-value, the larger the absolute value of the t-value and the greater the evidence of importance for the event.

The prior regression output shows that all the event rates are statistically considerable because their p-values are minimal compared to the usual significance level of 0.05 (less than 5%) and their corresponding t-values are greater than zero.

In particular, *inst_executed_per_cycle* represent beforehand the strongest event rate, while *gst_inst_32bit_per_cycle* and *l2_subp0_read_sector_misses_per_cycle* have a slightly lower contribution to power.

For the results above, we confirm that all the selected performance events are meaningful since variations in the event values are related to changes in power consumption.

6 Conclusions

This paper suggest a specific methodology for selecting hardware performance counters in order to build a real time cross-architectural GPU power model. Our work analyzes performance events on different GPU generations studying performance-power correlation to grant an universal subset of events. This counter selection process is even more difficult than in a classic CPU system due to the complexity and variety of its hardware structure.

Our experimental results reveal that choosing a common high correlated event for every hardware feature allows to generate an unique and robust data input to any power estimation scheme. Consequently, the performance events *inst_executed* (compute), *gst_inst_32bit* (memory), *l2_subp0_read_sector_misses* (cache) and *elapsed_cycles_sm* (cycles rate) can clearly explain power fluctuations in both GPU architectures.

In future work, we will develop a multi-architecture power model using common hardware performance counters to analyze and predict in real time the GPU power consumption.

References

1. M. Lobet, M. Haeefe, V. Soni, P. Tamain, J. Derouillat, and A. Beck. High performance computing at exascale: challenges and benefits. In *15me congrs de la Socit Franaise de Physique division Plasma*, pages 1–34, (2018).
2. J. Chen, A. Choudhary, S. Feldman, B. Hendrickson, C. R. Johnson, R. Mount, V. Sarkar, V. White, and D. Williams. Synergistic challenges in data-intensive science and exascale computing. In *DOE ASCAC Data Subcommittee Technical Report. Department of Energy Office of Science*, pages 1–10, (2013).
3. M. J. Schulte, M. Ignatowski, G. H. Loh, B. M. Beckmann, W. C. Brantley, S. Gurumurthi, N. Jayasena, I. Paul, S. K. Reinhardt, and G. Rodgers. Achieving exascale capabilities through heterogeneous computing. In *IEEE Micro*, volume 35, pages 26–36, (2015).
4. R. Springmeyer, C. Still, M. Schulz, J. Ahrens, S. Hemmert, R. Minnich, P. McCormick, L. Ward, and D. Knoll. From petascale to exascale: eight focus areas of r and d challenges for hpc simulation environments. In *Technical Report*, pages 1–11, (2011).
5. S. Fiore, M. Bakhouya, and Waleed W. Smari. On the road to exascale: Advances in high performance computing and simulationsan overview and editorial. In *Future Generation Computer Systems*, volume 82, pages 450–458, (2018).
6. The top 500 ranking, June 2019. Available at <https://www.top500.org/lists/2019/06/>.
7. L. N. Huynh, Y. Lee, and R. K. Balan. Deepmon: Mobile gpu-based deep learning framework for continuous vision applications. In *Proceedings of the 15th Annual International Conference on Mobile Systems, Applications, and Services*, pages 82–95, (2017).
8. M. McNaughton, C. Urmson, J. M. Dolan, and J Lee. Motion planning for autonomous driving with a conformal spatiotemporal lattice. In *IEEE International Conference on Robotics and Automation*, pages 4889–4895, (2011).

9. B. Haghghi, N. D. Ellingwood, Y. Yin, E. A. Hoffman, and C. Lin. A gpu-based symmetric non-rigid image registration method in human lung. In *Springer Medical and biological engineering and computing*, volume 56, pages 355–371, (2018).
10. M. Cárcamo, P. E. Román, S. Casassus, V. Moral, and F. R. Rannou. Multi-gpu maximum entropy image synthesis for radio astronomy. In *Astronomy and computing*, volume 22, pages 16–27, (2018).
11. S. Aljawarneh, M. B. Yassein, and W. A. Talafha. A resource-efficient encryption algorithm for multimedia big data. In *Springer Multimedia Tools and Applications*, pages 22703–22724, (2017).
12. A. Saxena, V. Agrawal, R. Chakrabarty, S. Singh, and J. S. Banu. Accelerating image encryption with aes using gpu: A quantitative analysis. In *Springer International Conference on Intelligent Systems Design and Applications*, pages (372–380), (2018).
13. Y. Wang, T. Mazur, O. Green, Y. Hu, H. Li, V. Rodriguez, H. Wooten, D. Yang, T. Zhao, S. Mutic, and H. Li. Thabbra07: Penelopebased gpuaccelerated dose calculation system applied to mriguided radiation therapy. In *Wiley Online Medical physics*, volume 43, pages 3855–3855, (2016).
14. M. Kim, H. Shin, J. Jung, S. Kim, D. Yoon, and T. S. Suh. Development of gpu-based fast reconstruction algorithm for gamma ray imaging with insufficient conditions. In *Wiley Online Medical physics*, volume 44, pages 3147–3147, (2017).
15. M. Pi Puig, L. C. De Giusti, and M. Naiouf. Are gpus non-green computing devices? In *Journal of Computer Science and Technology*, volume 18, pages 153–159, (2018).
16. M. Pi Puig, L. C. De Giusti, M. Naiouf, and A. E. De Giusti. Gpu performance and power consumption analysis: A dct based denoising application. In *XXIII Congreso Argentino de Ciencias de la Computación*, pages 185–195, (2017).
17. The green 500 ranking, June 2019. Available at <https://www.top500.org/green500/lists/2019/06/>.
18. C. Isci and M. Martonosi. Runtime power monitoring in high-end processors: methodology and empirical data. In *IEEE Proceedings of the 36th annual IEEE/ACM International Symposium on Microarchitecture*, pages 93–105, (2003).
19. M. Y. Lim, A. Porterfield, and R. Fowler. Softpower: Fine-grain power estimations using performance counters. In *Proceedings of the 19th ACM International Symposium on High Performance Distributed Computing*, pages 308–311, (2010).
20. F. Bellosa. The benefits of eventdriven energy accounting in power-sensitive systems. In *Proceedings of the 9th workshop on ACM SIGOPS European workshop: beyond the PC: new challenges for the operating system*, pages 37–42, (2000).
21. S. Song, C. Su, B. Rountree, and K. W. Cameron. A simplified and accurate model of power-performance efficiency on emergent gpu architectures. In *IEEE 27th International Symposium on Parallel and Distributed Processing*, pages 673–686, (2013).
22. Cupti api reference guide, May 2019. Available at <https://docs.nvidia.com/cuda/cupti/index.html>.
23. Cuda library, March 2019. Available at <https://developer.nvidia.com/cuda-zone>.
24. Papi cuda component, April 2019. Available at <https://developer.nvidia.com/papi-cuda-component>.
25. Profiler’s user guide, April 2019. Available at <https://docs.nvidia.com/cuda/profiler-users-guide/index.html>.
26. Nvml library, June 2019. Available at <https://developer.nvidia.com/nvidia-management-library-nvml>.

27. K. Kasichayanula, D. Terpstra, P. Luszczek, S. Tomov, S. Moore, and G. D. Peterson. Power aware computing on gpus. In *IEEE Symposium on Application Accelerators in High Performance Computing*, pages 64–73, (2012).
28. V. Weaver, M. Johnson, K. Kasichayanula, J. Ralph, P. Luszczek, D. Terpstra, and S. Moore. Measuring energy and power with papi. In *IEEE International Conference on Parallel Processing Workshops*, pages 262–268, (2012).
29. S. Che, M. Boyer, J. Meng, D. Tarjan, J. Sheaffer, S. Lee, and K. Skadron. Rodinia: A benchmark suite for heterogeneous computing. In *IEEE International Symposium on Workload Characterization*, pages 44–54, (2009).
30. Polybench benchmark, May 2019. Available at <http://web.cse.ohio-state.edu/~pouchet.2/software/polybench/>.
31. Cuda sdk samples, June 2019. Available at <https://docs.nvidia.com/cuda/cuda-samples/index.html>.
32. K. Asanovi, R. Bodik, B.n Catanzaro, J. Gebis, P. Husbands, K. Keutzer, D. Patterson, Lester P. W., J. Shalf, S. Williams, and K. Yelick. The landscape of parallel computing research: A view from berkeley. In *Technical Report, EECS Department, University of California, Berkeley*, pages 56–67, (2006).

Fire Weather Index assessment and visualization

Sigfrido Waidelich¹, Viviana Zimmerman², Karina Laneri^{3,4}, and Mónica Denham^{1,3}

¹ Laboratorio de Procesamiento de Señales Aplicado y Computación de Alto Rendimiento, Universidad Nacional de Río Negro, Sede Andina, Río Negro, Argentina.

² Centro Regional Universitario Bariloche, Universidad Nacional del Comahue, Argentina.

³ Consejo Nacional de Investigaciones Científicas y Técnicas. CONICET, Argentina.

⁴ Física Estadística e Interdisciplinaria, Centro Atómico Bariloche, Río Negro, Argentina.

`mdenham@unrn.edu.ar`

Abstract. The Fire Weather Index (FWI) is extensively used in Argentina to support operative preventive measures for forest fires management. This index describes the moisture content of three different fuel types and the effect of the wind in the behaviour of fire. During last years our team developed a wildfire simulator with several functionalities requested by firefighters. In this work we report the spatial visualization of the FWI on our simulator, in order to describe the dependence of forest fire risks on FWI and fuel type of Northwest Patagonian forests in Argentina. This additional functionality allows the user to visualize the local spatial configuration of fire risk. The potential of this tool is that it would provide the FWI to be readily used in all fire stations in Argentina. Moreover, it could be taken as input information of a fire propagation model under development. In this sense, the need of modelling the impact of weather on the fuel becomes essential for the accuracy of forest fire simulations.

Keywords: Fire Weather Index, Forest Fire Simulation, GPGPU

1 Introduction

In Argentinian Patagonia, climate has been changing during the last decades, increasing wildfire occurrences and severity. For instance, in the western Andean Patagonia, electric storms are more frequent today, usually causing wildfires if the conditions are favourable for burning. Hot and dry seasons are longer and more frequent. In addition, big wildfires increase global climate change causing a feedback between fires and climate that is important to study [9].

Every year, Argentina is the scenery of several and sometimes huge forest fires. Approximately 18 millions of hectares of native forest, shrubland, grassland and exotic species plantations were burned from 2000 to 2013 in our country [14]. Several lives were lost on January 1994 in a grassland-shrubland fire near Puerto

Madryn, Chubut, Argentina [5].

Due to this situation, in 1996 a National Plan for Fire Management (PNMF in its Spanish acronym) was implemented by the Secretary of Sustainable Environment Development [5]. The first objective within this Plan was to define indicators of fire risk that could be accurately applied to the different geographical regions in Argentina. The Fire Weather Index (FWI), originally published in 1970 by the Canadian Forest Service [1], was one of the indexes chosen to that aim, because is part of a modular system that could be gradually implemented. Also it was proved to accurately represent fire occurrences in different ecosystems all over Argentina [4, 5]. This is because FWI is related to the humidity content of three organic fuel layers where weather has similar impact. Hence, it does not depend on the species composing both the forest floor and hummus. Inputs for FWI calculation are several meteorological variables and it accounts for the effects of fuel moisture and wind on fire behavior [1]. Our team is working in collaboration with the staff of the Fire, Communications and Emergency Department (known in Spanish as ICE) of the Nahuel Huapi National Park (PNNH in its Spanish acronym), in San Carlos de Bariloche. In this department, firefighters combine FWI values with local vegetation information to obtain the local fire risk. This quantity is then categorized in five levels: low, moderate, high, very high and extreme.

Once a fire occurs, firefighters mainly make use of their experience to define the best operative strategy to stop fires. Therefore the need of a wildfire simulator tailored for local requirements becomes more evident every day. However, a proper fuel type classification (which is another crucial input for an accurate prediction system of fire behavior) is currently lacking in Argentina.

In previous works we developed a visual forest fire simulator [3] [13] that uses several layers, e. g. topography, weather and vegetation data, to simulate fire propagation. Our simulator was developed with a High Performance Computing technology and programmed in CUDA C and OpenGL. A cellular automaton computation, landscape and fire progress visualization, as well as user interaction are executed on Graphic Processing Units (GPU) to enhance efficiency.

In this work, we add the computation of the Fire Weather Index (FWI) to our simulator, allowing to display FWI values in a tabular format and fire risk as a coloured map. In this way firefighters will be able to visualize fire risk over large areas of interest and can calculate FWI every time that the availability of updated meteorological data allows it.

Standard weather station outputs are used as inputs for FWI computation: daily temperature, relative humidity, wind speed, precipitation and date. In the following sections we explain how we performed FWI assessment (section 3.1) and visualization (section 4) in detail.

2 State of the art

The FWI is one of the modules of the Canadian Forest Fire Danger Rating System (CFFDRS) developed by the Canadian Forest Services (CFS) [8, 10]. This

System is composed by four modules or subsystems: Fire Weather Index (FWI), Fire Behaviour Prediction (FBP), Fire Occurrence Prediction (FOP), and an accessory fuel moisture subsystem. The FWI relates meteorological conditions, fuel state and fire behavior. Humidification and drying process of fuels are essentially independent of location. In addition, fire behaviour always responds to the same physical factors of fuel, topography and weather [7, 5]. For these reasons, FWI can be successfully applied outside Canada.

In Argentina, the FWI has been used over different regions [5]. For scenarios presented in this work, historical FWI values were computed and its relationship with fire occurrences was studied. In reference [7] the author classified the FWI into 5 different fire risk levels (low, moderate, high, very high and extreme) based on risk classes defined for southeast British Columbia. Results for Chubut ($42^{\circ} 56' S$; $71^{\circ} 09' W$, region of Andean Patagonia) show that burned area and fire occurrences increased with fire risk levels, from low to very high levels. For very high and extreme risk levels it becomes necessary to include more accurate inputs. According to Dentoni et. al [5], in all Argentinean scenarios, the seasonal variation of FWI was found to be positively correlated with the monthly variation in burned areas for wildfires between 1999 and 2004. Once FWI was proved to be a useful indicator of fire risk on these Argentinean scenarios, an 8 stage plan was implemented for the FWI computation in our country, including personal for training, daily weather observations, local vegetation studies, etc. The Argentinean fire management system is based on the Canadian Forest Fire Danger Rating System (CFFDRS) and a very detailed report about its adaptation to Argentina can be found in [7]. This adaptation was possible despite the fact that many Argentinian natural habitats are different from the Canadian ones. It was possible to match Argentinian vegetation with Canadian fuel types in the case of grasslands, native cypress forests and plantations. However it is still necessary to define shrubland models for Argentinian scenarios. So far, we could not find evidences of development of FWI visualization tools over Argentina. There is however, some developed software for Córdoba Province, Argentina [2], that combines FWI with historical fires to predict fire risk using neural networks.

There are studies that link spatial variations in FWI and fire occurrences. For example [14] linked heat spots detected with the Moderate Resolution Imaging Spectroradiometer (MODIS) and FWI values for the north of Argentina. MODIS data is obtained from Terra and Aqua satellites (Earth Observation System EOS) using the electromagnetic mid-infrared spectrum for detecting focal points of heat. Using historical meteorological data, FWI and Fine Fuel Moisture Content (FFMC, see section 3.1) values were calculated from 2003 to 2012. A correlation analysis made for June to October (the fire season) between MODIS data and FWI showed that, from a given threshold, the occurrence of fire (estimated by the number of heat spots) increases with fire risk. However, given the high variability of collected data, other complementing factors like landscape vegetation cover, energy released, fuel type or burn proposes, should

be also taken into account.

Our simulator couples FWI and fire risk visualization in a geographical domain along with a simulation framework that allows the visualization of fire propagation in case of occurrence. Additionally, the user can navigate in simulation time to change the parameters of fire spreading.

3 Forest fire simulator and FWI implementation

Our application is based on a cellular automata to simulate forest fire propagation [6] [3]. The code was developed on GPGPU (General Purpose Graphic Processing Units) in order to achieve high performance, reducing execution time.

The fire scenario is modeled by a 2D array of cells that can be in one of the following three states: burned, burning or not burned. Moreover, each cell has specific features (topography, weather and vegetation) and the status of a given cell changes according to some probability that depends on the state and features of its 8 nearest neighbours as explained in [3].

In a sequential solution, the cellular automata goes cell by cell, executing the fire spread model across each row and column. Then, if the fire map is $N \times M$ (rows x columns), and the fire spread model has time complexity equal to $O(S)$, with S the number of arithmetic operations per site, then the simulator time complexity is $O(NMS)$. In this work S is constant in time, given that fire spreading is based on arithmetic operations over a target cell and its nearest 8 neighbors.

Our parallel cellular automata reduces the time complexity to $O(S)$, because loops that iterate over the rows and columns are replaced by a matrix of threads that executes all arithmetic operations simultaneously. However given that an infinite number of concurrent threads is not possible, a more realistic analysis is to consider $O(NMS/p)$ where p is the number of available GPU cores.

The first sequential version of this simulator (using SELES and presented through [6]) took 10 days to perform 9 hundred thousands simulations, running on an Intel(R) i7-4770K CPU. Our first version of the parallel simulator takes approximately 12 hours to perform a million of simulations using our CUDA parallel algorithm. More details of this parallel application and a scalability study is presented in [3].

Furthermore, visualization of fire progress and user interface was developed taking into account high performance requirements. Visualization of fire propagation landscape, main and secondary menus and layer information, were developed using OpenGL and executed in parallel on the GPU. In addition, user interaction with the simulator is solved with OpenGL and efficiently executed on GPU [13].

The visualization platform shows the fire spread, and the user can interact with the graphical interface in simulation time. The most important functionalities are: fire-cuts definition, new ignition points setting, zooming, rotating and shifting fire landscape. Our simulator stores the history of fire growth, then fire can go forward and backward. In particular the dual fire spread direction and

user customized firecuts are a very valuable tool for firefighters and fire control agencies.

Raster files of vegetation, aspect, slope, wind speed and direction, real fire map (if any) are some of our simulator input layers. The outputs are also raster files, all of them compatible with any GIS (QGIS, GoogleEarth, ArcGIS, etc). The raster files define the spatial resolution of the underlying cellular automaton. Inspired on GIS layer manipulation, our simulator displays a visual panel where each data layer can be managed. They can be turned on and off by the user. Additionally, layer transparencies can be changed in order to get the best fire landscape view during simulation.

The FWI computation and visualization was coupled with the original version of our simulator. As a first step, we implemented the set of equations for FWI in C and CUDA based on [11] and [12]. In a second step local firefighters helped us to define fire risks levels based on surrounding vegetation and local weather.

3.1 Fire Weather Index

The Fire Weather Index (FWI) is assessed with the following standard weather station readings as input: temperature, relative humidity and wind speed at 12h, and accumulated rain during the last 24 hours.

The index is made-up of 6 components: the first three are fuel moisture codes that follow daily changes in the moisture contents of three classes of forest fuel with different drying rates. The three last components are fire behavior indexes representing rate of spread, fuel weight consumed, and fire intensity [11].

Those three moisture contents codes are: the Fine Fuel Moisture Code (FFMC) that represents the moisture content of litter and other cured fine fuels, the Duff Moisture Code (DMC), which represents the moisture content of decomposing organic matter and the Drought Code (DC), which represents a deep layer of compact organic matter. For each of these three indexes an additional index is computed, one for wetting by rain and one for drying.

The two slow-reacting codes (DMC and DC) need date information because they depend on the variation on the day length according to the season.

The three moisture codes are combined with wind to form two intermediate indexes: Initial Spread Index (ISI) and Buildup Index (BUI). ISI is a combination of wind and the FFMC that represents rate of spread without fuel influence. BUI is a combination of DMC and DC that represents total fuel available for fire consumption.

Then, BUI and ISI are combined to compute FWI: the intensity of the spreading fire as energy output rate per unit length of fire front. In Fig. 1 we show the FWI block diagram and FWI required inputs [11] [12].

Codes and indexes of FWI are defined by mathematical functions presented in detail in the Canadian Forest Service technical reports: [11] and [12]. Furthermore, for each code (FFMC, DMC and DC), the drying phase and the rainfall phase equations are described. In particular the report [12] shows implementation procedures, inputs and outputs.

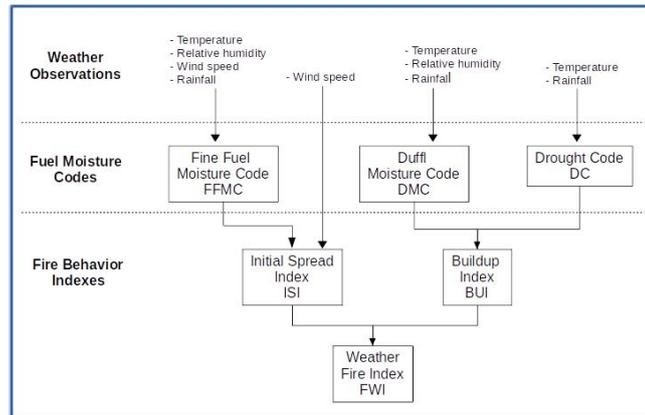


Fig. 1. Fire Weather Index block diagram.

An optional additional component of the FWI is the Daily Severity Rating (DSR). Severity rating provides a measure of wildfire control difficulty. The FWI itself is not considered suitable for averaging and should be used as its single daily value only. Any averaging (spatially over a number of station on a given day or at a single station over any period of time) is better accomplished using DSR [11].

4 FWI embedding in our Simulator

Our first FWI version was programmed in plane C language. It was based on the translation of FWI equations implemented in a spreadsheet to a more robust and solid application in C language. That initial implementation was based on [11] and [12].

It calculates the FWI for one day, or for a number of days, using an input file with weather stations data. Data input is date, temperature, relative humidity, wind and rain plus FFMC, DMC and DC of previous a day, and the outputs are: FFMC, DMC, DC, ISI, BUI and FWI.

Once this FWI implementation was sufficiently tested, the FWI computation was added to our visual parallel simulator as a new functionality. User can choose the input meteorological data file for FWI computation, including intermediate contents and indexes (FFMC, DCM, DC, ISI, BUI and DSR).

In Fig. 2 we show a simulation landscape and the corresponding table of fuel references. This scenery corresponds to a northwest Argentinian Patagonia area of 25km x 25km extension (at 41°21'59" S, 71°38'46"W). Simulator inputs are raster files with different layers of information [13].

Using FWI main menu item, the user can select an input meteorological data file (e. g., a list of: date, temperature, relative humidity, wind speed and direction and rainfall). Once the input data file is chosen, FWI equations are executed and

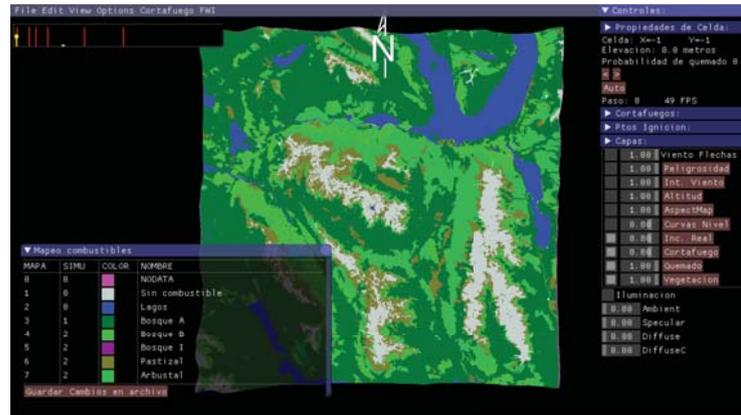


Fig. 2. Simulator main panels: landscape top view and table with fuel references. Landscape reflect effects were turned off to emphasize local vegetation categories.

subsidiary codes and indexes are computed. Results are presented in a tabular way (Fig. 3).

As shown in Fig. 3, the last four columns of the table display one of the 5 levels of fire risk for the corresponding fuel. For our area of interest 7 fuels are considered: lakes or rivers, no fuel, two type of native forest: nothofagus predominant mixed forest and cypress and lenga predominant mixed forest (columns BA and BB respectively), exotic plantations (column BI), shrubland (column Ar) and grassland (column Pa).

Then, fire danger or risk is calculated based on ISI, BUI and FWI values combined with fuel of each cell. Following previously mentioned works, fire risk was divided into 5 categories: low, moderate, high, very high and extreme (initials in figures are B, M, A, X, E respectively). These values are calculated for each day registered in the input file. Each risk level is properly coloured on the table. A window appears over the FWI table with the corresponding colours for reference (Fig. 3). This windows can be open or closed regardless of the display of the FWI table.

When the layer called "Peligrosidad" (fire risk) is turned on (using the right panel) and a specific day is selected from the FWI table, the underling map is coloured based on the fire risk of the selected day in combination with the cell fuel (Fig. 4 and Fig. 5). These two figures show the same map when different days are selected in the tabular FWI window.

To perform this test, 30x30m raster maps were used. The same resolution was used in order to generate raster maps with fire risk information. All these maps can be then managed using GIS systems. Nowadays daily FWI is calculated, but this temporal frequency can be increased in order to obtain hourly FWI results.

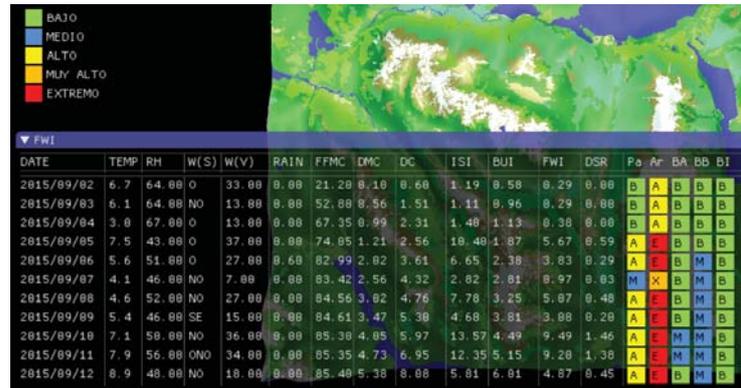


Fig. 3. FWI computation using a meteorological data as input. For each register through meteorological input file, FFMC, DMC, DC, ISI, BUI, FWI and DSR values are obtained. Then, fire risk levels are defined by the combination of local vegetation and FWI value.

5 Conclusions and Open Lines

Guided by the needs of our collaborators from ICE and the historical development of the fire management system in Argentina, we coupled to our forest fire simulator a tool for the assessment and visualization of the FWI. This computational tool was designed to be used by firefighters and is expected to be useful for fire management, training and communication. It was developed on graphic processing units to meet the requirements of a high performance real time application, with a friendly graphical interface. We are now developing a new model for fire propagation based on reaction-diffusion-convection differential equations. As opposed to previously implemented statistical models [6], this new model aims to understand the physical processes that drive fire propagation in our region. We are therefore studying the link between those physical processes and FWI, to include some of its components in our model, so at least part of the input will be measurable quantities already familiar to those involved in fire management. For example, from the implemented visualization of fire risk, one can start one or more fires in high level risk areas and see the final burned area. Stochasticity will be part of the models given that weather input is not deterministic. The study of the different possible outputs for fire propagation in those scenarios will be one of the open lines. Wind variability and its inclusion will be another very challenging open line, specially regarding the real time requirement for simulations. Finally the possibility of extending the use of the systems for other regions in Argentina will certainly represent a formidable challenge for the future.

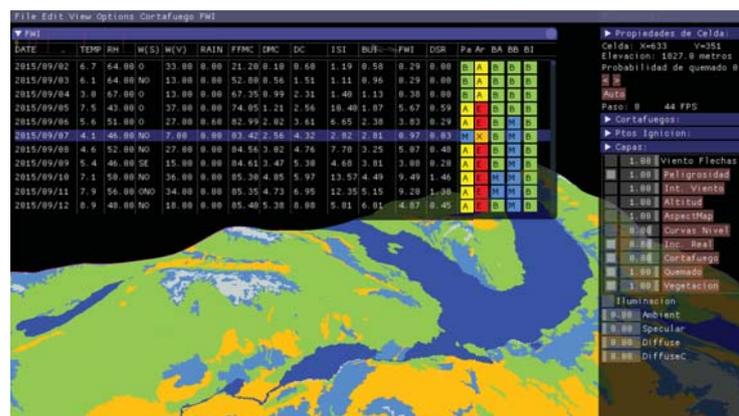


Fig. 4. Effect of selecting a specific day. Map is coloured according to fire risk level. In this figure landscape was zoomed and rotated in order to focus on Mascardi Lake area.

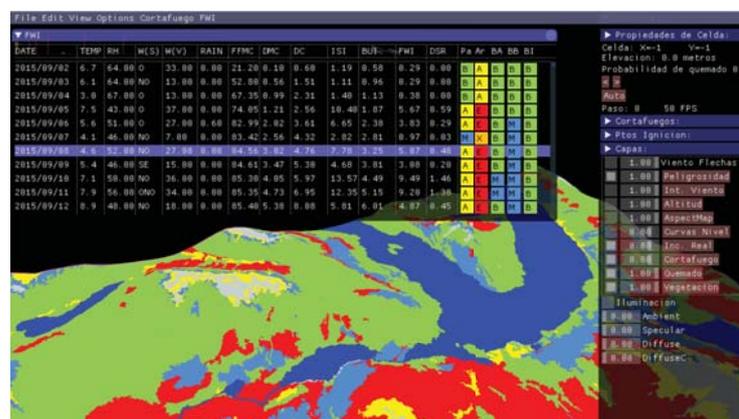


Fig. 5. Effect of selecting a specific day. Map is coloured according to fire risk level. In this figure landscape was zoomed and rotated in order to focus on Mascardi Lake area.

6 Acknowledgements

We acknowledge Marcelo Bari and collaborators from ICE (Departamento de Incendios, Comunicaciones y Emergencias del Parque Nacional Nahuel Huapi, APN: Administración de Parques Nacionales). We thank Gabriela Denham for helpful comments on maps and GIS and Emilio Bianchi for the careful reading of the manuscript. M. Denham and K. Laneri are members of Consejo Nacional de Investigaciones Científicas y Técnicas (CONICET). M. Denham, K. Laneri, V. Zimmerman and S. Waidelich are part of the project PI-UNRN "Simulador de Alto Rendimiento para la propagación de incendios forestales" (PI UNRN 40-B-745).

References

1. Canadian Forest Fire Weather Index (FWI) System, <https://cwfis.cfs.nrcan.gc.ca/background/summary/fwi>
2. Cárdenas, M., Castillo, J., Medel, R., Casco, O., Navarro, M., Gutierrez, S., Curti, A.: Sistema de predicción de incendios forestales para la provincia de Córdoba. Congreso Nacional de Ingeniería en Informática / Sistemas de información. Universidad Católica de Salta. Facultad de Ingeniería e Informática (2016), <http://bibliotecas.ucasal.edu.ar/>
3. Denham, M., Laneri, K.: Using efficient parallelization in graphic processing units to parameterize stochastic fire propagation models. *Journal of Computational Science* 25, 76 – 88 (2018)
4. Dentoni, M., Muñoz, M., Marek, D.: Fire danger rating as a tool for fire management. Implementation of the CFFDRS: The Argentine experience, pp. 101–120 (01 2015)
5. Dentoni, M.C., Muñoz, M.M., Epele, F.: Implementación de un sistema nacional de evaluación de peligro de incendios: la experiencia argentina (2007), <https://www.researchgate.net/profile/MariaDentoni/>
6. Morales, J.M., Mermoz, M., Gowda, J.H., Kitzberger, T.: A stochastic fire spread model for north patagonia based on fire occurrence maps. *Ecological Modelling* 300(0), 73 – 80 (2015)
7. Taylor, S.: Consideraciones para la aplicación del Sistema Canadiense de Evaluación de Peligro de Incendios forestales (cffdrs) en argentina. Tech. rep., Canadian Forest Service (2001)
8. Taylor, S., Pike, R.G., Alexander, M.E.: Field Guide to the Canadian Forest Behaviour Prediction (FBP) System. Canadian Forest Service (1996)
9. Tiribelli, F.: Phd Thesis: Cambios en la inflamabilidad con la edad post-fuego en bosques y matorrales del NO de la Patagonia: composición, estructura y combustibles finos. Universidad Nacional del Comahue. Centro Regional Universitario Bariloche. (2019)
10. Tymstra, C., Bryce, R., Wotton, B., Taylor, S., Armitage, O.: Development and structure of prometheus: the canadian wildland fire growth simulation model. information report nor-x-417. Tech. rep., Canadian Forest Service (2010)
11. Wagner, C.E.V.: Development and structure of the Canadian Forest Fire Weather Index System. Tech. rep., Canadian Forestry Service, Ottawa, Ontario. Forestry Technical Report 35 (1987)
12. Wagner, C.E.V., Pickett, T.L.: Equations and FORTRAN program for the Canadian Forest Fire Weather Index System. Tech. rep., Canadian Forestry Service, Petawawa National Forestry Institute, Forestry Technical Report 33 (1985)
13. Waidelich, S., Laneri, K., Denham, M.: Fire propagation visualization in real time. *Journal of Computer Science & Technology* 03, e27 (2018)
14. Zacconi, G.: Análisis preliminar de focos de calor MODIS con componentes del FWI en el norte de Argentina. Tech. rep., Servicio Nacional de Manejo del Fuego Ministerio de Ambiente y Desarrollo Sustentable de la Nación (2018)

Implementación y análisis de rendimiento de un algoritmo paralelo de multiplicación de matrices en 2,5D

Federico Sanchez¹, Ana Laura Molina¹, Nelson Rodriguez¹, María Murazzo¹

¹ Departamento de Informática, Facultad de Ciencias Exactas, Físicas y Naturales, Universidad Nacional de San Juan. San Juan, Argentina.
fgsanchez@unsj-cuim.edu.ar, amolina@unsj-cuim.edu.ar, nelson@iinfo.unsj.edu.ar,
marite@unsj-cuim.edu.ar

Resumen. En este trabajo se propone una implementación 2D-compatible del algoritmo paralelo de Cannon para multiplicación de matrices en su versión 2,5D. Dicha implementación fue realizada utilizando una distribución 2D de matrices en una grilla 2,5D de procesos. El objetivo consiste en evaluar el rendimiento de esta implementación en comparación con un algoritmo paralelo 1D previamente diseñado. Para tal fin, los desarrollos fueron ejecutados sobre un cluster homogéneo, conformado por 8 nodos, para diversos tamaños de problema. Los resultados obtenidos confirman que esta nueva alternativa 2D-compatible supera, en términos de rendimiento, a la solución 1D. La nueva implementación arroja una reducción del 6%, como mínimo, del tiempo de ejecución para todos los escenarios estudiados.

Palabras clave: programación paralela, multiplicación de matrices, algoritmo 2,5D.

1 Introducción

En el algoritmo clásico de multiplicación de matrices ideado por Strassen [1] cada una de las n^3 multiplicaciones puede realizarse de forma independiente. Es allí donde surge la sencillez de paralelizar su ejecución de forma balanceada. Debido a que el volumen de cómputo no puede ser reducido, se debe trabajar en optimizar la comunicación y el almacenamiento de datos.

Tras la investigación realizada en [2], donde se comparó el rendimiento obtenido al ejecutar diversas estrategias de multiplicación de matrices en arquitecturas distribuidas, se detectó la necesidad de paralelizar las comunicaciones, a fin de reducir el tiempo en el que las unidades de procesamiento disponibles permanecían ociosas y disminuir el costo de la paralelización.

Una alternativa estudiada, que permite paralelizar comunicaciones, es la presentada en [3]. En esta solución 2,5D, basada en el algoritmo de Cannon, los procesos no se

consideran como entidades aisladas, sino incluidos en múltiples grillas compuestas por un grupo de estos. Durante la ejecución, cada proceso multiplica un bloque 2D de matriz A con un bloque 2D de matriz B, en lugar de bloques unidimensionales (filas).

En este trabajo, se ha desarrollado una implementación 2D-compatible al algoritmo de multiplicación de matrices 2,5D, y se plantea como objetivo analizar su rendimiento en comparación con una implementación 1D desarrollada en trabajos previos. Para llevar a cabo dicho análisis, se ejecutaron diversas pruebas sobre los distintos algoritmos en diferentes escenarios definidos.

El presente artículo se organiza de la siguiente manera. En la sección 2 se describen soluciones desarrolladas previamente como estrategias para efectuar la multiplicación de matrices. En la sección 3 se presenta un resumen del algoritmo de Cannon en su versión 2,5D. En la sección 4 se incluye una descripción del desarrollo generado. En la sección 5 se enuncian las condiciones de hardware y software bajo las cuales se llevó a cabo el estudio de rendimiento. En la sección 6 se muestra gráficamente la reducción porcentual del tiempo de ejecución obtenida en los diferentes escenarios considerados, la cual da origen a las conclusiones, ofrecidas en la sección 7.

2 Trabajos Previos

De forma previa a la implementación presentada en este trabajo, se analizaron diversas estrategias de distribución de datos para efectuar multiplicaciones de matrices de gran tamaño [2].

Para la definición de dichas estrategias se propuso descomponer el cómputo considerando que, como cada elemento de C requiere la misma cantidad de cálculo, se puede balancear la carga mediante una distribución uniforme de bloques de C entre los p procesos disponibles.

En ambas soluciones, se realizó una distribución de datos unidimensional (filas) y uniforme. Cada proceso es responsable de calcular un bloque de n/p filas de C (siendo n la cantidad de filas y columnas de la matriz) y para ello, necesita obtener las correspondientes n/p filas de la matriz A y toda la matriz B.

Sin embargo, el número de filas de las matrices A y B no siempre es múltiplo de la cantidad de procesos disponibles. En estos casos, luego de hacer una distribución uniforme queda un número de filas de A sin distribuir y una porción de cálculo sin realizarse. Para resolver este inconveniente, se desarrollaron dos alternativas en las cuales se aborda el problema de las filas excedentes de distinta manera.

- Alternativa 1: las filas excedentes (en caso de existir) serán asignadas una a cada proceso hasta agotarlas, quedando así algunos procesos con una fila más para calcular que otros. En la Fig. 1. se puede observar que el proceso P0 trabaja sobre una fila más que los restantes.

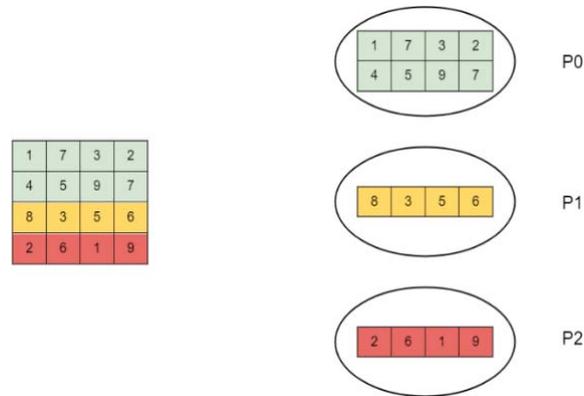


Fig.1. Distribución de filas de la matriz A entre p procesos realizada bajo la alternativa 1

- Alternativa 2: si el número de filas de la matriz A no es múltiplo de la cantidad de procesos disponibles, a dicha matriz se le agregan tantas filas, con componentes cero, como sean necesarias para que el nuevo número de filas sea múltiplo del número de procesos. Luego las filas se distribuyen equitativamente como se aprecia en la Fig. 2.

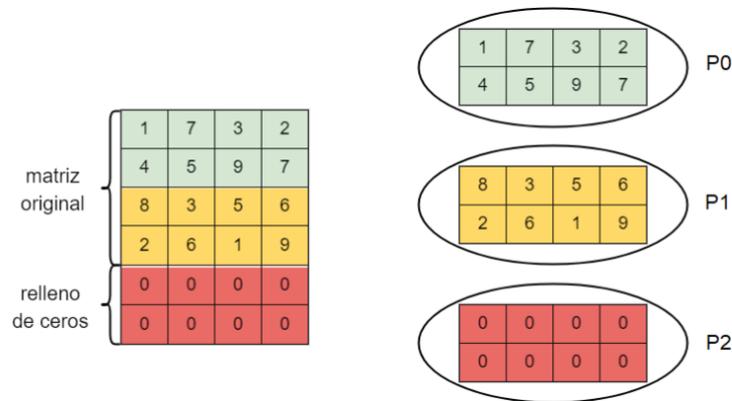


Fig.2. Distribución de filas de la matriz A entre p procesos realizada bajo la alternativa 2

La implementación de estas estrategias se basó en una modificación del modelo maestro-esclavo, donde el maestro es el encargado de distribuir los datos que necesita cada esclavo para llevar a cabo el cómputo y posteriormente ejecutar su porción del mismo. La deficiencia del modelo planteado, radica en que las comunicaciones se ejecutan de forma secuencial y previa al cómputo. En este caso, el último nodo esclavo en recibir datos permanece ocioso hasta que el maestro lleve a cabo los envíos

anteriores. Además, cada nodo debe esperar a que se envíen todas las filas de la matriz A para comenzar a recibir la matriz B. Esta misma problemática (comunicaciones secuenciales) se presenta cuando los esclavos envían los resultados obtenidos al maestro.

Tras comparar el rendimiento de ambas alternativas, se concluyó que si bien los resultados son similares, la alternativa 2 presenta un aumento mínimo en el tiempo de ejecución promedio y, en consecuencia, la alternativa 1 fue seleccionada para la comparación con la solución desarrollada en el presente artículo.

3 Algoritmo de Multiplicación de Matrices 2.5D

En esta sección se describe el algoritmo de multiplicación de matrices de Cannon [4] en su versión 2,5D presentada en [3].

La versión original del algoritmo de Cannon es considerada un algoritmo 2D ya que distribuye las matrices en una grilla de procesos de tamaño $\sqrt{p} * \sqrt{p}$ (suponiendo que la variable p representa el número total de procesos disponibles para realizar la ejecución). Posteriormente surgieron otras alternativas para calcular la multiplicación de matrices en donde se repartían los datos de entrada en una grilla de $\sqrt[3]{p} * \sqrt[3]{p} * \sqrt[3]{p}$ procesos, por lo que se denominan estrategias 3D.

En esta nueva versión del algoritmo de Cannon, la grilla está conformada por c niveles y cada uno de ellos compuesto por $\sqrt{\frac{p}{c}} * \sqrt{\frac{p}{c}}$ procesos. De acuerdo a la clasificación descrita anteriormente, es posible identificarla como un algoritmo 2,5D ya que se presenta como una solución intermedia entre las estrategias 2D y 3D.

Para llevar a cabo la multiplicación de matrices en 2,5D, inicialmente se distribuyen las matrices de entrada A y B entre los procesos ubicados en el primer nivel de la grilla. De esta manera, el proceso $P_{0,1,0}$ recibe los bloques: $A_{0,1,0}$ y $B_{0,1,0}$. Cada bloque intercambiado posee un tamaño de $\frac{n}{\sqrt{\frac{p}{c}}} * \frac{n}{\sqrt{\frac{p}{c}}}$. Luego, cada proceso se encarga de enviar los datos asignados a los siguientes niveles de la grilla, obteniéndose así c copias de los datos de entrada.

Posteriormente, se producen uno o múltiples reordenamientos inter-nivel puesto que cada proceso debe poseer los datos adecuados para realizar una o múltiples multiplicaciones de bloques. Por ejemplo, de nada sirve que $P_{0,1,0}$ realice la multiplicación de los bloques $A_{0,1,0}$ y $B_{0,1,0}$ recibidos ya que dichos bloques no deben ser multiplicados bajo el algoritmo de Strassen. Un proceso ejecutará múltiples multiplicaciones de bloques en caso de que $\sqrt{\frac{p}{c^3}}$ sea mayor que 1.

El algoritmo de Cannon está planteado de modo tal que aquellos c procesos que se encuentran en niveles distintos pero en la misma ubicación (x, y) procesen un único bloque de la matriz C resultante. Tal es así que los resultados parciales obtenidos en los c procesos, se suman en el proceso ubicado en el nivel 0.

Finalmente, se obtiene la matriz C completa unificando en una misma variable los resultados de cada proceso del primer nivel.

4 Implementación del Algoritmo de Multiplicación de Matrices 2,5D

De forma similar a la implementación 2D-compatible presentada en [5] para el algoritmo SUMMA, para el diseño de la solución 2D-compatible del algoritmo de Cannon se contempló una grilla 2,5D de procesos y una distribución 2D de bloques. La misma es ilustrada en la Fig. 3. para una grilla conformada por ocho procesos ubicados en dos niveles.

Por simplicidad, la implementación generada sólo es aplicable a matrices cuadradas y a una grilla de procesos cuadrada.

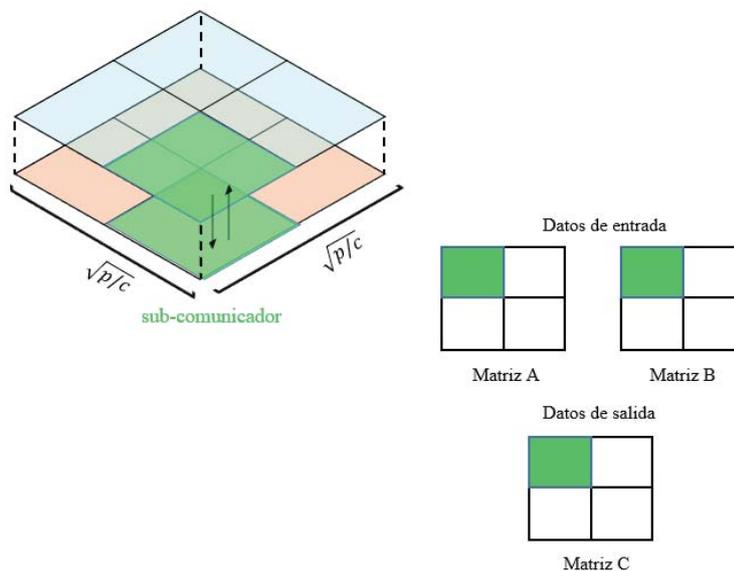


Fig. 3. Representación gráfica de la grilla 2,5D de procesos y distribución 2D de bloques de matrices

Inicialmente, se crean tantos procesos como nodos estén disponibles y haciendo uso de la operación `MPI_Comm_split` se crean sub-comunicadores para que cada proceso del nivel 0 pueda enviar los bloques de datos de entrada masivamente a sus procesos correspondientes en los restantes niveles. Para que cada proceso del primer nivel acceda a los datos de entrada, el proceso con `id=0`, reordena las matrices de entrada A y B y envía a través de una comunicación no bloqueante (`MPI_Isend`) los bloques correspondientes a los procesos de su mismo nivel. El reordenamiento de las matrices se vuelve obligatorio en esta implementación 2D-compatible, ya que el envío de datos vía MPI se puede producir únicamente sobre posiciones de memoria contiguas.

Una vez que cada proceso del primer nivel recibe los bloques de matriz, los envía mediante el sub-comunicador creado y haciendo uso de la operación colectiva

MPI_Bcast. Al finalizar estas tareas, se cuenta con 2 copias de las matrices A y B, una por cada nivel, en el caso particular analizado en la Fig. 3.

Posteriormente, se producen múltiples distribuciones de bloques inter-nivel mediante MPI_Isend y MPI_Irecv. Cada vez que un proceso determina que cuenta con los datos suficientes para la multiplicación, ya sea porque los recibió o porque contaba con ellos inicialmente, ejecuta el cómputo correspondiente y queda disponible para un nuevo intercambio de datos, en caso de corresponder. Por cada multiplicación de bloques ejecutada por un proceso, debe acumular los resultados obtenidos.

En este momento, aún resta sumar los resultados parciales de los procesos pertenecientes a un mismo sub-comunicador para obtener los valores definitivos de un bloque de la matriz C. Para esto, los procesos de cada nivel envían los bloques resultantes a los procesos del primer nivel, para que estos últimos ejecuten la suma de bloques y así se obtenga el bloque de C definitivo.

Finalmente cada proceso del nivel 0 debe enviar el bloque de matriz C obtenido al proceso 0, para que el mismo unifique todos los bloques en uno sólo. En lugar de que el proceso maestro reciba todos los bloques y luego reordene la matriz C, como se implementó en [5], se prefirió que cada proceso envíe fila por fila su bloque y cada fila sea ubicada en la matriz C resultante de forma ordenada, logrando así una mejor distribución del trabajo entre procesos a costa de un aumento en la comunicación.

Bajo esta implementación, la cantidad de memoria necesaria por proceso, a excepción del proceso 0 que requiere de más memoria para obtener el resultado final, es aproximadamente $3 * n^2 * \frac{c}{p}$. Si comparamos este valor, con la cantidad aproximada

de memoria requerida para la implementación 1D: $2 * \frac{n^2}{p} + n^2$, notamos que se ha obtenido una reducción significativa. Sin embargo, se produce un aumento en la cantidad de comunicaciones realizadas, ya que como inicialmente los procesos no reciben todos los datos necesarios, se deben realizar intercambios (inter-nivel) para obtenerlos. No obstante, ciertas comunicaciones se realizan en paralelo: comunicaciones por sub-comunicador y comunicaciones inter-nivel posteriores a la distribución original. Las consecuencias generadas por estas variaciones es lo que se pretende conocer con las múltiples ejecuciones de los algoritmos en diversos escenarios.

5 Diseño Experimental

Para evaluar el rendimiento de las implementaciones generadas, se utilizó un cluster homogéneo compuesto por 8 nodos. Dicho cluster pertenece al modelo de memoria distribuida debido a que los nodos (o computadoras) que lo forman están conectados a través de una red de área local con una velocidad de 100 Mb/seg, y cada uno de ellos cuenta con un espacio de memoria RAM de 8 GB. Además, cada nodo cuenta con un procesador multinúcleo Intel i5 3.2Ghz. Debido a la disponibilidad de computadoras existentes en el gabinete de computación, el máximo de nodos a añadir al cluster es 24. Sin embargo la implementación del algoritmo 2D-compatible requiere de una cantidad de nodos tal que $\sqrt{\frac{p}{c^3}}$ y $\sqrt{\frac{p}{c}}$ sean números enteros. Al seleccionar un valor $c=2$, el

mínimo valor de p posible es 8, mientras que el siguiente valor posible es 32. Aumentar el valor de c en una unidad implica la necesidad de aumentar el valor de p a 108. Al contar únicamente con 24 nodos, la única configuración posible es $c=2$ y $p=8$.

En la capa de software el cluster cuenta con un sistema operativo Ubuntu 14.04 LTS y NFS (Network File System) como sistema de archivos distribuido. En la capa de desarrollo, el lenguaje con el que se cuenta es C sumado a la librería MPICH2, necesaria para trabajar con MPI.

El rendimiento de las implementaciones se analizó para una cantidad fija de nodos (disponibles en el cluster) y con las siguientes variaciones en el tamaño de la matriz: 5000, 6000, 7000, 8000, 9000, 10000 filas (y columnas).

La medición del tiempo de ejecución de cada escenario se inició una vez finalizada la inicialización aleatoria de las matrices A y B, es decir previo a la distribución inicial de datos, y se finalizó una vez obtenida la matriz C por el proceso 0. Además de medir el tiempo total de ejecución, se analizó su descomposición en tiempo asociado a la comunicación y tiempo asociado al cómputo. Las mediciones fueron llevadas a cabo, en todos los casos, por el proceso 0.

Para contemplar las variaciones de rendimiento a lo largo de las múltiples ejecuciones, se midió el tiempo de ejecución de cada escenario cinco veces. El valor considerado como resultado en la siguiente sección se corresponde con el promedio de los cinco tiempos obtenidos.

6 Resultados

En esta sección se presentan los resultados obtenidos tras la ejecución de ambas implementaciones en el cluster para los seis escenarios planteados.

En la Tabla 1, se muestra el tiempo de ejecución promedio en segundos obtenido para cada escenario.

Tabla 1.: Comparación de tiempos de ejecución promedio en múltiples escenarios

	5000	6000	7000	8000	9000	10000
1D	62,667	102,333	153,000	237,000	307,000	405,333
2,5D	57,33	83,67	136,00	213,67	286,67	379,67

En la Fig. 4, se grafica la evolución de la reducción porcentual del tiempo de ejecución que otorga la nueva implementación para cada escenario evaluado. Se puede apreciar que en todos los casos analizados la reducción supera el 6%, alcanzando su máximo valor para un tamaño de matriz de 6000 filas y columnas. Es posible notar que mientras aumenta el tamaño de la matriz, la reducción disminuye. Para poder analizar este último fenómeno en detalle, resultó de interés conocer la descomposición de la reducción en: reducción asociada a la comunicación y reducción asociada al cómputo.

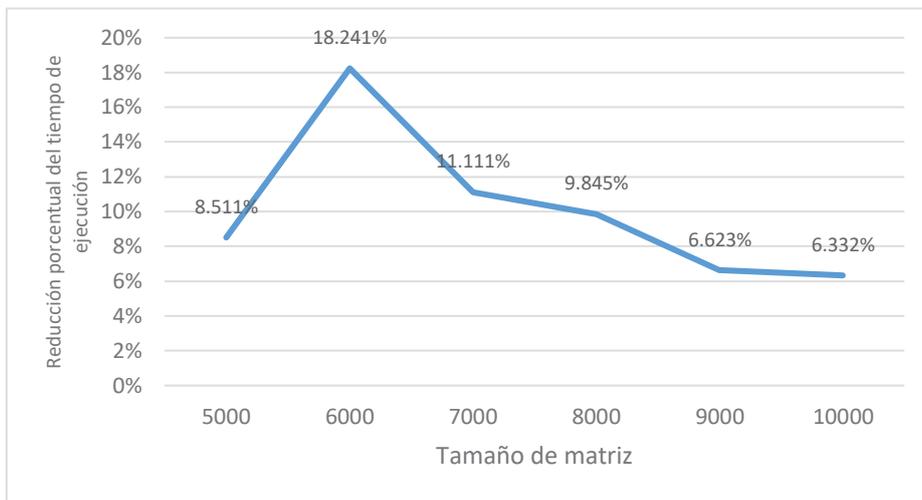


Fig. 4. Evolución de la reducción porcentual del tiempo de ejecución

La Fig. 5 permite afirmar que la reducción total conseguida es en gran parte consecuencia de la reducción asociada al cómputo.

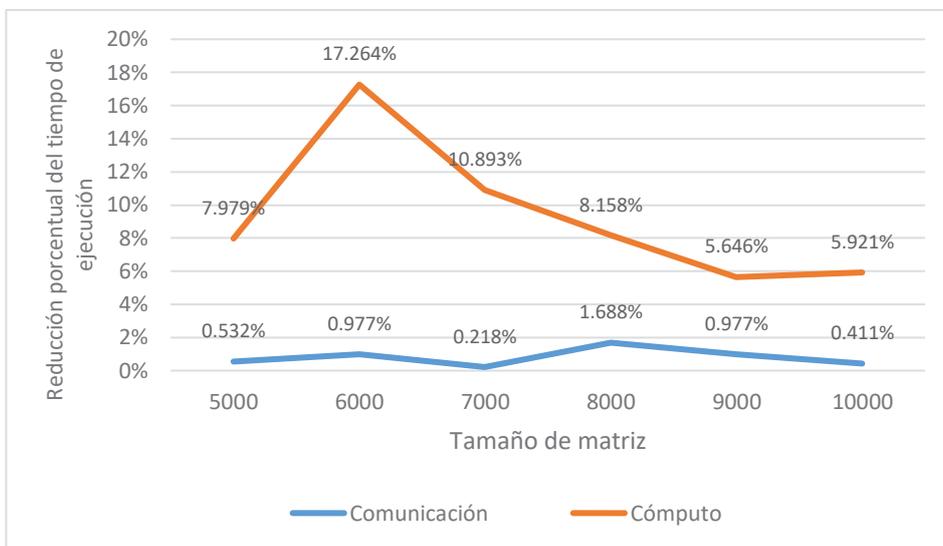


Fig. 5. Comparación de la reducción porcentual asociada a la comunicación y reducción porcentual asociada al cómputo

7 Conclusiones y Trabajos Futuros

Tras analizar la evolución y composición de la reducción total obtenida con la nueva implementación para los diversos escenarios, es posible obtener las siguientes conclusiones.

En primer lugar, la escasa reducción asociada a la comunicación permite determinar que la paralelización en las comunicaciones entre niveles y la reducción de la cantidad de datos intercambiados por cada comunicación se compensa con el aumento en la cantidad de comunicaciones, resultando así una reducción mínima (o incluso un aumento) de comunicación.

En segundo lugar, la reducción asociada al cómputo obtenida en cada caso, no se relaciona con la forma de realizar la multiplicación de bloques, que es idéntica en ambos casos, sino a la rapidez en el acceso a los datos al haberse conseguido una reducción en la cantidad de datos almacenados por proceso. Como a medida que aumenta el tamaño del problema se supera la capacidad de memoria caché de cada nodo, el rendimiento del algoritmo disminuye y se asemeja cada vez más al algoritmo 1D.

En tercer lugar, podemos atribuirle a la distribución 2D de bloques la capacidad de reducir la cantidad de datos almacenados y, en consecuencia, disminuir el tiempo asociado al cómputo.

Estudios futuros se van a enfocar en realizar pruebas en arquitecturas distribuidas que posean una mayor cantidad de nodos disponibles para la ejecución, permitiendo así probar el rendimiento del nuevo algoritmo para una grilla configurada con cuatro niveles. Bajo esta configuración, se evaluará el impacto de reducir la cantidad de datos almacenados por nodo en el rendimiento. Además, para maximizar el rendimiento mediante un enfoque híbrido, se implementará la multiplicación de bloques de matrices a través de OpenMP, permitiendo así una mejor utilización de los recursos de cada nodo.

Referencias

1. Strassen, V.: Gaussian elimination is not optimal. *Numer. Math.* 13, 354–356 (1969)
2. Sanchez, F.: *Paralelismo Híbrido para resolver problemas de cómputo intensivo*, (2018)
3. Solomonik, E., Demmel, J.: Communication-optimal parallel 2.5 D matrix multiplication and LU factorization algorithms. In: *European Conference on Parallel Processing*. pp. 90–109 (2011)
4. Cannon, L.E.: A cellular computer to implement the Kalman filter algorithm, (1969)
5. Mukunoki, D., Imamura, T.: Implementation and Performance Analysis of 2.5 D-PDGEMM on the K Computer. In: *International Conference on Parallel Processing and Applied Mathematics*. pp. 348–358 (2017)

Modelado de potencia en placas SBC: integración de diferentes generaciones Raspberry Pi

Juan Manuel Paniego¹, Leandro Libutti¹, Martin Pi Puig¹, Franco Chichizola¹,
Laura De Giusti^{1,2}, Marcelo Naiouf¹ y Armando De Giusti^{1,3}

¹ Instituto de Investigación en Informática LIDI (CEA-CIC), Universidad Nacional de La Plata,
1900 La Plata, Argentina

² Comisión de Investigaciones Científicas de la Provincia de Buenos Aires (CIC), La Plata,
Argentina.

³ Consejo Nacional de Investigaciones Científicas y Técnicas (CONICET),
CABA, Argentina.

{jmpaniego, llibutti, mpipuig, francoch, ldgiusti, mnaiouf,
degiusti}@lidi.info.unlp.edu.ar

Resumen. Monitorear la potencia de los procesadores es una tarea importante para definir estrategias que permitan disminuir los gastos de energía en los sistemas informáticos. Hoy en día, los procesadores disponen de un elevado número de contadores que permiten monitorear eventos del sistema tales como uso de CPU, memoria, cache, entre otros. Anteriormente se ha demostrado que es posible predecir el consumo de aplicaciones paralelas a través de estos eventos, pero únicamente para una determinada arquitectura de placas SBC. El presente trabajo analiza la portabilidad de un modelo estadístico de predicción de potencia sobre una nueva generación de placas Raspberry. La experimentación destaca las optimizaciones llevadas a cabo con el objetivo de reducir sistemáticamente el error final de estimación en las arquitecturas analizadas. El modelo final arroja un error promedio de 4.76% sobre ambas placas.

Palabras Claves: Potencia, Raspberry Pi, Contadores de Hardware, Modelado, Regresión estadística.

1 Introducción

Uno de los desafíos principales en el diseño de sistemas es la necesidad de una predicción rápida y precisa del consumo energético. En los últimos años, diversas innovaciones ayudaron a cumplir con este requisito.

La idea subyacente es que el consumo de energía no solo depende de las características del hardware, sino también del uso del software y las características internas del mismo. Por ejemplo, un software más complejo requerirá más ciclos de CPU, o una sola operación enorme de escritura en el disco puede consumir menos energía en lugar de varias operaciones de escritura pequeñas. Por lo general, si una persona es consciente de cuánto está consumiendo, puede encontrar su propia solución adecuada para ahorrar energía en el uso de su dispositivo [1][2].

Si bien es posible realizar mediciones por hardware al sistema para implementar un control de potencia de grano fino, tal instrumentación es costosa y no se encuentra disponible frecuentemente.

No obstante, obteniendo información de la ejecución de la aplicación, se puede estimar la energía consumida. Estos datos se recopilan a través de los contadores de hardware que se pueden utilizar para monitorear una amplia variedad de eventos relacionados con la performance del sistema con una alta precisión. Dado que cada evento está asociado con un determinado gasto de energía, cualquiera de ellos puede ser utilizado como parámetro en un modelo de rendimiento. Mientras que algunos eventos representan actividades con poco impacto en la energía, otros exhiben una alta correlación. Aunque el número de contadores suele ser limitado, los mismos permiten contabilizar una amplia diversidad de eventos.

Una forma de estimar el consumo de energía es generar un modelo estadístico basado en estos contadores. Con el objetivo de obtener una predicción en tiempo real se debe seleccionar un conjunto restringido de eventos que describa la mayor parte de la variación en potencia.

En el trabajo [13] se desarrolló un modelo de estimación de potencia utilizando contadores de hardware, el cual no contemplaba la variación en el número de hilos de la aplicación paralela. Asimismo, se utilizó el modelo estadístico de regresión lineal, el cual es intrínsecamente simple de aplicar, pero puede generar un mayor error en las estimaciones. A su vez, este modelo fue implementado utilizando como base la placa de desarrollo Raspberry Pi 3 modelo B (RPI3B).

En el presente trabajo se analiza el error obtenido usando el modelo desarrollado anteriormente, sobre la placa sucesora Raspberry Pi 3 modelo B+ (RPI3B+). Luego, con el objetivo de mejorar las predicciones, se optimiza el modelo de potencia desarrollado para aplicaciones multihilo con compatibilidad en ambas placas.

El artículo se organiza de la siguiente manera: en la Sección 2 se recopilan los trabajos relacionados en el ámbito de la predicción del consumo energético en diferentes arquitecturas CPU y GPU. En la Sección 3 se presenta la obtención de un modelo estadístico que sea compatible con las placas RPI3B y RPI3B+. En la Sección 4 se muestra la validación del modelo utilizando diferentes métodos estadísticos presentes en la herramienta de desarrollo. Finalmente, en la Sección 5 se comentan las conclusiones y trabajos futuros.

2 Trabajos Relacionados

En esta sección se presentan algunas de las investigaciones previas relacionadas con este trabajo. Lee et al. [3] propuso el modelado de regresión como un enfoque eficiente para predecir con precisión el rendimiento y la potencia para diversas aplicaciones que presentan cualquier configuración de microprocesador considerando varios diseños de microarquitectura, abordando como desafío fundamental el costo de simulación para obtener valores correctos en la predicción. Con la aparición de los contadores de hardware, Weaver et al [4] analizó los valores obtenidos de los mismos para comprobar si presentan una alta relación con lo que sucedía dentro de la arquitectura del procesador. A partir de sus resultados pudo observar que es razonable esperar que los contadores reflejen el comportamiento del procesador. Esto permitió que muchos investigadores utilicen herramientas para extraer los valores de rendimiento.

Singh et al [5] desarrolló un modelo de medición en tiempo real del consumo de energía del procesador recopilando la información provista por los contadores.

Por otra parte, Bircher et al. [6][7][8] exploró el uso de los contadores de rendimiento para predecir el consumo energético de diversos subsistemas como la CPU, memoria, chipset, E/S, disco y GPU. El mismo, fue desarrollado y validado sobre dos plataformas distintas. Asimismo, Rodrigues et al. [9] estudió el uso de los contadores de rendimiento aplicados en la estimación del consumo energético en tiempo real sobre dos arquitecturas diferentes: una orientada a alto rendimiento y la otra basada en bajo consumo.

Por otro lado, Lively et al. [10] desarrolló un conjunto de modelos híbridos de estimación de performance y consumo centrados en las aplicaciones. El mismo analiza un conjunto de códigos científicos en su implementación MPI/OpenMP y genera un procedimiento adecuado para llevar a cabo el modelado y la validación.

Las arquitecturas asimétricas de núcleos han surgido recientemente como una alternativa prometedora en un entorno con limitaciones de potencia y térmicas. Por lo general, integran núcleos con diferentes características de potencia y rendimiento, lo que hace que la asignación de cargas de trabajo a los núcleos apropiados sea una tarea desafiante. Pricopi et al [11] presentó un modelo para multi-cores asimétricos en el cual se puede obtener el rendimiento y consumo de energía de las cargas de trabajo asignadas a cada núcleo utilizando los contadores de hardware.

En este último tiempo, con la utilización de las FPGA, O'Neal et al [12] desarrolló modelos predictivos de rendimiento y consumo energético para CPU, GPU y FPGA permitiendo ahorrar costos de simulación.

Con la aparición de las placas de bajo consumo Single Board Computers (SBCs), en [13] se diseñó un modelo estadístico para predecir el consumo energético de aplicaciones ejecutadas sobre la placa RPI3B. La limitación que presenta este modelo es que solo permite predecir el consumo para la ejecución secuencial y con cuatro núcleos, sumado a la desventaja de que se utilizan coeficientes de predicción distintos de acuerdo al número de hilos. El actual artículo es una evolución de este trabajo previo, y se centra en desarrollar un modelo que permita evaluar las aplicaciones paralelas teniendo en cuenta la variación en cuanto al número de núcleos utilizados. Asimismo, el trabajo resalta la necesidad de construir un modelo multi-arquitectura que considere los cambios tecnológicos de nuevas generaciones de placas SBC.

3 Modelado

Debido a que el trabajo se basa en modelos estadísticos, se necesita extraer información de aplicaciones con diferentes comportamientos computacionales, con la finalidad de obtener datos relacionados con el rendimiento y consumo energético para las arquitecturas analizadas.

En particular, se utiliza la metodología empleada en el trabajo previo [13]: utilización de los benchmarks NAS [22] y RODINIA [23], los cuales presentan diferentes comportamientos computacionales; instrumentación y compilación del código fuente de las aplicaciones paralelas; recolección de contadores de performance y muestreo de potencia instantánea; correlación contadores-potencia y correlación mutua entre contadores; entrenamiento del modelo de regresión lineal y validación del mismo a través de la técnica de dejar uno afuera.

Teniendo en cuenta el modelo obtenido en [13], se aplican las modificaciones necesarias de acuerdo a la inclusión de la nueva generación Raspberry, generando un nuevo modelo estadístico que permita disminuir el error final de estimación.

Debido a que RPI3B y RPI3B+ son placas SBC compatibles, se reutiliza el código fuente instrumentado para las distintas aplicaciones.

El modelo se encuentra basado en regresión lineal, motor estadístico que presenta una variable dependiente, una constante de offset y variables predictoras o independientes. De acuerdo a la arquitectura utilizada, las variables independientes utilizadas son cinco, las cuales corresponden a los contadores de rendimiento *L2_DCM*, *L2_DCA*, *SR_INS*, *BR_INS* y *TOT_INS*. Además, se incluye el contador *TOT_CYC* con el objetivo de normalizar los eventos anteriores, obteniendo cinco ratios de performance.

En primer lugar, se estudian las predicciones obtenidas utilizando el modelo anterior sobre la nueva generación de placas Raspberry. Luego se realizan las optimizaciones necesarias para reducir el error y obtener de esta manera una estimación precisa.

3.1 Predicción con el modelo anterior

El modelo generado presenta una constante que se suma a los valores normalizados de los cinco contadores, los cuales tienen un peso asociado. En la Ecuación 1 puede observarse cómo obtener la potencia estimada de las aplicaciones paralelas.

$$Y_i = 1.595 + 14.696 L2_DCM + 2.308 L2_DCA + 0.108 SR_INS + 0.093 BR_INS + 0.058 TOT_INS \quad (1)$$

Para realizar la estimación de potencia sobre la placa RPI3B+, en primer lugar, se efectúa la compilación de todas las aplicaciones instrumentadas. Luego se procede a recolectar la potencia consumida y los contadores de rendimiento en cada aplicación ejecutando con 1, 2 y 4 hilos.

Una vez generada la información, se utilizan los valores obtenidos en cada aplicación paralela de los de los seis contadores usados en el modelo para estimar la potencia en la nueva placa.

La predicción arroja un error porcentual medio de 30% en la RPI3B+, a diferencia del 6.8% obtenido en la RPI3B [13]. Este incremento del error en las nuevas placas se explica por la diferencia arquitectónica entre generaciones, ya que la nueva versión presenta una frecuencia de reloj más elevada.

3.2 Generación del nuevo modelo

En la Figura 1 se puede observar la potencia real consumida por las diferentes aplicaciones paralelas en cada placa. El aumento de potencia a medida que se utilizan más hilos de ejecución es similar en ambas arquitecturas. Las 23 aplicaciones ejecutadas secuencialmente (1 hilo), ubicadas en el primer tercio de la figura, representan el menor consumo de potencia en ambas placas. Por otra parte, la ejecución paralela utilizando 2 y 4 hilos (segundo y tercer tercio de la figura respectivamente) genera un mayor aprovechamiento de los dispositivos y, por lo tanto, un mayor consumo energético.

Para la placa RPI3B, el rango de consumo se ubica entre 2 y 3 watts, mientras que para el modelo RPI3B+, el intervalo comprende 3 y 4 watts aproximadamente. El primer paso para encontrar un modelo de predicción de consumo unificado para ambas placas es estudiar la causa de esta diferencia.

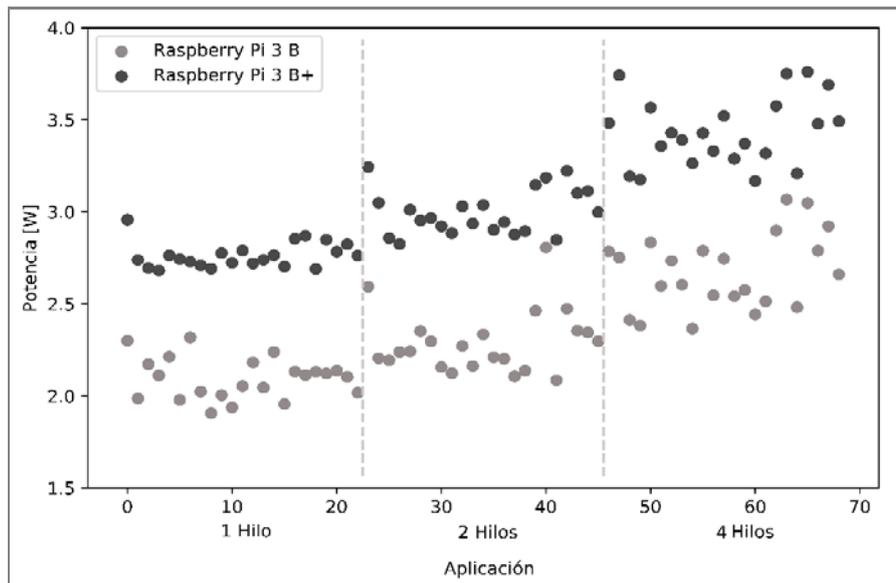


Fig. 1. Potencia real consumida de cada aplicación en cada placa considerando el número de hilos (1 - primer tercio, 2 - segundo tercio, 4 - tercer tercio)

En primer lugar, se analizan los valores que presentan los contadores de rendimiento que alimentan el modelo para cada aplicación. En este caso se elige como ejemplo la aplicación CFD, no obstante, los demás algoritmos presentan el mismo comportamiento en sus contadores. En la Figura 2 se puede observar que todos los eventos muestran la misma tendencia en ambas placas de desarrollo. Al ser placas que presentan varios aspectos arquitectónicos similares (niveles y tamaños de caché, memoria volátil, pipeline de ejecución, entre otras), la ejecución de las mismas aplicaciones arroja resultados semejantes respecto a su comportamiento. Por lo tanto, al no producirse variación en los valores de los contadores de rendimiento entre las placas, el modelo no permite distinguir la ejecución de la aplicación entre ellas.

Posteriormente, se procede a analizar los cambios generacionales entre ambas placas con el objetivo de evaluar su influencia en la predicción. La modificación significativa que se presenta es el aumento en la frecuencia del reloj de 1.2 GHz a 1.4 GHz. Esto genera que la ejecución de las aplicaciones requiera de un mayor consumo energético, lo que puede generar el aumento analizado en la Figura 1. Para comprobar esto se añade al modelo estadístico el predictor correspondiente al valor de la frecuencia máxima al momento de la ejecución. Este cambio disminuyó el error en la estimación, alcanzando el 14 % en promedio.

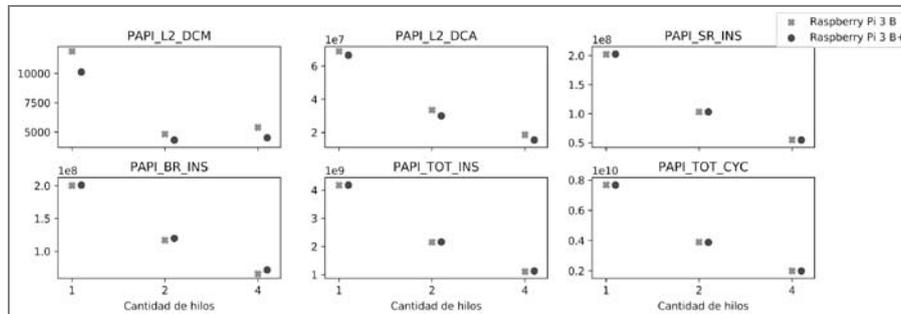


Fig. 2. Valores de contadores de rendimiento en cada placa de desarrollo considerando el número de hilos de ejecución en la aplicación CFD.

En el entrenamiento del modelo original solo se tuvo en cuenta la ejecución de las aplicaciones con 1 y 4 hilos, debido a que en [13] se buscaba evaluar la potencia para ejecuciones secuenciales y paralelas con la máxima cantidad de núcleos disponibles en el procesador. Por lo tanto, para considerar el impacto del nivel de paralelización utilizado en la aplicación, se incorpora el predictor que contempla el número de hilos (núcleos) necesarios para la ejecución ya que las aplicaciones están desarrolladas con la interfaz de programación multihilo de memoria compartida OpenMP [14].

Estas optimizaciones permiten estimar la potencia en ambas versiones de la placa de desarrollo. Por lo tanto, el modelo estadístico final basado en regresión lineal es descrito en la Ecuación 2.

$$Y_i = \beta_0 + \beta_1 L2_DCM + \beta_2 L2_DCA + \beta_3 SR_INS + \beta_4 BR_INS + \beta_5 TOT_INS + \beta_6 THREADS + \beta_7 MAX_FREQUENCY \quad (2)$$

Para la construcción del modelo estadístico, se sustituyó la herramienta de desarrollo RapidMiner por Python. Esto permite optimizar el tiempo de limpieza de datos y creación del modelo de predicción de acuerdo a la inclusión de librerías embebidas dentro de la herramienta. Asimismo, proporciona diferentes modelos estadísticos con un entrenamiento eficiente. Además, permite personalizar de manera sencilla los parámetros para el entrenamiento de cada modelo. La Tabla 1 muestra los valores de los pesos β_i una vez entrenado el modelo.

Tabla 1. Pesos obtenidos para cada predictor.

Predictor	Contador	Coefficiente	Valor
-	INTERCEPT	β_0	-2.656
X ₁	L2_DCM	β_1	18.437
X ₂	L2_DCA	β_2	4.381
X ₃	SR_INS	β_3	0.037
X ₄	BR_INS	β_4	-0.032
X ₅	TOT_INS	β_5	0.169
X ₆	THREADS	β_6	0.221
X ₇	MAX_FREQUENCY	β_7	3.546

4 Validación del modelo

4.1 Modelo de Regresión Lineal

Luego del entrenamiento del modelo, lo cual resulta en un conjunto de pesos que se aplican sobre los predictores mencionados, se evalúa el resultado del mismo ante los valores reales de potencia observados. De esta manera, en la Figura 3 puede apreciarse las predicciones que se obtuvieron para ambas generaciones Raspberry, junto a la potencia real consumida. Se observan dos tipos de muestras, las representadas en forma circular son las aplicaciones ejecutadas en la placa RPI3B mientras que las triangulares en la RPI3B+.

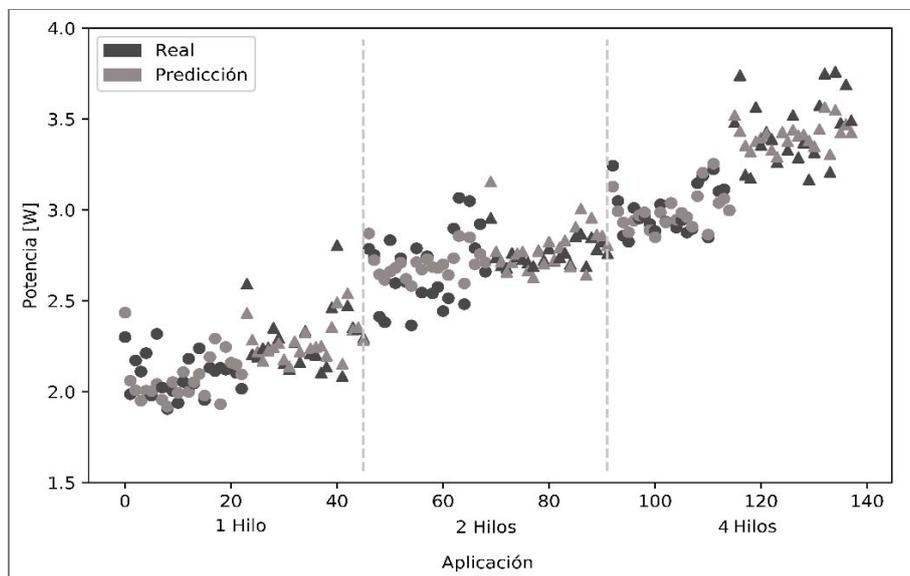


Fig. 3. Predicción y potencia consumida real de cada aplicación en cada placa (círculos - RPI3B, triángulos - RPI3B+) considerando el número de hilos (1 - primer tercio, 2 - segundo tercio, 4 - tercer tercio)

Para estimar la precisión del modelo se implementó una técnica que permite separar la información de manera tal que, para cada iteración, se destina una única muestra para los datos de prueba mientras que el conjunto restante conforma los datos de entrenamiento del modelo. La misma es conocida por el nombre de leave-one-out cross-validation o LOOCV.

En la Figura 4 puede observarse el error para cada iteración de la técnica de validación utilizada. El máximo error de predicción es de 18.46%. La dispersión del error o desviación estándar del conjunto de muestras es de 4.14%. El error promedio de las iteraciones es de 4.76%, el cual se considera aceptable y no genera grandes diferencias entre la potencia real y estimada de la ejecución de una aplicación paralela.

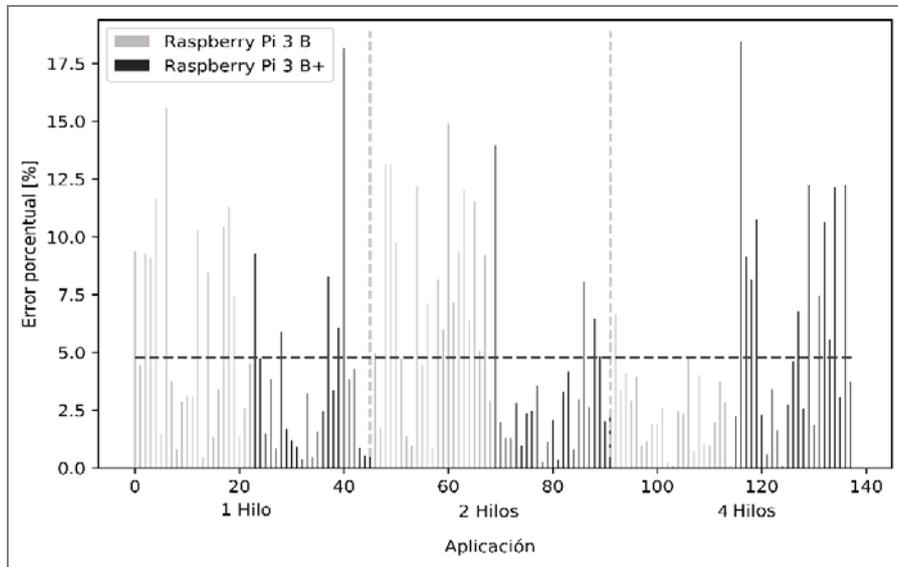


Fig. 4. Porcentaje de error de predicción en las ejecuciones de cada aplicación en cada placa considerando el número de hilos (1 - primer tercio, 2 – segundo tercio, 4 – tercer tercio)

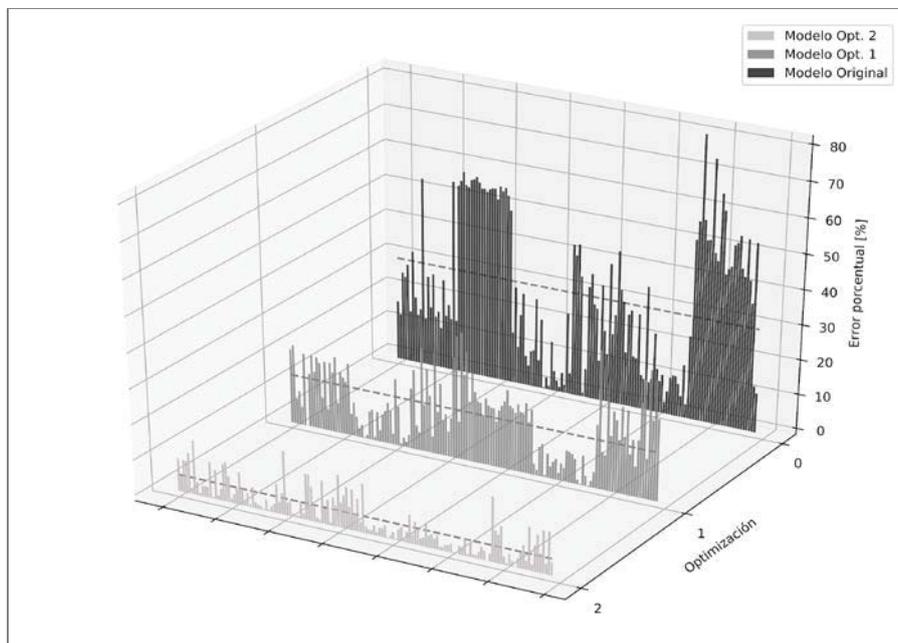


Fig. 5. Comparación de error de predicción de las aplicaciones entre las mejoras propuestas para el modelo estadístico.

Por último, puede observarse en la Figura 5 los errores de predicción para cada modelo evaluado. El eje X enumera cada aplicación analizada, el eje Y representa el modelo utilizado para la ejecución de las aplicaciones y el eje Z corresponde al porcentaje de error en la predicción. El modelo 0 corresponde al desarrollado en [13]. En el modelo 1 se agrega como predictor la frecuencia de reloj. Como última mejora, el modelo 2 adiciona la variable independiente que se corresponde con el número de hilos que utiliza la aplicación para su ejecución paralela.

4.2 Análisis de otros modelos estadísticos

Para minimizar el error de estimación producido por el método estadístico de regresión lineal se estudiaron diferentes modelos presentes en la librería Sklearn [15] de Python. Entre las técnicas analizadas se encuentran SVR (Support Vector Regression) [16], Regresión Gaussiana [17], Regresión Bayesiana [18], Regresión Kernel Ridge [19], Gradient Boosting [20] y Ridge CV [21]. En la Tabla 2 se detallan los errores de estimación obtenidos para cada modelo, en la cual se puede observar que no se lograron mejoras sustanciales en comparación con el modelo de regresión lineal.

Tabla 2. Error de predicción promedio para los modelos complementarios.

Técnica	Error Promedio
SVR	4.14
Regresión Gaussiana	4.38
Regresión Bayesiana	4.83
Regresión Kernel Ridge	4.41
Gradient Boosting	4.51
Ridge CV	4.86

5 Conclusiones y Trabajos Futuros

El presente artículo se basó en la predicción de la potencia consumida por la ejecución de diversas aplicaciones paralelas, utilizando como información de análisis los valores de los contadores de rendimiento presentes en las placas de desarrollo RPI3B y RPI3B+.

Se estudió la posibilidad de añadir a un modelo de potencia previamente construido (compatible únicamente con la placa RPI3B), nuevos predictores que permitan integrar las nuevas generaciones de la placa, así como también contemplar el nivel de paralelismo aplicado a cada algoritmo.

Se diseñó un modelo estadístico de potencia para aplicaciones multihilo basado en regresión lineal y validado a través de muestras obtenidas de las placas analizadas. Se optimizó el modelo previamente creado añadiendo el número de hilos utilizados por la aplicación y la frecuencia máxima de las placas como predictores.

La validación del modelo resultó en un error promedio en la predicción de 4.76%, permitiendo estimar el consumo de potencia de ambas placas con un alto grado de precisión.

Se analizaron diferentes técnicas estadísticas para buscar reducir el error de estimación: SVR, Kernel Ridge, Gaussian y Gradient Boosting. Las mismas disminuyen entre un 0.2% y 0.6% el error obtenido en la estimación, los cuales no producen mejoras significativas en la predicción.

Como línea de trabajo futuro se propone aplicar la metodología desarrollada sobre la nueva placa de desarrollo Raspberry Pi 4. Del mismo modo, se planea implementar una herramienta que recolecte en tiempo real la información necesaria, aplique el modelo estadístico propuesto e informe el consumo de potencia de una determinada aplicación paralela.

Referencias

1. Bekaroo, G., & Santokhee, A. Power consumption of the Raspberry Pi: A comparative analysis. In 2016 IEEE International Conference on Emerging Technologies and Innovative Business Practices for the Transformation of Societies (EmergiTech) (pp. 361-366). IEEE. (2016)
2. Procaccianti, G., Ardito, L., Vetro, A., Morisio, M., & Eissa, M. Energy efficiency in the ict-profiling power consumption in desktop computer systems. Energy Efficiency-The Innovative Ways for Smart Energy, the Future Towards Modern Utilities/InTech. Helwan: Prof. Moustafa Eissa, 353-372. (2012)
3. Lee, B. C., & Brooks, D. M. Accurate and efficient regression modeling for microarchitectural performance and power prediction. In *ACM SIGOPS Operating Systems Review* (Vol. 40, No. 5, pp. 185-194). ACM. (2006)
4. Weaver, V. M., & McKee, S. A. Can hardware performance counters be trusted?. In *2008 IEEE International Symposium on Workload Characterization* (pp. 141-150). IEEE. (2008)
5. Singh, K., Bhadauria, M., & McKee, S. A. Real time power estimation and thread scheduling via performance counters. *ACM SIGARCH Computer Architecture News*, 37(2), 46-55. (2009)
6. Bircher, W. L., & John, L. K. Complete system power estimation using processor performance events. *IEEE Transactions on Computers*, 61(4), 563-577. (2011)
7. Bircher, W. L., & John, L. K. Complete system power estimation: A trickle-down approach based on performance events. In *2007 IEEE International Symposium on Performance Analysis of Systems & Software* (pp. 158-168). IEEE. (2007)
8. Bircher, W., Law, J., Valluri, M., & John, L. K. Effective use of performance monitoring counters for run-time prediction of power. *University of Texas at Austin Technical Report TR-041104, 1*. (2004)
9. Rodrigues, R., Annamalai, A., Koren, I., & Kundu, S. A study on the use of performance counters to estimate power in microprocessors. *IEEE Transactions on Circuits and Systems II: Express Briefs*, 60(12), 882-886. (2013)
10. Lively, C., Wu, X., Taylor, V., Moore, S., Chang, H. C., Su, C. Y., & Cameron, K. Power-aware predictive models of hybrid (MPI/OpenMP) scientific applications on multicore systems. *Computer Science-Research and Development*, 27(4), 245-253. (2012)
11. Pricopi, M., Muthukaruppan, T. S., Venkataramani, V., Mitra, T., & Vishin, S. . Power-performance modeling on asymmetric multi-cores. In *Proceedings of the 2013 International Conference on Compilers, Architectures and Synthesis for Embedded Systems* (p. 15). IEEE Press. (2013)
12. O'Neal, K., & Brisk, P. Predictive Modeling for CPU, GPU, and FPGA Performance and Power Consumption: A Survey. In *2018 IEEE Computer Society Annual Symposium on VLSI (ISVLSI)* (pp. 763-768). IEEE. (2018)

13. Paniego, J. M., Libutti, L., Pi Puig, M., Chichizola, F., De Giusti, L. C., Naiouf, M., & De Giusti, A. E. (2018). Modelado estadístico de potencia usando contadores de rendimiento sobre Raspberry Pi. In XXIV Congreso Argentino de Ciencias de la Computación. (2018)
14. OpenMP, <https://www.openmp.org/>
15. Scikit-learn, <https://scikit-learn.org/stable/>
16. Regresión Gaussiana, https://scikit-learn.org/stable/modules/gaussian_process.html
17. Epsilon-Support Vector Regression, <https://scikit-learn.org/stable/modules/generated/sklearn.svm.SVR.html>
18. Ridge CV, https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.RidgeCV.html
19. Naive Bayes Classification with Sklearn, <https://blog.sicara.com/naive-bayes-classifier-sklearn-python-example-tips-42d100429e44>
20. Kernel Ridge Regression, https://scikit-learn.org/stable/modules/generated/sklearn.kernel_ridge.KernelRidge.html
21. Gradient Boosting Regressor, <https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.GradientBoostingRegressor.html>
22. D. H. Bailey, E. Barszcz, J. T. Barton, D. S. Browning, R. L. Carter, L. Dagum, R. A. Fatoohi, P. O. Frederickson, T. A. Lasinski, R. S. Schreiber, V. Venkatakrishan, S. K. Weeratunga, H. D. Simon. The NAS parallel benchmarks. The International Journal of Supercomputing Applications, (pp. 63-73), (1991).
23. S. Che, M. Boyer, J. Meng, D. Tarjan, J. W. Sheaffer, S. H. Lee, K. Skadron. Rodinia: A benchmark suite for heterogeneous computing. In: IEEE International Symposium on Workload Characterization (IISWC) (pp. 44-54), (2009).

SEDAR: Detectando y Recuperando Fallos Transitorios en Aplicaciones de HPC

Diego Montezanti^{1,4}, Enzo Rucci¹, Dolores Rexachs²,
Emilio Luque², Marcelo Naiouf¹ y Armando De Giusti^{1,3},

¹ III-LIDI, Facultad de Informática, UNLP. CEA CICPBA.

Calle 50 y 120, 1900 La Plata (Buenos Aires), Argentina

{[dmontezanti](mailto:dmontezanti@lidi.info.unlp.edu.ar), [erucci](mailto:erucci@lidi.info.unlp.edu.ar), [mnaiouf](mailto:mnaiouf@lidi.info.unlp.edu.ar), [degiusti](mailto:degiusti@lidi.info.unlp.edu.ar)}@lidi.info.unlp.edu.ar

² Departamento de Arquitectura de Computadoras y Sistemas Operativos, UAB

Campus UAB, Edifici Q, 08193 Bellaterra (Barcelona), Spain

{[dolores.rexachs](mailto:dolores.rexachs@uab.es), [emilio.luque](mailto:emilio.luque@uab.es)}@uab.es

³ Consejo Nacional de Investigaciones Científicas y Tecnológicas

⁴ Instituto de Ingeniería y Agronomía, UNAJ

Av. Calchaquí 6200, 1888 Florencio Varela (Buenos Aires), Argentina

Resumen. El manejo de fallos es una preocupación creciente en HPC; en el futuro, se esperan mayores variedades y tasas de errores, intervalos de detección más largos y fallos silenciosos. Se proyecta que, en sistemas de exa-escala, los errores ocurran varias veces al día y se propaguen para generar desde caídas de procesos hasta corrupciones de resultados debidas a fallos no detectados. En este trabajo se describe la utilización de SEDAR, una herramienta que permite detectar fallos transitorios en aplicaciones MPI, y recuperar automáticamente las ejecuciones, posibilitando su finalización con resultados fiables. La detección se basa en replicación de procesamiento y monitorización del envío de mensajes y del cómputo local, mientras que la recuperación se logra utilizando múltiples *checkpoints* de capa de sistema. El estudio del comportamiento de SEDAR en presencia de fallos, inyectados en distintos momentos durante la ejecución, permite evaluar su desempeño y caracterizar el *overhead* asociado a su utilización. Las posibilidades de configurar el modo de uso, adaptándolo a los requerimientos de cobertura y máximo *overhead* permitido de un sistema particular, hacen de SEDAR una metodología factible y viable para la tolerancia a fallos transitorios en sistemas de HPC.

Palabras clave: detección de fallos transitorios, recuperación automática, corrupción silenciosa de datos, aplicaciones de HPC, inyección de fallos.

1 Introducción

En el área del Cómputo de Altas Prestaciones (HPC - *High-Performance Computing*), el número de componentes de los sistemas paralelos continúa en aumento, en la búsqueda de mejorar la *performance*, y, como consecuencia, la confiabilidad se ha vuelto un aspecto crítico. En las plataformas actuales, las tasas de fallos son de pocas horas [1], y está previsto que en los sistemas de la exa-escala, las grandes aplicaciones paralelas tengan que lidiar con fallos que ocurran cada pocos minutos [2], por lo que requerirán ayuda externa para progresar eficientemente.

Algunos reportes recientes [3] han señalado a la Corrupción Silenciosa de Datos (SDC) como el tipo de fallo más peligroso que se puede presentar en una aplicación paralela, ya que producen que el programa aparente ejecutarse correctamente, pero finalice con resultados incorrectos. Algunas causas posibles de SDC son la radiación cósmica y la contaminación en los empaquetamientos de los circuitos, entre otros. Pueden afectar a bits de la caché, de la memoria principal o de los registros de la CPU [19]. Las aplicaciones científicas y las simulaciones a gran escala son las áreas más afectadas; en consecuencia, el tratamiento de los errores silenciosos es uno de los mayores desafíos en el camino hacia los sistemas resilientes.

En aplicaciones de paso de mensajes, y sin mecanismos de tolerancia apropiados, un fallo silencioso, que afecte a una única tarea, puede producir profundos efectos de corrupción de datos, causando un patrón de propagación en cascada, a través de los mensajes MPI, hacia todos los procesos que se comunican [4]; en el mejor escenario, los resultados finales erróneos podrán detectarse recién al finalizar la ejecución.

Una forma posible de tolerar estos fallos es diseñar hardware redundante (registros y ALU's), pero es una solución muy costosa y difícil de implementar [5]. Por lo tanto, se requieren soluciones de software específicas, con una relación costo/beneficio más adecuada, para evitar la necesidad de volver a realizar la ejecución completa en caso de detectar un fallo. Una de ellas es la ejecución redundante, en la cual un proceso es duplicado, y ambas réplicas realizan la misma secuencia de ejecución; en aplicaciones determinísticas, ambas copias producen la misma salida para la misma entrada [6]. En el contexto de HPC, la utilización de arquitecturas *multicore* con este fin representa una solución viable para detectar SDC's, debido a su redundancia natural [1].

Las técnicas de *Checkpoint & Restart* (C/R) constituyen una alternativa bien conocida para restaurar las ejecuciones de aplicaciones paralelas cuando existen fallos permanentes en los sistemas, que causan la caída de procesos o de nodos de cómputo completos (*fail-stop*) [7]. Los *checkpoints* coordinados, por ejemplo, almacenan periódicamente el estado completo de una aplicación, de forma de que todos los procesos pueden recomenzar desde el último *checkpoint* almacenado si ocurre un fallo. Lamentablemente, las técnicas de C/R introducen un alto *overhead* temporal, que crece a medida que aumenta el tamaño del sistema. Sin embargo, su mayor limitación consiste en que no hay garantía de que un *checkpoint* sea un punto seguro de recuperación, debido a que en su estado pueden haberse almacenado errores no detectados. Si el cómputo está fuertemente acoplado, esta situación empeora, debido a que un error en un nodo puede propagarse a los demás en pocos microsegundos [8].

En este escenario que conjuga resultados no fiables y altos costos de verificación, en los últimos años se ha diseñado la metodología SEDAR (*Soft Error Detection and Automatic Recovery*, anteriormente llamado SMCV [9]) para proveer tolerancia a fallos transitorios a sistemas formados por aplicaciones paralelas que utilizan paso de mensajes y se ejecutan en *clusters* de *multicores*. SEDAR es una solución basada en replicación de procesos y monitorización de los envíos de mensajes y el cómputo local, que aprovecha la redundancia de hardware de los *multicores*, en la búsqueda de brindar ayuda a programadores y usuarios de aplicaciones científicas a obtener ejecuciones confiables. SEDAR proporciona dos variantes para lograr la tolerancia a fallos: un mecanismo de detección y relanzamiento automático desde el comienzo; y un mecanismo de recuperación, también automático, basado en el almacenamiento de múltiples *checkpoints* de nivel de sistema. En este artículo se estudia el

comportamiento de SEDAR en presencia de fallos, inyectados en distintos momentos durante la ejecución, para evaluar su desempeño y caracterizar el *overhead* asociado a su utilización, demostrando así su eficacia y viabilidad para la tolerancia a fallos transitorios en sistemas de HPC.

El resto del artículo se organiza de la siguiente manera. La Sección 2 repasa algunos conceptos básicos y describe trabajos relacionados. La Sección 3 revisa conceptualmente el funcionamiento de las distintas variantes de la metodología propuesta. La Sección 4 describe SEDAR como herramienta, sus modos de uso y los fundamentos de su implementación. En la sección 5 se detallan los experimentos realizados para caracterizar el comportamiento de SEDAR, cuando es incorporado a una aplicación de prueba, en ausencia y en presencia de fallos. Por último, la Sección 6 presenta las conclusiones y las líneas de trabajo futuras.

2 Conceptos Básicos y Trabajos Relacionados

Los fallos transitorios se han clasificado (de acuerdo a sus efectos sobre la ejecución de las aplicaciones) en [9]: Errores Latentes (*LE-Latent Errors*), que afectan datos que no son utilizados posteriormente, no teniendo impacto en los resultados; Errores Detectados Irrecuperables (*DUE-Detected Unrecoverable Errors*), que causan anomalías detectables para el software del sistema, sin posibilidad de recuperación, produciendo que las aplicaciones finalicen abruptamente; Errores por *Time-Out* (*TOE*), que producen que el programa no finalice dentro de un lapso determinado; y Corrupciones Silenciosas de Datos (*SDC-Silent Data Corruptions*), que no son detectadas por el software del sistema, por lo que sus efectos se propagan para producir la finalización con salidas incorrectas. En aplicaciones paralelas que se comunican, pueden causar: Corrupción de Datos Transmitidos (*TDC-Transmitted Data Corruption*), que afecta a datos que se transmiten en algún momento (por lo que si no se detecta, se propaga hacia otros procesos); o Corrupción de Estado Final (*FSC-Final Status Corruption*), donde los datos alterados no se transmiten, pero se propagan localmente, corrompiendo los resultados finales del proceso afectado.

Para las tecnologías actuales es muy complejo lidiar con SDC frecuentes. Algunas soluciones algorítmicas existentes sólo pueden aplicarse a *kernels* específicos [10], en tanto que las propuestas de detección basadas en el compilador o en software de *runtime* son más generales, pero también significativamente más complejas. Por otra parte, las estrategias de contención buscan mitigar las consecuencias de los fallos, previniendo su propagación hacia otros nodos o hacia los datos almacenados en *checkpoints*, que son necesarios para la recuperación [7]. Los autores de [11] han mostrado que la replicación resulta más eficiente que *C/R* en contextos de altas tasas de fallo y altos *overheads* de *checkpointing*.

Tradicionalmente, la detección de SDC se ha abordado a través de replicación de procesamiento, combinada con algún grado de comparación de resultados durante la ejecución. Las soluciones basadas en redundancia de software posibilitan prescindir de hardware costoso, realizando la replicación a nivel de *threads* [12], procesos [5] o estado de la máquina. Otras soluciones, como la replicación aproximada [7] involucran menos recursos sacrificando precisión, por ejemplo, estableciendo límites superior e inferior para los resultados del cómputo. MR-MPI [13] propone replicación

parcial, que puede combinarse con C/R en los procesos que no se replican [14], para proporcionar una solución transparente para el ámbito de HPC. rMPI [11] se ocupa de fallos que causan la caída de procesos mediante ejecución redundante de aplicaciones MPI, por lo que el programa falla sólo si lo hacen dos nodos replicados; esta forma de redundancia escala, en el sentido de la probabilidad de que un nodo y su réplica fallen simultáneamente disminuye al aumentar el número de nodos; sin embargo, este beneficio tiene el costo de duplicar la cantidad de recursos y cuadruplicar el número de mensajes.

RedMPI [4] es una librería MPI que utiliza la replicación por proceso de rMPI para detectar y corregir SDC, comparando en el receptor los mensajes enviados por emisores replicados. Utiliza una optimización basada en *hashing* para evitar enviar y comparar los contenidos completos de los mensajes; no requiere cambios al código fuente y garantiza determinismo entre las réplicas. RedMPI puede ser utilizado en sistemas de gran escala, ya que provee protección aún en entornos con altas tasas de fallos. RedMPI permite que las réplicas sean mapeadas en los mismos nodos físicos que los procesos originales; en este sentido, SEDAR se comporta de manera similar. RedMPI también monitorea las comunicaciones, por lo que retrasa la detección hasta la transmisión de mensajes, pero, a diferencia de SEDAR, la validación se realiza del lado del receptor, lo que produce *overhead*, latencia y tráfico de red adicionales.

En [6] se abordan los fallos en sistemas de memoria compartida, mediante un esquema de múltiples hilos por proceso, que incluye el manejo del no-determinismo debido a los accesos a memoria. En [1] se describe un protocolo para programas híbridos que utilizan tareas y MPI, que permite la recuperación basada en *checkpointing* no coordinados y *logging* de mensajes: únicamente la tarea afectada por el error se reinicia, y todas las llamadas a MPI son manejadas dentro de ella.

También se han desarrollado protocolos tolerantes a fallos para otros modelos de programación paralela, como PGAS [15]. En [2] se explora la combinación de realizar *checkpointing* de los resultados de las tareas, y replicación para detección específica de la aplicación, en un contexto de flujos de trabajo lineales, tanto en presencia de fallos *fail-stop* como silenciosos.

3 Breve Revisión de la Metodología SEDAR

Como se mencionó en la introducción, SEDAR proporciona dos variantes para lograr la tolerancia a fallos: un mecanismo de detección y relanzamiento automático desde el comienzo, y un mecanismo de recuperación basado en el almacenamiento de múltiples *checkpoints* de nivel de sistema. A continuación se revisan conceptualmente ambas estrategias propuestas.

3.1 Detección y Relanzamiento Automático

Para detectar fallos transitorios durante la ejecución de aplicaciones paralelas determinísticas, SEDAR valida los contenidos de los mensajes que se van a enviar, acotando así la latencia de detección, aislando el fallo y evitando que se propague a los demás procesos. Para ello, duplica en un *thread* cada proceso de la aplicación. Al alcanzar el instante de envío de un mensaje, las réplicas utilizan un mecanismo de

sincronización para garantizar que se encuentran en el mismo punto, y allí se comparan los contenidos de los mensajes computados por ambos hilos; si hay coincidencia, sólo una réplica envía el mensaje (bajo la premisa de que no hay fallos en las comunicaciones), no requiriéndose ancho de banda adicional. El proceso receptor espera a que su réplica alcance el mismo punto y le copia los mensajes que ha recibido para luego reanudar la ejecución. Por otro lado, los fallos cuyos efectos no se transmiten a otros procesos, pero que se propagan localmente hasta el final son detectados en una operación de validación de resultados finales. En [9] se estudia compromiso entre la latencia de detección y la sobrecarga computacional involucrada en realizar validaciones más frecuentes.

Este mecanismo posibilita detectar fallos que causan TDC y FSC; además, bajo la premisa de que, en un sistema homogéneo dedicado, los tiempos de ejecución de dos réplicas que realizan el mismo cómputo deben ser similares, es posible detectar los TOE al notar una diferencia considerable entre los tiempos de procesamiento de ambas réplicas [16], a causa posiblemente de un fallo silencioso. La Fig. 1 esquematiza el funcionamiento del mecanismo de detección frente a la ocurrencia de un fallo.

El mecanismo de detección propuesto tiene situaciones de vulnerabilidad asociadas, que han sido descritas previamente en [16]; sin embargo, las situaciones que lo afectan son extremadamente improbables, de modo que pueden ignorarse sin riesgos considerables.

En ausencia de un mecanismo de recuperación, la ocurrencia de un fallo de los tipos mencionados causa que la aplicación detenga su ejecución al momento de la detección, evitando así la costosa e innecesaria espera a que finalice con resultados incorrectos, y permitiendo su relanzamiento inmediato y automático desde el comienzo [17].

3.2 Recuperación Basada en Múltiples *Checkpoints* de Capa de Sistema

El mecanismo de recuperación de SEDAR se basa en el almacenamiento de una cadena de múltiples *checkpoints* coordinados y distribuidos que se realizan por medio de una librería de *checkpointing* de capa de sistema. El hecho de guardar varios *checkpoints* es necesario debido a que, en el *checkpoint* de nivel de sistema, el usuario no tiene control sobre la información que se guarda. Por lo tanto, no es posible garantizar que un determinado *checkpoint* constituya un punto seguro de recuperación, debido a que los datos pueden haber sido alterados por la ocurrencia de un fallo transitorio antes de ser almacenados. En consecuencia, los *checkpoints* previos al último no pueden ser descartados, y, en el caso de que se detecte un fallo, se debe determinar la factibilidad de recuperar la ejecución desde el último *checkpoint*

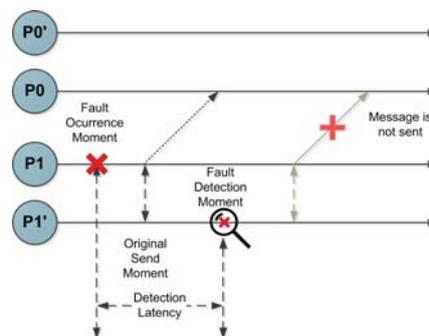


Fig. 1. Operación del mecanismo de detección frente a la ocurrencia de un fallo

(en el mejor caso) o desde alguno anterior [8]. Como casos particulares, si un fallo es detectado cuando aún no se ha almacenado ningún *checkpoint*, la aplicación debe ser relanzada desde el comienzo. En tanto, si en caso de detectar un fallo, se regresa a un *checkpoint* corrupto, en el intento de re-ejecución se vuelve a detectar el mismo fallo y es necesario retroceder hasta uno previo; esta situación puede ocurrir repetidas veces si la latencia de detección es muy alta [17]. En las Fig. 2 (a) y (b) se esquematizan estos comportamientos. El algoritmo de recuperación, en pseudocódigo, se muestra en la Fig. 3.

La automatización de este mecanismo reside en que un proceso externo a la aplicación pueda acceder al valor de `external_counter` y buscar el *script* de reinicio correspondiente (entre todos los generados por la librería de *checkpointing*) de acuerdo con el valor leído.

Este método tiene como principales ventajas la de ser transparente a la aplicación, en el sentido de que no es necesario conocer su estructura interna, ni disponer de *checkpointing* a medida de la aplicación; además, el mismo mecanismo de C/R es el que ya se usa para los fallos permanentes. Por lo tanto, la utilización de un mecanismo de detección basado en duplicación, y uno de recuperación basado en C/R posibilita detectar y corregir fallos silenciosos sin la necesidad de recurrir a la triplicación con mecanismo de votación [20]. En tanto, sus limitaciones consisten, por un lado, en la cantidad de almacenamiento requerido, al no poder eliminar los *checkpoints* previos; por otro, el gasto de tiempo involucrado en los múltiples intentos de reinicio posibles [8]; y, por último, la baja escalabilidad de los *checkpoints* coordinados al crecer la cantidad de procesos, además del hecho de que almacenan gran cantidad de información relacionada al sistema [1].

Cabe aclarar que la metodología SEDAR propone también un mecanismo de recuperación basado en *checkpoints* de capa de aplicación, que disminuye el tiempo y el trabajo asociados a la recuperación a costa de requerir validación de los *checkpoints* [17]

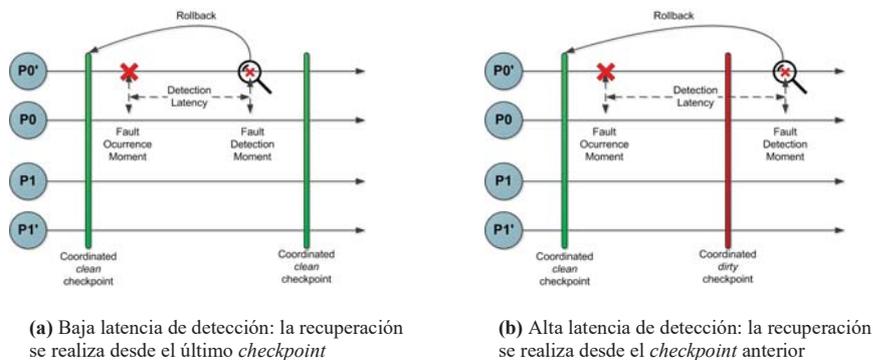


Fig. 2. Posibles casos de recuperación utilizando múltiples *checkpoints* de nivel de sistema, dependiendo de la latencia de detección

```

1 // contador externo que controla la cantidad de rollbacks
2 // (no se almacena en el checkpoint)
3 int external_counter = 0;
4 // variable Boolean que indica si se detectó una falla en la última ejecución
5 boolean fault_detected = FALSE;
6 // ejecutar aplicación monitorizada por SEDAR
7 SEDAR_exec(app)
8 // se controla si hubo falla y, en ese caso, se ejecuta el mecanismo de recuperación
9 while ( fault_detected == TRUE) do
10 // se incrementa en uno el contador externo
11 external_counter++;
12 // obtener cantidad de checkpoints realizados
13 num_ckpt = get_checkpoint_count();
14 // resetear variable boolean
15 fault_detected=FALSE;
16 // re-ejecutar aplicación desde el checkpoint num_ckpt - external_counter
17 SEDAR_restart(app,num_ckpt - external_counter);
18 end;

```

Fig. 3. Pseudocódigo del algoritmo de recuperación basado en múltiples *checkpoints* periódicos realizados desde el exterior

4 SEDAR como Herramienta

4.1 Modos de Utilización

En su estado actual, SEDAR permite 3 modos de utilización:

- (a) Sólo detección y relanzamiento automático
- (b) Recuperación basada en *checkpoints* de nivel de sistema disparados por eventos
- (c) Recuperación basada en *checkpoints* periódicos de nivel de sistema

El modo (a) permite evitar el coste asociado al almacenamiento de los *checkpoints*, posibilitando que la aplicación pueda ser relanzada desde el principio ni bien transcurre la latencia de detección. Este modo es útil cuando el *overhead* asociado a los *checkpoints* es muy alto, o cuando los sucesivos intentos fallidos de re-ejecución de manera regresiva representan un gasto de tiempo superior al de simplemente relanzar desde el comienzo. En la Sección 5.2 se verá una evaluación de estos costos para un caso de estudio.

El modo (b) es de utilidad cuando, en la búsqueda de obtener ejecuciones confiables, se tiene un conocimiento detallado del comportamiento de la aplicación a proteger, por ejemplo, en cuanto a la relación cómputo/comunicaciones. En este caso, SEDAR puede sintonizarse con la aplicación, minimizando el *overhead* introducido, ya que los *checkpoints* pueden sincronizarse con eventos significativos de la ejecución, como pueden ser determinadas fases de comunicación. Tomando en cuenta que los datos a comunicar son validados previamente, almacenar *checkpoints* en los instantes de comunicación permite minimizar los riesgos de almacenar valores corruptos y de propagarlos; esto lo hace un mecanismo muy viable para ser utilizado en aplicaciones tipo SPMD. Por otra parte, seleccionar cuántos *checkpoints* se realizan (y en qué instantes) posibilita optimizar la relación costo/beneficio del mecanismo de protección, y mantener acotado el *overhead*. Este modo fue el utilizado

principalmente en la etapa de validación funcional del mecanismo: basándose en el conocimiento de una aplicación de prueba, y mediante experimentos de inyección controlada de fallos y *checkpoints* sincronizados con las comunicaciones, pudo determinarse, en cada caso, qué dato resultaba afectado, en qué instancia se detectaba ese fallo y desde qué punto podía recuperarse la ejecución [17]. Como desventaja, este mecanismo no es transparente (ni automático), en el sentido de que el código de los *checkpoints* debe insertarse dentro de la aplicación.

Como contrapartida, el modo (c) permite proteger a la aplicación “desde el exterior”. Lanzando la ejecución bajo la órbita de un proceso coordinador de la librería de *checkpointing*, se establece un intervalo de almacenamiento periódico. En este caso, no es necesario conocer la aplicación en profundidad, a costa de no tener control sobre el instante preciso en que se realiza el *checkpoint*, ni de desde dónde podrá recuperarse la ejecución. El *overhead* tendrá un comportamiento asociado al costo de almacenar un *checkpoint* y a la cantidad de ellos que se almacenen dado el intervalo de periodicidad. Como ventaja, esta protección puede realizarse de forma completamente automática.

4.2 Implementación

El mecanismo de detección se implementa en la forma de una librería que modifica las funciones y los tipos de datos definidos en MPI (sintácticamente, sólo cambia el prefijo MPI por SEDAR), y agrega las funciones `SEDAR_Call` y `SEDAR_Validate`. Por otro lado, utiliza **Pthreads** para la replicación y sincronización de hilos. Para incorporar la detección, el código de la aplicación debe ser levemente modificado y recompilado, en un procedimiento automatizable.

La recuperación automática se implementa mediante la utilización de la librería DMTCP [21] (versión 2.4.4) que provee *checkpointing* coordinado y distribuido de nivel de sistema. El comportamiento por defecto de la librería consiste en almacenar un único *checkpoint*, sobre-escribiendo el archivo generado cada vez que se guarda un *checkpoint* nuevo; durante la instalación, puede modificarse este comportamiento mediante la configuración del *flag* `enable_unique_checkpoint = no`, de modo que se pueden almacenar varios *checkpoints* (con números correlativos) y sus correspondientes *scripts* de reinicio, sin perder ningún estado previo; en esta posibilidad está basado el mecanismo de recuperación de SEDAR. DMTCP genera un proceso coordinador, que es el encargado de monitorear la aplicación objetivo y realizar los *checkpoints* (en este caso periódicos) desde el exterior, lo cual produce que el mecanismo de recuperación sea transparente para la aplicación [17].

5 Trabajo Experimental

5.1 Diseño de la Experimentación

Se diseñaron experimentos destinados a determinar la incidencia de la incorporación de SEDAR a la ejecución de una aplicación paralela que utiliza MPI, mostrando además la eficacia de los mecanismos de detección y de recuperación automática. SEDAR se utilizó tanto en las variantes (a) y (c) descritas en la Sección 4.1. Para ello,

debió modificarse el código fuente de la aplicación para integrarla con la funcionalidad de SEDAR. La aplicación de prueba fue una multiplicación de matrices paralela, cuya implementación sigue el modelo *Master-Worker*, donde uno de los procesos es responsable de la distribución del trabajo y la recuperación de los resultados parciales. Para computar $C = A \times B$, el *Master* envía a cada proceso (inclusive a sí mismo) un bloque de filas de la matriz A y la matriz B en forma completa. Luego, cada proceso es responsable de calcular un bloque de filas de C. Por último, resulta importante mencionar que los procesos emplean operaciones de comunicación colectiva para el intercambio de datos (MPI_Scatter para distribuir A, MPI_Bcast para enviar B y MPI_Gather para recuperar los resultados parciales de C). Debe notarse que esta aplicación demanda gran cantidad de cómputo regular y que requiere poca cantidad de comunicaciones al inicio y al final de cada tarea. Para los experimentos, se utilizaron 8 procesos y matrices de 8192 x 8192 elementos.

Los experimentos consistieron en medir los tiempos de ejecución de ambas variantes de SEDAR en ausencia de fallos, para determinar el *overhead* introducido respecto de: (1) una ejecución sin ninguna protección y (2) una protegida manualmente por medio de ejecución simultánea de dos instancias independientes. Para el caso de la recuperación automática, se buscó establecer la relación entre el *overhead* y el intervalo de *checkpointing* (i), midiéndolo para tres valores distintos: $i = 60s$, $i = 90s$, $i = 120s$; para cada caso la cantidad de *checkpoints* almacenados fue $N_{ckpt} = 6$, $N_{ckpt} = 4$ y $N_{ckpt} = 3$, respectivamente. Estos intervalos fueron definidos sobre la base de que la ejecución, sin la incorporación de SEDAR, dura aproximadamente 6 minutos; tomando esto en cuenta, los intervalos seleccionados buscaron ofrecer distintas alternativas de compromiso entre la protección brindada y el *overhead* introducido, frente a distintos valores posibles de MTBF (*Mean Time Between Failures*).

Luego se realizaron experimentos en presencia de fallos, contemplando tres casos de inyección: uno al comienzo de la ejecución, uno aproximadamente a la mitad, y uno hacia el final con la variante de detección de SEDAR. En el caso de la recuperación, los experimentos para cada inyección se repitieron con cada uno de los valores de intervalo de *checkpoint*. El objetivo de estas pruebas fue determinar el comportamiento de SEDAR en presencia de fallos, y poder obtener conclusiones respecto de la conveniencia de utilizar cada variante, en función del momento de ocurrencia del fallo y el intervalo de *checkpoint* configurado. La información obtenida le permite al usuario configurar SEDAR para funcionar de manera óptima en conjunto con la aplicación monitorizada.

Los experimentos se llevaron a cabo utilizando dos nodos de un *cluster* Blade; cada nodo contiene dos procesadores Intel Xeon e5405 de 4 núcleos a 2.0 GHz, 10 GB de memoria RAM, disco local de 250 GB y una conexión de red Gigabit Ethernet. El sistema operativo es Debian 6.0.7 (64 bits, versión de kernel 2.6.32) y la versión de la librería de comunicaciones es OpenMPI 1.7.5.

5.2 Resultados Experimentales

Los resultados de todas las pruebas realizadas se resumen en la Tabla 1. En todos los casos, los tiempos obtenidos son el promedio de 5 repeticiones de cada experimento particular. El tiempo de la primera fila es el de la ejecución de la

aplicación MPI pura, con la particularidad de que el *mapping* se realizó con 4 procesos en cada nodo, aun quedando *cores* libres, de forma de forzar la utilización de la red, para que la comparación con SEDAR (que replica los procesos) fuese más justa. En la segunda fila, se muestra el tiempo de ejecutar dos instancias independientes simultáneas de la aplicación, con el mismo *mapping* del caso anterior. Esto representa un esquema de detección manual (que requiere acciones por parte del usuario), en el que todos los recursos de cómputo de ambos nodos del *cluster* se utilizan para tener ejecuciones redundantes, con posibilidad de comparar los resultados finales, posteriormente a la finalización de la ejecución (realizada como procesamiento *offline*); en caso de diferencia, es posible lanzar una tercera instancia, de forma de poder votar como resultado correcto el de dos de las ejecuciones que coincidan (al menos dos fallos en ejecuciones distintas deberían ocurrir para que no haya coincidencia). La diferencia de tiempo con la ejecución original se debe a la competencia por los recursos entre los procesos de las dos instancias. En caso de requerirse la tercera ejecución, sin modificar el *mapping*, se agregaría el tiempo de la primera fila (sin competencia por los recursos) al anterior. Sin embargo, esto sólo tiene en cuenta los tiempos de ejecución puros, sin contemplar el tiempo de comparación *offline* de los resultados, ni el de volver a lanzar la ejecución. Debe notarse que el tiempo total es independiente del instante de ocurrencia del fallo, debido a que, en cualquier caso, la ejecución debe concluir para poder verificar la validez de los resultados.

La tercera fila muestra el tiempo cuando se incorpora el mecanismo de detección de SEDAR. Se puede observar que el *overhead* introducido, debido a duplicación de procesos, la sincronización entre réplicas, la comparación y copia de los contenidos de los mensajes y la validación final de los resultados, es menor al 0.3%. En tanto, en caso de fallo, el tiempo total, que proviene de ejecutar hasta la detección, parar y relanzar desde el comienzo, es aproximadamente el doble del de la ejecución sin fallos, y representa un *overhead* aproximado del 6.5% respecto de realizar una tercera ejecución; sin embargo, SEDAR ofrece las ventajas de tener incluidos los tiempos de comparación y relanzamiento automático, que en el caso *offline* deberían ser llevados a cabo por el usuario. Además, debe notarse que este es el peor escenario posible, ya que todos los fallos inyectados son detectados casi sobre el final de la ejecución, como se explica a continuación. Por lo tanto, cuanto más tempranamente puedan ser detectados los fallos (menor latencia), mejor será el desempeño de la estrategia de detección y relanzamiento.

Tabla 1. Resultados de los experimentos

	Estrategia	Tiempo [s] según caso de inyección			
		Sin fallo	Al comienzo	Intermedio	Al final
1	Sin SEDAR – 1 instancia	329,57	-		
2	Sin SEDAR – 2 instancias simultáneas	373,16	702,73		
3	SEDAR – det	374,19	745,06	748,82	747,62
4	SEDAR – rec (i = 60s); Nckpt = 6	440,76	2066,23	1000,12	477,35
5	SEDAR – rec (i = 90s); Nckpt = 4	418,91	1569,85	822,93	459,63
6	SEDAR – rec (i = 120s); Nckpt = 3	407,06	1303,28	895,41	454,22

La situación mencionada se debe al comportamiento particular de la aplicación de prueba y su interacción con el mecanismo de detección. Como se mencionó en la sección 5.1, la multiplicación de matrices es fuertemente limitada por cómputo, y las fases de comunicación son breves y están ubicadas al comienzo y al final. Esto produce que, para todos los instantes de inyección seleccionados, la detección se produzca siempre durante la última comunicación colectiva. Como consecuencia, cuando el fallo se inyecta sobre el final (al finalizar el cómputo y antes de dicha comunicación), la latencia de detección es baja; pero cuando el fallo se inyecta antes de comenzar la fase de cómputo, o durante ella, permanece latente por mucho tiempo antes de ser detectado. En una aplicación con comunicaciones más frecuentes, la latencia de detección sería menor.

En la primera columna de las filas 4 a 6, se ven los tiempos correspondientes al agregado del mecanismo de recuperación, en ausencia de fallos. Se puede observar que el *overhead* es inversamente proporcional al intervalo de *checkpointing*. Este comportamiento es esperable, debido a que, si el intervalo es menor, aumenta el número de *checkpoints* y se consume más tiempo almacenándolos.

En tanto, el resto de las columnas muestra los tiempos de recuperación en presencia de los diversos casos de fallos. Puede notarse, en principio, que el mecanismo permite recuperar correctamente y finalizar la ejecución en todas las ocasiones, luego de distinta cantidad de reintentos según la latencia de detección. Sin embargo, sólo en el caso del fallo inyectado sobre el final, el tiempo de recuperación requerido es considerablemente menor que en la detección y relanzamiento automático desde el comienzo; pero en todos los demás, el almacenamiento de múltiples *checkpoints* para recuperar no representa una alternativa conveniente frente a simplemente detectar, parar y relanzar. Una vez más, este hecho está relacionado con la latencia de detección. Cuando la latencia es alta, los *checkpoints* que se almacenan mientras el fallo permanece latente están corruptos, y por lo tanto no son válidos para la recuperación. El algoritmo intenta recuperar sucesivamente hacia atrás desde cada uno de ellos, acumulando los tiempos involucrados en cada intento hasta encontrar un *checkpoint* válido¹. En cambio, en el caso de los fallos inyectados sobre el final, que son rápidamente detectados, la recuperación desde el último *checkpoint* es exitosa, mejorando notablemente el tiempo total. Nuevamente, en una aplicación que realizara comunicaciones más frecuentes, la latencia de detección sería menor y, por lo tanto, también la cantidad de intentos de recuperación.

Como conclusión, se puede destacar la flexibilidad de SEDAR para ajustarse a las necesidades de un sistema particular, proveyendo diferentes posibilidades de cobertura (detección y relanzamiento automático o recuperación basada en múltiples *checkpoints*), que le permiten adaptarse para obtener un compromiso en la relación costo/beneficio. El intervalo de *checkpoint* óptimo y la frecuencia de validación de las comunicaciones pueden ser determinados a partir de la caracterización de la aplicación a proteger.

¹ Por ejemplo, en la inyección intermedia (durante el cómputo) con intervalo de *checkpointing* $i = 60s$, al momento de la detección se han realizado 6 *checkpoints*; la recuperación se intenta sucesivamente, sin éxito, desde el 6, luego desde el 5 y por último desde el 4, rehaciendo cada vez todo el cómputo hasta volver a detectar (y rehaciendo también los *checkpoints* que ya se habían hecho); recién es exitosa la recuperación desde el *checkpoint* 3.

6 Conclusiones y Trabajo Futuro

Garantizar la fiabilidad de las ejecuciones es uno de los principales desafíos en el camino hacia los próximos sistemas de cómputo en la Exa-escala. La protección de las aplicaciones MPI a nivel de las comunicaciones es un método factible y efectivo para detectar y aislar la corrupción de datos, evitando su propagación hacia todos los procesos que se comunican. En este trabajo, se presenta la herramienta SEDAR, que proporciona tres modos de utilización: sólo detección y relanzamiento automático, y recuperación basada en múltiples *checkpoints* de capa de sistema, ya sea periódicos o disparados por eventos. El comportamiento temporal de SEDAR en ausencia y en presencia de fallos, inyectados en distintos momentos durante la ejecución, permite evaluar su desempeño y caracterizar el *overhead* asociado a su utilización. La posibilidad de seleccionar entre las estrategias de detección y la de recuperación, y configurar el intervalo de *checkpointing*, proporciona flexibilidad para adaptar SEDAR a los requerimientos de cobertura y máximo *overhead* permitido de un sistema particular.

Entre los principales trabajos futuros, se destacan: la ampliación del trabajo experimental hacia aplicaciones con diferentes patrones de comunicación; la emulación de llamadas no determinísticas (requerida para extender el horizonte de aplicaciones que se pueden proteger con SEDAR); la determinación del intervalo óptimo de *checkpoint*; y la implementación de una estrategia de adaptación dinámica del mecanismo de recuperación a la ejecución en curso, basada en la determinación del momento a partir del cual vale la pena comenzar a proteger dicha ejecución mediante *checkpoints* (antes de ese instante, es más efectivo simplemente detectar y relanzar desde el comienzo. Como objetivo final, se busca la integración con arquitecturas que utilicen C/R para tolerancia a fallos permanentes [18], de modo de poder soportar ambos tipos de fallos con una única herramienta funcional para plataformas de la Exa-escala.

Referencias

1. T. Martsinkevich, O. Subasi, O. Unsal, F. Cappello, and J. Labarta, "Fault-tolerant protocol for hybrid task-parallel message-passing applications," in 2015 IEEE International Conference on Cluster Computing, pp. 563-570.
2. A. Benoit, A. Cavelan, F. M. Ciorba, V. Le Fèvre, and Y. Robert, "Combining checkpointing and replication for reliable execution of linear workflows with fail-stop and silent errors," International Journal of Networking and Computing, vol. 9, no. 1, pp. 2-27, 2019.
3. J. Elliott, M. Hoemmen, and F. Mueller, "Evaluating the impact of sdc on the gmres iterative solver," in Parallel and Distributed Processing Symposium, 2014 IEEE 28th International, IEEE, 2014, pp. 1193-1202.
4. D. Fiala, F. Mueller, C. Engelmann, R. Riesen, K. Ferreira, and R. Brightwell, "Detection and correction of silent data corruption for large-scale high-performance computing," in Proceedings of the International Conference on High Performance Computing, Networking, Storage and Analysis. IEEE Computer Society Press, 2012, p. 78.
5. A. Shye, J. Blomstedt, T. Moseley, V. J. Reddi, and D. A. Connors, "PLR: A software approach to transient fault tolerance for multicore architectures," IEEE Transactions on Dependable and Secure Computing, vol. 6, no. 2, pp. 135-148, 2009.
6. H. Mushtaq, Z. Al-Ars, and K. Bertels, "Efficient software-based fault tolerance approach on multicore platforms," in Proceedings of the Conference on Design, Automation and Test in Europe. EDA Consortium, 2013, pp. 921-926.
7. F. Cappello, A. Geist, W. Gropp, S. Kale, B. Kramer, and M. Snir, "Toward exascale resilience: 2014 update," Supercomputing frontiers and innovations, vol. 1, no. 1, pp. 5-28, 2014.
8. G. Lu, Z. Zheng, and A. A. Chien, "When is multi-version checkpointing needed?" in Proceedings of the 3rd Workshop on Fault-tolerance for HPC at extreme scale. ACM, 2013, pp. 49-56.
9. Montezanti, D., Frati, F.E., Rexachs, D., Luque, E., Naiouf, M.R., De Giusti, A.: SMCV: a Methodology for Detecting Transient Faults in Multicore Clusters.; CLEI Electron. J. 15(3), pp. 1-11 (2012).

10. Chen, Z.: Algorithm-based recovery for iterative methods without checkpointing. In Proceedings of the 20th international symposium on High performance distributed computing (pp. 73-84). ACM (2011).
11. Ferreira, K., Riesen, R., Oldfield, R., Stearley, J., Laros, J., Pedretti, K., & Brightwell, T.: rMPI: increasing fault resiliency in a message-passing environment. Sandia National Laboratories, Albuquerque, NM, Tech. Rep. SAND2011-2488 (2011).
12. Yalcin, G., Unsal, O. S., & Cristal, A.: Fault tolerance for multi-threaded applications by leveraging hardware transactional memory. In Proceedings of the ACM International Conference on Computing Frontiers (p. 4). ACM (2013).
13. Engelmann, C., & Böhm, S.: Redundant execution of HPC applications with MR-MPI. In Proceedings of the 10th IASTED International Conference on Parallel and Distributed Computing and Networks (PDCN) (pp. 15-17) (2011).
14. Ni, X., Meneses, E., Jain, N., & Kalé, L. V.: ACR: Automatic checkpoint/restart for soft and hard error protection. In Proceedings of the International Conference on High Performance Computing, Networking, Storage and Analysis (p. 7). ACM (2013).
15. N. Ali, S. Krishnamoorthy, N. Govind, and B. Palmer, "A redundant communication approach to scalable fault tolerance in pgas programming models," in Parallel, Distributed and Network-Based Processing (PDP), 2011 19th Euromicro International Conference on. IEEE, 2011, pp. 24-31.
16. Montezanti, D., Rexachs, D., Rucci, E., Luque, E., Naiouf, M.R., De Giusti, A.: Characterizing a Detection Strategy for Transient Faults in HPC, in Computer Science & Technology Series, XXI Argentine Congress of Computer Science Selected Papers. ISBN 978-987-4127-00-6. La Plata. EDULP, 2016, pp. 77-90.
17. D. Montezanti, A. De Giusti, M. Naiouf, J. Villamayor, D. Rexachs, and E. Luque, "A methodology for soft errors detection and automatic recovery," in 2017 International Conference on High Performance Computing & Simulation (HPCS). IEEE, 2017, pp. 434-441.
18. M. Castro-León, H. Meyer, D. Rexachs, and E. Luque, "Fault tolerance at system level based on RADIC architecture," Journal of Parallel and Distributed Computing, vol. 86, pp. 98-111, 2015.
19. Benoit, A., Cavelan, A., Cappello, F., Raghavan, P., Robert, Y., & Sun, H. (2018). Coping with silent and fail-stop errors at scale by combining replication and checkpointing. Journal of Parallel and Distributed Computing, 122, 209-225.
20. F. Mathur and A. Avizienis, "Reliability analysis and architecture of a hybrid-redundant digital system: generalized triple modular redundancy with self-repair", in Proc. Spring Joint Computer Conference (AFIPS '70), New York, USA, May, 1970, pp. 375 - 383.
21. Ansel, J., Arya, K., & Cooperman, G. (2009, May). DMTCP: Transparent checkpointing for cluster computations and the desktop. In 2009 IEEE International Symposium on Parallel & Distributed Processing (pp. 1-12). IEEE.

Simulación de altas prestaciones (GPU) para la reproducción del mosquito *Aedes aegypti* en el cementerio de Santo Tomé, Corrientes

Erica Montes de Oca¹, Remo Suppi², Laura De Gisuti¹, Marina Leporace³,
Maria Victoria Micieli⁴, Maria Soledad Santini⁵ y Marcelo Naiouf¹

¹ Instituto de Investigación en Informática LIDI, Facultad de Informática, Universidad Nacional de La Plata (UNLP) - Comisión de Investigaciones Científicas de la Provincia de Buenos Aires (CICPBA), La Plata, Buenos Aires, Argentina
{emontesdeoca,ldgiusti,mnaiouf}@lidi.info.unlp.edu.ar

² Universitat Autònoma de Barcelona, Department Computer Architecture & Operating System, Facultad de Ingeniería, Campus Bellaterra, Barcelona, España
{remo.suppi}@uab.cat

³ Laboratorio de Control de Vectores Entomológicos de Importancia Sanitaria (LaCEVEIS), Fundación Barceló, Santo Tomé, Corrientes, Argentina
{marinaleporace}@barcelo.edu.ar

⁴ Centro de Estudios Parasitológicos y de Vectores (CEPAVE), Consejo Nacional de Investigaciones Científicas y Técnicas (CONICET) - Universidad Nacional de La Plata (UNLP), La Plata, Buenos Aires, Argentina
{victoria}@cepave.edu.ar

⁵ Centro Nacional de Diagnóstico e Investigación en Endemoepidemias (CeNDIE) - Administración Nacional de Laboratorios e Institutos de Salud (ANLIS), Capital Federal, Buenos Aires, Argentina
{mariasoledadsantini}@gmail.com

Abstract. El Dengue, Zika y Chikungunya, son las enfermedades reemergentes de mayor preocupación a nivel mundial. La carencia de tratamientos médicos obliga a los agentes de salud a abordar la contención de los focos infecciosos desde la identificación y eliminación de los criaderos del vector transmisor. Una herramienta tecnológica que ayude en la toma de decisiones representa la solución a los costos asociados al tiempo de recolección de muestras y el análisis de datos, además de la consiguiente reducción de los gastos económicos. En este trabajo se presenta un modelo basado en agentes en GPU para la evaluación de la reproducción del vector *Aedes aegypti*, orientado a la toma de decisiones. El modelo ha sido validado con datos de Santo Tomé, Corrientes, dando excelentes resultados. En el trabajo se muestran las ventajas de utilizar una arquitectura paralela (GPU) para reducir el tiempo de cómputo de las simulaciones y, se presentan los resultados de una predicción realizada y validada con datos del sistema real.

Keywords: ABM, FLAME GPU, GPU, Enfermedades Infecciosas, *Aedes aegypti*, Productividad Pupal.

1 Introducción

La modificación del ambiente circundante afecta a las especies animales, a tal punto que pueden producir su extinción, su sobre-población o su adaptación a otras latitudes que nunca habían tenido trascendencia. Algunas especies migran hacia territorios que no son sus espacios habituales de cría, adaptándose y colonizando esas nuevas áreas en forma rápida y con un gran impacto en el entorno. Estas invasiones suelen alterar el ecosistema regional, ocasionando competencia con las especies nativas y, en ciertas ocasiones, participar en la transmisión de enfermedades locales o de nuevas enfermedades. Su rápida expansión se ve beneficiada por las rutas comerciales y turísticas, la construcción de viviendas y asentamiento sin planificación urbana y ambiental, y la ineficiencia de las campañas de control y prevención de su reproducción [1] [2].

En nuestro país, la especie de mosquito *Aedes aegypti*, es la responsable de la transmisión de la fiebre del Dengue, entre otras. Su población se encuentra distribuida entre las provincias del noroeste y noreste argentino (NOA, NEA), centro y norte de la Patagonia [3].

Un aspecto muy importante en la contención de las enfermedades derivadas de estas especies, son los estudios de campo sobre la población de mosquitos que incluyen la recolección de los estadios inmaduros del vector durante cierto período de tiempo. Estos datos, una vez analizados y contextualizados, permiten llegar a conclusiones para determinar las tareas que se deberán llevar a cabo en la prevención de la propagación y transmisión de los agentes infecciosos de los cuales son responsables. Esta recolección y análisis no es una tarea fácil, ya que requiere profesionales capacitados para tal fin y, un presupuesto económico importante para que los datos extraídos puedan ser utilizados de forma eficiente para la toma de decisiones. Es evidente que dada la gran extensión a cubrir la tecnología, por medio de la simulación de altas prestaciones, puede ser de gran ayuda para la construcción de políticas públicas, permitiendo predecir y tomar decisiones en tiempos reducidos, o realizar experimentos de campo de una forma selectiva para confirmar los resultados obtenidos por simulación.

Claramente, este sistema real denota una gran complejidad y elementos que son difíciles de modelar a través de modelos clásicos, pero a través del uso del modelado basado en agentes (ABM), la ciencia computacional permitirá estudiar, conocer y comprender los vínculos y comportamientos de los individuos que componen el sistema y, fundamentalmente, modelarlos y reproducir su comportamiento [4] [5].

Un aspecto no menos importante de este tipo de simulaciones es la potencia de cómputo necesaria para modelos de muchos individuos (agentes) o con reglas de interacción complejas. Por esto la computación paralela es necesaria, no sólo para reducir el tiempo de cada simulación para un número importante de individuos, sino para lograr obtener el número adecuado de simulaciones con estabilidad estadística [6]. Arquitecturas como las Unidades de Procesamiento Gráfico (GPUs) son muy adecuadas para este tipo de problemas, ya que permiten la reducción significativa del tiempo de cómputo (con un costo por MFLOP más que aceptable), e incrementar la escalabilidad del problema, mejorar las prestaciones y reducir el tiempo de procesamiento de cada simulación o incrementar la complejidad del modelo para analizar diferentes tipos de interacciones/reglas de comportamiento [7] [8].

En este trabajo se presenta un simulador para modelos basados en agentes desarrollado en el entorno Flame GPU [9], que permite la evaluación de la productividad pupal de los criaderos del mosquito *Aedes aegypti*. Este entorno de simulación, funcionando sobre una arquitectura paralela de bajo costo, permite la simulación de situaciones/entornos que ayuden a la toma de decisiones sobre la prevención y el control de la población de *Aedes aegypti*, así como el análisis de casos para validar las decisiones tomadas o tratar de entender el comportamiento y aprender de estas.

2 Antecedentes

El mosquito *Aedes aegypti* es originario de África y se reproduce en latitudes tropicales y subtropicales. Es el responsable de la transmisión de los agentes causantes de enfermedades tales como: Dengue, Zika, y Chikungunya. Esta especie se adapta con facilidad a lugares poblados, beneficiándose de todo objeto que acumule agua para utilizarlo como criadero [10] [11] [12].

Las etapas de vida del mosquito comprenden los estados de huevo, larva, pupa y adulto y, el ciclo completo de vida ronda entre los 42 a 56 días.

El mosquito adquiere el virus cuando pica a una persona en estado virémico [13]. El virus sólo se transmite por la picadura del mosquito [2], pero existen casos comprobados de contagios de Zika a través de fluidos (líquido amniótico, semen, y sangre) [14]. Sólo las hembras ingieren sangre que utilizan para la producción de huevos [15] [16] [17] [18].

Aedes aegypti, utiliza una gran variedad de contenedores artificiales o naturales para su reproducción y la elección del recipiente para la oviposición es variada. Depende de la adaptación de la población del mosquito a cada zona, así como también, de los hábitos y actitudes de la población en relación a los recipientes [19].

La productividad del contenedor representa la tasa de mosquitos que tienen la probabilidad de llegar a la etapa adulta y, por consiguiente, ser un potencial transmisor de agentes infecciosos. El principal método que se propone para el control de estos insectos está centrado en el manejo integrado de vectores, siendo una de las estrategias, la eliminación de los criaderos del mosquito que resultan más productivos [7] [20] ya que, permiten el controlar sin efectos tóxicos o sin alterar el equilibrio genético de la especie, como proponen otros métodos de control.

La importancia sanitaria del control de la población del mosquito *Aedes aegypti* se traduce en las campañas de prevención y control que la Organización Mundial de la Salud (OMS) que recomienda llevar a cabo, ya que la disminución de la abundancia de mosquitos reduce sustancialmente los focos de infección y permitiría a largo plazo, a través de la aplicación sistemática de campañas de prevención, la erradicación de dichas enfermedades [11].

3 Modelado y simulación

El modelo desarrollado está compuesto por tres actores básicos: personas, mosquitos y contenedores. El modelo es paramétrico y el usuario puede definir un conjunto de parámetros como (entre los más habituales) poblaciones de personas y mosquitos, cantidad y tipos de contenedores, porcentaje de personas y/o mosquitos infectados, productividad de los contenedores, cantidad de días de simulación, dimensiones del mundo virtual; también se pueden modificar parámetros específicos como ciclo de vida, etapas, número de huevos, entre otros, para ajustar el modelo específicamente a la especie y el entorno bajo estudio. Los mosquitos pasan por todas las etapas del ciclo de vida del vector comenzando por los mosquitos en estado huevo y luego de los días correspondientes a las etapas de larva y pupa y finalmente a un mosquito adulto. Como vector adulto buscará picar una persona, y luego, depositará los huevos en un contenedor.

El mosquito contrae la enfermedad al picar una persona infectada. Una vez cumplido el período de incubación del virus (aprox. entre 8 a 12 días), el mosquito si pica a una persona sana, la contagiará, y el virus permanecerá en él el resto de su vida (ver Fig. 1).

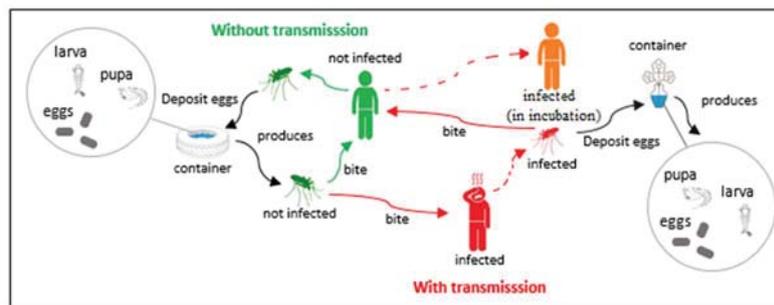


Fig. 1 Agentes y sus estados en el modelo de simulación propuesto

Los porcentajes de productividad de los contenedores que se utilizan se basan en el índice pupal que estima la cantidad de mosquitos adultos que emergerán de dicho contenedor. Sin embargo, el modelo es lo suficiente genérico para permitir que la productividad de los contenedores pueda ser definida a través de los índices larvales o pupales. Esta decisión es importante ya que, dependiendo de éste, la predicción variará (a causa de la tasa de mortalidad de cada etapa) por lo cual se recomienda trabajar con índices pupales siempre y cuando sea posible.

4 Experimentación y resultados

El modelo inicial, desarrollado en NetLogo, fue descartado para casos de uso real ya que dadas las limitaciones del entorno (número de agentes, dimensiones del terreno, reglas de interacción, ejecución secuencial, etc.) no cumplía con los objetivos propuestos. El modelo actual se diseñó bajo la premisa de acelerar las ejecuciones, llegar a un número de individuos de sistemas reales, y la decisión fue utilizar arquitecturas GPU por buena relación entre capacidad de cómputo/costo. Como núcleo de simulación para ejecución de agentes en paralelo se utilizó el framework Flame GPU, que es un entorno que le permite al programador concentrarse en los detalles del modelo y abstraerse de las particularidades de la arquitectura subyacente.

Las ventajas en la reducción de los tiempos de ejecución de las simulaciones entre los dos modelos son evidentes. Por ejemplo, teniendo una población de 100 personas, con un número inicial de mosquitos de 30,000 individuos considerando la productividad de los recipientes, el tiempo promedio de ejecución de la simulación en NetLogo es de 129 segundos, mientras que en GPU es de 16 segundos. La limitación del ambiente de simulación en NetLogo se aprecia claramente en la imposibilidad de ejecutar simulaciones con ésta misma cantidad de individuos pero sin considerar la productividad de los contenedores, ya que la población mosquitos tiene un crecimiento exponencial.

Un paso crucial para estas simulaciones fue la validación del modelo que se realizó tanto para el modelo inicial (desarrollado en NetLogo), como para el modelo actual implementado sobre Flame GPU [21]. La validación se realizó con datos de un entorno real sobre un estudio de campo llevado a cabo en Sao Sebastiao en Brasil, en un área de 400,4 Km² donde se realizó un análisis de muestras exhaustivo con recolección de los estadios inmaduros del vector durante la temporada de reproducción entre los años 2002-2004 [22].

Sin embargo, los recipientes relacionados a los hábitos de la población son distintos, así como también la ecología del vector y las condiciones ambientales, por tal razón resulta imprescindible que la herramienta sea validada con datos de nuestro país para que pueda ser utilizada efectivamente.

Para validar el modelo con información de Argentina, los datos utilizados fueron recolectados a través de un estudio de campo en el cementerio de Santo Tomé, Corrientes, durante los meses de agosto de 2017 y mayo a junio de 2018. El modelo fue modificado y adaptado a los recipientes utilizados comúnmente en un cementerio: floreros de plástico, de vidrio, de cerámica, de cemento, macetas, u otros recipientes, realizando una clasificación de siete tipos de contenedores diferentes donde a partir de la cantidad de pupas recolectadas se obtuvo la productividad pupal de cada recipiente.

Como se ha mencionado anteriormente, el modelo permite ser configurado para analizar el comportamiento de la población del mosquito considerando la productividad pupal de los contenedores, por lo que la cantidad de mosquitos total durante la simulación se verá reducida. Este dato es importante ya que, si no se la tiene en cuenta, la población del vector crecerá de manera exponencial, ya que el 100% de los huevos depositados en cada contenedor llegarán a la etapa adulta.

Una vez calibrado el modelo, se realizó una serie de experimentos, para validar la información obtenida con los datos del sistema real. Como arquitectura base se utilizó

una GPU GeForce GTX 960 Ti con 2 GB GDDR5, con 1024 cores y un ancho de memoria de 128 bits. En la Fig. 2, se puede observar el porcentaje de pupas por contenedor para el modelo de referencia (sistema real) y el modelo propuesto (desarrollado en Flame GPU) para una población de mosquitos inicial de 3000 individuos, simulando un total de 100 días y siete tipos de contenedores (Container ID). Los resultados aquí presentados representan un promedio de cincuenta ejecuciones de la simulación.

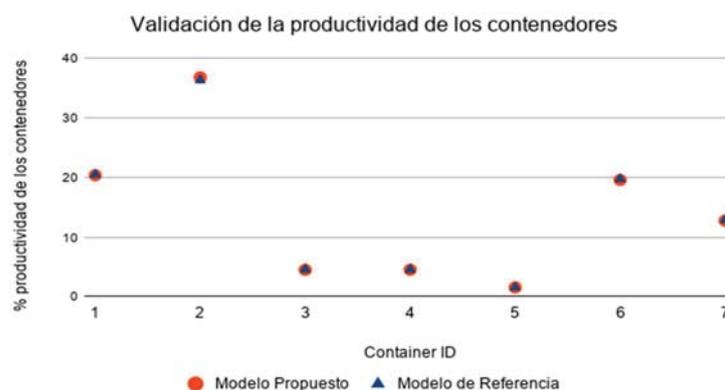


Fig. 2 Validación de la productividad de los contenedores del modelo desarrollado en FLAME GPU con el sistema real

Las diferencias entre los porcentajes de productividad entre los modelos (Fig. 2), no superan el 1% en todos los casos. La mayor diferencia se encontró en el contenedor 2 (floreros de plástico negro), que representa el recipiente con mayor productividad del sistema, y fue del 0,46%. Claramente, los resultados obtenidos muestran la capacidad del modelo de simulación computacional de reproducir resultados con un error mínimo en relación al sistema real.

En la Fig. 3, se puede observar el tiempo de ejecución medido en segundos para la simulación considerando o no la productividad de los contenedores. Los resultados representan el promedio de cincuenta ejecuciones de la simulación, simulando cada una de ellas un total de 80 días, para los tamaños de entrada iniciales de la población de mosquitos de 300 a 50.000.

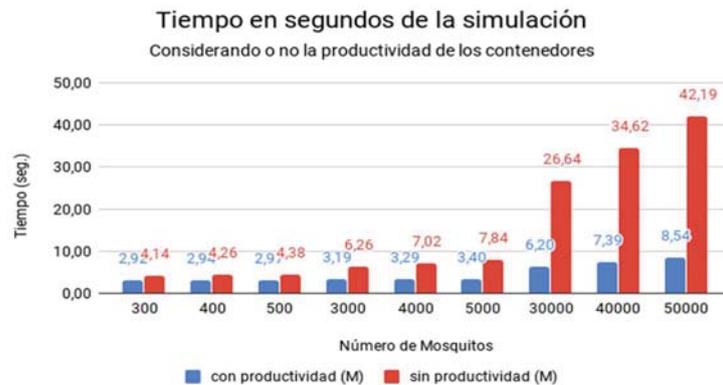


Fig. 3 Tiempo en segundos de la simulación considerando o no la productividad de los contenedores para 80 días de simulación.

Claramente se puede apreciar que el modelo escala con respecto al tamaño de entrada de la simulación. En el presente experimento, el análisis del modelo se ve limitado por la memoria de la GPU ya que el entorno (FlamGPU) realiza una pre-carga de los agentes en memoria al inicio de la simulación limitando a 50.000 individuos y 80 días de simulación la combinación para el hardware disponible. Con el fin de estimar el tiempo requerido para simular poblaciones más grandes, se ha utilizado un modelo de regresión lineal donde se representa el tiempo de ejecución en segundos de la simulación considerando o no la productividad de los contenedores, simulando un total de 80 días.

La Fig. 4, muestra el tiempo de ejecución medido (M) en segundos de la simulación para los tamaños de entrada que van de 300 a 50.000 mosquitos simulando 80 días, los modelos de tiempo obtenidos por regresión lineal para la simulación y los tiempos estimados (E) en segundos calculados con dichos modelos.

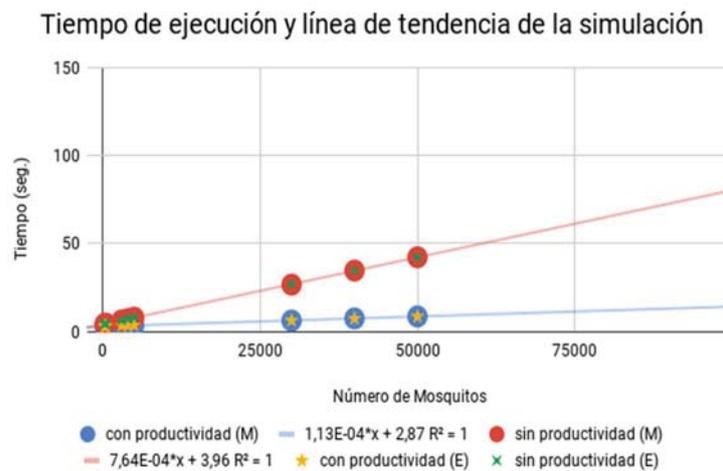


Fig. 4 Modelos de estimación de tiempo obtenidos por regresión lineal para la simulación considerando o no la productividad de los contenedores

El porcentaje de error mayor cometido al estimar con los modelos de tiempo de ejecución de la simulación (Fig. 4), fue de: 1,5%.

En la Fig. 5, se muestra cómo responde el modelo al cambiar la productividad de los contenedores con mayor productividad del sistema. Se escogieron los contenedores 1, 2 y 6. Manteniendo la población de 3000 mosquitos fija, simulando en todos los casos 100 días. Los resultados presentados son el promedio de los tiempos medidos en segundos (50 ejecuciones). Se clasificó a las combinaciones en cuatro casos posibles: Caso A, representa el tiempo de ejecución en segundos conservando el porcentaje de productividad original del sistema (20,45%, 36,36% y 19,70%, de manera respectiva para los contenedores 1, 2 y 6); Caso B, incrementando al 50,0% de productividad uno de los tres contenedores seleccionados, por lo que el Caso se subdivide en tres subcasos (Caso B-1: 50,0%, 36,36% y 19,70%, para los contenedores 1, 2 y 6, respectivamente; Caso B-2: 20,45%, 50,0% y 19,70%, contenedores 1, 2 y 6 respectivamente; y, Caso B-3: 20,70%, 36,36% y 50,0%, contenedores 1, 2 y 6 respectivamente); Caso C: combinación de dos de los tres contenedores con 50,0% de productividad (da como resultado tres subcasos: Caso C-1: 50,0%, 50,0% y 19,70%, para los contenedores 1, 2 y 6; Caso C-2: 50,0%, 36,36% y 50,0%, para los contenedores 1, 2 y 6 respectivamente; y, Caso C-3: 20,45%, 50,0% y 50,0%, para los contenedores 1, 2 y 6 respectivamente); y, Caso D: los tres contenedores con productividad al 50,0%.

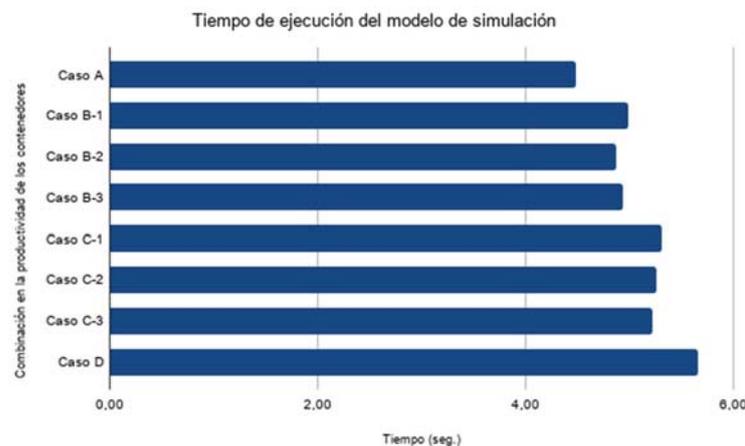


Fig. 5 Tiempo de ejecución medidos en segundos del modelo de simulación, incrementando el porcentaje de productividad de los contenedores 1, 2 y 6.

Se observa que el modelo se mantiene estable aún modificando las productividades de los contenedores.

Para comprobar si el modelo puede predecir la cantidad de pupas de *Aedes aegypti* se dividieron los datos de campo por año en dos conjuntos: uno para ajustar el modelo (2017) y otro para predecir la cantidad (2018) y ver la desviación con la realidad. Tomando los tipos de contenedores para los cuales se disponían datos en dos años

diferentes (2017 y 2018) de 7 contenedores diferentes (recipientes de plástico en general, floreros de plástico negro, recipientes de cemento, frascos cementados, recipientes de metal, recipientes de vidrio, macetas y recipientes de cerámica) se escogieron 4 contenedores (floreros plásticos negros, recipientes de metal, recipientes de vidrio y recipientes de cerámica).

Para ambos conjuntos de datos, se calculó el total de pupas de los cuatro recipientes que se tuvieron en cuenta para el siguiente experimento, obteniendo el porcentaje de productividad de los contenedores. Con los datos del año 2017 se configuró el modelo teniendo en cuenta sólo los cuatro contenedores mencionados (con sus respectivas productividades 2017) y después de realizada la simulación se volvió a validar para estos datos, obteniéndose una diferencia menor al 2% en todos los contenedores.

Posteriormente, se inicializó solo la población de mosquitos del modelo y se realizó la simulación para 100 días. De la comparación de los datos simulados, se calculó la diferencia entre el porcentaje de pupas obtenido y los datos reales siendo los resultados muy satisfactorios, observándose la mayor diferencia en el contenedor 4 (floreros de cerámica) de 3,39%, seguido por 0,80% en los contenedores 2 y 3 (recipientes de metal y vidrio respectivamente), y 0,09% para el contenedor 1 (floreros de plástico negro). En la Fig. 6, se presentan los porcentajes de productividad para los cuatro contenedores para el modelo propuesto, el sistema real (modelo de referencia) y el modelo con los datos del 2018 (Predicción 2018).

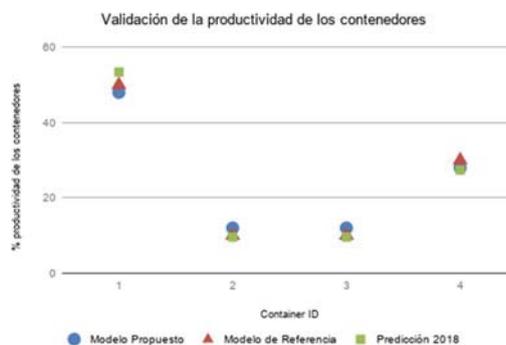


Fig. 6 Validación del porcentaje de productividad entre el modelo propuesto con el sistema real, y, la predicción con respecto a los datos de 2018.

Como se observa el modelo puede considerarse válido y capaz de simular situaciones de reproducción pupal para los ecosistemas considerados.

4 Conclusiones y Trabajos Futuros

El aumento de casos de dengue en las últimas décadas se ha convertido en una preocupación a nivel mundial. Su expansión se debe a la proliferación de la población de su vector transmisor: *Aedes aegypti*. La recomendación de la OMS es de vigilancia

y control de la población del mosquito como solución para la erradicación de las enfermedades infecciosas.

La diversidad de recipientes que utiliza para su reproducción varía según los hábitos de los residentes de cada zona, y las características particulares del vector que allí se desarrolla. La identificación de los recipientes más productivos es la acción principal para lograr el control de la población del vector y la disminución del foco infeccioso. Una herramienta de simulación que facilite las tareas predicción y ayuda en la toma de decisiones del personal sanitario, acortando los tiempos de análisis y estudio de la información del vector sería de una ayuda inestimable.

El modelo desarrollado ejecutándose sobre GPU, muestra un excelente comportamiento desde el punto de vista de las altas prestaciones, así como también, en relación al sistema real. La obtención de resultados en tiempo de respuesta acotados es sumamente importante en problemas epidemiológicos, y la posibilidad de predecir, a partir de experiencias pasadas, el futuro, permite mejorar la toma de decisiones y llevar a cabo las acciones que se podrían traducir en la disminución del contagio de enfermedades infecciosas.

Como trabajos futuros se pueden mencionar: la adaptación del modelo a otras zonas de Argentina donde el riesgo de contagio también es crítico, la valoración del consumo energético del modelo y la visualización interactiva que lo transforme en una herramienta para la toma de decisiones y su posible migración al cloud a fin de que pueda ser utilizado como M&SaaS (Modeling & Simulation as a Service).

Referencias

1. Álvarez, E.M.C., Tórres, A.A., Tórres, A.A., Semper, G.A.I., Romeo, A.D.: Dengue, Chikungunya, virus del Zika. Determinantes sociales. Rev. Médica Electrónica, Año 2018, nro. 1 (2018)
2. Contigiani, M.S., Díaz, L.A., Spinsanti, L.I., Tauro, L.B.: Arbovirus. Proceedings de Jornadas Regionales sobre Mosquitos. Mar del Plata, Buenos, Aires, Argentina (2016)
3. Stein, M., Rossi, G.C., Almirón, W.R.: Distribución geográfica de Culicidae de Argentina. Proceedings de Jornadas Regionales sobre Mosquitos. Mar del Plata, Buenos, Aires, Argentina (2016)
4. Ginovart, M.: ¿Qué pueden ofrecer los modelos basados en agentes vivos en el contexto docente? Modelling in Science Education and Learning, Vol. 8 (2015)
5. Izquierdos-Espinosa, E.V.: Implementación de un modelo de agentes para estudiar la propagación del virus de la fiebre Chikungunya. Tesis de Master en Matemáticas Aplicadas. Quito, Ecuador (2016)
6. Borges, F., Gutierrez-Milla, A., Suppi, R., Lague, E., de Brito Arduino, M.: An Agent-Based Model for assessment for Aedes aegypti pupal productivity. Proceeding 2015 Winter Simulation Conference, pp. 159-170 (2015)
7. Montes de Oca, E., De Giusti, L., De Giusti, A., Naiouf, M.: Análisis de consumo energético en Cluster de GPU y MultiGPU en un problema de alta demanda computacional. Congreso Argentino de Ciencias de la Computación, Tantil, Buenos Aires, Argentina (2018)
8. Nvidia, Qué es GPU Computing?: <https://la.nvidia.com/object/what-is-gpu-computing-la.html> fecha de acceso mayo 2019
9. FLAME GPU Documentation: <https://buildmedia.readthedocs.org/media/pdf/flamegpu/latest/flamegpu.pdf>
10. Ruíz-López, F., González-Mazo, A., Vélez-Mira, A., Gómez, G.F., Zuleta, L., Uribe, S., Vélez-Bernal, I.D.: Presencia de Aedes (Stegomyia) aegypti (Linnaeus, 1762) y su infección

- natural con el virus del dengue en alturas no registradas para Colombia. *Biomédica*, vol. 36, nro. 2 (2016)
11. Organización Mundial de la Salud: <https://www.who.int/es/news-room/fact-sheets/detail/dengue-and-severe-dengue> fecha de acceso mayo 2019
 12. Organización Mundial de la Salud: <https://www.who.int/denguecontrol/mosquito/es/> fecha de acceso mayo 2019
 13. Albricou-Llinás, G., Chiappero, M.B., Rodán-Dueñas, J.C., Gardenal, C.N.: Reconstrucción de una invasión: pasado y presente de poblaciones de *Aedes (Stegomyia) aegypti* en Argentina. *Proceedings de Jornadas Regionales sobre Mosquitos*. Mar del Plata, Buenos Aires, Argentina (2016)
 14. Gorodner, J.O.: Dengue, fiebre Zika y fiebre Chikungunya. *Patologías contaminantes y cambio climático en América*. *Rev. Asociación Médica Argentina*, Vol. 129, nro. 1 (2016)
 15. González-Escalona, M., Rojas-Izquierdo, M., Santos-Romero, A.I.: Vigilancia y lucha antivectorial en la enfermedad del dengue. Papel del profesional de la salud en la atención primaria. *Rev. Uruguya de Enfermería*, vol. 12, nro. 1, pp. 269-275 (2017)
 16. Vezzani, D.: El hábita del *Aedes aegypti* (Diptera: Culicidae) en Buenos Aires para distintas escalas espaciales de estudio. Tesis de Doctorado en Ciencias Biológicas. Universidad Nacional de Buenos Aires, Buenos Aires, Argentina (2003)
 17. De-Majo, M.S., Byttebier, B., Fischer, S.: Predisposición de los huevos de *Aedes aegypti* (Diptera: Culicidae) a eclosionar a bajas temperaturas: efectos de las condiciones ambientales de los meses previos. VIII Congreso Argentino de Entomología, Bariloche, Río Negro, Argentina (2012)
 18. Ngugi, H.M., Mutuku, F.M., Ndenga, B.A., Musunzaji, P.F., Mbakaya, J.O., Aswani, P., Irungu, M.W., Mukoko, D., Kitron, U., LaBeaud, A.D.: Characterization and productivity profiles of *Aedes aegypti* (L.) breeding habitats across rural and urban landscapes in western and coastal Kenya. *Rev. Parasites & Vectors*, vol. 10, nro. 1 (2017)
 19. Organización Mundial de la Salud: Dengue: guía para el diagnóstico, tratamiento, prevención y control. https://www.who.int/denguecontrol/resources/dengue_guidelines_2009/es/ (2009)
 20. Cromwell, E.A., Stoddard, S.T., Barker, C.M., Van Rie, A., Messeer, W.B., Meshnick, S.R., Morrison, A.C., Scott, T.W.: The relationship between entomological indicators of *Aedes aegypti* abundance and dengue virus infection. *PLoS Negl Trop Dis*, vol. 11, nro. 3 (2017)
 21. Montes de Oca, E., Suppi, R., De Gisuti, L., Naiouf, M.: Agent-Model Based on Flame GPU for assessing pupal productivity of the transmitting vector of *Aedes Aegypti* infectious diseases. Aceptado para publicar en *Proceedings de Modelling and Applied Simulation*, Lisboa, Portugal. 18-20 de Septiembre (2019)
 22. de Brito Arduino, M.: Assessment of *Aedes aegypti* Pupal Productivity during the Dengue Vector Control Program in a Costal Urban Centre of São Paulo State, Brazil. *Journal of Insects* Volume 2014, Article ID 301083, 9 pages.

Simulación de N Cuerpos Computacionales sobre Intel Xeon Phi KNL

Enzo Rucci¹ , Ezequiel Moreno², Malek Camilo², Adrián Pousa¹ and Franco Chichizola¹ 

¹ III-LIDI, Facultad de Informática, UNLP – CEA CICPBA. La Plata (1900), Bs As, Argentina
{erucci, apousa, francoch}@lidi.info.unlp.edu.ar

² Facultad de Informática, UNLP. La Plata (1900), Bs As, Argentina
{morenoezequiel8, malek.camilo}@gmail.com

Resumen. En la comunidad HPC, el uso de aceleradores se ha consolidado como estrategia para mejorar el rendimiento de los sistemas al mismo tiempo que la eficiencia energética. Recientemente, Intel introdujo Knights Landing (KNL), la segunda generación de aceleradores Xeon Phi. Entre sus características destacadas, se puede mencionar su gran cantidad de núcleos, la incorporación de las instrucciones vectoriales AVX-512 y la integración de una memoria de alto ancho de banda. Este trabajo se enfoca en la paralelización de la simulación de N cuerpos computacionales sobre un acelerador Xeon Phi KNL. Además de representar la base de un gran número de aplicaciones de la astrofísica, esta simulación requiere de alto poder computacional para ser procesada con un tiempo de respuesta aceptable. Comenzando por una implementación secuencial, se muestra cómo es posible que la implementación paralela alcance 2355 GFLOPS a través de diferentes optimizaciones.

Keywords: Xeon Phi, Knights Landing, N body, AVX-512, MCDRAM.

1 Introducción

En la actualidad, el problema del consumo energético se presenta como uno de los mayores obstáculos para el diseño de sistemas de cómputo de alto rendimiento (HPC) que sean capaces de alcanzar la escala de los Exaflops. En ese sentido, el uso de aceleradores (como pueden ser las GPUs de NVIDIA o los procesadores Xeon Phi de Intel) se ha consolidado como estrategia para mejorar la eficiencia energética. El factor clave en estos dispositivos radica en que ofrecen un pico de rendimiento muy superior al de las CPUs al mismo tiempo que limitan el consumo de potencia, lo que les permiten obtener mejores cocientes FLOPS/Watt [1].

Recientemente, Intel ha presentado la segunda generación de sus procesadores Xeon Phi, con nombre clave Knights Landing (KNL). Entre sus principales características, se pueden mencionar la gran cantidad de núcleos con soporte para *hyper-threading*, la incorporación de las instrucciones vectoriales AVX-512 y la integración de una memoria de alto ancho de banda [2].

Entre las áreas que se ven afectadas por los problemas actuales de los sistemas HPC se encuentra la física, debido a que cuenta con un número creciente de aplicaciones que requieren de cómputo de altas prestaciones para alcanzar tiempos de respuesta aceptables. Una de esas aplicaciones es el clásico problema de la simulación de N cuerpos computacionales, la cual aproxima en forma numérica la evolución de un sistema de cuerpos en el que cada uno interactúa con todos los restantes [3].

El uso más conocido de esta simulación quizás sea en la astrofísica, donde cada cuerpo representa una galaxia o una estrella particular que se atraen entre sí debido a la fuerza gravitacional. Sin embargo, también se ha empleado en otras áreas muy diferentes. Por ejemplo, para el plegado de proteínas en la biología computacional [4] o para la iluminación global de una imagen en computación gráfica [5].

Existen diferentes métodos para computar la simulación de los N cuerpos [6]. La forma más sencilla se denomina de *all-pairs* (o directa) y consiste en evaluar todas las interacciones entre todos los pares de cuerpos. Es un método de fuerza bruta que posee alta demanda computacional ($O(n^2)$). Debido a su complejidad, la versión directa sólo es empleada cuando la cantidad de cuerpos es moderada, o bien para computar las interacciones entre cuerpos cercanos en combinación con una estrategia para los que se encuentran lejanos entre sí. Esta segunda opción es el enfoque empleado por métodos avanzados que permiten simular la interacción entre una gran cantidad de cuerpos, como pueden ser el de Barnes-Hut ($O(n \log n)$) o el Fast Multipole Method ($O(n)$). Por lo tanto, al acelerar la versión directa no solo se mejora a la misma sino también a las otras que la emplean como componente.

Este trabajo se enfoca en la paralelización de la simulación de N cuerpos computacionales sobre un acelerador Intel Xeon Phi KNL. Comenzando por una implementación secuencial, se muestra cómo es posible que la implementación paralela alcance 2355 GFLOPS a través de diferentes técnicas de optimización.

El resto del artículo se organiza de la siguiente forma. La Sección 2 introduce el problema de los N cuerpos computacionales mientras que la Sección 3 describe la arquitectura del Intel Xeon Phi KNL. La Sección 4 detalla las implementaciones realizadas. En la Sección 5 se analizan los resultados experimentales mientras que en la Sección 6 se discuten los trabajos relacionales. Finalmente, la Sección 7 resume las conclusiones junto a los posibles trabajos futuros.

2 Simulación de N Cuerpos Computacionales

El problema consiste en simular la evolución de un sistema compuesto por N cuerpos durante una cantidad de tiempo determinada. Dada la masa y el estado inicial (velocidad y posición) de cada cuerpo, se simula el movimiento del sistema a través de instantes discretos de tiempo. En cada uno de ellos, todo cuerpo experimenta una aceleración que surge de la atracción gravitacional del resto, lo que afecta a su estado.

La física subyacente a la simulación es fundamentalmente la mecánica Newtoniana [7]. La simulación se realiza en 3 dimensiones espaciales y la atracción gravitacional entre dos cuerpos C_1 y C_2 se computa usando la ley de gravitación universal de Newton:

$$F = \frac{G \times m_1 \times m_2}{r^2}$$

donde F corresponde a la magnitud de la fuerza gravitacional entre los cuerpos, G corresponde a la constante de gravitación universal¹, m_1 corresponde a la masa del cuerpo C_1 , m_2 corresponde a la masa del cuerpo C_2 , y r corresponde a la distancia Euclídea² entre los cuerpos C_1 y C_2 .

Cuando N es mayor a 2, la fuerza de gravitación sobre un cuerpo, se obtiene con la sumatoria de todas las fuerzas de gravitación ejercidas por los $N-1$ cuerpos restantes. La fuerza de atracción se traduce entonces en una aceleración del cuerpo mediante la aplicación de la segunda ley de Newton, la cual está dada por la siguiente ecuación:

$$F = m \times a$$

donde F es el vector fuerza, calculado utilizando la magnitud obtenida con la ecuación de gravitación y la dirección y sentido del vector que va desde el cuerpo afectado hacia el cuerpo que ejerce la atracción.

De la ecuación anterior, se despeja que se puede calcular la aceleración de un cuerpo dividiendo la fuerza total por su masa. Durante un pequeño intervalo de tiempo dt , la aceleración a_i del cuerpo C_i es aproximadamente constante, por lo que el cambio en velocidad es aproximadamente:

$$dv_i = a_i dt$$

El cambio en la posición de un cuerpo es la integral de su velocidad y aceleración sobre el intervalo de tiempo dt , el cual es aproximadamente

$$dp_i = v_i dt + \frac{a_i}{2} dt^2 = \left(v_i + \frac{dv_i}{2} \right) dt$$

Esta fórmula emplea el esquema de integración Leapfrog [8], en el cual una mitad del cambio de posición emplea la velocidad *vieja* mientras que la otra considera la velocidad *nueva*.

3 Intel Xeon Phi KNL

KNL es la segunda generación de aceleradores Xeon Phi y la primera capaz de funcionar en forma autónoma (no requiere de un *host*). El diseño de su arquitectura se basa en el concepto de *tile* (unidad básica de replicación), donde cada uno cuenta con dos núcleos y una caché L2 de 1MB compartida entre ellos. Los *tiles* se encuentran físicamente replicados hasta 38 veces³, estando interconectados por una malla 2D con

¹ Equivalente a $6,674 \times 10^{-11}$

² Se calcula utilizando la fórmula $\sqrt{(x_2-x_1)^2+(y_2-y_1)^2+(z_2-z_1)^2}$, siendo (x_1, y_1, z_1) las coordenadas de C_1 y (x_2, y_2, z_2) las coordenadas de C_2 .

³ Aunque sólo 36 de ellos pueden estar activos al mismo tiempo

coherencia de caché. A nivel de núcleo, la micro-arquitectura se basa en la de los Intel Atom con 4 hilos hw y 2 unidades de procesamiento vectorial (VPU). Estas VPU introducen las nuevas instrucciones Intel AVX-512 de 512 bits, además de dar soporte a las ya conocidas SSEx (de 128 bits) y AVXx (de 256 bits). Como una instrucción AVX-512 puede realizar 8 operaciones de suma/multiplicación con operandos de doble precisión (DP) o 16 operaciones con operandos de simple precisión (SP), el pico de rendimiento se encuentra por encima de 3(6) TFLOPS en DP(SP) [2].

Además de una memoria DDR4 convencional, KNL cuenta con una memoria de alto ancho de banda denominada MCDRAM. Esta memoria puede ser configurada en uno de entre tres modos (1) modo *Cache*, donde se la utiliza como memoria cache L3 (transparente al programador); (2) modo *Flat*, donde es tratada como una memoria direccionable con alto ancho de banda y baja latencia; y (3) modo *Híbrido*, donde una parte es utilizada en modo *Cache* y el resto en modo *Flat*. Si bien el modo *Flat* permite alcanzar un mayor rendimiento, requiere de intervención del programador.

Por su parte, la malla 2D configurable le permite a KNL ofrecer tres modos de operación de cluster diferentes: (1) *All-to-all*; (2) *Quadrant*; y (3) *sub-NUMA*. La diferencia principal entre estos modos radica en si los núcleos tendrán acceso UMA o NUMA a una memoria particular.

Desde el punto de vista de la programación, KNL soporta modelos tradicionales en HPC como MPI y OpenMP, lo que representa una ventaja frente a otros aceleradores que requieren del aprendizaje de lenguajes específicos como CUDA u OpenCL. Sin embargo, para obtener alto rendimiento, resulta necesario que los programas sean capaces de explotar eficientemente la jerarquía de memoria, además de las capacidades vectoriales de los núcleos [9].

4 Implementación

En esta sección se describen las optimizaciones realizadas a la implementación para el Xeon Phi KNL.

4.1 Implementación *Naive*

Inicialmente se desarrolló una implementación *naive*, la cual servirá como referencia para evaluar las mejoras introducidas por las técnicas de optimización posteriores. En la Fig. 1 se muestra el pseudo-código de la implementación *naive*.

4.2 Multi-hilado

La primera optimización consiste en introducir paralelismo a nivel de hilos a través de directivas OpenMP. Los bucles de las líneas 4 y 26 son paralelizados mediante la inserción de directivas *parallel for*. De esta forma, se respetan las dependencias del problema ya que un cuerpo no se puede mover hasta que el resto no haya terminado de calcular sus interacciones y tampoco puede avanzar al paso siguiente hasta que los restantes no hayan completado el paso actual. Por último, la opción *static* para la

cláusula *schedule* permite distribuir equitativamente la cantidad de cuerpos entre los hilos logrando un balance de carga de costo mínimo.

```

1 // Para cada instante discreto de tiempo
2 for (t = 1; t <= D; t += DT){
3     // Para cada cuerpo que experimenta una fuerza
4     for (i = 0; i < N; i++){
5         // Componentes de la fuerza del cuerpo i
6         forcesx[i] = forcesy[i] = forcesz[i] = 0.0;
7         // Para cada cuerpo que ejerce una fuerza
8         for (j = 0; j < N; j++){
9             // Ley de atracción gravitacional de Newton
10            dx = xpos[j] - xpos[i]; dy = ypos[j] - ypos[i]; dz = zpos[j] - zpos[i];
11            dsquared = (dx*dx) + (dy*dy) + (dz*dz) + SOFT;
12            F = G * masses[i] * masses[j]; d32 = 1/POW(dsquared,1.5);
13            // Calcular la fuerza total
14            forcesx[i] += F*dx*d32; forcesy[i] += F*dy*d32; forcesz[i] += F*dz*d32;
15        }
16        // Calcular aceleración
17        ax = forcesx[i] / masses[i]; ay = forcesy[i] / masses[i]; az = forcesz[i] / masses[i];
18        // Calcular velocidad
19        dvx = (xvi[i] + (ax*DT*0.5)); dvy = (yvi[i] + (ay*DT*0.5)); dvz = (zvi[i] + (az*DT*0.5));
20        // Calcular posición
21        dpx[i] = dvx * DT; dpy[i] = dvy * DT; dpz[i] = dvz * DT;
22        // Actualizar velocidad
23        xvi[i] = dvx; yvi[i] = dvy; zvi[i] = dvz;
24    }
25    // Actualizar posiciones
26    for(i = 0; i < N; i++){
27        xpos[i] += dpx[i]; ypos[i] += dpy[i]; zpos[i] += dpz[i];
28    }
29 }

```

Fig. 1. Pseudo-código para la implementación *naive*

4.3 Vectorización

El reporte de optimización del compilador ICC permite identificar qué bucles son vectorizados en forma automática. A partir del mismo, se pudo saber que el compilador detecta dependencias falsas en algunas operaciones, imposibilitando la generación de instrucciones SIMD. En consecuencia, para garantizar la vectorización de operaciones se optó por un enfoque guiado a través del uso de la directiva *simd* de OpenMP 4.0. En particular, los bucles vectorizados son los de las líneas 8 y 26, siendo el último combinado con la directiva *parallel for*, como se mencionó en la sección anterior. Por último, para favorecer el uso de instrucciones SIMD, se alinearon los datos a 64-bytes en su alocação, agregando la cláusula *aligned* a la directiva *simd*.

4.4 Procesamiento por Bloques

Para explotar la localidad de datos, es posible implementar un procesamiento por bloques. En la Fig. 2 se muestra el pseudo-código de la implementación paralela por bloques. El cambio con respecto a la implementación *naive*, consiste en desdoblar el bucle *i* (línea 4) y colocarlo dentro del bucle *j* (línea 8). En consecuencia, se reemplaza un bucle por otros dos: uno que itera sobre todos los bloques (línea 5) y uno más interno que itera sobre los cuerpos de cada bloque (línea 12). De esta manera se mi-

nimiza el tráfico entre cache y memoria principal al aumentar la cantidad de veces que se usa cada dato en el bucle más interno.

```

1 // Para cada instante discreto de tiempo
2 for (t = 1; t <= D; t += DT){
3 // Para cada bloque de cuerpos de tamaño BS
4 #pragma omp parallel for schedule(static) private(i,j)
5 for(ii = 0; ii < N; ii+=BS){
6 // Componentes de la fuerza del cuerpo i
7 forcesx[BS] = forcesy[BS] = forcesz[BS] = {0.0};
8 // Para cada cuerpo que ejerce una fuerza
9 for(j = 0; j < N; j++){
10 // Para cada cuerpo que experimenta una fuerza
11 #pragma omp simd aligned
12 for (i = ii; i < ii+BS; i++){
13 // Ley de atracción gravitacional de Newton
14 dx = xpos[j] - xpos[i]; dy = ypos[j] - ypos[i]; dz = zpos[j] - zpos[i];
15 dsquared = (dx*dx) + (dy*dy) + (dz*dz) + SOFT;
16 F = G * masses[i] * masses[j]; d32 = 1/POW(dsquared,1.5);
17 // Calcular la fuerza total
18 forcesx[i-ii] += F*dx*d32; forcesy[i-ii] += F*dy*d32; forcesz[i-ii] += F*dz*d32;
19 }
20 }
21 #pragma omp simd aligned
22 for (i = ii; i < ii+BS; i++){
23 // Calcular aceleración |M = F * A| ---> |A = F / M|
24 ax = forcesx[i-ii] / masses[i]; ay = forcesy[i-ii] / masses[i]; az = forcesz[i-ii] / masses[i];
25 // Calcular velocidad
26 dvx = (xvi[i] + (ax*DT*0.5)); dvy = (yvi[i] + (ay*DT*0.5)); dvz = (zvi[i] + (az*DT*0.5));
27 // Calcular posición
28 dpv[i] = dvx * DT; dpy[i] = dvy * DT; dpz[i] = dvz * DT;
29 // Actualizar velocidad
30 xvi[i] = dvx; yvi[i] = dvy; zvi[i] = dvz;
31 }
32 }
33 // Actualizar posiciones
34 #pragma omp parallel for simd aligned
35 for(int i = 0; i < N; i++){
36 xpos[i] += dpv[i]; ypos[i] += dpy[i]; zpos[i] += dpz[i];
37 }
38 }

```

Fig. 2. Pseudo-código para la implementación paralela por bloques

4.5 Desenrollado de Bucles

El desenrollado de bucles es otra técnica de optimización que puede mejorar el rendimiento de un programa. En particular, se encontró beneficioso desenrollar completamente el bucle más interno de la implementación presentada en la Fig. 2 (línea 9), además del que actualiza las posiciones de los cuerpos posteriormente (línea 35).

5 Resultados Experimentales

5.1 Diseño Experimental

Todas las pruebas fueron realizadas en un sistema equipado con un Intel Xeon Phi 7230 de 64 núcleos (4 hilos hw por núcleo), 192 GB de memoria RAM y 16 GB de

memoria MCDRAM. En ese sentido, el procesador fue usado en modo cluster *All-to-all* y la memoria MCDRAM en modo *Flat*⁴.

Respecto al software, el sistema operativo es Ubuntu 16.04.3 LTS mientras que el compilador es el ICC (versión 19.0.0.117). Para acelerar el cómputo de operaciones en punto flotante se usó el *flag -fp-mode fast=2* mientras que para utilizar las instrucciones vectoriales AVX2 y AVX-512, se emplearon los *flags -xAVX2* y *-xMIC-AVX512*, respectivamente. Además, se empleó el comando *numactl* para poder explotar la memoria MCDRAM (no requiere modificaciones al código fuente). Por último, se varió la carga de trabajo al usar diferentes números de cuerpos: $N = \{65536, 131072, 262144, 524288, 1048576\}$ ⁵.

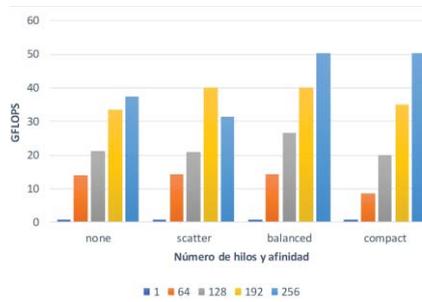


Fig. 3. Rendimientos obtenidos para diferentes tipos de afinidad y número de hilos cuando $N=65536$.

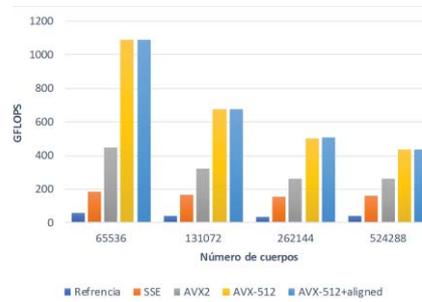


Fig. 4. Rendimientos obtenidos para diferentes niveles de vectorización al variar el número de cuerpos (N).

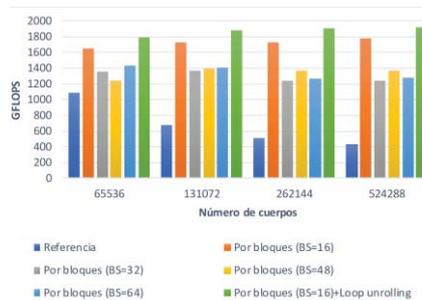


Fig. 5. Rendimientos obtenidos para la técnica de bloques y desenrollado de bucles al variar la carga de trabajo (N).

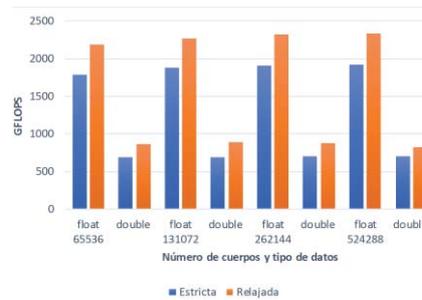


Fig. 6. Rendimientos obtenidos para la relajación de precisión al variar el tipo de dato y la carga de trabajo (N).

⁴ Debido a que los módulos de memoria DDR tienen diferente tamaño, no fue posible configurar el procesador en un modo diferente.

⁵ El número de pasos de simulación se mantuvo fijo ($I=100$).

5.2 Resultados de Rendimiento

Para evaluar el rendimiento se emplea la métrica GFLOPS (mil millones de FLOPS), utilizando la fórmula $GFLOPS = \frac{20 \times N^2 \times I}{T \times 10^9}$, donde N es el número de cuerpos, I es el número de pasos, T es el tiempo de ejecución (en segundos) y el factor 20 representa la cantidad de operaciones en punto flotante requerida por cada interacción⁶.

En la Fig. 3 se pueden ver los rendimientos al variar el tipo de afinidad y el número de hilos cuando $N=65536$. Al emplear paralelismo a nivel de hilos, el rendimiento mejora considerablemente, notando que un mayor número de hilos lleva a mejores prestaciones (excepto con afinidad *scatter*). Respecto a la afinidad, se puede apreciar que resulta conveniente elegir alguna de las estrategias disponibles en lugar de delegar la distribución en el sistema operativo (*none*). A diferencia de *scatter*, *balanced* y *compact* garantizan la contigüidad de hilos OpenMP con identificadores consecutivos, lo que favorece a la comunicación de los datos que cada uno requiere⁷.

Como se mencionó en la Sección 4.3, el compilador no es capaz de vectorizar todas las operaciones por su cuenta. Se puede notar en la Fig. 4 que, al forzar la vectorización de operaciones, se produce una mejora de aproximadamente 3.9×. Al agregar los flags `-xAVX2` y `-xMIC-AVX512`, el compilador emplea las instrucciones AVX2 y AVX-512, respectivamente. Como estas extensiones tienen mayor ancho vectorial, el rendimiento se incrementa aún más, logrando mejoras de 7.4× para AVX2 y de 15.1× para AVX-512. Por lo tanto, resulta claro que este problema se beneficia de instrucciones vectoriales más anchas. Adicionalmente, no se observan mejoras significativas por el alineamiento de datos a memoria.

Como se puede apreciar en la Fig. 5, la técnica de procesamiento por bloques aumenta considerablemente el rendimiento, siendo mayor la ganancia a medida que la cantidad de cuerpos crece. En particular, se logra una mejora promedio de 2.9× y una máxima de 4.1× ($BS=16$). En forma adicional, las prestaciones mejoran aproximadamente un 9% al desarrollar los bucles mencionados en la sección 4.5.

En la Fig. 6 se muestran los rendimientos obtenidos para la relajación de precisión al variar el tipo de dato y la carga de trabajo (N). Se puede notar que las prestaciones mejoran un 22% en promedio al aplicar la optimización del



Fig. 7. Rendimientos obtenidos para la explotación de MCDRAM al variar la carga de trabajo (N).

⁶ Convención ampliamente aceptada en la literatura disponible sobre este problema.

⁷ Como todos los núcleos del procesador se encuentran en el mismo paquete, *balanced* y *compact* producen la misma asignación cuando se emplean 4 hilos por núcleo.

compilador para este fin. En sentido contrario, el rendimiento decae aproximadamente un 60% al duplicar la precisión numérica usando el tipo de dato *double*.

Por último, a partir de la Fig.7, se puede notar que el rendimiento se mantiene (casi) constante al aumentar el número de cuerpos, obteniendo como máximo 2355 GFLOPS. También se puede apreciar una pequeña mejora cercana al 2% por el uso de la memoria MCDRAM. Este resultado es similar al observado en [10], estando relacionado con el hecho de que el rendimiento de esta aplicación se ve más influenciado por la latencia de la memoria que por el ancho de banda⁸.

6 Trabajos Relacionados

La aceleración de la simulación de N cuerpos ha sido ampliamente estudiada en la literatura. Sin embargo, pocos trabajos lo hacen sobre la arquitectura Xeon Phi, usando en su mayoría la primera generación de esta familia (KNC) [6] [11] [12]. Con respecto a KNL, sólo se puede mencionar el trabajo [10], el cual presenta algunas similitudes y diferencias con la presente investigación. Al igual que en este artículo, los autores estudian la paralelización de la versión directa de la simulación mostrando las mejoras obtenidas a través de diferentes optimizaciones, aunque la implementación final alcanza un pico de rendimiento mayor (2875 GFLOPS). A diferencia de esta investigación, el trabajo prioriza las optimizaciones para KNL, haciendo algunas simplificaciones en el cálculo de la simulación, como emplear un mecanismo de integración más sencillo (menos operaciones) y sólo computar un único paso en el tiempo. Al usar la misma métrica de evaluación⁹, sobreestiman la cantidad de FLOPS obtenidos. En relación con el análisis de prestaciones, en este artículo se consideraron varios aspectos adicionales como la cantidad y la afinidad de los hilos, el uso de todos los conjuntos de instrucciones vectoriales del KNL, la búsqueda del tamaño de bloque óptimo, el impacto del aumento de precisión por el uso del tipo *double*, además de la variación en la cantidad de cuerpos.

7 Conclusiones y Trabajos Futuros

Este trabajo se enfoca en la paralelización de la simulación de N cuerpos computacionales sobre un acelerador Intel Xeon Phi KNL. Comenzando por una implementación secuencial, se mostró cómo es posible que la implementación paralela alcance 2355 GFLOPS a través de diferentes técnicas de optimización. Entre las principales conclusiones de esta investigación se pueden mencionar:

- En general, un número mayor de hilos llevó a un mejor rendimiento. Respecto a la afinidad, se encontraron diferencias significativas entre las distintas opciones, por lo que es un factor que no se debe omitir al momento de ejecutar una aplicación paralela.

⁸ La latencia de la memoria DDR4 es similar a la de la MCDRAM.

⁹ También asumen que se realizan 20 operaciones de punto flotante por interacción pero el número real es inferior por usar un esquema de integración más simple (menos operaciones).

- La vectorización de operaciones representó un factor fundamental para la mejora de rendimiento. En ese sentido, se lograron aceleraciones cercanas al número de operaciones simultáneas que cada repertorio SIMD permite, a un bajo costo de programación.
- La explotación de la localidad de datos mediante el procesamiento por bloques resultó otro aspecto clave para obtener alto desempeño. No sólo permitió incrementar los FLOPS obtenidos en cada caso sino también que el rendimiento escale al aumentar la cantidad de cuerpos.
- Si la precisión en el resultado final no es una prioridad, el compilador puede ofrecer mejoras significativas en el rendimiento. En sentido contrario, duplicar la precisión puede reducir el rendimiento por debajo de la mitad.
- El uso de la memoria MCDRAM puede no proveer mejoras significativas en el rendimiento, especialmente cuando la aplicación no tiene una alta demanda de ancho de banda.

Entre los trabajos futuros, se pueden mencionar dos posibles líneas de investigación:

- Considerando los resultados obtenidos, se espera avanzar en la implementación de métodos avanzados para esta simulación.
- Dado que las GPUs son el acelerador dominante en la actualidad, interesa realizar una comparación de rendimiento y eficiencia energética entre estas arquitecturas.

Referencias

- [1] M. B. Giles y I. Reguly, «Trends in HPC for engineering calculations,» *Phil. Trans. of the Royal Soc. of London A: Mathematical, Physical and Engineering Sciences*, vol. 372, n° 2022, 2014.
- [2] J. Reinders, J. Jeffers y A. Sodani, Intel Xeon Phi Processor High Performance Programming Knights Landing Edition, Boston, MA, USA: Morgan Kaufmann , 2016.
- [3] G. Andrews, Foundations of Multithreaded, Parallel, and Distributed Programming, Addison Wesley, 2000.
- [4] P. L. Freddolino, C. B. Harrison, Y. Liu y K. Schulten, «Challenges in protein-folding simulations,» *Nature Physics*, vol. 6, pp. 751-758, 2010.
- [5] R. Goradia, «Global Illumination for Point Models,» 2008. [En línea]. Disponible en: <https://tinyurl.com/y5nqel39>.
- [6] R. Yokota y M. Abduljabbar, «N-Body methods,» de *High Performance Parallelism Pearls - Multicore and Many-core Programming Approaches*, Morgan-Kaufmann, 2015, pp. 175-183.
- [7] P. Tipler, Physics for Scientists and Engineers: Mechanics, Oscillations and Waves, Thermodynamics, Freeman & Co, 2004.
- [8] P. Young, «The leapfrog method and other "symplectic" algorithms for integrating Newton's laws of motion,» 21 04 2014. [En línea]. Disponible en:

- <https://young.physics.ucsc.edu/115/leapfrog.pdf>. [Último acceso: 23 07 2019].
- [9] E. Rucci, M. Naiouf y A. D. Giusti, «Blocked All-pairs Shortest Paths Algorithm on Intel Xeon Phi KNL Processor: A Case Study,» de *Computer Science - CACIC 2017*, La Plata, Argentina, Springer, 2018, pp. 47-57.
 - [10] A. Nikolaev y R. Asai, «N-Body simulation,» de *Intel Xeon Phi Processor HPC - KNL edition*, Morgan Kaufmann, 2016, p. 638.
 - [11] A. Vladimirov y V. Karpusenko, «Test-driving Intel Xeon Phi coprocessors with a basic N-Body simulation,» 2013. Reporte técnico Colfax Research International. Disponible en: <https://tinyurl.com/y5vtj34a>
 - [12] B. Lange y P. Fortin, «Parallel and dual tree traversal on multi-core and many-core architectures for astrophysical N-body simulations,» de *Euro-Par 2014 Parallel Processing. Lecture Notes in Computer Science, vol 8632*, 2014, pp. 716-727.



CACIC 2019
Congreso Argentino de Ciencias de la Computación

XVIII Workshop Tecnología Informática aplicada en Educación

Coordinadores

Gorga, Gladys (UNLP)
Malberti, Alejandra (UNSJ)
Russo, Claudia (UNNOBA)

Comité Científico

Baldasarri, Sandra (España)
Cukierman, Uriel (Argentina)
Depetriz, Beatriz (Argentina)
Gorga, Gladys (Argentina)

Lanzarini, Laura (Argentina)
Madoz, Cristina (Argentina)
Malberti, Alejandra (Argentina)
Malbran, María (Argentina)

Manresa Yee, Cristina (España)
Rueda, Sonia (Argentina)
Russo, Claudia (Argentina)
Sanz, Cecilia (Argentina)

Temas

Diseños y desarrollos de TICs orientadas a la Educación
Análisis de nuevas TICs para aplicar en la educación
Entornos y ambientes de soporte para Educación a Distancia
Hipermedias educativas
Laboratorios Virtuales
Simulación, Visualización y Realidad Virtual aplicadas en Educación
Aplicaciones educativas de la Inteligencia Artificial
Laboratorios Remotos
Métricas de Calidad de productos TICs aplicados a educación
Tecnología para ambientes e-learning
Tecnología para ambientes m-learning

Análisis de la experiencia de utilización del juego serio Desafiate para la autoevaluación de los alumnos

Federico Héctor Archuby¹, Cecilia Sanz^{1,2}, and Patricia Pesado¹

¹Instituto de Investigación en Informática LIDI
Facultad de Informática, Universidad Nacional de La Plata
La Plata, Argentina

²Investigador Asociado de la Comisión de Investigaciones Científicas de la Prov. de Buenos Aires

{farchuby, csanz, ppesado}@lidi.info.unlp.edu.ar

Abstract. En los últimos años ha proliferado el desarrollo y uso de juegos serios para el ámbito educativo. Uno de los aspectos que se vinculan con este fenómeno es la motivación que suele traer el uso de este tipo de herramienta. Por otra parte, otra tecnología que ha tenido un crecimiento desde ya hace tiempo son los entornos virtuales de enseñanza y aprendizaje (EVEA). Este tipo de software permite realizar distintas actividades, entre las que suele encontrarse la autoevaluación. En este artículo se presenta un análisis de la utilización del juego serio móvil Desafiate, creado por los autores, e integrado a un EVEA. El análisis se realiza sobre la motivación intrínseca que despierta el uso de este juego y se contrasta con la realización de la misma autoevaluación en lápiz y papel y en el EVEA IDEAS, pero sin el uso del juego. Para ello, durante la experiencia se ha trabajado con tres grupos, que realizaron la autoevaluación siguiendo las estrategias antes mencionadas. Como instrumento para medir la motivación intrínseca se utilizó el cuestionario IMI, que considera como variables que inciden en la motivación a: la utilidad, competencia percibida, presión/tensión, y esfuerzo durante la realización de la actividad, entre otras que no se contemplan en este trabajo. Los resultados muestran que los alumnos valoran a la utilidad de autoevaluarse y han demostrado mayor interés en este proceso cuando trabajan con el juego Desafiate.

Keywords: Uso de Juegos Serios, Educación, Autoevaluación, Dispositivos Móviles, Entornos Virtuales de Enseñanza y Aprendizaje

1 Introducción

Hoy en día las tecnologías de la información y la comunicación (TICs) se encuentran presentes en varios aspectos de la vida cotidiana y de la sociedad en general. El ámbito educativo a un ritmo más lento va integrando algunas herramientas e incluso para llevar adelante procesos de enseñar y aprender. Tal es el caso de

los entornos virtuales de enseñanza y aprendizaje (EVEA) que hoy son parte de la vida diaria de numerosas instituciones educativas. Los EVEAs son sistemas centrados en la web que permitieron que tanto alumnos como docentes puedan vivenciar procesos educativos mediados por tecnologías digitales. Esto se logra gracias a la integración de las distintas herramientas que se encuentran dentro de los EVEAs, lo que favorece además el seguimiento y trazabilidad de las actividades del alumno para un mejor acompañamiento y personalización (Cheng & Yuen, 2018; Ramírez-Correa, Rondan-Cataluña, Arenas-Gaitán, & Alfaro-Perez, 2017; Sanz, 2015; Cavus & Alhih, 2014). En la actualidad existen una gran cantidad de EVEAs, algunos incluso cuentan con gran difusión en el ámbito académico y otros han sido desarrollados ad-hoc por algunas universidades. Es posible observar experiencias en las que estos entornos sirven de complemento al trabajo realizado en las aulas, donde se abordan modalidades del tipo aula extendida, o modalidades que son híbridas y que combinan diferentes estrategias de mediación con la presencialidad. Entre las herramientas que se encuentran dentro de estos entornos, es frecuente encontrar algunas orientadas a la autoevaluación de los estudiantes. Éstas posibilitan crear distintos tipos de consignas (de opción múltiple, de dos opciones, de relación, etc). En este trabajo se aprovechan las funcionalidades de este tipo de herramientas de autoevaluación de los EVEAs para integrarlas a un juego.

Por otra parte, los juegos serios se han constituido como una alternativa actual en la planificación de actividades educativas que proponen los docentes para alcanzar cambios de comportamiento y actitud, y/o lograr determinados resultados de aprendizaje. Los juegos en general siempre han sido utilizados en la educación, y especialmente en el nivel inicial y primaria (Marcano Lárez, 2014). Es en el año 1970 que en Abt (1970) surge el concepto de juego serio. En este libro, se vincula a los juegos serios directamente con la educación, ya que se los define como aquellos que poseen un propósito educacional explícito y bien pensado. En el año 1970, cuando se acuña el término juego serio, no se orientaba a juegos digitales. Sin embargo, la popularidad que han ganado los videojuegos en los últimos años, y que se vio reflejada en un auge del uso de los videojuegos con fines educativos, hace que muchos autores utilicen esta terminología considerando solo a juegos digitales con un fin más allá del entretenimiento (Gallego et al., 2014). En consideración con las posibilidades de los juegos serios y las posibilidades de autoevaluación que ofrecen los EVEA, se desarrolló el juego serio móvil Desafiate, orientado a la autoevaluación de los alumnos.

En este artículo se presentan los resultados del análisis de lo ocurrido en una serie de sesiones en las que se integró Desafiate como estrategia de autoevaluación para los alumnos. La evaluación realizada se enfoca en la motivación intrínseca de los alumnos frente a una autoevaluación realizada siguiendo otras estrategias (lápiz y papel y EVEA). Se plantean dos hipótesis a analizar: 1- Realizar una autoevaluación mediante el uso del juego serio Desafiate mejora la motivación de los estudiantes y 2- Realizar una autoevaluación mediante el uso de un juego serio (Desafiate) reduce la presión que sienten los estudiantes durante ese proceso. Los resultados se analizarán para poder corroborar la validez de estas hipótesis para

la muestra con la que se ha trabajado. A partir de aquí el artículo se organiza de la siguiente manera: en la sección 2 se analizan los antecedentes relevantes para este trabajo. En la sección 3 se describe brevemente Desafiate, el juego serio implementado y utilizado en el marco de estas sesiones con alumnos. En la sección 4 se presentan la organización y descripción general de las sesiones con los alumnos, y las herramientas utilizadas para la medición y el análisis de los datos. En la sección 5 se analizan los resultados obtenidos y se discuten las hipótesis, y por último en la sección 6 se presentan las conclusiones y trabajos futuros.

2 Antecedentes

Como se pudo ver anteriormente, los juegos serios se encuentran vinculados a la educación desde la acuñación del término en 1970. A pesar de eso, el término ha ido cambiando y se ha expandido su significado. Por ejemplo, en Michael & Chen (2005) se da una definición de juego serios más amplia, al decir que son aquellos que se usan para educar, entrenar e informar. De esta forma, se agregan una relación con el entrenamiento y el dar información al usuario. Pero los juegos serios no se limitan solamente a esos objetivos. En Bione & Miceli (2017), se rescata la definición dada por Dörner, Göbel, Effelsberg, & Wiemeyer (2016) que dice que un juego serio es un juego digital creado con al menos un objetivo caracterizante, además del de entretener. Con esta definición se da a los juegos serios una característica más general, ya que no los limita solamente al ámbito educativo. A pesar de eso, existen numerosas investigaciones y experiencias que evalúan las ventajas del uso de este tipo de herramientas y su impacto en el ámbito educativo Chittaro & Buttussi (2015); Giannakos (2013); Rutten, van Joolingen, & van der Veen (2012); Vos, van der Meijden, & Denessen (2011).

Ya en (Abt, 1970) se nota que las personas se enfrentan a aprender las reglas y las dinámicas para poder jugar. En (Marcano Lárez, 2014) se afirma que los últimos avances en las ciencias cognitivas en general apoyan a los principios del aprendizaje que los juegos incorporan. Por ejemplo, destacan el carácter activo y el rol protagónico que adquiere el alumno en la resolución de problemas en tiempo real y con un *feedback* inmediato. En (Crawford, 2003) se explica que los videojuegos utilizan estrategias de diseño para generar ambientes virtuales que atrapen las capacidades perceptivas del usuario, que les produzcan gratificación sensorial y que generen la sensación de inmersión en la que el jugador “sienta” que puede “participar”. Y es mediante esta interactividad que se logran factores fundamentales en el proceso de aprendizaje: la atracción, la inmersión, la emoción y la motivación.

Teniendo en cuenta que los juegos serios pueden usarse para el aprendizaje, es necesario evaluar también cuáles son las ventajas que trae el uso de este tipo de herramienta en este sentido. Se han revisado algunas investigaciones que han explorado esto. En el trabajo de Chittaro & Buttussi (2015) se realiza una experiencia para aprender sobre las medidas de seguridad a tomar cuando ocurre un accidente de avión, en la cual se comprobó que el juego serio utilizado favoreció

el aprendizaje a más largo plazo. En el caso de la investigación de Castel de Haro et al. (2009), se desarrolló el juego serio PLMAN, para realizar prácticas de inteligencia artificial. Entre los resultados de esta investigación se menciona que se logró reducir la deserción de los estudiantes de la materia, de un 13% a un 9% en vinculación con el uso de estas herramientas y la motivación que generan. También es posible reducir la ansiedad en los alumnos, como puede verse en Kiili, Devlin, Perttula, Tuomi, & Lindstedt (2015), en donde se utiliza Semideus, un juego serio para evaluar los conocimientos de los estudiantes acerca de las fracciones. Otra ventaja que ha sido referida por diversos autores al integrar juegos serios en su propuesta educativa, es el aumentar el disfrute y la diversión de los estudiantes mientras realizan la experiencia. Un ejemplo, de esto se presenta en Giannakos (2013), que describen el uso del juego Gem-Game.

Como se puede ver existen diferentes ventajas de la aplicación de juegos serios. Sin embargo, la motivación es una ventaja que suele aparecer frecuentemente en distintas investigaciones (Kiili & Ketamo, 2018; Rutten et al., 2012; Boyle, Connolly, & Hainey, 2011; Vos et al., 2011), incluyendo algunas anteriormente citadas (Chittaro & Buttussi, 2015; Kiili et al., 2015; Giannakos, 2013; Castel de Haro et al., 2009). Esto es importante, ya que la motivación es uno de los factores fundamentales en el aprendizaje. Esto muestra que los juegos serios pueden ser una herramienta muy útil para ser considerados en la planificación didáctica, para su posterior integración.

Aún así, es importante notar que la mayoría de estas investigaciones se encuentran apuntadas al proceso de aprendizaje propiamente dicho, y solo algunas se enfocan específicamente en el proceso de evaluación de los alumnos, como la investigación de Kiili et al. (2015) o la de Castel de Haro et al. (2009). Si se considera que las evaluaciones suelen ser un punto de estrés para los estudiantes universitarios (Ferrel, 2012), la inclusión de juegos serios en esta instancia podría ser de interés.

Es por esto que en este trabajo se plantea que el uso de juegos serios en el proceso de evaluación y se considera que puede mejorar la motivación de los alumnos, y reducir la presión que sienten.

3 Presentación de Desafiate

Desafiate es un juego serio que se desarrolló específicamente para dispositivos móviles con el sistema operativo Android (al momento), y que se encuentra orientado a la autoevaluación de los alumnos. Se trata de un juego de preguntas y respuestas con la particularidad de que las preguntas se obtienen a partir de su integración con la herramienta de autoevaluación de un EVEA en particular.

En este juego el jugador personifica a un pirata que ha encontrado unos mapas de tesoros. Debido a esto, el pirata decide salir a navegar a través de un archipiélago de islas para comenzar su aventura. Esta aventura tiene el objetivo de recolectar todos los tesoros indicados en los mapas. Para lograrlo, en cada isla lo espera una persona que le plantea un desafío. Solo si el pirata resuelve correctamente este desafío podrá obtener el tesoro de dicha isla. La aventura

finaliza cuando el pirata resuelve los desafíos al recorrer las islas y vuelve a su puerto original, donde se muestran los resultados de la aventura.

Como se dijo anteriormente, Desafiate obtiene sus preguntas a través de la comunicación con un EVEA. Pero la comunicación no se limita solamente a esto. Varios elementos que componen Desafiate se encuentran relacionados con elementos del EVEA con el que se integra, en particular con aquellos que componen el módulo de autoevaluación. Para empezar, es necesario que el jugador inicie sesión en el juego, desde un dispositivo móvil, utilizando los datos de usuario que ya posee en el EVEA. Esta pantalla de inicio puede verse en la figura 1.



Fig. 1. Pantalla de inicio de sesión de Desafiate

Luego de esto, el jugador podrá ver diferentes aventuras a su disposición. Estas aventuras se corresponden con las autoevaluaciones que el estudiante posee en el EVEA. El estado de la autoevaluación, indica si el jugador puede realizar la aventura o solo ver sus resultados (en caso de ya estar resuelta). Cada uno de las preguntas que componen una autoevaluación, se relacionan con los desafíos que integran una aventura en Desafiate, y que se presentan en las islas. La nota que el estudiante tenga en la autoevaluación, dependerá de los resultados de los desafíos. En la figura 2 se muestran gráficamente estas relaciones, utilizando el EVEA IDEAS, el cual es el sistema con el que se encuentra relacionado actualmente Desafiate. Cabe aclarar que diseño del juego fue realizado en capas, y en la capa de comunicación con el EVEA podría desarrollarse la integración con otro que no sea IDEAS, sin afectar a la lógica del juego. Se puede encontrar más información acerca de este desarrollo en Archuby, Sanz, & Pesado (2017).



Fig. 2. La relación de Desafiate con IDEAS.

4 Metodología utilizada para las sesiones

El objetivo de esta investigación es evaluar si el juego Desafiate, utilizado para realizar una autoevaluación, impacta en la motivación de los estudiantes. Más concretamente, se plantean dos hipótesis diferentes:

H1: Realizar una autoevaluación mediante el uso del juego serio Desafiate mejora la motivación de los estudiantes.

H2: Realizar una autoevaluación mediante el uso de un juego serio (Desafiate) reduce la presión que sienten los estudiantes durante ese proceso.

Para saber si estas hipótesis se cumplen, se trabaja con grupos contrastados, donde se compara con la resolución de la misma autoevaluación en lápiz y papel, y *on-line* mediante el EVEA IDEAS.

Las sesiones se realizaron en el contexto de la materia Programación I de la carrera de Ingeniería en Computación. Dicha carrera pertenece a la Facultad de Informática de la Universidad Nacional de La Plata. Junto con los docentes de la materia se creó una autoevaluación en base a los contenidos que los estudiantes habían abordado hasta el momento. Todos los estudiantes que fueron a clase ese día, realizaron dicha autoevaluación. Se aprovechó el hecho de que la materia se encuentra dividida en tres comisiones, para asignar a cada comisión una estrategia de autoevaluación diferentes. Así, una comisión realizó la autoevaluación con lápiz y papel, otra comisión la realizó *on-line* usando el EVEA IDEAS, y la última a través del juego Desafiate en dispositivos móviles.

Para el caso de las comisiones que utilizaron medios electrónicos, fue necesario organizar las herramientas necesarias para poder realizar la prueba. Así, para los

estudiantes que utilizaron IDEAS, la prueba se realizó en una de las salas con PC, disponible en la propia institución. Fue necesario asegurar que dicha sala poseía acceso a internet y todos los alumnos podían acceder tanto a IDEAS, como a la autoevaluación. Para el caso de la comisión que utilizó Desafiate, se les proveyó un enlace de descarga para que puedan instalar el juego en sus celulares. Además, se prepararon *tablets* con el sistema operativo Android, por si era necesario darle a algún alumno.

Para la medición de la motivación de los estudiantes, se utilizó el instrumento *Intrinsic Motivation Inventory* (IMI), el cual analiza y mide diferentes dimensiones vinculadas a la motivación intrínseca. Este instrumento ha sido utilizado en las investigaciones de (McAuley, Duncan, & Tammen, 1989) y de (Tsigilis & Theodosiou, 2003). El instrumento consiste en la realización de un cuestionario con afirmaciones que dan un puntaje general en diferentes subescalas o variables de análisis, cada una relacionada a una dimensión. Cada afirmación se relaciona directamente a una variable de análisis particular, y es posible seleccionar solo algunas de las preguntas de cada variable. En total existen siete variables, pero en esta prueba solo se utilizaron cuatro: interés, presión, esfuerzo y utilidad.

La variable de análisis de interés es la más importante ya que es la que mide la motivación intrínseca propiamente dicha. La presión se considera un indicador negativo, por lo que es deseable obtener un resultado bajo en este punto. En el caso de la variable de esfuerzo, es deseable un resultado alto, ya que es indicador de que los participantes de la actividad se sintieron motivados a esforzarse más. Por último, la utilidad es un indicador de cuán útil para si mismo ve cada participante el realizar la actividad.

Para obtener un resultado en cada variable de análisis, es necesario evaluar el puntaje obtenido en cada respuesta. Se utilizó una escala de Lickert de 7, para indicar qué tan de acuerdo se encuentra una persona con respecto a la afirmación que se le da. En base a esto, se saca la media de los resultados obtenidos para cada variable. Existen algunas afirmaciones que su puntaje se obtiene a partir de hacer la resta de 8 menos el resultado de la respuesta, ya que las afirmaciones están negadas (reversa). Entre este tipo de afirmaciones y el cálculo del promedio de cada variable, se busca minimizar las inconsistencias que pueda haber entre las respuestas.

Para el análisis de los resultados se evalúan las variables de análisis de interés y presión por separado. Se considera para validar las hipótesis que la diferencia entre los grupos contrastados sea estadísticamente significativa. Esto se hará mediante un test de ANOVA entre los 3 grupos, y un test de Welch entre pares de grupos. Por otro lado, se hará un análisis general observando los resultados generales de cada una de las comisiones participantes. Esto se hará, obteniendo la media de cada una de las variables para cada una de las comisiones.

5 Resultados

De las sesiones participaron un total de 64 estudiantes pertenecientes a cada una de las tres comisiones que forman parte de la materia. La distribución de

los estudiantes fue la siguiente: 23 estudiantes realizaron la autoevaluación en formato de lápiz y papel, 19 lo hicieron a través de IDEAS, mientras que 22 estudiantes lo hicieron por medio de Desafiate. Luego de la autoevaluación se les dio a cada estudiante el cuestionario IMI conformado por 24 preguntas en total. De entre estos cuestionarios se utilizaron solo aquellos que fueron completados correctamente. De esta forma, se utilizaron en el procesamiento un total de 57 cuestionarios. Un cuestionario de la comisión que utilizó lápiz y papel no fue utilizado por poseer una respuesta ambigua con dos posibles valores. Mientras que 6 cuestionarios de la comisión que utilizó IDEAS, no fueron contestados en su totalidad, por lo que tampoco se los utilizó. En resumen, se utilizaron 22 cuestionarios correspondientes a la comisión que utilizó lápiz y papel, 13 correspondientes a la que utilizó IDEAS, y 22 que corresponden a la comisión de Desafiate.

5.1 Resultados obtenidos y discusión sobre las hipótesis

En esta subsección se presentan los resultados obtenidos.

Los resultados de las medias en cada una de las cuatro variables de análisis, se pueden ver en la tabla 1. Como se puede observar, el interés entre cada uno de los grupos es diferente, siendo Desafiate la estrategia de evaluación que genera más interés, e IDEAS el que menos lo hace. Si a esto lo combinamos con que éste fue el proceso de autoevaluación con el que menos presión sintieron los participantes, puede parecer que a grandes rasgos, que el uso de Desafiate es más motivante para los estudiantes. Sin embargo, más adelante se analizarán cada uno de estas variables de análisis por separados, para verificar si estas diferencias son lo suficientemente significativas para aceptarse las hipótesis.

Table 1. Tabla de medias de cada comisión de los cuestionarios realizados.

Método utilizado	Interés	Presión	Esfuerzo	Utilidad
Lápiz y papel	5.2208	3.2273	4.9818	6.4351
IDEAS	4.8132	2.9538	4.6769	6.1209
Desafiate	5.7532	2.5364	4.9545	6.2208

A la hora de analizar las siguientes dos variables de análisis, se puede notar que no hay diferencias entre los tres métodos utilizados (tres grupos contrastados). Si tomamos la utilidad, se puede observar que en todos los métodos se obtuvieron resultados muy altos. Esto indica que los alumnos en general, ven a

la autoevaluación como útil para su proceso de aprendizaje. Teniendo en cuenta el esfuerzo, en los tres métodos se obtuvieron valores altos que indican que los estudiantes consideran un esfuerzo la realización de la autoevaluación. Con estas dos variables, se puede notar la importancia que le dan los alumnos a la autoevaluación, ya que la consideran de mucha utilidad para su proceso de aprendizaje, más allá del esfuerzo que esto implica.

Retomando con el análisis del interés, es necesario realizar una comparación de los grupos para verificar si las diferencias son estadísticamente significativas. De esta forma, se podrá comprobar si se cumple la hipótesis H1 (realizar una autoevaluación mediante el uso del juego serio Desafiate mejora la motivación de los estudiantes). Primero se realizó un test de ANOVA para evaluar las diferencias entre los tres grupos con un nivel de significancia de 0,05. En este caso, el test arrojó como resultado un p-valor de 0,054378, dando como resultado que no se pueda comprobar la hipótesis. Analizando de a pares los grupos, utilizando un test de Welch, se puede observar que la diferencia entre el método de lápiz y papel no tiene diferencias significativas con respecto a los otros métodos ya que ningún resultado es inferior al 0,05. La diferencia de este método con Desafiate arroja un p-valor de 0,150223, mientras que la diferencia con IDEAS es de un p-valor 0,259359. Aún así la diferencia entre los métodos de Desafiate e IDEAS sí llegan a ser estadísticamente significativas ya que el p-valor entre estas poblaciones arroja un valor de 0,00375037. Con estos valores, no se puede afirmar la hipótesis H1, pero es posible que la conformación de los grupos haya influido en los resultados. Por lo pronto se puede afirmar que el uso de Desafiate para la autoevaluación, es de mayor interés que el uso de IDEAS. Esto abre las puertas para seguir indagando en el futuro en este aspecto, y para analizar otros factores que puedan influir en el experimento como la conformación de los grupos, la cercanía con el uso del EVEA, entre otras variables de entrada que pueden afectar.

Con respecto a la hipótesis H2 (realizar una autoevaluación mediante el uso de un juego serio reduce la presión que sienten los estudiantes), se obtuvieron los siguientes resultados. A la hora de aplicar el test ANOVA, se obtuvo un p-valor de 0,139781, el cual es superior al 0,05, necesario para confirmar la hipótesis. En el caso del test de Welch por pares, los resultados se encuentran en concordancia con los del test ANOVA ya que no se encontraron diferencias estadísticamente significativas entre pares. Teniendo en cuenta el método de lápiz y papel, éste arrojó un p-valor de 0,471610 al comparar su población con la de IDEAS, y un p-valor 0,0593150 al comparar con la población de Desafiate, lo que se podría acercar más a la aceptación de la hipótesis. Con respecto a la comparación entre IDEAS y Desafiate, ésta arrojó un p-valor de 0,262056. En este caso ningún resultado puede comprobar la hipótesis. Aún así, es necesario seguir indagando en la diferencia entre la presión que siente el alumno al utilizar lápiz y papel y un juego serio, a la hora de realizar una autoevaluación, a partir de entrevistas con los alumnos que ayuden a profundizar en estos resultados.

En un análisis más general, se puede ver como el uso de juegos serios en el proceso de autoevaluación, ha mostrado incidir en el interés y la presión que

sienten los alumnos al realizar una autoevaluación. Aún así, los resultados de aplicar el test de ANOVA y el test de Welch, muestran que es necesario seguir indagando sobre estos aspectos, y en lo posible con una población mayor. En el caso de las dos hipótesis, ninguna puede ser aceptada, pero abren nuevas preguntas para seguir investigando en el futuro.

6 Conclusión

Los juegos serios son una de las herramientas que han proliferado gracias al uso de TICs en Educación. Si bien cuentan con varias ventajas, la motivación suele ser la que más comúnmente se encuentra. En este artículo se realizó una comparación entre tres métodos diferentes a la hora de realizar una autoevaluación. Se utilizó el cuestionario IMI para medir el interés, la presión, el esfuerzo y la utilidad percibida por los alumnos a la hora de realizar esta actividad. Este cuestionario permite medir la motivación intrínseca de los participantes de una actividad.

Los resultados muestran que los estudiantes en general perciben a la autoevaluación como una herramienta útil para su proceso de aprendizaje. Esto se traduce en que se esfuerzan más en realizar la actividad. Teniendo en cuenta el interés y la presión, se plantearon dos hipótesis diferentes. Por un lado, la hipótesis H1 planteaba que realizar una autoevaluación mediante el uso de un juego serio (Desafiate en este caso) mejora la motivación de los estudiantes. Luego, se encontraba la hipótesis H2, la cual planteaba que realizar una autoevaluación mediante el uso del juego serio Desafiate reduce la presión que sienten los estudiantes. Si bien el resultado general muestra que hay mejoras en cuanto a estos niveles cuando se usa el juego serio, los test realizados mostraron que estos no eran estadísticamente significativos como para confirmar las hipótesis. Aún así, a la hora de analizar por pares, se encontraron diferencias significativas entre el uso de Desafiate, y el uso del EVEA IDEAS.

Estos resultados también mostraron la necesidad de seguir indagando sobre esta cuestión. Se abre así la puerta para continuar con la investigación con una mayor población y considerando la incidencia de algunas variables de entrada como la conformación de los grupos, género, experiencia previa en el uso de del EVEA, entre otros.

Referencias

- Abt, C. (1970). *Serious games*. Viking Press.
- Archuby, F. H., Sanz, C. V., & Pesado, P. M. (2017). Desafiate: juego serio para la autoevaluación de los alumnos y su integración con un entorno virtual de enseñanza y aprendizaje. , 23.
- Bione, J., & Miceli, P. (2017). *Astrocódigo* (Unpublished doctoral dissertation). Facultad de Informática.
- Boyle, E., Connolly, T. M., & Hailey, T. (2011). The role of psychology in understanding the impact of computer games. *Entertainment Computing*, 2, 69-74.
- Castel de Haro, M. J., Durán, F. J., Pomares Puig, C., Suau Pérez, P., Villagrà, C., & Cortés Vaíllo, S. (2009, 07). Evaluación en tiempo real (comunicación). *Lógica Computacional*.
- Cavus, N., & Alhih, M. S. (2014). Learning management systems use in science education. *Procedia - Social and Behavioral Sciences*, 143, 517 - 520. Retrieved from <http://www.sciencedirect.com/science/article/pii/S1877042814043572> (3rd Cyprus International Conference on Educational Research, CY-ICER 2014, 30 January – 1 February 2014, Lefkosa, North Cyprus) doi: <https://doi.org/10.1016/j.sbspro.2014.07.429>
- Cheng, M., & Yuen, A. H. K. (2018). Student continuance of learning management system use: A longitudinal exploration. *Computers & Education*, 120, 241 - 253. Retrieved from <http://www.sciencedirect.com/science/article/pii/S036013151830037X> doi: <https://doi.org/10.1016/j.compedu.2018.02.004>
- Chittaro, L., & Buttussi, F. (2015, April). Assessing knowledge retention of an immersive serious game vs. a traditional education method in aviation safety. *IEEE Transactions on Visualization and Computer Graphics*, 21(4), 529-538. doi: 10.1109/TVCG.2015.23918532
- Crawford, C. (2003). *On game design*. New Riders.
- Dörner, R., Göbel, S., Effelsberg, W., & Wiemeyer, J. (2016). *Serious games: Foundations, concepts and practice*.
- Ferrel, G. (2012). A view of the assessment and feedback landscape: baseline analysis of policy and practice from the jisc assessment & feedback programme. *JISC*.
- Gallego, F. J., Villagrà, C. J., Satorre, R., Compañ, P., Molina, R., & Llorens Largo, F. (2014). Panoràmica: serious games, gamification y mucho más. *Revisión*, 7(2).
- Giannakos, M. (2013). Enjoy and learn with educational games: Examining factors affecting learning performance. *Computers & Education*, 68, 429 - 439.
- Kiili, K., Devlin, K., Perttula, A., Tuomi, P., & Lindstedt, A. (2015). Using video games to combine learning and assessment in mathematics education. *International Journal of Serious Games*, 2(4). doi: 10.17083/ijsg.v2i4.98

- Kiili, K., & Ketamo, H. (2018). Evaluating Cognitive and Affective Outcomes of a Digital Game-Based Math Test. *IEEE Transactions on Learning Technologies*, *11*(2), 255–263. doi: 10.1109/TLT.2017.2687458
- Marcano Lárez, B. E. (2014). *Factores emocionales en el diseño y la ejecución de videojuegos y su valor formativo en la sociedad digital.: El caso de los videojuegos bélicos* (1st ed.). Ediciones Universidad de Salamanca.
- McAuley, E., Duncan, T., & Tammen, V. V. (1989). Psychometric properties of the intrinsic motivation inventory in a competitive sport setting: A confirmatory factor analysis. *Research quarterly for exercise and sport*, *60*(1), 48–58.
- Michael, D., & Chen, S. (2005). *Serious games: Games that educate, train, and inform*. Thomson Course Technology.
- Ramírez-Correa, P. E., Rondan-Cataluña, F. J., Arenas-Gaitán, J., & Alfaro-Perez, J. L. (2017). Moderating effect of learning styles on a learning management system's success. *Telematics and Informatics*, *34*(1), 272 - 286. Retrieved from <http://www.sciencedirect.com/science/article/pii/S0736585316301241> doi: <https://doi.org/10.1016/j.tele.2016.04.006>
- Rutten, N., van Joolingen, W. R., & van der Veen, J. T. (2012). The learning effects of computer simulations in science education. *Computers & Education*, *58*(1), 136 - 153. Retrieved from <http://www.sciencedirect.com/science/article/pii/S0360131511001758> doi: <https://doi.org/10.1016/j.compedu.2011.07.017>
- Sanz, C. V. (2015, Junio). Los objetos de aprendizaje, un debate abierto y necesario. *Bit & Byte*, *1*(1), 33-35.
- Tsigilis, N., & Theodosiou, A. (2003). Temporal stability of the intrinsic motivation inventory. *Perceptual and motor skills*, *97*(1), 271–280.
- Vos, N., van der Meijden, H., & Denessen, E. (2011). Effects of constructing versus playing an educational game on student motivation and deep learning strategy use. *Computers & Education*, *56*(1), 127 - 137. Retrieved from <http://www.sciencedirect.com/science/article/pii/S0360131510002344> (Serious Games) doi: <https://doi.org/10.1016/j.compedu.2010.08.013>

AppCase4Blind: una herramienta para asistir a personas no videntes en el proceso de enseñanza-aprendizaje de Grafos

Marcelo Uva, Marcela Daniele y Dario Astorga
Universidad Nacional de Río Cuarto
{uva,marcela,astorgad}@dc.exa.unrc.edu.ar

Resumen. La incorporación, integración y aprovechamiento pedagógico de las Tecnologías de la Información y Comunicación (TIC) en del Sistema Educativo, se ha convertido en una necesidad esencial para lograr aprendizajes significativos. Las TIC se han transformado en una herramienta de apoyo para la inclusión e integración de personas con discapacidad. Se estima que en el mundo, existen al menos seis millones de personas con algún tipo de discapacidad visual. La Universidad Pública debe asegurar el acceso de los conocimientos a todos aquellos que tomen la decisión de formarse en ella. En este trabajo se presenta una propuesta didáctica para la enseñanza de Grafos, con el fin de integrar estudiantes no videntes o disminuidos visualmente con el resto de los actores intervinientes en el proceso de enseñanza-aprendizaje. Como resultado se presenta una aplicación móvil atendiendo fundamentalmente a reducir las barreras de accesibilidad presentes en estudiantes ciegos o disminuidos visualmente.

Palabras Clave: TIC, Accesibilidad, Integración e Inclusión.

1 Introducción

Durante la última década, el uso cada vez más generalizado de las Tecnologías de la Información y Comunicación (TIC) han sido determinantes en múltiples aspectos de la sociedad actual. La economía, las relaciones personales, la forma en que se desarrollan los procesos de enseñanza y de aprendizaje, inevitablemente han sido atravesados y modificados por estos cambios tecnológicos. La incorporación, integración y aprovechamiento pedagógico de las TIC dentro del Sistema Educativo, se ha convertido en una necesidad esencial a fin de lograr aprendizajes significativos en el marco de la sociedad de conocimiento. Programas tales como Conectar Igualdad [1], se implementaron buscando reducir las brechas sociales, digitales y educativas, a fin de garantizar la igualdad de oportunidades de acceso y uso de estas tecnologías, propiciando mayores y mejores accesos a la información y al conocimiento [2]. En este

mismo sentido, y a fin de favorecer la igualdad de oportunidades a todos los estudiantes, las TIC se transforman en un herramienta de apoyo para la inclusión e integración de personas con discapacidad. En un documento entregado en el marco de la Conferencia Mundial de Telecomunicaciones Internacionales, la Unión Internacional de Telecomunicaciones (UIT) [3], órgano dependiente de las Naciones Unidas, expresa que “para poder alcanzar una sociedad integradora, todos deben ser capaces de usar las TIC con total confianza, lo que significa que la mejora de la “accesibilidad” a las TIC[4] debe ser un punto fundamental en el orden del día de los proveedores de servicios y equipos”. Se estima que en el mundo, existen al menos seis millones de personas con algún tipo de discapacidad visual que les impide utilizar las pantallas de las computadoras tradicionales. Muchos de los usuarios con discapacidad visual utilizan tecnología de asistencia, un término que hace referencia al hardware y software diseñado para facilitar el uso de computadoras por parte de personas con discapacidad. La tecnología de asistencia para discapacitados visuales incluye dispositivos tales como lectores de pantalla, tamaños de fuente ajustables, reconocimiento de voz, pantalla braille, etc. Dentro de este marco, la Universidad Pública tiene la necesidad y la obligación de asegurar el acceso de los conocimientos a todas aquellas personas que tomen la decisión de formarse en ella [2,5]. Esto implica, adaptar contenidos, materiales y metodologías de enseñanza y de aprendizaje, entre otras cuestiones.

Las carreras de Analista en Computación y Licenciatura en Ciencias de la Computación de la Universidad Nacional de Río Cuarto, han tenido estudiantes con disminución visual severa que han culminado sus estudios y son graduados de la Universidad Pública. Actualmente una estudiante ciega está cursando el primer año de la carrera de Analista de Computación, convirtiéndose en la principal motivación para que los docentes desarrollen nuevas estrategias y modalidades para la enseñanza de los contenidos de cada asignatura.

En este trabajo, se plantea una propuesta didáctica para la enseñanza de Grafos [7], siendo éste un contenido que es abordado en más de una asignatura de la currícula de las carreras de computación, con el principal objetivo de integrar estudiantes ciegos, o disminuidos visualmente, con el resto de los actores intervinientes en el proceso de enseñanza y de aprendizaje. Para el desarrollo de esta propuesta, se investigaron y estudiaron diversas propuestas pedagógicas y herramientas que abordan la enseñanza de diferentes temas para personas ciegas. Como resultado de esta propuesta, se ha desarrollado AppCase4Blind, una aplicación móvil para la enseñanza de grafos que aporta, fundamentalmente, a la reducción de barreras de accesibilidad[8] presentes en estudiantes ciegos.

El presente trabajo está organizado de la siguiente manera: en la sección 2 se realiza una revisión de trabajos vinculados al desarrollo de herramientas tecnológicas orientadas específicamente a integrar e incluir a personas ciegas, la sección 3 y 4 se presenta la propuesta y la herramienta AppCase4Blind que la implementa. En la sección 5 se

presentan algunas pruebas realizadas a la herramienta y observaciones por parte de los usuarios y finalmente, en la sección 6 se presentan conclusiones y trabajos futuros.

2. Trabajos relacionados / Antecedentes

El término accesibilidad se utiliza para describir el grado en que un producto, dispositivo, servicio o ambiente está disponible para los usuarios[9]. Dentro de las características deseadas se requiere que una herramienta sea:

- Perceptible: la información y los componentes de la Interfaz de Usuario (IU) deben presentarse de manera sencilla y fácil de percibir,
- Operable: los componentes de la IU y la navegación deben utilizarse de forma adecuada,
- Entendible: la información y operación de la IU debe ser comprensible,
- Robusto: el contenido debe ser interpretado de manera confiable por una amplia variedad de usuarios.

Dentro del ámbito académico no se han podido encontrar herramientas de las características que se proponen en este trabajo. En la actualidad, la mayoría de los sistemas operativos masivos tales como Android poseen funcionalidades de accesibilidad como por ejemplo: TalkBack que proporciona mensajes de voz a usuarios ciegos o con baja visión indicando qué función o botón está activo. Configuración de tamaño de fuentes, de ampliación y alejamiento de imágenes, entre otras. Estas características facilitan el acceso a aplicaciones (en especial en ambientes móviles), a pesar de ello, el uso de TIC aplicadas al desarrollo de herramientas para la integración e inclusión de personas con discapacidad aún no ha sido explotada. En este sentido coincidimos con lo que expresa el Ingeniero Luis Campos, investigador del Instituto de Neurociencias de Guadalajara, México, profesor de la Universidad Nacional de Chilecito, La Rioja y Director del Instituto de Tecnología para la Inclusión de la Universidad de Morón, Provincia de Buenos Aires. “El mundo de los discapacitados es individual. Cada uno de ellos tiene distintas características que les permiten hacer ciertas cosas o no. Es un mundo en donde no se puede generalizar. Cada computadora tiene que ser adaptada a esa persona que la va a utilizar. No se puede diseñar algo para ‘las personas con parálisis cerebral’ porque cada uno tiene distintos grados. Para el mercado sería muy caro hacer tecnología especial y se las dan a todos por igual. Es como que a todos los chicos de un aula les des zapatillas de calzado 33”.

Mencionaremos a continuación algunas aplicaciones informáticas que proponen contribuciones, para abordar problemas que no tienen que ver con la enseñanza y el aprendizaje, sino con ayudar personas. Esta no pretende ser una lista acabada de las herramientas disponibles, sino que son algunas de las que a criterio de los autores son relevantes y se vinculan con la propuesta.

Lazzus [10], es un asistente que acompaña a las personas ciegas y con discapacidad visual en sus desplazamientos diarios, creando un campo de visión auditivo. Proporciona información relevante del entorno como pasos de peatones, cruces entre calles, escaleras, establecimientos, etc. ayudando a caminar de una manera más segura y explorando de forma intuitiva lo que nos rodea. Lazzus es una aplicación paga.

What's In My Wallet? [11], Es una aplicación que permite escanear distintos billetes y tarjetas, con el fin de que los usuarios no videntes puedan realizar transacciones de manera más fácil y fluida. Lucila Lang, de 25 años, responsable del proyecto, se inspiró en un compañero de Universidad no vidente quien le señaló que a pesar de haber logrado bastante independencia y desarrollo en su día a día, el tema de la billetera era algo que aún lo limitaba, al no poder reconocer el valor de cada billete o la tarjeta que estaba usando. La aplicación, desarrollada en Android Studio, funciona en base al escaneo de los distintos billetes y tarjetas, utilizando la librería de procesamientos de imágenes.

Be My Eyes [12]. Aplicación informática colaborativa. Esta app está integrada por personas no videntes como también personas videntes, las cuales de manera gratuita y sin ningún tipo de fin de lucro pueden ser sus lazarillos informáticos, por medio del celular, asistiendo a las personas no videntes en diversas ocasiones, más precisamente cuando ellos lo necesiten.

AGS-R4. [13] es una aplicación desarrollada íntegramente en Río Cuarto por estudiante de la carrera de Lic. en Ciencias de la Computación, para dispositivos que cuenten con el sistema operativo Android, la cual nos permite registrar destinos visitados con anterioridad para volver a visitarlos guiados por la App, como también nos brinda información de nuevos destinos para visitar, haciendo uso de Internet, GPS y TalkBack Activado. AGS-R4 es gratuita y puede ser descargada desde el centro de software Google Play Store, para dispositivos móviles Android.

3. Descripción de la propuesta

Este trabajo fue motivado ante la necesidad de integrar a una estudiante ciega perteneciente al primer año de la carrera de Analista en Computación de la UNRC. Los

docentes actuales de la estudiante junto con aquellos que en el transcurso de la carrera lo serían se plantearon la inquietud de cómo favorecer el proceso de enseñanza-aprendizaje de la estudiante. Los docentes de las asignaturas de Análisis y Diseño de Software e Ingeniería de Software junto con un estudiante de grado, el cual estaba interesado en la temática y deseaba realizar su trabajo de tesis de Licenciatura iniciaron un trabajo de investigación junto con el Centro de Producción de Información Accesible (CEPIA), ente perteneciente a la UNRC. El objetivo propuesto fue el desarrollo una herramienta CASE (Computer Aided Software Engineering, Ingeniería de Software Asistida por Computadora) soportada en una aplicación móvil para trabajar una temática particular que pudiera ser extensible y extrapolada a otros campos [6]. El tema seleccionado fue Grafos y el enfoque fue dirigido a integrar todos los actores: estudiante, pares y docentes.

3.1 Grafos

Un grafo es un conjunto de objetos llamados nodos o vértices, que pueden estar unidos (conectados) por líneas llamada aristas[7]. Los grafos definen estructuras de datos que permiten modelar una infinidad de problemas expresables de manera computacional. Problemas como representación de computadoras conectadas a una red, relaciones entre personas pertenecientes a una red social, mapas de distribución de productos a través de rutas, diagramas de clases, etc. son algunas de las situaciones representables por este tipo abstracto de datos. La teoría de grafos analiza propiedades presentes en estas estructuras y define algoritmos que permiten realizar cálculos sobre las mismas.

3.2. Requerimientos funcionalidades

Para el desarrollo de la aplicación se establecieron una serie de requerimientos funcionales definidos por el equipo docente. En este sentido, podríamos decir que los requisitos funcionales estaban claramente definidos desde un comienzo, a diferencia de los requisitos no funcionales ya que estos últimos exigían características esenciales de accesibilidad [14,16]. Las funcionalidades a desarrollar deberían permitir: crear, editar, eliminar y recorrer grafos, lo cual implicaría la definición de nodos y arcos o relaciones con sus respectivos atributos. También planteó la necesidad de incorporar funcionalidades de exportación e importación de grafos. Esto posibilitaría el intercambio de modelos entre los estudiantes (videntes y no videntes) y docentes. La herramienta CASE a la cual denominamos APP-Case4Blind debía implementarse para ejecutarse en un teléfono celular con sistema operativo Android. Desde un comienzo se pensó en que APP-Case4Blind debía requerir lo mínimo con respecto a hardware y software a fin de que pudiese ser utilizada en la mayor cantidad de dispositivos posible. APP-Case4Blind

no pretende ser una herramienta compleja en cantidad de funcionalidades, sino que más bien todo lo contrario. El objetivo central es proveer una herramienta minimalista que permita trabajar con grafos mediante una interfaz de usuario accesible para personas ciegas, estudiantes en general y docentes. A través de APP-Case4Blind los estudiantes podrán modelar soluciones a situaciones problemáticas mediante la definición de grafos. Éstas podrán ser enviadas y analizadas por el docente, el cual realizará una devolución mediante la misma aplicación. APP-Case4Blind deberá contar con un módulo de configuración que posibilite activar/desactivar el modo lectura/vibración de la pantalla además de otras características las cuales pueden ser vitales para poder interactuar con personas ciegas.

3.2. Requerimientos no funcionales

Los requerimientos no funcionales fueron definidos por el equipo docente en colaboración con especialistas de CEPIA y junto con la estudiante no vidente. Se tomó la decisión de incorporar al proyecto a la estudiante no vidente debido a los aportes que ella podía realizar por su experiencia en el uso de aplicaciones móviles. Luego de una serie de reuniones se llegaron a las siguientes definiciones:

- La aplicación se desarrollaría para Android debido a la masividad y características del sistema operativo. Tanto la estudiante ciega, como el resto de los alumnos, en su gran mayoría, utilizan este sistema operativo en sus teléfonos smartphones.

Con respecto a la interfaz de usuario:

- El número máximo de botones por pantalla no debería superar los seis, para evitar un scroll excesivo.
- La disposición de los botones en la pantalla deberían ocupar el ancho total del dispositivo cuando este se encuentra en posición vertical u horizontal, para agilizar la búsqueda táctil.
- Se pudo determinar que no resultaría beneficioso integrar una barra de desplazamiento vertical por no resultar práctica de manipular por su dimensión y posicionamiento en pantalla.
- Cada una de las pantallas deberá poseer un título descriptivo, claro y conciso.
- El tamaño de letra para un dispositivo con pantalla de 4.5" no debiera ser menor a 16px, facilitando de esta manera la lectura.

- El posicionamiento del botón "Volver Atrás" (para la navegación) , debería ubicarse por debajo de los demás sin importar en qué pantalla nos encontremos trabajando. Las personas ciegas o con visión reducida apelan principalmente a la memoria.
- Para la creación, edición, eliminación de un grafo, debería definirse un esquema o dinámica de navegación similar, en los que respecta a secuenciamiento de operaciones y disposición de botones en la pantalla.
- La búsqueda de un grafo creado con anterioridad debería poder realizarse y visualizarse de manera cronológica.
- Es de vital importancia la utilización del lector de pantalla [17] TalkBack (lector de pantalla de Google incluido en los dispositivos Android) en las funcionalidades implementadas.

Como se expuso en la sección anterior, los requerimientos funcionales de la aplicación fueron más sencillos de definir que los no funcionales. La lista anterior proporciona algunas de las características más interesantes de la propuesta debido a que recopilan las necesidades de todos los actores involucrados.

4. Arquitectura y diseño de AppCase-4Blind

La arquitectura de la aplicación fue definida a través del patrón arquitectural Modelo-Vista-Controlador (MVC), el cual separa los datos, la lógica de negocio de una aplicación de la interfaz de usuario y el controlador encargado de gestionar los eventos y las comunicaciones. AppCase4Blind está estructurada en bloques de construcción Android [15,18]. Este tipo de componentes básicos son utilizados por desarrolladores de aplicaciones en otros lenguajes, tales como Java o .NET. A partir de estos componentes los desarrolladores definen ventanas, controladores, eventos o servicios. A continuación se presentan algunos de los componentes Android utilizados para definir AppCase4Blind:

Actividad (Activity): Una actividad normalmente es una pantalla de la aplicación que se visualiza en el dispositivo móvil, siendo ésta el componente principal de la interfaz gráfica de una aplicación Android. Una aplicación típicamente cuenta con múltiples actividades, donde el usuario puede navegar. Toda actividad hereda de una clase principal Activity.

Intent: Los intents son mensajes o peticiones que son enviados entre los bloques de construcción. Estos disparan el inicio de una actividad, pasan información de una actividad a otra, inician o detienen un servicio o bien simplemente juegan el papel de "broadcast". Un intent es asíncrono, lo cual significa que el código que lo mandó no

necesariamente tiene que esperar a que la acción del intent se ejecute. Este tipo de bloque de construcción puede ser explícito o implícito.

Vista (View) :Las vistas son los elementos que componen la interfaz de usuario de una aplicación. Por ejemplo, un botón, una entrada de texto, entre otros. Todas las vistas van a ser objetos dependientes de la clase View, y por tanto pueden ser definidos utilizando código Java. Sin embargo, lo habitual va a ser definir las vistas utilizando un fichero XML y dejar que el sistema cree los objetos por nosotros a partir de ese fichero.

Layout: Un layout es un conjunto de vistas agrupadas de una determinada forma. Se va a disponer de diferentes tipos de layouts para organizar las vistas de forma lineal, en cuadrícula o indicando la posición absoluta de cada vista.

En Android el Controlador es definido mediante una actividad. Para el manejo de los diferentes eventos lanzados por las entradas realizadas por el usuario, Android provee la interfaz “EventListener” de la clase “View”, dicha interfaz cuenta con un método callback, que será invocado por Android al momento de producirse una acción determinada. La figura 1 muestra el esquema de instanciación del patrón MVC para la funcionalidad “crear grafos”. En la figura 1 se presenta la interfaz inicial de la aplicación la cual implementa la View de “Crear grafos” de la figura 1.



Figura 1 - Interfaz inicial de la AppCase4-Blind

5. Implementación y pruebas de funcionamiento

Para evaluar APPCASE-4Blind se llevaron a cabo múltiples pruebas dentro del emulador Android Studio AVD (Android Virtual Device) para determinar su correcto

funcionamiento en distintas versiones del sistema Android. Del mismo modo se efectuaron pruebas periódicas el emulador de Android Genymotion . AppCase4Blind fue sometida al test de accesibilidad de Google. Este último escanea cada pantalla activa y sugiere mejoras en la accesibilidad de la aplicación. Para ello, se tienen en cuenta: etiquetas de contenido, tamaño de los elementos táctiles, elementos en los que se puede hacer clic y contraste de texto e imagen. Los resultados de las pruebas realizadas a AppCase-4Blind han sido todas satisfactorias, lo que proporciona un alto grado de confianza en la accesibilidad de la herramienta. Los especialistas representantes de CEPIA junto con la estudiante ciega expresaron sobre la herramienta una inmediata aceptación, de fácil descarga e instalación, intuitiva y perceptible.

6. Conclusiones y trabajos futuros

AppCase-4Blind es una herramienta perteneciente a las TIC destinada a colaborar en el proceso de enseñanza-aprendizaje de grafos dirigidos. Es una herramienta simple, versátil y útil para lograr integrar a todos los actores del proceso de enseñanza-aprendizaje, especialmente a estudiantes con disminución visual o ceguera. AppCase-4Blind fue diseñado para proveer una didáctica dinámica, interactiva, táctil y auditiva que permite crear grafos dirigidos, facilitando la enseñanza de diversos contenidos curriculares en asignaturas de las carreras de computación. Su distribución es libre y gratuita. La aplicación fue desarrollada para ejecutarse en el sistema operativo Android abarcando una gran variedad de dispositivos móviles y tablets accesibles hoy en día por la mayoría de la población estudiantil. La arquitectura de la herramienta fue pensada de forma modular para poder ser extendida fácilmente, ampliando su campo de aplicación a otras temáticas y a otras áreas de enseñanza. Nuestros aportes han sido dirigidos a generar dentro de las TIC herramientas destinadas a reducir las barreras de accesibilidad presentes en estudiantes con discapacidad. AppCASE-4Blind es una consecuencia del trabajo de investigación multidisciplinario llevado adelante. Como trabajos futuros se plantea el desarrollo de un framework configurable que permita el desarrollo de este tipo de aplicaciones en otras temáticas, aplicables a otras carreras e incluso, posible de utilizar en distintos niveles educativos.

Bibliografía

- [1] Programa Conectar Igualdad .
<https://www.argentina.gob.ar/educacion/aprender-conectados/conectar-igualdad>
- [2] Julio Cabero Almenara. "Las TIC y las universidades: retos, posibilidades y preocupaciones".
Rev. educ. sup vol.34 no.135 México jul./sep. 2005

- [3] Unión Internacional de Telecomunicaciones (UIT). <https://www.itu.int/es>
- [4] Accesibilidad de Aplicaciones Móviles, "Accesibilidad en Aplicaciones Móviles", Observatorio Accesibilidad TIC disc@pnet y fundación ONCE. Versión detallada. Agosto 2013. discapnet.es Pág. 6-11, 108-109, 109-111
- [5] Julio Cabero Almenara. "TICs para la igualdad: la brecha digital en la discapacidad." ANALES de la Universidad Metropolitana- Universidad de Sevilla (España – UE). <http://tecnologiaedu.us.es>
- [6] Paul Blenkhorn y David Gareth Evans, "A Method of Access to Computer Aided Software Engineering (CASE) Tools for Blind Software Engineers", Pág.321-328, M60 1QD, England. 08 Junio de 2005. Springer.com
- [7] Introduction to Graph Theory, Robin J. Wilson "Introduction, Definitions and examples, Paths and cycles", Introduction to Graph Theory, Fourth Edition. England, Edinburgh Gate, Harlow, Essex CM20 2JE. Cuarta edición 1996. Pág. 1- 7, 8-21, 26-38.
- [8] Pablo Santana-Mansilla, Germán Lescano, y Rosanna Costaguta, "Accesibilidad de aplicaciones móviles para discapacitados visuales: problemas y estrategias de solución." (Simposio Argentino sobre Tecnología y Sociedad). Facultad de Ciencias Exactas y Tecnologías (FCEyT) Universidad Nacional de Santiago del Estero (UNSE). Pág. 321-328 unlp.edu.ar
- [9] Stella Maris Massa, Silvia Vassolo, Hernán Fino, María Dolores Finochietto y Luciano Wehrli "Aplicaciones accesibles para dispositivos móviles: diseño e implementación", Pág. 1-2-3, Último acceso 12 de Julio de 2018. Sedici.unlp.edu.ar
- [10] Lazzus app. <http://www.lazzus.com/es/>
- [11] What's in my Wallet? app. https://play.google.com/store/apps/details?id=com.uns.moneyreader&hl=es_AR
- [12] Be My Eyes <https://www.bemyeyes.com/>
- [13] AGS-R4 <https://desarrollarinclusion.cilsa.org/ags-r4/>
- [14] Moreno López, L. (2010). L. (2010) Marco metodológico específico en el dominio de la accesibilidad para el desarrollo de aplicaciones web. Tesis Doctoral. Universidad Carlos III de Madrid. Departamento de Informática
- [15] Jesús Tomás Vicente Carbonell Carsten Vogt Miguel García Pineda Jordi Bataller Mascarell Daniel Ferri, "El gran libro de Android Avanzado, Diseño personalizado de vistas, Aplicaciones web en Android". Barcelona. Marcombo, S.A, Primera Edición, 2013. Pág. 29-47, 98-121
- [16] Instituto Nacional de Tecnologías de la comunicación (INTECO). (2009) Accesibilidad en contenidos Web accesibles móviles. Centro de Referencia en Accesibilidad y Estándares Web. Disponible en: www.cyldigital.es
- [17] Azenkot, S., Lee, N.B. "Exploring the Use of Speech Input by Blind People on Mobile Devices" En.: 15th International ACM SIGACCESS . Conference on Computers and Accessibility. ACM, Nueva York (2013)
- [18] Eric Maxwell MVC on Android "Buenas prácticas en la organización de aplicaciones Android", Último acceso 9 de Julio de 2018. academy.realm.io

Arquitectura basada en microservicios en un sistema de corrección automatizada de exámenes

Karina Ligorria, Analía Guzmán, Martín Casatti, Nicolás Horenstein, María Alejandra Paz Menvielle, Cynthia Corso

Facultad Regional Córdoba, Universidad Tecnológica Nacional, Maestro López y Cruz Roja Argentina s/n, Córdoba, Argentina.
{karinaligorria, analia.casatti, mcasatti, nicolashorenstein, pazmalejandra, corso.cynthia}@gmail.com

Resumen. El presente trabajo describe la arquitectura de un sistema de corrección automatizada de exámenes. Se toma como caso testigo los contenidos curriculares de la Cátedra Paradigmas de Programación de la Carrera Ingeniería en Sistemas de Información. Se comienza referenciando la arquitectura monolítica diseñada en 2015 cuando se comenzó este proyecto. Luego se describen los cambios realizados en ella para conformar una arquitectura basada en servicios, describiendo las distintas situaciones e inconvenientes que se presentaron y que condujeron al diseño e implementación de la nueva arquitectura.

Palabras clave: Arquitectura de software, arquitectura monolítica, arquitectura basada en microservicios, API REST

1 Contexto

Este trabajo se desarrolla en el marco del proyecto de investigación y desarrollo denominado “Análisis y detección de patrones en un grafo conceptual construido a partir de respuestas escritas en forma textual a preguntas sobre un tema específico”, proyecto homologado por la Secretaría de Investigación, Desarrollo y Posgrado de la Universidad Tecnológica Nacional, reconocido con el código SIUTICO4812TC.

Este proyecto, actualmente en ejecución, continúa y amplía los trabajos realizados durante el desarrollo del Proyecto “Metodología para determinar la exactitud de una respuesta, escrita en forma textual, a un interrogatorio sobre un tema específico”, (PID EIUTNCO0003592). Durante el transcurso del mismo se construyó el prototipo denominado Sibila 1.0 para realizar la corrección automatizada de los exámenes. Este prototipo accede a una base de conocimiento modelada como grafo dirigido. Inicialmente la base de conocimiento fue generada por los docentes de la cátedra Paradigmas de Programación, perteneciente al segundo año de la carrera de Ingeniería en Sistemas de Información, dictada en la Facultad Regional Córdoba, de la

Universidad Tecnológica Nacional. Cátedra que actúa como caso testigo de este proyecto y ha utilizado el prototipo para evaluar a los alumnos.

El proyecto actual tiene como objetivo analizar, detectar y evaluar patrones topológicos frecuentes en un grafo conceptual para determinar la exactitud de las respuestas, escritas en forma textual sobre un tema específico, utilizando una base de conocimiento diseñada como un grafo dirigido.

La arquitectura de Sibila 1.0 se diseñó e implementó inicialmente en forma monolítica, sin embargo con el paso del tiempo y con el objetivo del proyecto actual se tuvo la necesidad de actualizar el prototipo y por ende su arquitectura.

Así surge Sibila 2.0, que se modela y diseña con una arquitectura basada en microservicios.

El presente trabajo describe la evolución en el diseño de la arquitectura del sistema Sibila para adaptarse a los nuevos requerimientos y a los desafíos planteados por el equipo de trabajo.

2 Introducción

La Arquitectura de Software, es una disciplina de gran importancia dentro de la ingeniería de software. Se refiere a la estructura de un sistema compuesta de elementos con propiedades visibles de forma externa y las relaciones que existen entre ellos [1]. Los aportes más destacados de una arquitectura son la mejora en la comprensión de los sistemas grandes y complejos, la mejora en las posibilidades de reutilización, la consideración de la evolución del sistema y al definirse principalmente a través de diagramas proporciona una forma de interpretar y documentar la estructura [2].

A continuación se describen las características principales de las arquitecturas monolítica y basada en microservicios.

2.1 Arquitectura monolítica

En una arquitectura monolítica “toda la lógica se ejecuta en un único servidor de aplicaciones” (Daya y Col., 2015). Todos los módulos corren en un único desplegable bajo la misma máquina virtual [3].

Si bien algunos enfoques plantean esta arquitectura implementando el patrón Modelo Vista Controlador (MVC), por lo general, se tiene como resultado una aplicación grande y compleja. Contar con una base grande de código [4] hace que sea un sistema difícil de mantener, para realizar algún cambio o mejora se debe determinar el impacto de dicho cambio para que no se vea afectada la ejecución del resto del sistema.

En la Figura 1 se representa una arquitectura monolítica.

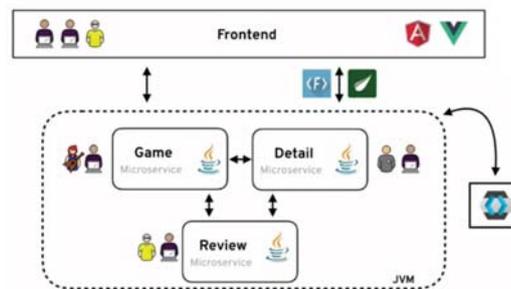


Figura 1: Arquitectura monolítica [3]

2.2 Arquitectura basada en microservicios

La arquitectura basada en microservicios es un estilo de arquitectura en donde la aplicación se encuentra constituida por un conjunto de microservicios o componentes, y se trata cada componente como si fuera una aplicación por sí misma [4, 5, 6]. La respuesta del sistema requiere la ejecución de los microservicios adecuados de acuerdo al proceso que se haya activado.

Algunas de las características de las aplicaciones basadas en microservicios que se pueden mencionar son [7, 8]:

- La funcionalidad del sistema es provista por un conjunto de servicios.
- Cada microservicio o servicio no debe cubrir una funcionalidad muy grande.
- Cada microservicio puede ser desarrollado en un lenguaje de programación diferente.
- Cada componente debe poder implementarse en forma independiente.
- La comunicación entre microservicios se realizará a través de métodos como REST (Representational State Transfer / transferencia de estado representacional) o similares [9].

En la Figura 2 se representa la arquitectura basada en microservicios:

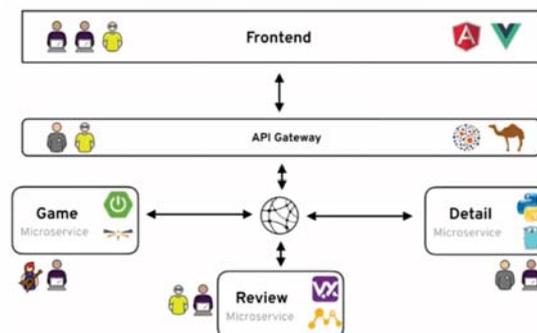


Figura 2: Arquitectura basada en microservicios [3]

A continuación se presentan algunas de las ventajas e inconvenientes que ofrecen las arquitecturas basadas en microservicios [8, 10]:

Beneficios:

- Componentes que pueden ser reutilizados en distintos escenarios.
- Mayor flexibilidad al ser más fácil reemplazar los microservicios.
- Componentes especializados en los niveles de usuario, modelo de datos y lógica de negocio.
- Mayor capacidad para adaptarse a cambios tecnológicos.
- Los equipos de desarrollo pueden trabajar en forma independiente, lo que propicia el trabajo en paralelo.
- Poca necesidad de coordinación entre los equipos, si se requiere un acuerdo de la información que espera y devuelve un servicio, y condiciones técnicas que se deben respetar.

Inconvenientes:

- Requiere de un buen diseño para poder determinar en todo momento qué componentes o microservicios participan en un proceso.
- Se puede ver afectado el rendimiento del sistema por la cantidad de intermediarios que se necesitan para ejecutar un proceso.

3 Rediseño de la arquitectura

3.1 Arquitectura Sibila 1.0

La arquitectura elegida inicialmente para Sibila 1.0, fue una arquitectura monolítica que se compone de tres capas horizontales (ver Figura 3) las cuales modelan niveles crecientes de abstracción y se utilizan interfaces de software para aislar los distintos módulos de efectos no deseados al realizar modificaciones [11].

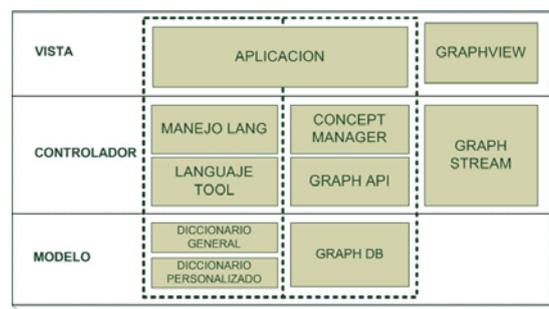


Figura 3: Arquitectura monolítica en tres capas del Sistema Sibila 1.0

En Sibila 1.0 cada módulo fue desarrollado en el mismo lenguaje de programación, utilizando una base de datos orientada a grafos para la evaluación de las respuestas obtenidas de las instancias de evaluación ejecutados por los alumnos de la cátedra de Paradigmas de Programación.

En la infraestructura planteada cada bloque tiene responsabilidades bien definidas que posibilitan un alto grado de especialización y brindan la capacidad, existente en todo sistema de información que está modularizado, de reemplazar un módulo por otro implementado de manera diferente pero que cumpla con los mismos requisitos de funcionamiento.

La elección de una arquitectura monolítica ha posibilitado determinar fácil y rápidamente la factibilidad de representar un texto como un conjunto de conceptos y relaciones en forma de grafo, pudiendo tempranamente validar las fórmulas propuestas para determinar el grado de exactitud de una respuesta.

A continuación se realiza una breve descripción de la funcionalidad de cada módulo:

- **Módulo de corrección ortográfica:** Tiene la responsabilidad de analizar las respuestas escritas en forma de texto libre y detectar los errores ortográficos, informarlos y sugerir las correcciones necesarias.
- **Módulo de gestión de conceptos:** Tiene la responsabilidad de administrar la base de conocimientos y sirve de capa de abstracción a las librerías de gestión de la base de datos de grafos en la que se implementa la base de conocimiento.
- **Módulo GraphView:** Permite visualizar un conjunto de conceptos y relaciones como un grafo dirigido, información que recibe en forma de lista del Módulo de Conceptos. Utiliza una librería Open Source para realizar el gráfico propiamente dicho, implementa mecanismos propios para establecer el formato visual del gráfico.
- **Módulo aplicación:** Brinda la interfaz de iteración con el usuario, a través de Forms SWI en Java. Aplicando el patrón de diseño MVC, se plantea una capa de aplicación desacoplada e independiente, para futuros cambios, como por ejemplo, el desarrollo de vistas web.

3.2 Arquitectura Sibila 2.0

La evolución a una arquitectura basada en microservicios permite incorporar herramientas nuevas en cuanto a procesamiento de texto, utilizar diferentes lenguajes de programación en una misma aplicación y utilizar la colaboración de otros servicios disponibles en la nube. Esta evolución asegura un sistema más flexible, sobre todo previendo su crecimiento y su aplicación en otros ámbitos.

Se propone para Sibila 2.0 una nueva estructura que cuenta con los siguientes componentes:

- Servidor de procesamiento de texto libre
- Servidor de gestión del conocimiento
- Servidor de patrones

- Cliente de gestión de evaluaciones
- Cliente de visualización de grafos

Cada componente se comunica con los demás a través de una API REST (Application Programming Interface / interfaz de programación de aplicaciones).

En la Figura 4 se muestran los componentes que intervienen en la nueva arquitectura propuesta.

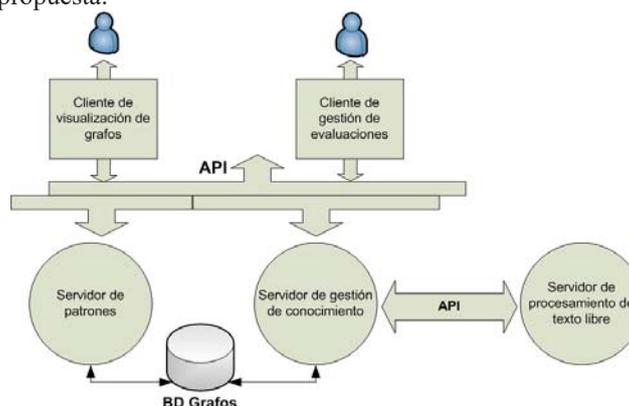


Figura 4: Arquitectura basada en microservicios de Sibila 2.0

4 Evolución de los componentes del sistema

Como parte de la evolución del sistema Sibila se planteó el reemplazo paulatino y gradual de los componentes monolíticos que componían el sistema en su versión 1.0 por un conjunto de servidores livianos de fines específicos que componen una arquitectura de microservicios.

La arquitectura basada en microservicios ofrece un conjunto de servicios de alto nivel, que se implementan a través de APIs que posibilitan el desarrollo de aplicaciones cliente que utilicen los algoritmos y datos de la base de conocimientos.

A continuación se describe la funcionalidad de cada componente:

4.1 Servidor de procesamiento de texto libre

Este servidor reemplaza al módulo de corrección ortográfica de Sibila 1.0. Se diseñó para poder realizar la validación ortográfica y el análisis del texto de tal forma que pueda enviar, al servicio que lo solicite, una cadena con el texto procesado.

Realiza dos operaciones básicas:

- Control de ortografía: analiza el texto para detectar errores de ortografía.
- Análisis de texto: identifica las entidades fundamentales para la construcción de la base de conocimiento del sistema.

4.2 Servidor de gestión de conocimiento

Este servidor tiene la misma funcionalidad que el módulo de gestión de conceptos del prototipo Sibila 1.0, su función es gestionar y consultar la base de conocimiento [11].

Algunas de las funcionalidades que implementa son la detección de tipo de término, la consulta de términos compuestos, la administración de conceptos y relaciones, la existencia de estructuras y, la generación y ejecución de consultas.

4.3 Servidor de patrones

Este servidor es el encargado de realizar la búsqueda de información en la base de datos de grafos, estas búsquedas serán realizadas a través de algoritmos que permitan detectar patrones por medios del análisis de las métricas sobre la base de conocimiento [12, 13]. Las métricas permiten analizar sobre qué componentes del grafo se realizan las mediciones, por ejemplo si son globales toman como referencia el grafo completo, con todos los nodos y arcos que lo conforman; si son locales toman como referencia un nodo o subconjunto de nodos para realizar los cálculos.

4.4 Cliente de gestión de evaluaciones

Este cliente reemplaza el módulo de aplicación de Sibila 1.0 y, a través de un sitio Web actúa de interfaz con el usuario. Realiza la gestión de alumnos, cursos, docentes, preguntas, permite la configuración y evaluación de instancias de evaluación y la visualización de notas e informes. Almacena dicha información en una base de datos relacional.

Este cliente se comunica con el servidor de gestión de conocimiento a través de la API de procesamiento de respuestas, para obtener y enviar información desde y hacia la base de datos de grafos.

4.5 Cliente de visualización de grafos

Este cliente reemplaza al módulo GraphView de Sibila 1.0 y permite la visualización de los datos de la base de datos que pueden ser enviados tanto desde el servidor de gestión de conocimiento como desde el servidor de patrones.

Esto implica que se podrá visualizar toda la información almacenada en la base de datos de grafos y además visualizar los resultados de las búsquedas de patrones definidos por los docentes para realizar su análisis.

5 Interfaz de comunicación entre los componentes

La comunicación entre los diferentes componentes de la arquitectura se logra a través de diferentes APIs, los cuales envían y reciben archivos JSON (Acrónimo de JavaScript Object). En este archivo la información se encuentra en un formato de texto que es completamente independiente del lenguaje de programación pero utiliza convenciones conocidos por los programadores de la familia de lenguajes C, C++, C#, Java, JavaScript, Python, y muchos otros [14].

Sibila 2.0 propone el uso de dos APIs para realizar la comunicación entre los servicios. Las APIs diseñadas son:

- API de interpretación de texto: Actúa de interfaz entre el servicio de procesamiento de texto libre y el servicio de gestión de conocimiento, la cual brinda una interfaz HTTP.
- API de procesamiento de respuestas: Actúa de interfaz entre el resto de los servicios de la arquitectura, es decir, el servidor de gestión de conocimiento, el servidor de patrones, el cliente de gestión de evaluaciones y el cliente de visualización de grafos.

6 Conclusiones y trabajos futuros

El presente trabajo muestra la nueva arquitectura propuesta para el sistema Sibila la cual consiste en una arquitectura basada en microservicios. A través de esta arquitectura se logra abstraer la complejidad técnica del sistema, logrando un sistema más flexible que permite la reutilización de los componentes como cajas negras, y es más sencillo utilizar soluciones open source disponibles en la nube.

Se propone seguir utilizando herramientas modernas de bases de datos, como lo es OrientDB, interfaz Web con el usuario e intercomunicación entre los diferentes componentes del sistema a través de API REST.

Se puede concluir que con esta nueva arquitectura basada en microservicios, el sistema ofrece un conjunto de servicios ligeros tanto para uso de la propia cátedra como de terceros, a la vez que se mantiene compacto y eficiente el núcleo que contiene la lógica de corrección, la base de conocimientos y la lógica de detección de patrones.

A través de esta arquitectura de software, se puede desarrollar cada microservicio con el lenguaje de programación más adecuado según la funcionalidad que debe ofrecer. Por otra parte, este enfoque facilita la convivencia de diferentes tipos de bases de datos, relacional y orientada a grafos, logrando representar o modelar los datos respetando sus características.

Como trabajo futuro se propone investigar nuevas herramientas que permitan la interpretación de texto libre escrito con mayor complejidad expresiva. Esto será posible ya que se cuenta con una arquitectura basada en microservicios para la que hay disponibles herramientas en lenguajes de programación más modernos facilitando el trabajo de base de este sistema.

También, se trabajará en la implementación de la lógica necesaria para el mantenimiento de la información para poner a prueba los algoritmos de detección y análisis de patrones sobre grafos dirigidos.

7 Referencias

1. L. Bass, P. Clements, R. Kazman. "Software Architecture in Practice, 2nd Edition". Addison Wesley. 2003.
2. Fernández, L. F. "Arquitectura de Software". Software Guru cono, 40-42. 2006
3. Soto, A. "Diferencia entre arquitectura monolítica y de microservicios". <https://openwebinars.net/blog/diferencia-entre-arquitectura-monolitica-y-microservicios/>
4. Daya,S. y Col. "Microservices from Theory to Practice Creating Applications in IBM Bluemix Using the Microservices Approach". 2015.
5. "DEVOPS: Arquitectura monolítica vs. microservicios", 2016. <https://www.chakray.com/es/devops-arquitectura-monolitica-vs-microservicios>
6. López, D. et al. "Arquitectura de Software basada en Microservicios para Desarrollo de Aplicaciones Web". 2017.
7. Esaú, A. "Microservicios: Beneficios y funcionamiento", 2016. <https://openwebinars.net/blog/microservicios-beneficios-y-funcionamiento/>
8. Wolff, E. "Microservices: Flexible Software Architecture". 2017.
9. Masse, M. "REST API Design Rulebook: Designing Consistent RESTful Web Service Interfaces. O'Reilly Media, Inc.". 2011.
10. Cortés, G. "IEDGE: Ventajas e inconvenientes de diferentes tipos de arquitecturas", 2019. <https://www.iedge.eu/gregorio-cortes-ventajas-e-inconvenientes-tipos-arquitecturas-de-sistemas>
11. Paz Menvielle,A. y col. "Arquitectura y operatoria de un sistema de corrección de exámenes automatizado, utilizando grafos dirigidos", en IV Congreso Nacional de Ingeniería Informática y Sistemas de Información, CONAIISI 2016, Universidad Católica de Salta, Facultad de Ingeniería, Argentina, 2016.
12. Pavlidis, Theodosios. "Structural pattern recognition". Springer. 2013.
13. Watanabe, S. "Pattern recognition: human and mechanical". John Wiley & Sons, Inc. 1985.
14. "Using JSON (JavaScript Object Notation) with Yahoo! Web Services", 2019. <https://web.archive.org/web/20100106010113/http://developer.yahoo.com/common/json.html>
15. Sowa, J. "Conceptual graph summary, Conceptual Structures: Current Research and Practice". Ellis Horwood, New York London Toront, pp. 3-66. 1992.

Calidad en uso en plataformas educativas: estudio centrado en la Accesibilidad Web

Sonia I. Mariño, Pedro L. Alfonzo

Facultad de Ciencias Exactas y Naturales y Agrimensura, Universidad Nacional del Nordeste, Corrientes, Argentina
{simarinio@yahoo.com, plalfonzo@hotmail.com}

Resumen. Las plataformas educativas son software diseñados y construidos según métodos, estándares y herramientas de la Ingeniería del Software (IS). La Accesibilidad Web es un criterio de calidad que debe aplicarse considerando a todos los usuarios, dado que se desconoce quiénes acceden a las plataformas. Por lo expuesto, en este trabajo se evalúa una plataforma educativa ampliamente utilizada para apoyar cursos de posgrado. En la presente investigación, el método se caracterizó por: i) Dada su finalidad se trata de una investigación aplicada. Se centró en localizar estándares y normativas que aborden aspectos de calidad y en particular aquellos orientados al proceso y producto software. Este conocimiento puede verse en la aplicación de los estándares, métodos y herramientas de Accesibilidad Web en los procedimientos y productos tecnológicos que se diseñan y desarrollan. ii) Según su alcance temporal, es seccional. Trata como momento específico el actual, es decir, el que transcurre para evaluar la plataforma. iii) Considerando las fuentes utilizadas, es una investigación primaria, los datos provienen de la revisión de las normativas vinculadas al objeto de estudio descripto. iv) Según su carácter, es una investigación cualitativa de investigación-acción, dado que la revisión de estos estándares y su implementación, permitirán lograr transformaciones en los desarrollos tecnológicos desde sus primeras fases. v) La investigación de campo se justifica en las evidencias relevadas de examinar el objeto de estudio: una plataforma web. Los resultados dan cuenta del interés de normar la accesibilidad web según diversas iniciativas internacionales. Además, la verificación expuesta sobre un sitio educativo muestra la importancia de difundir y aplicarlas, y así incidir positivamente para asegurar la calidad en uso de las soluciones web.

Palabras clave: Estándares, WCAG 2.0, Accesibilidad Web, plataformas educativas.

1 Introducción

Las plataformas educativas son software diseñados con fines específicos en el contexto formativo. Estos productos tecnológicos se construyen aplicando estándares, métodos, y herramientas de la Ingeniería del Software (IS). En [1] se sintetizan diversas denominaciones referentes a las plataformas virtuales “ (a) Entorno de

Aprendizaje Virtual (EAV/VLE), (b) Sistema de Gestión de Aprendizajes (SGA/LMS), (c) Sistema de Gestión de Cursos (SGC/CMS), (d) Entorno de Gestión de Aprendizajes (EGA/MLE), (e) Plataforma de Aprendizaje (PA/LP) y (f) Aula Virtual (AV)”.

Se coincide con [2] que “el software en general y por lo tanto, el software para la instrucción y el aprendizaje en particular, son productos que deben satisfacer los requerimientos de calidad del cliente-usuario”. En un contexto globalizado, donde el sujeto se transforma en el centro de los diseños mediados por computadora, es menester asegurar la calidad en uso, para lograr la aceptación de los potenciales usuarios de la industria, la academia y el gobierno.

En [3] se considera “a la calidad en uso como la capacidad del producto software para permitir que usuarios específicos logren realizar tareas específicas con productividad, efectividad, seguridad y satisfacción, en determinados escenarios de uso”.

Se coincide con [4] quienes expresan que “la web ha permeado todos los aspectos de nuestra vida cotidiana”. En distintas estadísticas se indican que es una de las tecnologías más utilizadas para mediar diversas actividades.

Además, las sociedades del conocimiento, plantean nuevas exigencias, roles e incursiones mediáticas que permiten transformar la información codificada en conocimiento real transmutado [5]. Particularmente en educación, el modelo *b-learning* representa la combinación de estrategias presenciales tradicionales y virtuales de la educación a distancia. Para aplicar estrategias virtuales, actualmente se opta por plataformas, ya sea aquellas específicamente construidas o aquellas que se adaptan sustentadas en gestores de contenidos.

1.2 La Accesibilidad Web como un criterio de calidad de la información

Las plataformas educativas son recursos “de apoyo al aprendizaje, por lo que la ausencia de calidad afecta directa o indirectamente el aprendizaje alcanzado por los estudiantes” [2]. La Accesibilidad Web (AW) es un criterio de calidad de un producto software, dado que asegura a todos los usuarios de las plataformas el acceso a los contenidos. La AW aborda el acceso universal a este servicio de Internet, en este sentido, es importante que la Web sea accesible para así proporcionar un acceso equitativo e igualdad de oportunidades a las personas con diversas discapacidades; permitiendo además, que los ciudadanos participen más activamente en la sociedad ([6]).

En este sentido, el diseño y construcción de sitios accesibles requiere implementar buenas prácticas relacionadas con las tecnologías de la información y la comunicación, particularmente con las vinculadas a los estándares de la Accesibilidad Web.

Por lo expuesto, un criterio a medir en un software en términos de calidad, es la accesibilidad. La accesibilidad Web (AW) “se ha entendido siempre como responsabilidad de los desarrolladores Web”. Sin embargo, “el software Web tiene también un papel importante en la accesibilidad Web” ([6]). La AW puede considerarse como parte de la accesibilidad del software, pero supone una mayor relevancia, debido a las enormes repercusiones de Internet en la sociedad ([7, 8]).

La W3C [9] es un organismo mundial que aborda ésta temática, siendo su función primordial desarrollar estándares para mejorar la calidad de los sitios Web y maximizar su potencial. Entre los principios que rigen en el ámbito del consorcio de W3C, se menciona el objetivo de hacer que los beneficios de la web estén disponibles para todo el mundo, independientemente del hardware, software, infraestructura de red, idioma, cultura, localización geográfica o habilidad física o mental.

Se elaboraron pautas que permiten construir productos de calidad centrado en los principios de la WCAG [10]) para presentar los contenidos.

La AW, ha cobrado relevancia en el diseño de sistemas web dirigidos a todos los ciudadanos digitales. Se sancionaron normativas que regulan la accesibilidad a nivel mundial, como en EE.UU., Portugal, España, Brasil, Chile, Perú, entre otros ([11, 12], 13]). La AW en la Argentina ha logrado una mayor difusión desde la implementación de la Ley 26.653 de Accesibilidad de la información ([14]).

Además, deben contemplarse las políticas públicas que se orientan a promover el sector de Servicios y Sistemas Informáticos. Entre ellas se menciona Ley de promoción de la Industria del Software ([15]) y la Ley de la Economía del Conocimiento ([16]).

Dada que la AW aborda el acceso a los contenidos, en el presente trabajo se plantea la relevancia de su evaluación para asegurar que se cumplan los estándares en las plataformas educativas debido a su amplia difusión en procesos de enseñanza.

El artículo se complementa de las siguientes secciones. En la segunda se describe la metodología utilizada en la elaboración de este trabajo. En la tercera: i) se sintetizan algunas normas y estándares que median la evaluación de la calidad del producto software, particularmente la AW y ii) se analiza la calidad de una plataforma educativa parametrizable y ampliamente utilizada, centrándose en el estándar W3C. Finalmente se exponen las conclusiones y futuras líneas de trabajo.

2 Metodología

En esta sección se caracteriza la presente investigación. Dada su finalidad se trata de una investigación aplicada. Se centró en localizar estándares y normativas que aborden aspectos de calidad y en particular aquellos orientados al proceso y producto software. Este conocimiento puede verse en la aplicación de los estándares, métodos y herramientas de AW en los procedimientos y productos tecnológicos que se diseñan y desarrollan.

Según su alcance temporal, es seccional. Trata como momento específico el actual.

Considerando las fuentes utilizadas, es una investigación primaria, los datos provienen de la revisión de las normativas vinculadas al objeto de estudio descripto y el uso de un validador automático aplicado a una plataforma educativa.

Según su carácter, es una investigación cualitativa de investigación-acción, dado que la revisión de estos estándares y su implementación, permitirán lograr transformaciones en los desarrollos tecnológicos desde sus primeras fases.

La investigación de campo se justifica dado que “se basa en informaciones obtenidas directamente de la realidad, permitiéndole al investigador cerciorarse de las condiciones reales en que se han conseguido los datos” [17].

El carácter documental de la propuesta se sustenta en que se trata de “un proceso basado en la búsqueda, recuperación, análisis, crítica e interpretación de datos secundarios, es decir, los obtenidos y registrados por otros investigadores en fuentes documentales: impresas, audiovisuales o electrónicas” [17].

El método aplicado para la elaboración del presente trabajo constó de las siguientes fases:

- Revisión de antecedentes de normas y estándares referentes a la AW.
- Selección y validación automática de la página de cursos de una plataforma educativa.
- Selección y estudio de validadores automáticos [18, 19, 20]. Se optó por TAW [18], dado que automatiza la valoración de las pautas WCAG 2.0 [10] propuestas por el W3C [9] y determinación de métricas de evaluación.
- Análisis de los resultados y elaboración de conclusiones preliminares.

3 Resultados

En la Ingeniería del Software, la satisfacción del cliente es fundamental y se representa a través de sus productos y servicios. El software como producto intangible, complejiza la medición de su calidad. Un aspecto a medir en los productos de software es la AW, es decir, la viabilidad de acceder a los contenidos desde la Web.

Dada la importancia de medir la calidad de los productos software, se diseñan estándares, siendo algunos de ellos los definidos por asociaciones particulares.

3.1 Estándares de Accesibilidad Web

Los estándares de calidad son reglas internacionales que garantizan la calidad de los productos. Las organizaciones que las promueven, constantemente las revisan por lo que están en evolución de acuerdo al campo de aplicación.

Particularmente en torno a estándares y normas de AW se mencionan [12, 20]:

1) Propuestas por AENOR:

- UNE 139803: Aplicaciones informáticas para personas con discapacidad. Requisitos de accesibilidad para contenidos en la Web.
- UNE 139804: Requisitos para el uso de la Lengua de Signos Española en redes informáticas 2007.
- UNE 66181. Calidad de la Formación Virtual. (Niveles de accesibilidad web basados en WCAG) 2012.
- UNE 139802: Requisitos de accesibilidad de software. (Traducción de ISO 9241-171) 2009. Accesibilidad del software embebido en páginas web (RIA).
- UNE-EN ISO/IEC 24751. Adaptabilidad y accesibilidad individualizadas en e-aprendizaje, en educación y formación. (Traducción de ISO/IEC 24751) 2012.

2) Propuestas del W3C

- WCAG 1.0: Web Content Accessibility Guidelines 1999. Accesibilidad del contenido de páginas web.
- WCAG 2.0: Web Content Accessibility Guidelines 2008. Accesibilidad del contenido de páginas web.
- ATAG 1.0: Authoring Tool Accessibility Guidelines 2000. Accesibilidad de editores de páginas web.
- User Agent Accessibility Guidelines 2002. Accesibilidad de navegadores web.
- WAI-ARIA 1.0: Accessible Rich Internet Applications 2011. Accesibilidad del software embebido en páginas web.
- WCAG-EM 1.0 Website Accessibility Conformance Evaluation Methodology. Metodología de evaluación de conformidad de accesibilidad de sitios web
- EARL 1.0 Evaluation and Report Language. Lenguaje e informe de evaluación.
- ATAG 2.0: Authoring Tool Accessibility Guidelines. Pautas de accesibilidad de la herramienta de autor.
- UAAG 2.0: User Agent Accessibility Guidelines. Pautas de accesibilidad para el usuario.

3.2 Accesibilidad Web en plataformas educativas

Según la WAI [21] la Accesibilidad Web aborda:

- Requisitos técnicos, se relacionan con el código subyacente en lugar de la apariencia visual.
- Requisitos relacionados con la interacción del usuario y el diseño visual.

El presente trabajo centra la evaluación en los requisitos técnicos, Se plantea como estudio de caso preliminar con miras de argumentar la formación profesional y así aportar desde espacios de Educación Superior al sector de la Industria del Software [22, 23, 24] y la sociedad.

En las plataformas diseñadas para apoyar procesos de aprendizaje sustentados en herramientas TIC, sean semi-presenciales o virtuales, es un requisito asegurar una grata experiencia de usuario, siendo un criterio de calidad la AW.

Dado que la AW tiene como objetivo garantizar el acceso e interacción de los sujetos en la web. Por ello se recurre al estudio y aplicación de los estándares y normas para lograrlo.

La WCAG [10] define un conjunto de principios que agrupan a las pautas que establecen los criterios de éxito a verificar. Estas pautas de accesibilidad establecen aquellas características específicas que todo sitio web debería contemplar para asegurar el acceso a la información a los usuarios con discapacidades.

- A. Principio PERCEPTIBLE: son aquellas condiciones que buscan que la información y los componentes de la interfaz del usuario sean presentadas, de modo que pueda percibirse de la manera más inteligible u óptima:

- Alternativas textuales, alternativas para convertir texto a otros formatos dependiendo la capacidad de la persona que los necesite;
 - Medios tempodependiente, para proporcionar acceso a los multimedia dependientes del tiempo y sincronizados, como son sólo audio, sólo vídeo, audio y vídeo, audio y/o video combinado con interacción;
 - Adaptable, contenido que pueda presentarse de diferentes formas sin perder información o estructura;
 - Distinguible, se busca facilitar a los usuarios ver y oír el contenido, incluyendo la separación entre el primer plano y el fondo;
- B. Principio OPERABLE: garantizar que los componentes de usuario y la interfaz de navegación deben ser fáciles:
- Accesible por teclado, proporcionar acceso a toda la funcionalidad mediante el teclado;
 - Tiempo suficiente, proporcionar el tiempo suficiente para leer y usar el contenido;
 - Convulsiones, no diseñar contenido de un modo que se sepa podría provocar ataques, espasmos o convulsiones;
 - Navegable, proporcionar medios para ayudar a navegar, encontrar contenido y determinar dónde se encuentran.
- C. Principio COMPENSIBLE: la información y el manejo de la interfaz de usuario deben ser claros. Se enfoca en características como:
- Legibilidad, hacer que los contenidos textuales resulten claros y comprensibles;
 - Predecible, hacer que las páginas web aparezcan y operen de manera previsible;
 - Entrada de datos asistida, para ayudar a evitar y corregir los errores.
- D. Principio ROBUSTO: El contenido debe ser lo suficientemente consistente y fiable como para permitir su uso con una amplia variedad de agentes de usuario, ayudas técnicas y estar preparado para las tecnologías posteriores.
- Compatible, para maximizar la semejanza con las aplicaciones de usuario actuales y futuras, incluyendo las ayudas técnicas.

A continuación se exponen los resultados de evaluar la página de cursos de una plataforma educativa. Se seleccionó y aplicó como validador TAW [18], dado que automatiza la valoración de las pautas WCAG [10]. Esta herramienta comprueba el nivel de accesibilidad alcanzado en el diseño y desarrollo de páginas web; y genera un informe en formato HTML.

Las comprobaciones que realiza TAW [18] se dividen en dos categorías: i) Automáticos. Problemas de accesibilidad que la herramienta detecta por sí sola y que deben ser solucionados; ii) Manuales. La herramienta señala la existencia de un posible problema que se debe verificar.

El informe generado por TAW [18] resume el total de los problemas (indica las correcciones necesarias), las advertencias (determina aspectos a revisar manualmente)

y los puntos no verificados (los criterios que requieren un análisis manual completo); y los organiza por los principios: Perceptible, Operable, Comprensible y Robusto.

Por lo expuesto, las siguientes métricas permiten determinar el total general de problemas (TGP) detectados de forma automática y el total general de problemas que requiere revisión manual (TPRM), incluyendo advertencias y puntos no verificados.

$$\mathbf{TGP} = \sum_{i=1}^N \mathbf{CPPP} \quad \begin{array}{l} N = \text{Cantidad de principios} \\ \mathbf{CPPP} = \text{Cantidad de problemas} \\ \text{por principio} \end{array} \quad (1)$$

$$\mathbf{TPRM} = \sum_{i=1}^N \mathbf{PRR} \quad \begin{array}{l} N = \text{Cantidad de principios} \\ \mathbf{PRR} = \text{Cantidad de problemas} \\ \text{que requieren revisión manual} \\ \text{por principio} \end{array} \quad (2)$$

La Fig. 1 representa el número de problemas automáticos detectados en el sitio analizado por TAW [18] agrupados por principios y a las que se deben realizar correcciones de una página de cursos de la plataforma elegida. Sobresale el error correspondiente al principio “Perceptible”, es decir, aquellas condiciones que buscan que la información y los componentes de la interfaz del usuario sean presentadas, de modo que pueda percibirse de la manera más inteligible u optima. Seguida por el principio “Robusto”, es decir, se requiere que el contenido sea lo suficientemente consistente y fiable para permitir ser utilizado por una amplia variedad de agentes de usuario y ayudas técnicas. Posteriormente los errores respondientes al principio “Comprensible”, éste necesita que la información y el manejo de la interfaz de usuario sean claros. La aplicación de la métrica TGP indica el total problemas que requieren correcciones.

$$\mathbf{TGP} = 9 + 0 + 4 + 5 = 19$$

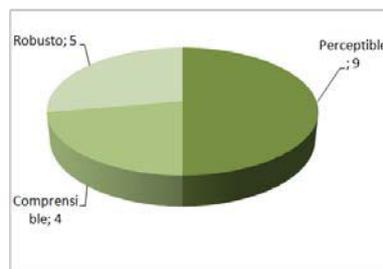


Fig. 1. Problemas detectados por principios.

La Fig. 2 presenta los posibles problemas detectados por la herramienta que se deben verificar de manera manual, agrupados en “Advertencias” (es necesario revisar manualmente) y “No verificados” (comprobación completamente manual). La métrica TPRM indica el total de problemas que requieren revisión y en caso de ser necesario realizar las correcciones pertinentes.

$$\mathbf{TPRM} = 9 + 13 + 11 + 0 = 33$$

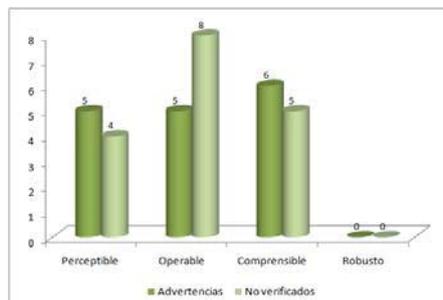


Fig. 2. Problemas a verificar por principios.

4 Conclusiones

Las plataformas educativas, como productos software, resultan de aplicar sistemáticamente modelos, métodos y herramientas de la Ingeniería del Software. La calidad en uso aborda la percepción del usuario en referencia al empleo de un software.

Particularmente, la Accesibilidad Web es una característica de calidad medible que se debe incorporar desde las fases iniciales de la creación de un software. Por ello, resulta relevante conocer los estándares y las normas, que ligados con la accesibilidad, aseguran una grata experiencia del usuario, y además sus fundamentos para evaluar la AW.

En el presente estudio, se expuso una síntesis de los estándares relacionados con la AW propuestos por asociaciones como AENOR y W3C. Se seleccionó como guía de verificación las pautas establecidas por la WAI [21] del consorcio W3C.

Los estudios y métricas obtenidas, así como los resultados expuestos en otros artículos [25, 26], dan cuenta de la necesidad de aplicar estos estándares internacionales en la construcción de las plataformas educativas, con la finalidad de asegurar la calidad en uso considerando a todos los usuarios de la web.

Referencias

1. Ramírez Valdez, W. y Barajas Villarruel, I.: Uso de las plataformas educativas y su impacto en la práctica pedagógica en instituciones de educación superior de San Luis Potosí. EDUTEC, Revista Electrónica de Tecnología Educativa, Nro. 60 (2017).
2. Segura, A., Vidal, Ch. y Prieto, M.: Evaluación de la Calidad del Software para el aprendizaje, <https://pdfs.semanticscholar.org/2bd0/232cedd7c9e74b13039ecc1fd17bdafd02a3.pdf>
3. González Rodríguez, J. y Olsina, L.: Hacia la Medición de Calidad en Uso Web, https://www.researchgate.net/publication/255601471_Hacia_la_Medicion_de_Calidad_en_Uso_Web

4. Ramakrishnan I., Ashok V., Billah S.M.: Non-visual Web Browsing: Beyond Web Accessibility. *Universal Access in Human-Computer Interaction*. Vol, 10278, pp. 322-334 (2017).
5. Rodríguez Camargo, M., Latorre Barragán, C., Hurtado Castro, M., Vera Díaz, C. y Ibarra, M.: El blended-learning, una alternativa de calidad educativa para la educación virtual y a distancia. *Virtual Educa*, <https://acceso.virtualeduca.red/documentos/ponencias/puerto-rico/1100-5f05.pdf>
6. W3C: Introducción a la Accesibilidad Web, <https://www.w3c.es/Traducciones/es/WAI/intro/accessibility>
7. Rodríguez Fuentes, A. y García Guzmán, A.: Medios de comunicación y discapacidad. *Revista Icono* 14(15), pp. 303-319 (2009).
8. Abid Ismail, K. K.: Accessibility of Indian universities' home pages: An exploratory study. *Journal of King Saud University – Computer and Information Sciences*, n° 30, pp. 268–278 (2018).
9. W3C: Consorcio World Wide Web. Disponible: <http://www.w3c.es/>
10. WCAG: Content Accessibility Guidelines (WCAG) 2.0., <http://www.w3.org/TR/WCAG20>
11. Mariño, S. I., Godoy, M. V., Alfonzo, P. L., Acevedo, J. J., Solís L. G. y Vázquez, A. F.: Accesibilidad en la definición de requerimientos. *Multiciencias*, 12(3), pp. 305 – 312 (2012).
12. Varela, C., Miñán, A., Hilera, J., Restrepo, F., Amado, H., Córdova, M. y Villaverde, A.: Estándares y legislación sobre accesibilidad web, *Actas del IV Congreso Internacional ATICA 2012 - Loja (Ecuador)*, pp. 46-53 (2012).
13. Fuertes, J. y Martínez, L.: Accesibilidad web. *Legislación y estándares*, <https://core.ac.uk/download/pdf/61912290.pdf>
14. Senado de la Nación Argentina (2010). Ley 26.653: Accesibilidad de la Información en las Páginas Web. Autoridad de Aplicación. Plazos. Reglamentación.
15. Ley 25.922: Ley de Promoción de la Industria del Software, <http://infoleg.mecon.gov.ar/infolegInternet/anexos/95000-99999/98433/norma.htm>.
16. Ley 27.506: Régimen de Promoción de la Economía del Conocimiento, http://www.parlamentario.com/db/000/000759_ley_de_edc_8.3.19.pdf.
17. Arias, 2015: citado por IEEE STD 610-1990: IEEE Standard Glossary of Software Engineering Terminology.
18. TAW: validador de Accesibilidad Web, <https://www.tawdis.net/>
19. Cynthia Says: Test de Accesibilidad Web, <http://www.cynthiasays.com/Home.aspx>
20. Tanaguru: Test de Accesibilidad Web, <https://my.tanaguru.com/>
21. WAI. Accessibility, Usability, and Inclusion, <https://www.w3.org/WAI/fundamentals/accessibility-usability-inclusion/>
22. Mariño, S. I. y Alfonzo, P. L.: Evaluación de la accesibilidad web. Una mirada para asegurar la formación en la temática, *Campus Virtuales*, 6(2), pp. 21-30 (2016).
23. Mariño, S. I., Alfonzo, P. L. y Pagnoni, V. K.: Accesibilidad web visual. Formación de RRHH para fomentar su desarrollo en la región NEA, WICC 2019.
24. Mariño, S. I., Alfonzo, P. L., Pagnoni, V. K., Gronda, L., Galain García, C., Maidana, J., Gallardo, J. P. Blanco, V. y Godoy, M. V.: Formación Universitaria en torno a la Accesibilidad Web. Algunas experiencias en la FaCENA-UNNE. JAIIO 2019.
25. Mariño, S. I., Alfonzo, P., Gómez Codutti A. y Godoy, M. V.: Automatic evaluation of WCAG 2.0 guidelines in a Drupal-based platform, *International Journal of Information Science and Intelligent System*, 4(1), pp. 35-42 (2014).
26. Mariño, S. I., Alfonzo, P., Escalante, J., Alderete R. y Godoy, M. V.: Las pautas WCAG 2.0 para determinar el nivel de accesibilidad en dos plataformas educativas, *Revista Internacional de Tecnología, Conocimiento y Sociedad*, 2(2) pp. 139-149 (2015).

Definición de un estándar para la evaluación de calidad de software educativo

Luciano Straccia¹, Paula Zanetti¹, María Florencia Pollo-Cattaneo¹

¹Grupo de Estudio en Metodologías de Ingeniería de Software (GEMIS). Universidad Tecnológica Nacional. Facultad Regional Buenos Aires. Argentina.
lstraccia@frba.utn.edu.ar, paula.zanetti@gmail.com, flo.pollo@gmail.com

Resumen. La evaluación de calidad de producto es parte inherente de los procesos de desarrollo de software. Existe un modelo de calidad para realizar una evaluación que asegure maximizar la calidad de producto, pero no existe ningún estándar que permita la evaluación de calidad específicamente para el ámbito educativo considerando aspectos de los procesos de enseñanza y aprendizaje. En el presente trabajo se desarrolla una propuesta de normalización basada en las directivas para la construcción de estándares de ISO/IEC y su aplicación para la evaluación de un software en actividades de una asignatura de carrera de grado.

Palabras Claves: calidad, software, educación, normas, ISO.

1 Introducción

Dentro del ámbito de la Facultad Regional Buenos Aires de la Universidad Tecnológica Nacional (UTN.BA) se encuentra el Grupo GEMIS (Grupo de Estudios de Metodología para Ingeniería de Software y Sistemas de Información), que lleva adelante el Proyecto de Investigación y Desarrollo “Intervenciones tecnológicas en dispositivos didácticos con herramientas de tecnología informática” entre cuyos objetivos se encuentra describir y analizar el uso de la tecnología informática en las intervenciones didácticas de los profesores de las asignaturas de la carrera de Ingeniería en Sistemas de Información (ISI) y desarrollar nuevos artefactos tecnológicos que favorezcan la mejora en las intervenciones didácticas.

El proyecto ha implicado la construcción de software, cuya evaluación, basada en las normas y estándares de calidad, resulta incompleta respecto de las necesidades disciplinares educativas. Por ello, en el presente trabajo se desarrolla una propuesta para la normalización de la evaluación de calidad de software en el ámbito educativo, tomando como referencia modelos preexistentes en la industria del software y se proponen características y subcaracterísticas de calidad que, utilizados conjuntamente con la ISO/IEC 25000:2005, permiten evaluar la calidad de un software educativo.

Este trabajo presenta conceptos vinculados a calidad de producto software (sección 2), la calidad de producto para software educativo (sección 3), la presentación de la propuesta de normalización (sección 4) y la validación e implementación de la

normalización (sección 5). Finalmente, en la sección 6, se presentan las conclusiones y las futuras líneas de trabajo.

2 Calidad de producto software

2.1. Concepto de calidad

La calidad es compleja y multifacética. El software es complicado, y se vuelve aún más complejo, conforme evoluciona e incorpora nuevas funcionalidades e interconexiones. Sin embargo, es correcto pensar que los desarrolladores de software deben tratar de construir sistemas de alta calidad; pero, en la vida real, esto representa un dilema, inclusive para las mejores y más experimentadas organizaciones. Esta disyuntiva está relacionada con la búsqueda de un software suficientemente bueno; el arte de buscar ese punto intermedio, en donde la calidad es suficientemente buena (*good enough*), es parte esencial del planeamiento de un desarrollo, que debe tratar de aplicar las mejores prácticas en cada paso del proceso [1].

Pressman [2] define calidad de software como “el cumplimiento de los requisitos de funcionalidad y desempeño explícitamente establecidos, de los estándares de desarrollo explícitamente documentados, y de las características implícitas que se espera de todo software desarrollado profesionalmente”, es decir, que los estándares definen un conjunto de criterios de desarrollo que guían la forma en que se aplica la Ingeniería del Software; por lo tanto, si no se siguen esos criterios, casi siempre habrá falta de calidad.

Colaborando en estas definiciones, Garvin [3] estableció 5 diferentes visiones pragmáticas para la calidad: trascendental, del usuario, del fabricante, del producto y del valor. La visión trascendental refiere que si bien la calidad es algo que se reconoce rápidamente, no es fácil de definir; resulta una suerte de cualidad innata que se percibe pero no necesariamente se describe. La visión del usuario, que es subjetiva y define la calidad en términos del fin particular del destinatario; de la atención de sus necesidades y conveniencias; si el destinatario está satisfecho, esta visión asume que el producto tiene calidad. La visión del fabricante, que indica que si el producto cumple con sus especificaciones originales entonces tiene calidad. La visión del producto, que apunta a que la calidad está directamente relacionada con las funciones y características del producto, siendo medible y precisa a través de atributos del mismo. Y, para finalizar, la visión del valor: la cual indica que la calidad es función del nivel de conformidad del producto, a un costo razonable.

2.2. Modelo de calidad de producto

El trabajo de preparación de las normas internacionales normalmente se realiza a través de los comités técnicos de la Organización Internacional de Normalización (ISO). Cada organismo, miembro interesado en una materia para la cual se haya

establecido un comité técnico, tiene el derecho de estar representado en dicho comité. Las organizaciones internacionales, públicas y privadas, en coordinación con ISO, también participan en el trabajo. ISO colabora estrechamente con la Comisión Electrotécnica Internacional (IEC) en todas las materias de normalización electrotécnica. La tarea principal de los comités técnicos es preparar Normas Internacionales.

Algunos de los estándares internacionales de evaluación de producto software son: la ISO/IEC 9126:2001 [4], que permite evaluar la calidad del producto software y establece las características de la calidad; la ISO/IEC 14598:1999 [5], que presenta pautas que ayudan al proceso de evaluación del producto software y la ISO/IEC 25000:2005 [6], que proporciona una guía para el uso de las nuevas series de estándares internacionales, llamados Requisitos y Evaluación de Calidad de Productos de Software (SQuARE), constituye una serie de normas basadas en la ISO 9126 y en la ISO 14598 y ofrece términos y definiciones, modelos referencia, guía general, guías de división individual y los estándares para fines de especificación, planificación y gestión, medición y evaluación.

El objetivo principal de la ISO/IEC 25000 es guiar el desarrollo de los productos de software con la especificación y evaluación de requisitos de calidad. La familia ISO 25000 está orientada al producto software, permitiendo definir el modelo de calidad y el proceso a seguir para evaluar dicho producto.

El estándar ISO/IEC 25010, que reemplaza y actualiza el estándar ISO9126-1, define un modelo de calidad del producto [7]. El modelo de calidad de producto abarca cualidades internas y externas del sistema y está compuesto por 8 características (funcionalidad, rendimiento, usabilidad, fiabilidad, seguridad, mantenibilidad, portabilidad y compatibilidad, cada una de ellas con sus correspondientes subcaracterísticas) y 30 subcaracterísticas.

3 Calidad de producto software educativo

3.1. Antecedentes en la evaluación de calidad de software educativo

Díaz y otros [8] analizan diversas metodologías y realizan una propuesta metodológica nueva basada en un Modelo Sistémico de Calidad de Software, aunque plantea una propuesta centrada en aspectos informáticos dejando de lado aspectos pedagógicos. Por otro lado, Flores y Rivero [9] plantean la necesidad de generar instrumentos unificados para la evaluación del software educativo, aunque no realizan propuestas concretas.

En [10], se realiza un relevamiento de diversas metodologías y herramientas para la evaluación de software educativo. Su lectura permite comprender las diferentes propuestas generadas en la segunda mitad del siglo pasado. En tanto, en [11] se presentan diversos modelos de evaluación para el área. Según el análisis realizado en [12], la mayoría de estos métodos se basan en las propuestas de Pere Marques Graells, quien en su trabajo [13], considera tres aspectos: funcionales, técnicos y pedagógicos.

Por otro lado, en [12] se presenta un esquema general del Modelo de Aspectos Funcionales, Pedagógicos y Técnicos del Software Educativo, basado en el trabajo de Marques, pero resultando de mayor complejidad dada la falta de información que permita su implementación.

Reyes-Caballero [14], propone la evaluación de software según tres aspectos fundamentales: pedagógicos, técnicos y tecnológicos. En una evaluación detallada de dichos aspectos, puede observarse que el modelo de Reyes-Caballero resulta incompleto en comparación con el de Marques [15].

En tanto, en [16] se realiza un análisis y comparación de diversos modelos de evaluación de software educativo que han sido trabajados por diversos autores [8; 17-21]. El trabajo presenta en sus conclusiones la categorización de variables en dos grupos: los aspectos técnicos y los aspectos pedagógicos. Dicho trabajo propone esta categorización, pero no expresa cuáles elementos de cada uno de los modelos de evaluación de software educativo corresponden a estas categorías, por lo cual requieren un mayor grado de análisis para su operatividad [15].

Según un trabajo de Alba García-Barrera [22] “desde la inclusión de las TIC en las aulas, han sido numerosos los autores que han recomendado el análisis de ciertas dimensiones de los recursos digitales por medio de indicadores de credibilidad que permitan evaluar su calidad y utilidad educativa. En general, las guías de evaluación que analizan estos factores surgieron en el ámbito de la biblioteconomía y en un principio se enfocaron a los recursos de información disponibles en internet” dando baja prioridad a aspectos educativos y limitando el análisis a ciertos tipos de recursos.

3.2. Propuesta preliminar

En [15] se presenta la primera propuesta de normalización de la evaluación de calidad de producto software para el ámbito educativo, definiendo las características y subcaracterísticas de un modelo de calidad. En dicha propuesta se define un modelo de calidad con cuatro características, en base al trabajo de Cova [16]: condiciones referidas a los destinatarios, enseñanza, contenido y aprendizaje, con sus correspondientes subcaracterísticas. Sin embargo, dicho trabajo no contempla la terminología adoptada por los estándares ISO/IEC ni los principios y reglas para la estructura de un documento de estandarización de calidad. En dicho trabajo preliminar se realiza la validación de la normalización sobre la herramienta Google Forms y su aplicación en el ámbito educativo, pero no sobre un software específicamente destinado a los procesos de enseñanza., tal como se presenta en este trabajo con la herramienta WinQSB.

4 Propuesta de normalización

En [23] se puede encontrar el documento con la normalización de calidad siguiendo los principios y reglas de la Directiva ISO/IEC Parte 2 [24], que define las siguientes subdivisiones del documento: título; prólogo; introducción; alcance; normativas referenciadas; términos y definiciones; símbolos y abreviaturas; métodos

de medición y prueba; marca, etiqueta y prueba (que no aplica al tipo de producto presentado en este trabajo); anexos y bibliografía.

A continuación, se presentan las características y subcaracterísticas definidas en el modelo:

1. Condiciones referidas a los destinatarios: la capacidad del producto software para adaptarse al medio y los destinatarios.
 - 1.1. Individualización: la capacidad del software para adaptarse a las características individuales de los usuarios y al progreso individual.
 - 1.2. Adaptación social, cultural e ideológica: la capacidad del producto software para adaptarse a las características del medio sociocultural, respetando las visiones de sociedad, familia y cotidianidad que se intentan abordar en el proceso de enseñanza.
2. Enseñanza: la capacidad del producto software para acompañar técnicas y metodologías de enseñanza.
 - 2.1. Cumplimiento de objetivos: la capacidad del producto software para satisfacer los objetivos del proceso de enseñanza.
 - 2.2. Motivación: la capacidad del producto software para brindar motivaciones al usuario al conseguir una meta global.
 - 2.3. Retroalimentación: la capacidad del producto software para brindar al usuario una retroalimentación positiva, amigable y convincente.
 - 2.4. Adaptación metodológica: la capacidad del producto software para adaptarse a la metodología de enseñanza y estilos docentes.
 - 2.5. Adaptación a enfoques no memorísticos: la capacidad del producto software para adaptarse a un enfoque crítico, aplicativo y creativo de las actividades dirigido a la construcción de conocimiento (no memorístico) y con inclusión de actividades colaborativas.
3. Contenidos: la capacidad del producto software para favorecer la generación de contenidos apropiados a la metodología y técnica de enseñanza.
 - 3.1. Presentación: la capacidad del producto software para presentar los contenidos de manera clara, precisa y actualizada.
 - 3.2. Tutorización: la capacidad del producto software para tutorizar itinerarios, sugiriendo contenidos diferentes en función de las respuestas y actividades del usuario.
 - 3.3. Ejemplificación: la capacidad del producto software para presentar ejemplos claros y relevantes para ilustrar y comprender el contenido.
 - 3.4. Ayuda: la capacidad del producto software para presentar glosarios de ayuda y ofrecer pistas para resolver situaciones problemáticas.
 - 3.5. Evaluación automática: la capacidad del producto software para facilitar la evaluación, con corrección automática de actividades.
 - 3.6. Registro de datos: la capacidad del producto software para realizar registro del desempeño del usuario y suministro de informes.

- 3.7. Desarrollo de valores: la capacidad del producto software para desarrollar actitudes favorables hacia los valores del ser humano y su ambiente.
4. Aprendizaje: la capacidad del producto software para contribuir al proceso de aprendizaje.
 - 4.1. Metodología de aprendizaje: la capacidad del producto software para permitir que el estudiante sea un participante activo en el proceso de aprendizaje y le permita aprender a partir de su propia experiencia.
 - 4.2. Flexibilidad: la capacidad del producto software para ofrecer diversos niveles y/o itinerarios.

5 Validación e implementación de la normalización propuesta

5.1. Producto evaluado

El objetivo del software WinSQB es permitir el tratamiento de los problemas de gestión mediante herramientas cuantitativas, las que en su conjunto constituyen los métodos de la investigación de operaciones. WINQSB es "una aplicación versátil que permite la solución de una gran cantidad de problemas: administrativos, de producción, de recursos humano (y) dirección de proyectos" [25]. Debido a su facilidad y potencia de manejo, se convierte en una herramienta útil para temáticas como la investigación de operaciones, los métodos de trabajo, planeación de la producción, evaluación de proyectos, control de calidad, simulación, estadística, entre otras.

5.2. Contexto de evaluación

La evaluación del producto se consideró en un ambiente educativo de carreras de grado, en la cátedra de Investigación Operativa de la carrera Ingeniería en Sistemas de Información de la UTN.BA. Dicha asignatura, ubicada en el cuarto nivel del plan de estudios, realiza actividades prácticas (tanto vinculadas a prácticas específicas de cada clase, como el trabajo práctico integrador) sobre diferentes temáticas, que son resueltas a partir del uso de WinQSB, donde los alumnos resuelven problemas abiertos de programación lineal, con dual, simplex dual y análisis de sensibilidad y paramétrico.

El software es utilizado en el marco de una actividad basada en la siguiente metodología de trabajo:

- definir un problema real, seleccionado por el alumno. Se debe identificar el dominio y el problema que se busca resolver;

- identificar las variables, restricciones y capacidades del problema, determinando además el enfoque de resolución (maximización o minimización);
- utilizar el software con la carga del modelo;
- visualizar en el software de las iteraciones de ejecución del proceso, con presentación de tablas intermedias de resultados, hasta que el software presenta el resultado óptimo;
- analizar por parte de los alumnos del resultado obtenido poniéndolo en relación con el problema y dominio especificado;
- realizar un análisis de sensibilidad y paramétrico (analizando el impacto en el resultado de cambios en los valores de coeficientes de las variables).

El objetivo de aplicación concreto de la herramienta es la ayuda al estudiante para resolver y encontrar resultados óptimos de problemas específicos, mediante métodos cuantitativos de gestión para su posterior análisis y uso en la toma de decisiones.

5.3. Evaluación de calidad de producto software educativo

Para la evaluación de calidad del WinQSB en el contexto de actividades detallado en 5.2 se utiliza:

- la norma de calidad presentada en 5.1 y disponible en [23];
- un documento de trabajo (planilla) para facilitar la evaluación [26].

El documento de trabajo utilizado posee el listado de características y subcaracterísticas evaluables, las cuales deben ser evaluadas según los siguientes valores: Cumple, Cumple parcialmente, No Cumple o No Aplica (N/A). En los casos que se trate de cumplimiento parcial, no cumplimiento o, si se considera que no aplica para la evaluación, se registra en la columna observaciones un comentario que permita completar la evaluación.

El resultado de la evaluación es presentado en la Tabla 1 y sus observaciones son presentadas a continuación.

Tabla 1. Evaluación de calidad de WinQSB.

Característica	Subcaracterística	Evaluación
Condiciones referidas a los destinatarios	Individualización	Cumple
	Adaptación social, cultural e ideológica	N/A
Enseñanza	Cumplimiento de objetivos	Cumple
	Motivación	No Cumple
	Retroalimentación	Cumple
	Adaptación metodológica	Cumple
	Adaptación a enfoques memorísticos	Cumple
Contenidos	Contenidos	Cumple

Característica	Subcaracterística	Evaluación
	Tutorización	No Cumple
	Ejemplificación	No Cumple
	Ayuda	Cumple Parcialmente
	Evaluación automática	No Cumple
	Registro de datos	No Cumple
	Desarrollo de valores	N/A
Aprendizaje	Metodología de aprendizaje	Cumple
	Flexibilidad	N/A

Según lo observable en la tabla 1, los hallazgos realizados son:

- cumple satisfactoriamente con 7 subcaracterísticas (individualización, cumplimiento de objetivos, retroalimentación, adaptación metodológica, adaptación a enfoques no memorísticos, contenidos y metodología de aprendizaje);
- cumple parcialmente con 1 subcaracterística (ayuda);
- no cumple con 5 subcaracterísticas (motivación, tutorización, ejemplificación, evaluación y registro de datos);
- 3 subcaracterísticas se considera que no aplican para la evaluación (adaptación social, cultural e ideológica; desarrollo de valores; flexibilidad)

A partir de las observaciones realizadas para cada subcaracterísticas se presenta, a continuación, un análisis de los aspectos que cumple parcialmente o no cumple.

El software no realiza análisis contextualizado, sino que sólo presenta resultados matemáticos, que deben ser puestos en contexto por los propios usuarios, siendo además que esta puesta en contexto y dominio resulta la actividad analítica más importante para la materia, por lo cual si el software realizara dicho análisis influiría negativamente en los objetivos de aprendizaje. Respecto de los aspectos vinculados a sociedad, cultura e ideología y al desarrollo de valores, están relacionados al dominio seleccionado y no al uso de la herramienta. De igual forma, si bien la herramienta no resulta flexible, permite una implementación flexible en el marco de las actividades que defina el docente. Puede observarse en los aspectos mencionados la relevancia que cobra la preparación de la actividad por parte del docente, una estrategia adecuada para el uso del software y el dominio y problema seleccionado por el alumno, todos aspectos de relevancia para el aprendizaje, pero no vinculables al software en sí.

Se han hallado algunas falencias del producto:

- No ofrece premios ni ningún tipo de recompensas, ni tiene niveles de dificultad diferenciados; estos aspectos pueden ser cubiertos por la propia estrategia didáctica del docente, pero no por la herramienta en sí.
- No presenta ejemplos y las ayudas que presenta a veces resultan ineficientes.
- No permite evaluación automática ni registro de datos

6 Conclusiones

La aplicación del modelo de calidad y sus atributos definidos en la familia de Normas ISO 25000 permiten abordar la evaluación de calidad de un producto software. Sin embargo, cuando se espera que este software sea evaluado en función de características propias de la disciplina a la cual sirve, se requiere un modelo propio de evaluación de producto en el marco disciplinar correspondiente. En este trabajo se presenta un modelo de calidad para el software educativo y su validación e implementación con el software WinQSB. A partir de los resultados hallados, se sugiere la generación de herramientas que permitan reducir el impacto de las falencias del producto.

Se prevé la ampliación de su implementación y, en consonancia con la Directiva ISO/IEC Parte 1 [27], que define el uso de tres idiomas oficiales: inglés, francés y ruso y que los estándares internacionales publicados por ISO y IEC se encuentran en inglés y francés y en algunos casos en ruso, se prevé generar una versión en idioma inglés con la normalización de calidad propuesto en este trabajo.

Referencias

1. Venners, B. Design by Contract: A Conversation with Bertrand Meyer. Artima Developer" (2003)
2. Pressman, R. Ingeniería del Software. Ed: Mc Graw Hill. (2005).
3. Garvin, D. What Does Product Quality Really Mean? *Sloan Management Review*. (1984)
4. International Organization for Standardization. *ISO/IEC 9126-1*. Software engineering. Product Quality (2001).
5. International Organization for Standardization. *ISO/IEC 14598-1*. Information Technology. Software Product Evaluation (1999).
6. International Organization for Standardization. *ISO/IEC 25000-1*. Software Engineering. Software Product Quality Requirements and Evaluation (SQuaRE). (2005).
7. International Organization for Standardization. *ISO/IEC 25010*. *Systems and Software Engineering. System and Software Quality Requirements and Evaluation (SQuaRE). System and software quality models*. (2011).
8. Diaz, M; Perez, M; Mendoza, L; Griman, A. Calidad Sistémica del Software Educativo. Presentado en el Congreso Internacional Edutec 2003, Gestión de las Tecnologías de la Información y la Comunicación en los diferentes ámbitos educativos, Caracas (2003).
9. Flores, H. y Rivero, P. ¿Cómo selecciona el profesorado recursos digitales para enseñar Historia? *History and History teaching*. (2014)
10. Cataldi, Z. Metodología de diseño, desarrollo y evaluación de software educativo. (2000)
11. Angel, O. y Lopez, S. Estudio comparativo de modelos para evaluar software educativo. Tesis de Grado, Universidad Juárez Autónoma de Tabasco, Cunduacán, Tabasco, México. (2014)

12. Yedra, R.; Ramos Mendez, E. y Ferrer Sanchez, J. Análisis para determinar cómo se valúa una herramienta de software de tipo educativo. *CONAIC*. (2015)
13. Marques, P. *Software educativo: guía de uso y metodología de diseño*. Barcelona: Editorial ESTEL (1995)
14. Reyes-Caballero, F; Fernandez-Morales, F.; Duarte, J. Herramienta para la selección de software educativo aplicable al área de tecnología en el nivel de educación básica. *Revista Entramado*. (2015)
15. Straccia, L; Zanetti, P.; Pollo-Cattaneo, M. F. Normalización de la evaluación de calidad de producto software para el ámbito educativo. En *XXIV Congreso Argentino de Ciencias de la Computación, CACIC 2018: Libro de Actas*, Tandil (2018).
16. Cova, A.; Arrieta, X. Análisis y comparación de diversos modelos de evaluación de software educativo. *Enl@ce: Revista Venezolana de Información, Tecnología y Conocimiento*. (2008)
17. McDougal, A. y Squires, D. Cómo elegir y utilizar software educativo. Madrid: Morata (2001).
18. Clarke, P.; Pete, M.; Naidoo, A. Evaluation of Software and its effect on Learning. Project: Kwazulu Concept Burger: HTML mind map / model of evaluation of educational software. En SITE Conference, Orlando, U.S. (1997).
19. Marques, P. Entorno formativo multimedia: elementos, plantilla de evaluación/criterios de calidad (2005).
20. Gonzalez, M. Evaluación de Software Educativo. (Orientaciones para su uso pedagógico). En *Conexiones, Informática y Escuela: Un enfoque global.*, C. Zea Ed. Medellín: Universidad EAFIT (2003).
21. OTA. Power on! New tools for teaching and learning. Ed. Washington D.C.: U.S.Congress, Office of Technology Assesment, SET-379, Government Printing Office (1988).
22. Garcia-Barrera, A. Evaluación de recursos tecnológicos didácticos mediante e-rúbricas. *Revista de Educación a Distancia*. (2016)
23. Straccia, L.; Zanetti, P. Ingeniería de Software: calidad de producto educativo. Disponible en <http://cor.to/gemisnormaES> (2019)
24. International Organization for Standarization. *ISO/IEC Directives Part 2*, Geneva, Suiza. (2018).
25. Quesada Ibarguen, V.; Vergara Schmalbach, J.C. Análisis cuantitativo con WINQSB. (2000).
26. GEMIS, "Evaluación de calidad de software educativo. Planilla de evaluación de producto" Disponible en <http://cor.to/gemisplanilla> (2019)
27. International Organization for Standarization. *ISO/IEC Directives Part 1*, Geneva, Suiza. (2016).

Diseño de plantillas para la creación de actividades educativas con Realidad Aumentada en AuthorAR

Salazar Mesía, Natali¹, Sanz, Cecilia^{1,2}, Gorga Gladys¹

¹Instituto de Investigación en Informática LIDI. Facultad de Informática – Universidad Nacional de La Plata

²Investigador Asociado de la Comisión de Investigaciones Científicas de la Prov. de Buenos Aires

{nsalazar, csanz, ggorga}@lidi.info.unlp.edu.ar

Abstract. En este trabajo se presenta un estudio de librerías de Realidad Aumentada (RA) y de un conjunto de herramientas de autor para RA, lo que ha dado lugar al diseño y desarrollo de plantillas de actividades educativas con esta tecnología, en una herramienta de autor llamada AuthorAR. Se inicia con el análisis de una selección de librerías de RA donde se consideran los tipos de reconocimiento, detección y seguimiento que posibilitan, sus licencias, plataformas para desarrollo y versionado. Luego, se presenta una revisión de herramientas de autor que hace referencia a su uso en el ámbito educativo, en particular con actividades educativas con RA. A partir de este estudio, se comparan y se analizan funcionalidades y aspectos no considerados en estas herramientas para luego, proponer el diseño de plantillas para la creación de actividades educativas con RA. En particular, se ha visto la necesidad de incluir en las herramientas de autor, facilidades para la creación de actividades educativas que incorporen geoposicionamiento y reconocimiento del rostro. El artículo presenta el diseño de plantillas para la creación de este tipo de actividades, a partir del uso de determinadas librerías estudiadas. Estas plantillas se integrarán en la herramienta de autor AuthorAR.

Keywords: Realidad Aumentada, Actividades Educativas, Librerías de Realidad Aumentada, Herramientas de Autor, Plantillas de actividades educativas con Realidad Aumentada.

1 Introducción

La utilización de las Tecnologías de la Información y la Comunicación (TIC) ha ido integrándose en los procesos de enseñar y aprender. Muchos docentes trabajan para incorporar diferentes tipos de actividades que involucren tecnología en el aula, con el objetivo de mejorar tanto la motivación de los estudiantes como el tratamiento de contenidos. La Realidad Aumentada (RA) es una tecnología prometedora utilizada frecuentemente en diferentes áreas como turismo, medicina, entrenamiento militar, etc. Particularmente, en el ámbito educativo, el uso de la RA permite incluir actividades de exploración de nuevos contenidos o actividades de repaso para analizar contenidos vistos, entre otros [1]. Su utilización en este campo ha aumentado en los

últimos años. Los recursos involucrados en su implementación son cada vez menos costosos, gracias a la evolución de la tecnología. Un ejemplo de ello es la mejora continua en las funcionalidades que ofrecen los dispositivos móviles.

En [2] se realiza un análisis de 57 experiencias educativas que utilizan RA en dispositivos móviles. Algunos de los resultados obtenidos identifican el crecimiento exponencial del uso de esta tecnología en educación, y confirman los presentados en el *New Media Consortium Horizon Report* [3]. También se destaca como resultado, que la tendencia es realizar desarrollos para sistemas operativos móviles como iOS y Android.

Por otro lado, las experiencias educativas utilizan aplicaciones con diferentes niveles de RA. La aplicación puede involucrar RA de nivel 0, donde el reconocimiento parte de un código QR, de nivel 1 basado en marcadores, de nivel 2 basado en la ubicación o de nivel 3 basado en la visión aumentada [4]. En particular, aquellas basadas en la localización trabajan sin marcadores. Detectan la posición del usuario con la ayuda de: GPS, acelerómetro, brújula digital y la superposición de objetos virtuales sobre los espacios físicos.

En [5] se describe una experiencia con una aplicación de RA basada en geolocalización referida a la reconstrucción de un campo de concentración destruido después la segunda guerra mundial con el propósito de avanzar en el campo del patrimonio digital. El área que utiliza la aplicación en la actualidad no tiene ningún rastro de ese campo. Para la reconstrucción se utiliza información de documentos y registros históricos. La experiencia se llevó a cabo con dos grupos de estudiantes que utilizaron la aplicación de RA en dos modalidades: uno de ellos, realizó una exploración libre y el otro grupo experimentó un recorrido guiado. Los autores valoran positivamente la experiencia, con algunos resultados ventajosos en la exploración libre.

En [6] se presenta una experiencia de enseñanza geolocalizada de proyectos urbanos con el objetivo de incorporar modelos virtuales 3D de edificios generados por los estudiantes y visionarlos sobre el lugar. Los resultados obtenidos fueron positivos dado que los estudiantes encontraron útil la tecnología de RA y el software utilizado, a pesar de no tener conocimientos previos acerca de las aplicaciones utilizadas.

En [7] se describe una experiencia que utiliza la aplicación TraceReaders. Se localizan puntos de referencia que los estudiantes identifican mediante dispositivos móviles y pueden aumentar la información del lugar con el objetivo de mostrar los pasos de investigación para resolver una situación problemática.

En [8] se describe una aplicación de RA para estudiantes de Anestesiología que, a través del reconocimiento del rostro, permite determinar el grado de dolor del paciente en una escala de 1 a 10. El docente evalúa si el grado de dolor calculado coincide con el sugerido por el estudiante.

En este artículo se describe el diseño de dos plantillas de actividades educativas con RA que incluyen el geoposicionamiento y el reconocimiento de rostro, de manera que puedan aprovecharse estas posibilidades para experiencias educativas. Estas plantillas se diseñan a partir de un análisis llevado a cabo desde tres ejes: el análisis de librerías de RA, de herramientas de autor que posibilitan crear contenidos de RA y de una clasificación de actividades educativas tomadas de diferentes autores y presentada en [9].

De aquí en más este artículo se organiza de la siguiente manera: en la sección 2 se describen y analizan un conjunto de las librerías de RA seleccionadas, en la sección 3 se analizan 6 herramientas de autor seleccionadas a partir de la literatura para revisar sus posibilidades para la construcción de actividades educativas de RA. En la sección 4, se discuten aspectos aún no considerados o escasamente atendidos en las herramientas de autor y en las librerías de RA. En la sección 5, se especifica la funcionalidad de AuthorAR y se presenta el diseño de dos nuevas plantillas para la creación de actividades educativas con RA y su incorporación en esta herramienta de autor. Dichas plantillas se diseñan considerando los estudios previos. Finalmente, en la sección 6, se presentan la discusión y las conclusiones de este trabajo.

2 Librerías de Realidad Aumentada

Las librerías de RA son específicas para el desarrollo de aplicaciones que integran la RA. Presentan diferentes características como el tipo de reconocimiento, la forma de aumentar la escena con elementos virtuales y la plataforma de ejecución.

En [9] y [10] se presenta el conjunto de librerías analizadas a partir de una selección de criterios tecnológicos que permiten clasificarlas y obtener resultados útiles para el desarrollo de aplicaciones con RA. Aquí se resumen algunos resultados. Se seleccionaron 13 librerías: Vuforia, Wikitude, EasyAR, LayAR, KudanAR, ARMedia, DeepAR, MaxST, ARToolKit, XZIMG, VisionLib, ARKit y ARCore. Es importante señalar que, más allá de la amplia variedad de librerías de RA, se seleccionan aquellas que resultan representativas, poseen diferentes características y se evalúan a partir de los siguientes criterios: documentación y versionado, tipos de reconocimiento, detección y seguimiento, plataforma de ejecución (dispositivos móviles o PC) y tipos de licencias que posee (libres, comerciales o educativas).

En la figura 1 se pueden observar los tipos de reconocimiento que caracterizan a las librerías seleccionadas.



Fig. 1. Gráfico sobre tipos de reconocimiento de las librerías seleccionadas. Extraído de [9].

En general, se observa que sólo el 15% cuenta con reconocimiento facial, mientras que alrededor del 62% cuenta con los reconocimientos de marcadores, imágenes y

objetos 3D, que resultan más comunes. El 38% tiene la posibilidad de geolocalización.

Las licencias de software son generalmente comerciales, sólo un 15% posee licencias educativas, mientras que el 46% ofrece licencias libres con características más limitadas y marca de agua en las producciones.

En cuanto a la plataforma de ejecución, el 15% son específicas para dispositivos móviles, mientras que otras resultan híbridas y pueden ejecutarse tanto en PC como en dispositivos móviles.

Este análisis permite determinar para cada aplicación de RA, la/s librería/s más apropiada/s para su desarrollo, dado que cada librería posee puntos fuertes, pero no todas reúnen todas las características.

3 Herramientas de Autor

Las herramientas de autor son sistemas que permiten la creación de aplicaciones informáticas sin necesidad de poseer conocimientos avanzados de programación. Además, en [11] se presenta la siguiente definición: “las herramientas de autor son aquellas que permiten, mediante un proceso más o menos complejo de compilado, la generación de un programa que funciona independientemente del software que lo generó”, y en un sentido más específico, se puede incluir dentro de este grupo a “todas aquellas herramientas que permiten generar actividades, materiales y recursos en formato multimedia”.

Las herramientas de autor generalmente cuentan con facilidades para organizar actividades o interconectar diferentes componentes que permiten adecuar el contenido a los objetivos, los conocimientos y habilidades que se busca desarrollar. La posibilidad de participar en decisiones de diseño, sin necesidad de conocimientos de programación, y a partir de plantillas prediseñadas, convierte a las herramientas de autor en instrumentos de uso cada vez más frecuente en el ámbito educativo.

En [12] se realiza un análisis comparativo de herramientas de autor para la creación de contenidos con RA. Las conclusiones de este trabajo indican que la mayoría no se orientan a la construcción de materiales y/o actividades educativas, sino que se centran en la tecnología RA. En consecuencia, carecen de opciones tales como la presentación de la consigna de la actividad y el *feedback* a los estudiantes, ambas relevantes en los procesos educativos.

En [13] se realiza un análisis de herramientas de autor que posibilitan crear contenidos de RA con objetivos comerciales y educativos. Se distingue la forma en que incorporan RA en el diseño y desarrollo de actividades educativas. Particularmente, se señala el tipo de reconocimiento y el tipo de actividad que plantean. Las conclusiones de este trabajo destacan que estas herramientas permiten a personas que no están familiarizadas a usar este tipo de tecnología, acercarse fácilmente a ésta. De este modo, se constituyen en un aspecto clave para ayudar a la RA a ganar popularidad en diferentes áreas. Sin embargo, no se describen los aspectos específicos para el diseño de actividades educativas.

Profundizando los trabajos previos, se revisan a continuación 6 herramientas de autor que permiten generar contenidos de RA. Para la revisión se tienen en cuenta los

siguientes aspectos: plataformas sobre las que se ejecuta, licencias, actividad de RA que posee y sus características principales. Las herramientas seleccionadas fueron consideradas por el tipo de reconocimiento que ofrecen, su utilización en experiencias educativas y la documentación disponible. Las herramientas seleccionadas son: Aumentaty Author¹, AppyPie², ZapWorks³, AugmentedClass⁴, ActionBound⁵ y AuthorAR [14]. En la Tabla 1 se muestra un resumen de este análisis.

Tabla 1. Herramientas de autor con Realidad Aumentada.

Nombre	Características principales	Plataforma	Licencias	Actividad de RA
Aumentaty Author	Es web para la creación de contenido con técnicas <i>drag-and-drop</i> y móvil para la visualización	Windows y Mac, iOS y Android	Gratuita y comercial	Actividades exploratorias con reconocimiento de códigos QR, marcadores, imágenes y geolocalización
AppyPie	Cuenta con un programa para que los estudiantes diseñen aplicaciones con RA	iOS y Android	Gratuita y comercial	Reconocimiento de imágenes, Geoposicionamiento, visor panorámico y de video 360
ZapWorks	Tiene dos partes para la creación de RA: diseño de escenas y plantillas prediseñadas	iOS y Android	Comercial, gratuita y educativa	Dispone de marcadores propios para aumentar la escena a través de formatos multimedia: imagen, video, botones, galerías
AumentedClass	Crea contenido en base a múltiples marcadores que interactúan con objetos virtuales	Android	Versión Beta de acceso gratuito	Permite crear proyectos con diferentes actividades asociando imágenes o códigos QR con recursos multimedia
ActionBound	Diseñada para crear 'mapas del tesoro' y visitas interactivas	Windows y Android	Comercial, Gratuita y educativa	Pregunta-Respuesta, información sobre el objeto apuntado o misiones que debe completar
AuthorAR	Presenta diferentes plantillas que incluyen presentación de la consigna de la actividad y <i>feedback</i>	Windows y Android	Creative Commons Educativa	Presentación, exploración y estructuración de frases basado en marcadores.

En el siguiente apartado se realiza una discusión de los aspectos analizados en relación a las librerías y herramientas de RA.

¹ Aumentaty Author. Disponible en <http://www.aumentaty.com/>

² AppyPie. Disponible en <https://www.appypie.com/vr-ar-app-builder>

³ ZapWorks. Disponible en <https://zap.works/>

⁴ AugmentedClass. Disponible en <http://www.augmentedclass.com/>

⁵ ActionBound. Disponible en <https://en.actionbound.com/>

4 Discusión sobre librerías y herramientas de autor de RA

Como ya se ha presentado, las herramientas de autor permiten construir aplicaciones sin tener conocimientos en informática, se orientan al público en general, y suelen guiar a través de plantillas o guías paso a paso la tarea a realizar. En cambio, las librerías conforman un conjunto de funcionalidades orientadas a ayudar al profesional informático a abstraerse de las capas de desarrollo de más bajo nivel necesarias para crear una aplicación, en este caso de RA.

Las herramientas de autor suelen integrar las funcionalidades de diversas librerías de RA, que aportan a la creación de las actividades/contenidos que estas herramientas posibilitan.

De las 6 herramientas de autor analizadas, 4 de ellas permiten asociar un recurso multimedia tal como objeto 3D, video o imagen a un marcador o a un código QR, es decir posibilitan la creación de actividades exploratorias principalmente. Luego, solo 2 de ellas, AumentedClass y AuthorAR, permiten crear una actividad que incluya más de un marcador en simultáneo. Aumentaty author, AppyPie y ActionBound poseen la funcionalidad de crear actividades basadas en geoposicionamiento, aunque éstas no se orientan específicamente al ámbito educativo.

Además, tampoco se encuentran herramientas de autor relacionadas directamente al ámbito educativo, que posibiliten crear actividades con reconocimiento de rostro. Sin embargo, pueden ser de utilidad para conocer sobre las emociones de los estudiantes, aumentando la realidad con información sobre esto, también para aumentar con contenidos del área de Medicina y Ciencias Naturales, por ejemplo. También ha sido empleada en dinámicas como la presentada en EPRA⁶.

Como se mencionó en la primera sección, la utilización de la geolocalización y reconocimiento del rostro, se realizan en experiencias ad-hoc y específicas en su desarrollo. Las herramientas de autor si bien permiten crear este tipo de actividades sin tener conocimientos específicos de programación, no han explorado esta área para la creación de actividades educativas con estas tecnologías.

En cuanto al análisis de las 13 librerías de RA se destacan las posibilidades de reconocimiento que brindan. Dos de ellas son específicas para el reconocimiento de rostro, mientras que 4 poseen el geoposicionamiento como parte de sus características principales. El reconocimiento basado en marcadores lo poseen casi todas las librerías y se destaca el reconocimiento sin marcador en 6 de ellas.

Respecto a las licencias, solo 2 librerías son de código abierto mientras que las otras ofrecen licencias comerciales, gratuitas limitadas o educativas para su uso.

Las plataformas de ejecución varían de acuerdo con la librería, pero la mayoría permite su uso en dispositivos móviles con los sistemas operativos iOS y Android.

Estas librerías favorecen crear diferentes actividades de RA, pero requieren de la *expertise* de los profesionales informáticos para integrarlas en una aplicación. Por esto, se considera que este estudio puede servir para poder crear una herramienta de autor que pueda considerar la creación de actividades basadas en diferentes tipos de reconocimiento, que pueden ser interés para el escenario educativo. Por ejemplo, de las experiencias analizadas en la sección 1, muchas de ellas basadas en geoposicionamiento, se realizan en general a partir de desarrollos ad-hoc. Resulta de

⁶ EPRA. Disponible en <https://epra.info.unlp.edu.ar/>

interés aportar al diseño y desarrollo de herramientas de autor orientadas a crear contenidos y/o actividades educativas con RA y con estos tipos de reconocimiento. En la próxima sección se describe AuthorAR y el diseño de dos nuevas plantillas que se integrarán para ofrecer las posibilidades antes mencionadas.

5 AuthorAR y el diseño de nuevas plantillas de actividades educativas con RA

5.1 Descripción de AuthorAR

La herramienta de autor AuthorAR está diseñada para la creación de actividades educativas con RA. Ha sido creada dentro de un proyecto institucional en el III-LIDI en el marco de una tesis de maestría [14]. Se caracteriza por la creación de proyectos que incluyen actividades educativas con RA, permite también la inclusión de otros recursos multimedia para generar contenido interactivo.

Particularmente, se ha planificado su diseño en dos aplicaciones separadas:

- Editor de proyectos con actividades de RA, llamado AuthorAR. Se utiliza para la creación de proyectos de RA con diferentes actividades educativas.
- Visualizador (*player*) de las actividades. Posibilita al estudiante/destinatario la visualización de las actividades, previamente creadas, en un proyecto de RA desde la herramienta de autor.

AuthorAR ofrece plantillas para el diseño de las actividades, donde se van completando datos tales como el objetivo de la actividad, su consigna, utilizando diferentes recursos multimedia (Texto, Audio, Imagen y video). Al momento se cuenta con dos tipos de plantillas, pero no presenta la posibilidad de crear actividades educativas basadas en geoposicionamiento y en reconocimiento del rostro. Por esto, en este trabajo se presentan, a continuación, dos nuevas plantillas que se han diseñado para su integración en AuthorAR.

5.2 Plantilla de actividad educativa basada en geoposicionamiento

Esta plantilla de actividad educativa se basa en el geoposicionamiento a partir del uso de dispositivos móviles. El objetivo es posibilitar la creación de juegos de preguntas y respuestas, donde las respuestas se basan en la geolocalización del estudiante participante. La configuración de la actividad contempla tres etapas:

- 1) Se registra el nombre de la actividad, se describe la consigna que el estudiante debe seguir y se definen los objetivos para los cuales fue creada.
- 2) Se selecciona la cantidad de preguntas y la cantidad de espacios físicos diferentes (geoposiciones) que intervienen en el juego. Se establece un máximo de cinco preguntas y debe tener la misma cantidad de ubicaciones físicas diferentes para seleccionar. Las selecciones realizadas habilitan la configuración de las preguntas y la descripción de cada posición. Cada una de las preguntas debe llevar el texto de la pregunta y dos respuestas posibles. Cada respuesta se asocia a una geoposición diferente, y representan la

respuesta correcta y la incorrecta. En cada una además se debe seleccionar un recurso multimedia que puede ser un texto, imagen, video u objeto 3D que se muestra como *feedback* cuando el participante llega a la posición. Además, se deben describir las posiciones mediante un nombre y las coordenadas de latitud y longitud. Se brinda la opción de acceder a Google Maps para asociar las coordenadas.

- 3) Se define el orden de las preguntas y el lugar de inicio del juego en unas coordenadas específicas, donde se encontrará con la presentación descrita en la primera etapa (objetivos, consigna, etc.) en conjunto con la primera pregunta.

En la figura 2 se muestra el formulario que se debe completar para agregar una geoposición a la actividad.

Fig. 2. Plantilla de geoposicionamiento para agregar una nueva geoposición que se asocia a la actividad.

Una vez finalizada la configuración de la actividad, el docente exporta el proyecto con el conjunto de actividades creadas en la herramienta de autor. El visualizador es una aplicación móvil que aún está en desarrollo y posibilitará que los estudiantes tengan acceso al proyecto con las actividades de RA creadas.

En la ejecución, el primer paso es seleccionar el juego basado en geoposicionamiento con el nombre específico configurado de la actividad.

El estudiante debe ingresar su nombre y un correo electrónico para registrar su jugada. Luego debe posicionarse en el lugar de inicio para empezar el recorrido. Se muestra, posteriormente, la consigna y los objetivos del juego. Luego se continúa con la primera pregunta, que tiene dos opciones de respuesta que los lleva a ubicarse en dos espacios físicos distintos. Ejemplo: si se encuentran ubicados en el hall central del edificio y las opciones de respuesta hacen referencia a la biblioteca para la respuesta A y al patio para la respuesta B, entonces el estudiante debe ir al lugar asociado a la respuesta que considera correcta. Una vez ubicado en el lugar asociado a la respuesta, se muestra un botón “estoy aquí” para que la aplicación le devuelva el *feedback* respecto a su elección. Si la respuesta es incorrecta, aparece el recurso multimedia seleccionado con la indicación de que vuelva a la ubicación anterior para continuar con el juego. En el caso que la respuesta sea correcta, se muestra el *feedback* y accede a la próxima pregunta para avanzar en el juego hasta completar todas las preguntas de la actividad.

Los aumentos de la escena real debieran guardar relación con el contexto físico en que se presenta la respuesta para que el concepto de RA tome sentido.

Esta plantilla se realiza utilizando las librerías de RA Vuforia y layAR para la visualización de los objetos aumentados en la escena real y el geoposicionamiento.

Cuenta con una base de datos que registra la actividad realizada y las respuestas dadas según el nombre ingresado. Además, se utiliza la API de GoogleMaps para la localización de los espacios físicos seleccionados.

5.3 Plantilla de actividad educativa basada en reconocimiento del rostro

Esta plantilla se basa en el reconocimiento del rostro para mostrar contenido aumentado con RA. Tiene como objetivo aumentar diferentes partes del rostro con información aumentada que pueden ser imágenes u objetos 3D. Esta plantilla puede ser aplicada a áreas de Ciencias Naturales y Medicina, entre otras.

La plantilla permite incluir preguntas con dos posibles respuestas que le permita indagar algún tipo de conocimiento previamente tratado al reconocer una parte del rostro.

La configuración de esta actividad se divide también en etapas, en este caso en dos: en la primera etapa se ingresa el nombre de la actividad, la consigna y se definen los objetivos para los cuales fue creada.

En la segunda etapa se seleccionan las partes del rostro. En este caso, la plantilla permite configurar las siguientes zonas del rostro: frente, ojos, nariz, mejillas, boca y mentón. Se seleccionan las zonas que se van a utilizar y se habilita la configuración de cada una.

Para cada una de las zonas del rostro se permite asociar un aumento a partir de un recurso multimedia o una pregunta. En el caso de ser una pregunta se habilitan las dos respuestas con los recursos multimedia asociados como *feedback*.

De igual manera que en la configuración de la plantilla basada en geoposicionamiento, al finalizar se exporta el proyecto a la aplicación móvil donde los estudiantes tienen acceso para realizar la actividad.

Los pasos a seguir por el estudiante que utiliza una actividad generada con esta plantilla son: registrar un nombre y correo electrónico en la aplicación y seleccionar la actividad a realizar. Luego, se revisa la consigna y los objetivos previstos para esta actividad y, por último, se trabaja con la cámara frontal del dispositivo móvil para visualizar el rostro del estudiante que ejecuta la aplicación.

Se dispone de un menú sándwich para seleccionar la parte del rostro con la que quiere interactuar. No se muestran todos los elementos juntos para evitar la superposición de recursos multimedia. Se seleccionan las zonas una por una, y se aumenta la escena con el recurso multimedia o la pregunta asociada y el *feedback* correspondiente a la respuesta elegida.

Este tipo de plantillas toma como antecedente el estudio realizado con el material educativo digital EPRA para el reconocimiento del rostro [15]. Se busca mejorar tipo de reconocimiento que utiliza EPRA, y la plataforma de ejecución con el uso de librerías que permiten su visualización desde dispositivos móviles, como DeepAR.

6 Discusiones y Conclusiones

En general, la RA no es una tecnología de uso común en las aulas, pero en los últimos años se observa un incremento. Para planificar actividades educativas con

estas tecnologías, es necesario disponer de recursos como notebooks o computadoras de escritorio con cámaras, o con dispositivos móviles, si la aplicación lo permite. Si bien esta puede ser una barrera, la imposibilidad de realizar ajustes a los contenidos que ya se ofrecen de RA, se constituye en una dificultad también. En [16] se menciona la dificultad de docentes e investigadores no informáticos para apropiarse de estas tecnologías.

Uno de los aportes que tiene la utilización de actividades educativas con RA en el aula es la motivación y la posibilidad de abordar e interactuar con contenidos que pueden resultar abstractos o difíciles de comprender. También recrear escenas no existentes en la realidad, y fomentar el aprendizaje en contextos significativos. En los análisis realizados por parte de experiencias que han utilizado RA se observa una mejora en la comprensión de los contenidos, la ejercitación y el análisis de los problemas planteados [10]. Es por esto, que resulta de interés tender puentes para llegar a que los docentes planifiquen actividades educativas que tomen ventaja de estas tecnologías. Se cree entonces, que el uso de herramientas de autor como AuthorAR ayudará a eliminar las barreras en relación a la creación de contenido y actividades educativas con RA que se ajusten a las necesidades de los docentes.

Referencias

1. Saltan, F., Arslan, Ö. (2017). The Use of Augmented Reality in Formal Education: A Scoping Review. *Eurasia Journal of Mathematics, Science and Technology Education*, 13(2), 503-520. <https://doi.org/10.12973/eurasia.2017.00628a>
2. Herpich, F., Nunes, F. B., Petri, G., & Tarouco, L. M. R. (2019). How Mobile Augmented Reality Is Applied in Education? A Systematic Literature Review. *Creative Education*, 10, 1589-1627. <https://doi.org/10.4236/ce.2019.107115>
3. Becker, S. A., Cummins, M., Davis, A., Freeman, A., Giesinger, C. H., y Ananthanarayanan, V. (2017). *NMC Horizon Report: 2017 Higher Education Edition*. Austin, TX: The New Media Consortium.
4. Lens-Fitzgerald, M. (2009). Sprxmobile, augmented reality hype cycle. Descargado de http://realidadaugmentadaparaestudiantes.blogspot.com.ar/2017/04/niveles-de-la-realidad-aumentada_29.html
5. Pacheco, D.; Wierenga, S.; Omedas, P.; Oliva, L.S.; Wilbricht, S.; Billib, S.; Knoch, H.; Verschure, P.F.M.J. (2015). A Location-Based Augmented Reality System for the Spatial Interaction with Historical Datasets. In *Proceedings of the 2015 Digital Heritage, Granada, Spain, 28 September–2 October 2015*; IEEE: Washington, DC, USA, 2016; pp. 393–396.
6. Redondo, E., Sanchez Riera, A., Fonseca, D., y Navarro, I. (2014). Enseñanza geolocalizada de los proyectos urbanos. *Nuevas estrategias educativas con ayuda de dispositivos móviles. Un estudio de caso de investigación educativa*, 8(24), 11–32.
7. Kyza, E.A., Georgiou, Y. (2016). Digital tools for enriching informal inquiry-based mobile learning: the design of the TraceReaders location-based augmented reality learning platform. In *Proceedings of the 3rd Asia-Europe Symposium on Simulation & Serious Gaming (VRCAI '16)*. ACM, New York, NY, USA, 195-198. DOI: <https://doi.org/10.1145/3014033.3017432>
8. Xiang, L., Mingbao, Z., Yidi, L., Chuanqing, L. (2019). Application of Augmented Reality Technology for Anesthesiology Major. 4th International Conference on Contemporary Education, Social Sciences and Humanities (ICCESSH 2019). *Advances in Social Science, Education and Humanities Research*. July 2019. ISBN 978-94-6252-752-2

9. Salazar Mesia, N., Sanz, C. y Gorga, G. (2019). Posibilidades de las librerías de Realidad Aumentada en el desarrollo de actividades educativas. XI Congreso sobre Tecnología en Educación & Educación en Tecnología (TE & ET) (San Luis, 2019).
10. Salazar Mesia, N. (2019). Análisis comparativo de librerías de realidad aumentada. Sus posibilidades para la creación de actividades educativas. Descargado de: <http://sedici.unlp.edu.ar/handle/10915/76545>
11. Gómez Villa, Franco Morales, A., Martínez Valenzuela, J., Pastor Marín, P., Marín Saorín, S., Reyes Camacho Marín, S. y Villalba del Baño. J. (2002). "Herramientas de Autor e integración Curricular: 'Las Aventuras de Topy', una aplicación multimedia para el desarrollo de la comunicación alternativa y aumentativa en el aula". Actas del II Congreso Nacional de Nuevas Tecnologías y Necesidades Educativas Especiales, Murcia, España.
12. Moralejo, L. (2014). Análisis comparativo de herramientas de autor para la creación de actividades de realidad aumentada. Descargado de <http://hdl.handle.net/10915/38497>
13. Roberto, R., Paulo Lima, J., Mota, R., Teichrieb, V. (2016). Authoring Tools for Augmented Reality: An Analysis and Classification of Content Design Tools. 9748. 237-248. 10.1007/978-3-319-40406-6_22.
14. Moralejo, L., Sanz, C., Pesado, P., y Baldasarri, S. (2013). Authorar: Authoring tool for building educational activities based on augmented reality. En 2013 international conference on collaboration technologies and systems (cts) (p. 503-507).
15. Salazar Mesia, N., Gorga, G., y Sanz, C. (2016). Experiencia de enseñanza de programación con realidad aumentada. Jornadas de la Enseñanza Universitaria de la Informática JENUI 2016. Almería, España (3300-008612/18-000), 213-220.
16. Garzón, J., Acevedo, J. (2019). Meta-analysis of the impact of Augmented Reality on students' learning gains. Educational Research Review, Volume 27, 2019, Pages 244-260, ISSN 1747-938X.

Diseño Instruccional para fortalecer las competencias digitales en ingresantes: el caso de la FaCENA-UNNE

Fernandez, Mirta G.^{1,2}; Godoy Ma. Viviana²; Barrios, Walter G²; Mariño Sonia I.²

¹ Facultad de Artes, Diseño y Ciencias de la Cultura, Juan B. Justo 365. 3500. Chaco- Argentina.

² Departamento de Informática. Facultad de Ciencias Exactas y Naturales y Agrimensura. 9 de Julio 1449. 3400. Corrientes- Argentina
Universidad Nacional del Nordeste.

{mirtagf@hotmail.com, mvgy@yahoo.com, simarinio@yahoo.com, waltegarbarrios@yahoo.com.ar, }

Resumen: Las Instituciones de Educación Superior (IES), además de su rol tradicional y preponderante en el desarrollo de las sociedades democráticas, en el progreso de la ciencia y la tecnología y en el avance cultural, deben favorecer la inclusión de sus actores plenamente en la sociedad de la información, del conocimiento y propiciar la innovación. Para esto último, es importante planificar y desarrollar acciones que fomenten la adquisición de las competencias mediáticas, digitales e informacionales necesarias para la inserción de los nuevos profesionales en un mundo donde dominan cada vez más las Tecnología de la Información y la Comunicación (TIC). El trabajo presenta una propuesta educativa orientada a facilitar el acceso a las TIC y la formación en su uso, destinada a los ingresantes de una Unidad Académica de una IES. Cuyo objetivo fue introducir a los destinatarios en un abordaje articulado con otros dos módulos, de Pensamiento matemático y de Competencias de lectura y escritura. Consta de una breve reseña de lineamientos a seguir, el análisis para su implementación, su aplicación y evaluación de los resultados basado en las prescripciones metodológicas consensuada por diversos autores del Modelo ADDIE.

Palabras Clave: Gestión de competencias, Competencias digitales, Educación Superior, Modelo ADDIE

1 Introducción

Uno de los reportes de referencia sobre tendencias en Tecnología Educativa para la Educación Superior [1] insiste en que uno de los desafíos importantes que impiden la adopción de tecnología en este nivel educativo tiene que ver con la “la mejora de la competencia digital”. Lo cual no se trata sólo de proporcionar acceso a las TIC sino de formar para una utilización adecuada [2].

El término de competencia digital es relativamente nuevo y su conceptualización varía dependiendo del organismo u organización que la formula [3]. Sin embargo, la literatura afirma que entraña el uso seguro y crítico de las tecnologías de la sociedad de la información para el trabajo, el ocio y la comunicación [4].

Una definición más amplia y que se aproxima al marco de este trabajo es la que expone a la “*Competencia Digital como la combinación de conocimientos, habilidades y capacidades, en conjunción con valores y actitudes, para alcanzar objetivos con eficacia y eficiencia en contextos y con herramientas digitales*” [5] y [6], y se expresa en términos de cinco dimensiones a las que el alumnado debe aspirar y los docentes promover:

- i. La dimensión del **aprendizaje** abarca la transformación de la información en conocimiento y su adquisición.
- ii. La dimensión **informacional** abarca la obtención, la evaluación y el tratamiento de la información, en entornos digitales.
- iii. La dimensión **comunicativa** abarca la comunicación interpersonal y la social.
- iv. La dimensión de la **cultura digital** abarca las prácticas sociales y culturales de la sociedad del conocimiento y la ciudadanía digital.
- v. La dimensión **tecnológica** abarca la alfabetización tecnológica y el conocimiento y dominio de los entornos digitales.

Con el objeto de coadyuvar a ingresantes en la adquisición de un conjunto de competencias para el tránsito por la universidad, se realizó en la Facultad de Ciencias Exactas y Naturales y Agrimensura (FaCENA) de la Universidad Nacional del Nordeste (UNNE), al inicio del ciclo lectivo 2019, una propuesta de trabajo articulada que les permitió introducirse en nociones de:

- Competencias digitales para el tránsito en la universidad
- Pensamiento matemático
- Competencias de lectura y escritura.

Para ello, se estableció una planeación de las diversas decisiones técnicas, didácticas y pedagógicas, dirigidas a la concreción de los objetivos formulados. En la implementación se desarrollaron encuentros presenciales combinados con actividades virtuales para extender y reforzar la experiencia. Se apoyó de esta manera el desarrollo articulado de los tres módulos. Se instrumentó una encuesta para la valoración de los resultados; se analizaron los mismos y se compartieron en una reunión de puesta en común con los tutores, docentes y gestores académicos participantes de la propuesta, con la intención de retroalimentar y aplicar mejoras a futuro del proyecto iniciado.

2 Competencias digitales en Educación Superior

Las instituciones educativas concuerdan en que las competencias digitales básicas se han convertido en herramientas fundamentales para el desarrollo humano y académico [6]; en [7] se afirma que “...*en todos los territorios, el uso de las nuevas herramientas permite y facilita el aprendizaje de cualquier tópico...*”. Por lo cual adquisición las mismas (aquellas competencias digitales mediacionales que sirven de base para adquirir otras competencias) constituyen hoy una condición indispensable para lograr que los individuos alcancen un pleno desarrollo personal, social y profesional que se ajuste a las demandas de un mundo globalizado, promueva el desarrollo económico-social y que “...*los prepare para desenvolverse en las sociedades del conocimiento...*” [3].

En la actualidad, los niños y jóvenes realizan un profuso uso de las TIC, especialmente basados en dispositivos móviles y redes sociales [7] y [8]. No obstante, diversos estudios profundizan la hipótesis que afirma que, a la hora de su utilización en tareas formales, tales como la búsqueda de información, comparación, pensamiento crítico, fuentes confiables de información digital, etc., los jóvenes demuestran desconocimiento o cierta dificultad en estos aspectos [9], [11], [12] y [13].

En este sentido, se coincide con el metaanálisis [13] que indica que, no hay que hacer suposiciones sobre las competencias y conocimientos digitales de los aprendices cuando ingresan a la Educación Superior. En general, no se destina un tiempo específico en el calendario académico para trabajar de forma explícita estos conocimientos; más bien, se introducen en las metodologías de trabajo del aula y se demandan como competencias que deberían tenerse previamente o adquirirse en el transcurso de la formación [15].

2.1. Planeación educativa para alcanzar resultados esperados

De acuerdo con [16], “... *planear es un acto de inteligencia cuyo propósito es racionalizar la selección de alternativas para el futuro. Implica un ejercicio libre de la razón para definir con claridad los fines a los que se orienta la acción, y desentrañar los mejores medios para alcanzarlos...*”.

La planeación educativa, se encarga de especificar los fines, objetivos y metas de la educación. Se refiere a planear sobre una actividad a realizar para obtener los mejores resultados con el mínimo de tiempo y de recursos. Además, implica la interacción de diversas dimensiones y actores; y prever si las metas planteadas se van logrando desde la propuesta curricular y si esta responde a los requerimientos del contexto y de cada miembro de la comunidad educativa [17].

Diversos son los marcos metodológicos que proveen prescripciones en la planeación educativa. Bates en su libro “La enseñanza en la era digital” propone ADDIE (Analizar-Diseñar-Desarrollar-Implementar-Evaluar) [18].

Para ser llevada a la práctica, se ayuda de la didáctica, es decir, del conjunto de técnicas que se emplean en la enseñanza. A su vez estará influida por múltiples factores: el contexto de la educación basada en las TIC en el que se aplicará. Si se utiliza un entorno virtual de aprendizaje, será necesario conocer sus capacidades, potenciales y sus limitaciones, así, por ejemplo, habrá que conocer si se pueden desarrollar trabajos en grupo, si hay posibilidad de plantear debates, si se pueden realizar un determinado tipo de actividades y de prácticas, los tipos de interacción que pueden darse, etc. [19].

2.2. ¿Cuándo utilizar ADDIE?

La competencia digital deberá ser considerada como una de las competencias profesionales fundamentales en el perfil del egresado universitario [13]. Para ello es importante planificar y desarrollar acciones que fomenten la adquisición de las competencias mediáticas, digitales e informacionales necesarias para la inserción de los nuevos profesionistas en un mundo donde dominan cada vez más las TIC.

ADDIE es un modelo utilizado por muchos diseñadores instruccionales profesionales para la enseñanza basada en la tecnología, particularmente para casos que involucran proyectos complejos. Dicho modelo, ha sido casi un estándar para los programas de educación a distancia de alta calidad desarrollados por especialistas ya sea online o impresos y ahora está siendo utilizado por muchas instituciones para el rediseño de las clases magistrales masivas, el aprendizaje híbrido, y para los cursos totalmente online [18].

3 Marco metodológico:

El presente trabajo, de carácter empírico mediante la sistematización de las fases abordadas, describe de manera cualitativa los aspectos relevantes en la aplicación de ADDIE [18], considerando:

- i. **Analizar:** identificar todas las variables que deben tenerse en cuenta al diseñar el curso, tales como las características del alumno, el conocimiento previo de los alumnos, los recursos disponibles, etc.
- ii. **Diseñar:** esta etapa se centra en la identificación de los objetivos de aprendizaje para el curso y cómo se crearán y diseñarán los materiales (por ejemplo, puede incluir la descripción de las áreas de contenido que se deben incorporar y el guion gráfico que defina qué contenidos se presentarán en formato de texto, audio y video y en qué orden), y decidir sobre la elección y el uso de la tecnología.
- iii. **Desarrollar:** la creación de contenidos, que incluye la decisión si será tercerizado, la tramitación de los derechos de autor para los materiales de terceros, la carga de contenido en un sitio web o LMS, etc.
- iv. **Implementar:** se refiere a la modalidad de distribución del curso, que incluye cualquier formación previa o capacitación del personal que brinda apoyo al estudiante, y evaluación de los alumnos;
- v. **Evaluar:** la retroalimentación y los datos se analizan para identificar las áreas que requieren mejoras y que se considerarán en el diseño, desarrollo y puesta en práctica de la siguiente edición del curso.

Las etapas antes mencionadas, se consideraron como un marco adecuado para sustentar las cuestiones referidas al proyecto formulado.

4 Resultados

4.1. Actividades en las etapas iniciales (análisis del contexto)

Se indagó las características, condiciones y necesidades del contexto educativo, del educando, de los recursos disponibles y requeridos. Se reunió información de cantidad de alumnos ingresantes, su distribución por carrera, días y horarios de cursado, espacios físicos disponibles, conectividad de red Wifi, cantidad de docentes tutores participantes, se designaron coordinadores y se especificaron los requerimientos de plataforma virtual de apoyo para la realización de actividades extendidas.

La Unidad Académica, en su ciclo lectivo 2019 posee aproximadamente 1300 estudiantes inscriptos en primer año, casi un 90 % participaron del proyecto (en tanto que porcentaje restante, son alumnos que reinciden en el cursado del primer año); distribuidos por carrera, como se muestra en la **Tabla 1**.

Tabla 1. Cantidad de ingresantes por carrera, en 2019, de la FaCENA

Carrera	Cantidad de alum.
Profesorado en Biología	138
Licenciatura en Ciencias Biológicas	96
Profesorado en Física	17
Licenciatura en Ciencias Físicas	51
Ingeniería en Agrimensura	57
Ingeniería en Electrónica	109
Ingeniería Eléctrica	92
Profesorado en Matemática	88
Licenciatura en Matemática	29
Bioquímica	167
Licenciatura en Ciencias Químicas	23
Profesorado en Ciencias Químicas y del Ambiente	38
Licenciatura en Sistemas de Información	327
Total	1232

Para las actividades presenciales y de acción tutorial del módulo Competencias Digitales participaron diecisiete tutores y un coordinador, del plantel docente del departamento de Informática, distribuidos de a pares en comisiones de cuarenta alumnos.

Se definió que el aula virtual debía concentrar a todos los ingresantes en un solo espacio. Para lo cual se planteó la posibilidad de configurar “aula anidada”, una opción que provee Moodle, en la que todos los usuarios acceden mediante una misma interfaz, pero que a su vez pueden estar distribuidos en diferentes grupos con una clave de acceso diferenciada. Dicha configuración se fundamenta en que las notificaciones y la tutorización de las actividades se realizan por comisiones individuales y que posteriormente puede efectuarse análisis conjunto mediante las herramientas disponibles en Moodle. La selección de esta plataforma se basó en la decisión que institucionalmente desde la Universidad, ha sido adoptada por UNNE Virtual¹ y por lo tanto los estudiantes tendrán contacto con ella a lo largo de distintas asignaturas de su carrera.

4.2. Actividades de diseño de la propuesta educativa

Basados en la intencionalidad educativa, se delinearon objetivos de logros de aprendizajes para los estudiantes, para ello se convocó a reuniones con autoridades del área académica y estudiantil, y docentes de la facultad, y se especificaron:

i. Objetivo general

- “Introducir al alumno en competencias digitales para su desenvolvimiento apropiado en la universidad”, articulando las actividades con los demás módulos para ingresantes, que se desarrollaron en forma simultánea.

ii. Objetivos específicos

- Proporcionar información general de las vías de comunicación que posee la facultad.
- Desarrollar competencias básicas en el uso adecuado de herramientas de interacción educativa. En particular de las plataformas SIU Guaraní, Aula Virtual en Moodle y el sitio Institucional de la FaCENA.
- Promover el sentido de pertenencia a una institución universitaria.

Se considera que, a partir de los objetivos y de la experiencia con tales herramientas, se favorecerá al alumno en su inserción y permanencia en la universidad y en el desarrollo del sentido de pertenencia con la institución a la que se incorpora.

Por otra parte, cuando se pensó en el “entorno de aprendizaje” se lo concibió como las diversas ubicaciones físicas, contextos y culturas en las que los estudiantes aprenden. Para ello, se promovió un

¹ <http://virtual.unne.edu.ar/web2015/index.php>

formato de aula extendida a través del entorno virtual como apoyo a los encuentros presenciales. Se diseñó una estructura simple, en la cual los estudiantes pudieran explorar con facilidad los materiales, actividades y foros, principalmente. El prototipo de aula en Moodle se muestra en la **Fig. 2**.

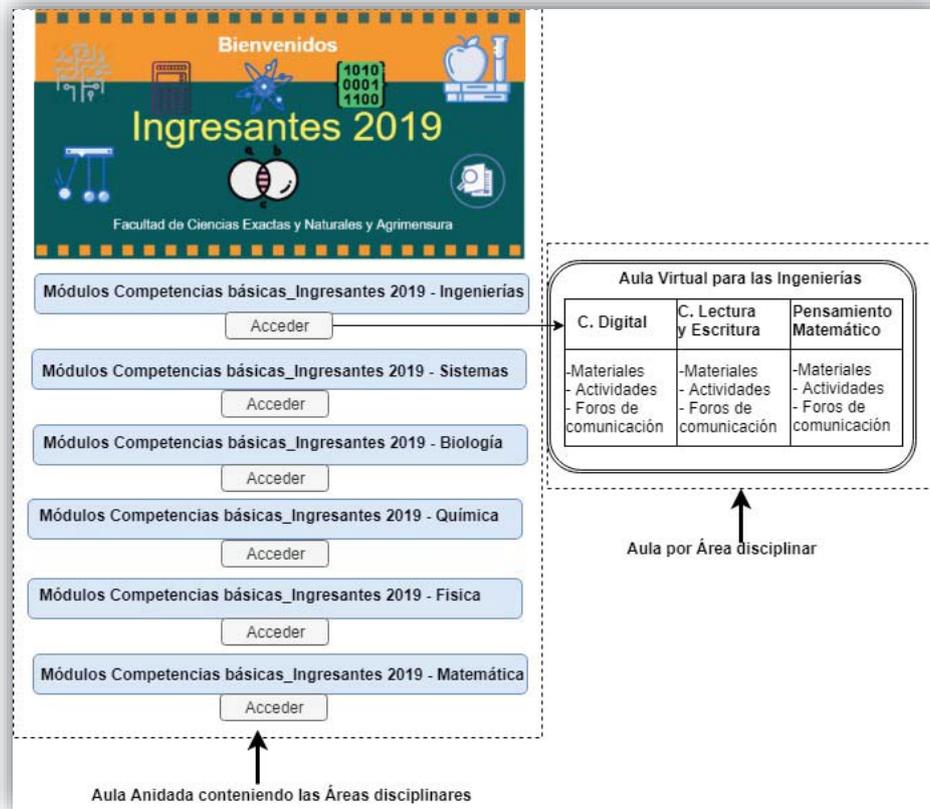


Fig. 2. Prototipo de aula virtual en Moodle

Se consensuó el prototipo con los coordinadores de los otros módulos y en encuentros sucesivos se comunicó las responsabilidades, en la administración de las comisiones de alumnos y la articulación de toda la propuesta.

4.3. Desarrollo de contenidos

En el desarrollo de los contenidos se elaboraron materiales especiales para cada temática, como apoyo. En la **Fig. 3**, se muestra la interfaz de aula virtual, así como los materiales incluidos.

Además, se incluyeron objetivos, foros, actividades, información y enlaces a la web de la facultad, que pudieran resultar de interés para los alumnos.

Competencias digitales

Objetivos del Módulo "Competencias Digitales"

Desarrollar habilidades en el manejo de herramientas de interacción educativa. En particular el uso de SIU Guarani, el Aula Virtual en Moodle y el sitio Institucional de la FaCENA.

¿Qué es el SIU Guarani- UNNE?

Es un sistema de gestión
ON LINE
en el que podrás...

¿Que es el Aula Virtual Moodle- UNNE?

y mucho más...
Es un espacio de
Interacción
con tus compañeros
y profesores;
puedes acceder a
material
de estudios de tus
asignaturas...

Sitio Intitucional FaCENA- UNNE

<http://exa.unne.edu.ar/alumnos/competencia-basica.php>

Fig. 3. Interfaz desarrollada para el espacio virtual en Moodle.

Se elaboraron tutoriales instruccionales e informativos acerca del sistema SIU Guarani y el aula virtual en Moodle, en formato video y documento de texto, para su descarga (Fig. 4.).

Bienvenido

Facultad de Ciencias Exactas y Naturales y Agrimensura

Web FaCENA	SIU Guarani	Aula Virtual
http://exa.unne.edu.ar	http://guarani.exa.unne.edu.ar	http://virtual-moodle.unne.edu.ar/

Competencias Digitales

Facultad de Ciencias Exactas y Naturales y Agrimensura
Lic. Fernández Mirta

Fig. 4. Video informativo e instructivo disponible en web <https://www.youtube.com/watch?v=nipOS0ZbYI4>

Asimismo, se creó en la web <http://exa.unne.edu.ar/alumnos/competencia-basica.php>, un área de consulta para todos los materiales de los módulos.

En el caso de los módulos Pensamiento Matemático y Competencias Lectoras, se utilizaron materiales digitales, especialmente diseñados para un programa formativo de la UNNE denominado “Nexos²” del subprograma “Universidad y Escuela Secundaria” aprobado por la Secretaría de Políticas Universitarias dependiente del Ministerio de Educación de la Nación [20], [21] y [22].

4.4. Desarrollo de la acción formativa (implementación)

En lo que refiere a la organización, se coordinó la utilización de los espacios áulicos, el acceso a internet, el apoyo técnico para la disposición de utilitarios, se establecieron mecanismos de acceso a las aulas y las correspondientes comisiones. Se realizó la difusión del calendario de las aulas y horarios; en las clases inaugurales se presentaron autoridades y docentes representantes de las distintas carreras.

La implementación se llevó a cabo de manera presencial en el mes de febrero con actividades virtualizadas que se extendieron hasta el mes de abril. El trabajo se centró en elucidar cuestiones reiteradas, según las temáticas:

- Principales recursos para utilizar en los módulos en la plataforma Moodle de la universidad: gestión de usuarios, accesos, foros y tareas.
- En relación a la gestión de información académica en el entorno SIU Guaraní, se puntualizó en las consultas más frecuentes recogidas desde el Centro de Estudiantes y el área Gestión Estudios; vinculadas con la administración de usuarios, fechas de cursado y de exámenes, obtención de constancias, consulta de historia académica, etc.
- Además, se abordó el uso de la Web institucional de la Facultad, como canal oficial de comunicación y de obtención de información para la realización de trámites tales como:
 - requisitos para la inscripción definitiva del ciclo lectivo,
 - becas,
 - guía de trámites (solicitud de libreta universitaria, legalización de programas de asignaturas aprobadas, carga horaria de asignaturas, reconocimiento de asignaturas aprobadas en otras carreras o facultades, reinscripción anual obligatoria, etc.)

4.4.1. Realización de actividades virtualizadas.

Se desarrollaron en el espacio virtual de Moodle, diseñado para tal fin. En una primera instancia, los estudiantes, participaron en un foro de presentación, en el que comentaron sus expectativas respecto a la carrera que eligieron. En una segunda, se abocaron a las tareas dispuestas por los demás módulos y se atendieron consultas acerca de las herramientas y recursos del aula virtual, como una primera aproximación a su utilización.

4.4.2. Criterios generales de evaluación para las actividades

En relación a la evaluación de las actividades se definieron los siguientes criterios de acuerdo a los objetivos:

- Considerar la participación tanto individual como grupal, en los distintos módulos a lo largo de todo del proceso de enseñanza.
- Valorar la adquisición de conocimientos con un tratamiento didáctico de las consultas más frecuentes, a través de mecanismos de interacción y retroalimentación.
- Realizar tareas voluntarias propuestas, como parte de los módulos.

² http://www.bnm.me.gov.ar/giga1/normas/RCFE_321-17.pdf

4.5. Evaluación de la propuesta

Mediante la disposición de encuestas en línea, se recogieron aspectos de interés para el mejoramiento de la propuesta. Los instrumentos diseñados se aplicaron a todos los participantes en el aula virtual y con respecto a todos los módulos. Sin embargo, para la finalidad del presente trabajo, se exponen los valores obtenidos para el módulo de Competencias Digitales. Estas permitieron recopilar información acerca de la utilidad del mismo desde el punto de vista de los estudiantes y docentes, detectar las dificultades, indagar y comprender experiencias en el marco del ingreso a la universidad.

4.5.1. Resultados de encuestas de estudiantes

Cuando se indagó acerca de la experiencia en general, los estudiantes calificaron en un **87,6%**, como **Útil** y **Muy Útil**, la implementación de los módulos; y sólo un **12,4%** opinó lo contrario.

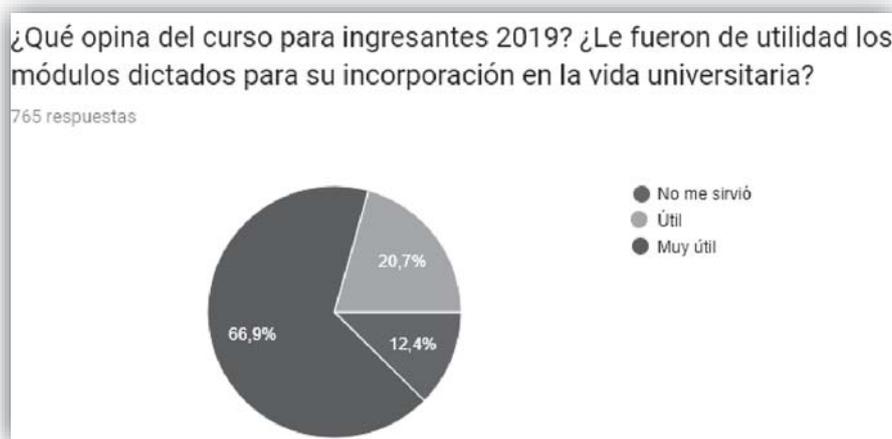


Fig. 5. Utilidad de los módulos desarrollados según los estudiantes

Al solicitar que califiquen los módulos, respecto a Competencias Digitales un **79,47** respondió que **Buena y Excelente**. En tanto que sólo un **19,6** opinó **Regular**.

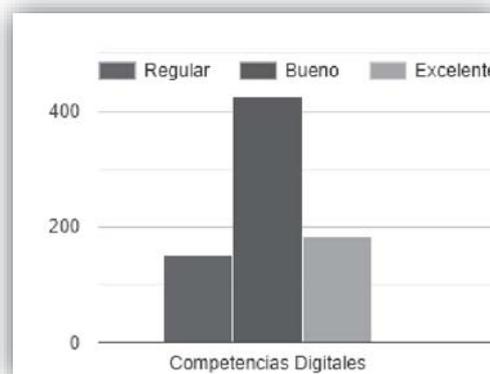


Fig. 6. Opinión de los alumnos del módulo de Competencias Digitales

Además, se consultó sobre la adecuación de la duración de los módulos; el 20% de los alumnos se inclinaron por el módulo de Competencias Digitales, tal como se muestra en la **Fig. 7**.



Fig. 7. Apreciación de los alumnos respecto a la duración de los módulos

Otra de las cuestiones que se indagó, fue acerca de los **contenidos** (expectativas, profundidad y claridad), en que en general los alumnos respondieron positivamente en un promedio del **80%**.



Fig. 8. Experiencia de los alumnos respecto contenido desarrollado en Competencias Digitales.

Desde un punto de vista cualitativo, los alumnos fueron consultados acerca de **¿Qué aspectos de los módulos se deberían mejorar?**, refirieron entonces con mayor frecuencia de aparición a las siguientes cuestiones:

- “Mayor profundidad de contenidos, por ejemplo, en el Módulo Pensamiento matemático”.
- “Deberían iniciar con mayor anticipación al cursado de las asignaturas”.
- “Podrían haber sido un poco más exigentes/más dinámicos”
- “Dividir en comisiones más pequeñas”.
- “Promover instancias de interacción entre ingresantes”.
- “Mayor extensión horaria de los módulos”.

4.5.2. Resultado de encuesta a docentes

Se llevó a cabo también, la instrumentación de una encuesta abierta, de autoevaluación dirigida a los docentes y se consultó acerca de inconvenientes, fortalezas, sugerencias y otras observaciones (situaciones particulares, tanto académicas como disciplinares, entre otras).

En relación con los inconvenientes, surgieron cuestiones vinculadas a la cuestión edilicia. Dadas las altas temperaturas que acontecen en época estival en la zona, se notó una insuficiencia de los equipos de ventilación en algunas aulas. Otras de las cuestiones mencionadas y a mejorar, fue la señal de internet que, pese a los esfuerzos, se vieron saturadas por la utilización masiva. Otros docentes, no indicaron dificultades algunas.

Acerca de las fortalezas, la mayoría de los docentes subrayaron:

- "Excelente coordinación y organización"
- "Muy buena predisposición de los alumnos, mucho interés en el desarrollo de las actividades"
- "El trabajo en equipo del plantel docente"
- "El material ofrecido desde la coordinación"
- "La distribución de las comisiones y duración de los talleres"

En cuanto a las sugerencias y observaciones, se centralizaron en:

- "Mejorar las conexiones y accesibilidad a internet dentro de las aulas".
- "Mejorar la distribución de alumnos en comisiones más reducidas a fin de brindar atención más personalizada a las consultas de alumnos".

4.5.3. Análisis de la evaluación

Las valoraciones obtenidas indican la repercusión de la acción formativa en los estudiantes, que en general respondieron positivamente.

Una de las consideraciones observadas a partir de la revisión de los resultados y que se destaca, es la actitud y buena predisposición de los estudiantes respecto a la realización de los módulos.

Por otra parte, uno de los puntos a reconsiderar está vinculado al cronograma de distribución de las actividades, dado que los módulos se extendieron con el objeto de ofrecer la posibilidad de desarrollar las actividades posteriores a los encuentros presenciales, lo cual resulta desacertado en algún punto para los alumnos, al superponerse al cursado de las primeras asignaturas.

A su vez, la contrastación de las apreciaciones tanto de los estudiantes como de los docentes indican cierta coherencia en la experiencia y los resultados, por lo que se deduce que los instrumentos son válidos a fin de retroalimentar la propuesta.

5 Conclusiones

Los sistemas de educación y formación deben generar nuevas competencias digitales, con ello lograr mejores oportunidades de incorporación y adaptación al campo laboral desde los inicios en la educación / Educación Superior.

A partir del trabajo y coincidentemente con la literatura revisada, los estudiantes poseen las destrezas básicas de manejo, sin embargo, han desarrollado escasa reflexión sobre la importancia de las tecnologías en aspectos formales a su vida universitaria futura. En la sociedad actual, la competencia digital debe ser un aprendizaje múltiple, global e integrado de las distintas formas y lenguajes de representación, mediante el uso de diferentes tecnologías, en diferentes contextos y situaciones de interacción social.

El conocimiento de las herramientas digitales de comunicación que serán utilizadas durante su carrera académica presenta múltiples ventajas para los alumnos. Para aprovechar estas ventajas se comenzó a trabajar en la temática de formación por competencias, en este caso que aquí se presenta, las digitales. Estas constituyen el pilar, para el logro de otras como: las del aprendizaje, las informacionales, la comunicativa, la cultura digital y las tecnológicas.

En este sentido, el trabajo presenta una experiencia de trabajo articulado con otras temáticas consideradas transversales para los perfiles de formación de la unidad académica en virtud de aportar de manera transdisciplinaria al desarrollo de las dichas temáticas.

Además, es posible resaltar que diversas carreras de las disciplinas Informática, como las comprendidas por el CONFEDI (Consejo Federal de Decanos de Ingeniería) tienden a redactar sus planes de estudios basados en el enfoque por competencias. Lo expuesto se sustenta, que en el ámbito universitario se presenta como un nuevo paradigma de aprendizaje, basado no solo en el marco teórico, sino también en el saber hacer, y en el saber ser.

Referencias

1. NMC, (2019). EDUCAUSE Horizon Report 2019 Higher Education Edition. Recuperado de: <https://library.educause.edu/-/media/files/library/2019/4/2019horizonreport.pdf?la=en&hash=C8E8D444AF372E705FA1BF9D4FF0DD4CC6F0FDD1>
2. Gros Salvat, B., & Contreras, D. (2006). La alfabetización digital y el desarrollo de competencias ciudadanas. *Revista Iberoamericana de Educación (OEI)*, 2006, num. 42, p. 103-125.
3. UNESCO, 2011. Educación de calidad en la era digital. <http://www.unesco.org/new/fileadmin/MULTIMEDIA/FIELD/Santiago/pdf/educacion-digital-Buenos-Aires.pdf>
4. Comisión Europea. Dirección General de Educación y Cultura. (2007). Competencias clave para el aprendizaje permanente: Un Marco de Referencia Europeo.
5. Marquès, P. (2009). Aportaciones sobre el documento puente: Competencia digital. Disponible On line: <http://www.pangea.org/peremarques/competen.htm>.
6. Freire, J., & Brunet, K. S. (2016). Políticas y prácticas para la construcción de una Universidad Digital. *La cuestión universitaria*, (6), 85-94.
7. Piscitelli, A. (2008). Nativos digitales. *Contratexto*, (016), 43-56.
8. UNESCO, Las competencias digitales son esenciales para el empleo y la inclusión social. <https://es.unesco.org/news/competencias-digitales-son-esenciales-empleo-y-inclusion-social>, Accedido el 3 de Julio de 2019.
9. Chiecher, A. C., & Melgar, M. F. (2018). ¿Lo saben todo? Innovaciones educativas orientadas a promover competencias digitales en universitarios. *Apertura (Guadalajara, Jal.)*, 10(2), 110-123.
10. Cortés, J. M., Orozco, G. L., Rodríguez, E. M. O., & Luna, V. R. (2015). Impacto de las Competencias Digitales en el Desarrollo Académico del Estudiante Universitario. *Revista Iberoamericana para la Investigación y el Desarrollo Educativo* ISSN: 2007-2619, (10).
11. Cabra Torres, F. & Marciales-Vivas, G. P. (2009). Mitos, realidades y preguntas de investigación sobre los 'nativos digitales': una revisión. *Universitas Psychologica*, 8(2), 323-338.
12. Bautista, G., Escofet, A., Forés Miravalles, A., López Costa, M., & Marimon Martí, M. (2013). Superando el concepto de nativo digital. Análisis de las prácticas digitales del estudiantado universitario
13. Edel, Rubén & Iram Zúñiga Lobato, José. (2016). Competencias digitales y educación superior. Número temático: Educación mediada por tecnología: alternativas digitales y virtuales. *Revista de Transformación Educativa*. 1. 158-189.
14. Sharpe, R. (2010). Conceptualizing differences in learners' experiences of e-learning: A review of contextual models. Reporte de la Higher Education Academy LearnerDifference (heald) Synthesis Project.
15. Márquez, E. F., Olivencia, J. J. L., & Meneses, E. J. L. (2017). Formación en competencias digitales en la universidad. Percepciones del alumnado. *Campus Virtuales*, 6(2), 79-89.
16. Villareal, E. (1980). "La planeación académica integral", en cuadernos de planeación universitaria, Dirección General de Planeación, UNAM, México.
17. Díaz-Barriga Arceo, Frida, Coaut (1990). Metodología de diseño curricular para educación superior, México. Trillas. 175 p. 23 cm.
18. Bates, A. W. (2015). La Enseñanza en la Era Digital. Una guía para la enseñanza y el aprendizaje. BC Campus. <https://laescuelacolaborativa.files.wordpress.com/2017/07/la-ensec3blanza-en-la-era-digital.pdf>, Accedido el 12 de junio de 2019.
19. Williams, P., Schrum, L., Sangrà, A., & Guàrdia, L. (2004). Fundamentos del diseño técnico-pedagógico. Modelos de diseño instruccional en e-learning.
20. Nexos: UNNE y la Educación Secundaria y las jurisdicciones de Chaco y Corrientes, su articulación desde una perspectiva regional para la democratización del acceso a la educación superior Institución: Ministerio de Educación. PROYECTOS DE ARTICULACIÓN UNIVERSIDAD – ESCUELA SECUNDARIA (Resolución N° 4462-2017-APN-SECPU#ME) del Programa "NEXOS: Articulación y Cooperación Educativa" (Resolución Ministerial 3117-E/2017). Aprobado por Res. 2016-2055-E-APN-SECPU#ME
21. Saiz, I. E., Gorostegui, E. N., Sosa, J. J., Musimessi, O., Farías, L. R., & Vallejos, M. E. (2015) MATEMÁTICO. Accedido desde http://exa.unne.edu.ar/alumnos/docs/UNNE_pensamiento-matematico.pdf
22. Lineras, A., Ayala, Y., Capovilla, N., Musimessi, O., Farías, L. R., & Vallejos, M. E. (2015). Aprendizaje y Estudio en la Universidad. Accedido desde http://exa.unne.edu.ar/alumnos/docs/UNNE_de_lectores_y_escritos.pdf

Empoderando Personas a Través del Uso de Internet - VIDUC

Gustavo Calcaterra, Fatima Romero y María E. García-Díaz

Universidad Nacional de Asunción-Facultad Politécnica

Asunción, Paraguay

gcalcaterra@pol.una.py, flromero@est.pol.una.py, mgarcia@pol.una.py

<http://www.pol.una.py>

Resumen Existe un porcentaje significativo de personas con acceso a Internet o con capacidad económica de acceder a la red que no lo utiliza de una manera consciente para mejorar su calidad de vida; esto ocurre por distintas razones, entre ellas el conocimiento de uso, impacto y la aceptación cultural así como la adopción local.[5]

Internet es una herramienta poderosa para aprender y desarrollar habilidades de cualquier índole como cocina, deportes, manualidades; solo por citar algunas. Mediante este trabajo se busca descubrir una forma en la que se pueda potenciar y empoderar a las personas a través del uso de Internet de manera educativa, consciente y útil.

Para poder experimentar la idea, se desarrolló una primera versión de la plataforma web, VIDUC (Videos Educativos, <http://viduc.net>), cuyo objetivo es mostrar el potencial de Internet para aprender. La plataforma filtra y ordena contenidos específicos de videos de Youtube, mostrando contenidos que pueden ser de interés a los usuarios. El experimento está dirigido a personas de cualquier edad que puedan acceder a Internet y que tengan pocos conocimientos sobre como utilizar la red de manera más efectiva.

A través de encuestas realizadas se encontró que a un gran porcentaje de personas le pareció de utilidad la aplicación. Potenciar el uso eficiente de Internet, buscando la educación, el empoderamiento, la autonomía, el desarrollo y la alfabetización digital de las personas es útil, ya que ayuda a disminuir la brecha digital existente, entre otros factores positivos. La plataforma VIDUC es un caso básico que se enfoca en el contenido de videos para aprender nuevas habilidades así como conocer experiencias de otras personas que han podido utilizar sus habilidades para emprender negocios o resolver algunos de sus problemas.

Palabras Claves: Educación Digital · Alfabetización Digital · Herramienta Digital · Brecha Digital · Habilidades Técnicas · Aprendizaje en Red

1. Introducción

Los datos publicados por la Dirección General de Estadísticas, Encuesta y Censo (DGEEC) en setiembre del 2018 confirma que la población juvenil para-

guaya de entre 15 a 29 años de edad representa el 27,79 % de la población total, del cual 1.909.947 personas distribuidas entre hombres y mujeres tienen acceso a Internet, la encuesta demuestra que el 82,48 % del total de personas tiene acceso a una herramienta digital, y el 17,50 % no tiene la posibilidad de acceder a Internet.[1]

Hoy en día en el Paraguay el 96,7 % de la población total cuenta con un teléfono móvil, superando ampliando a la telefonía fija y el 24,8 % tiene acceso a una red de Internet en su hogar, lo cual demuestra el crecimiento a nivel de acceso al mundo web de la población paraguaya. Además, se afirma que un alto porcentaje de esta población accede a las redes sociales.[2]

Así mismo, es importante mencionar que 6 de cada 10 paraguayos se dedican a actividades informales, como sub empleos, lo cual implica que muchas veces estas personas no perciban ni siquiera el salario mínimo y trabajen en condiciones no deseadas, sin tener acceso a los servicios básicos como salud y educación.[12]

Este trabajo surge debido a la problemática observada de que la falta de instrumentos para identificar y segmentar aprendizajes a través de la web, con la finalidad de adquirir conocimientos y habilidades técnicas respecto a oficios en general. Según el BID, el futuro del trabajo está marcado por dos grandes tendencias, el “tsunami” tecnológico y el envejecimiento poblacional, para América Latina y el Caribe. Ambas tendencias tienen una naturaleza positiva (nos dan la posibilidad de vivir más años, abandonar los trabajos más repetitivos y aumentar nuestra calidad de vida) y presenta una gran oportunidad para la región. El futuro del trabajo no es un escenario predefinido, sino una realidad en construcción. El mercado laboral del mañana en nuestra región depende, en realidad, de cómo actuemos en todo los niveles: los estados, las empresas, los trabajadores, las escuelas, etc. [3]

Se estima que la propuesta permitirá advertir no solo el potencial de las TIC en los procesos de enseñanza y aprendizaje sino también el amplio camino de ensayo y prueba que los docentes pueden recorrer en bien de las experiencias de formación profesional de los estudiantes. [4]

El resto del documento está estructurado de la siguiente manera: La Sección 2 presenta información de antecedentes y trabajos relacionados en el área. La Sección 3 presenta el experimento y la solución planteada. La sección 4 presenta la validez de los datos y la interpretación de los resultados. Finalmente, en la sección 5 se presentan las conclusiones y algunas propuestas futuras para la continuidad de esta investigación.

2. Estado del Arte

En esta sección se detallan los trabajos relacionados con la propuesta y que sirven de antecedentes para fundamentar la importancia de realizar este proyecto.

Internet for All A Framework for Accelerating Internet Access and Adoption

Este documento, que es parte del pilar Acceso/Adopción de la iniciativa Desafío Global del Foro Económico Mundial en el Futuro de Internet, provee un marco

de referencia para tratar las barreras relacionadas con lograr un Internet para todos. La solución que plantea se divide en estos 4 puntos sobre como atacar la brecha digital: a) Infraestructura, b) Costo, c) Conocimiento de uso e impacto y aceptación cultural y d) Adopción local y uso. Los puntos más relevantes para esta investigación son los puntos c y d.[5]

Desafíos de la Educación Digital

La nueva Ley de Protección de Datos (Ley Orgánica 3/2018, de 5 de diciembre) que rige en la Unión Europea indica que, además de adaptar la normativa interna en materia del derecho fundamental de protección de datos personales al RGPD (Reglamento General de Protección de Datos), recogió un amplio catálogo de “derechos digitales”, entre los cuales se encuentra uno de particular relevancia para hacer frente a la revolución tecnológica que está llamando a la puerta: el derecho a la educación digital, regulado en el artículo 83 del citado texto normativo. Explica el redactor, de acuerdo al artículo, el sistema educativo garantizará la plena inserción del alumnado en la sociedad digital y que el aprendizaje del uso de los medios digitales sea seguro.[6]

Los vídeos que animan al estudio en YouTube crecen un 120 % en España durante 2018

Los vídeos publicados en España desde la plataforma YouTube con temática relacionada con el estudio aumentaron en un 120 % su número de visualizaciones durante el pasado año 2018 con respecto a los datos de 2017. Los vídeos de YouTube que incluyen en su título los términos ‘estudia conmigo’ o ‘*study with me*’, que suelen ser las búsquedas más frecuentes de los estudiantes, crecieron un 120 % entre 2017 y 2018, según datos compartidos en el estudio. Además, siete de cada diez personas que acceden a YouTube lo hacen buscando aprender algo.[7] Esta publicación fundamenta con mayor fuerza la propuesta de trabajo que se presenta en este artículo.

YouTube y su popularidad como plataforma educativa

En esta plataforma se puede encontrar contenidos educativos de materias como física, matemáticas, tecnología e inglés y sus picos más altos de popularidad se dan en época de exámenes. El incremento en la popularidad de este tipo de contenidos va de la mano con la idea de la educación continua, ya que pone al alcance el conocimiento de distintas áreas, sin tener que salir de su casa.[8]

El Futuro del Trabajo en América Latina y el Caribe. ¿Una gran oportunidad para la región?

La tecnología y la demografía son dos puntos que tienen un impacto en el presente y en el futuro en cuanto al trabajo. Los autores presentan algunos puntos sobre cómo enfrentar los desafíos, indican que se debe aplicar los cambios a nivel de estado, empresas e individuos, entre ellos mencionan que todos los individuos deben aprender todo el tiempo. Su objetivo es presentar problemáticas e ideas sobre el Trabajo en América Latina y el Caribe, para generar conciencia y acciones en los organismos más importantes. Plantea ideas sobre la automatización, la demografía y la cuarta revolución de la tecnología y cómo esto puede influir en el trabajo de mucha gente.[3]

Impacto y Aprovechamiento de las Tecnologías de la Información y las Comunicaciones en la Educación Superior

En este estudio, se comprobó mediante una encuesta que el Internet es un recurso muy útil a nivel de educación, se encontró que algunos de los factores que contribuyen al uso del Internet son: el uso de la computadora del hogar, la forma divertida de usar la computadora, así como el uso de las redes sociales para chatear, también el uso de la computadora en la escuela y las redes sociales como medio de comunicación. Sin embargo, no puede desconocerse que las redes sociales y vídeos como pasatiempos en horas de ocio y horario académico actúan como distractores negativos.[9]

Construcción de un instrumento para el aprendizaje en red de estudiantes universitarios

Este artículo justifica la investigación basándose en la falta de instrumentos para identificar aprendizajes en la red y desarrollar habilidades de búsqueda de información al interactuar en ese medio, se procura orientar las prácticas de enseñanza de los docentes. En tal sentido, se presentan los datos de una prueba piloto que se realizó para identificar cómo los estudiantes de la Universidad Autónoma de Sinaloa se apropian de los aprendizajes cuando interactúan en Internet y cuáles habilidades desarrollan cuando buscan información en ese medio digital. Los resultados demuestran que aproximadamente 50 % de los jóvenes conectados a la Red autorregula su aprendizaje, más de 90 % se siente motivado cuando busca información en la Web y más de la mitad comprende cómo las TIC fomentan el aprendizaje.[10]

Invertirá YouTube 20 millones de dólares en recursos para EduTubers?

YouTube financiará a EduTubers para crear contenido en temas referentes a habilidades profesionales, como tutoriales para el desarrollo de currículums, cursos de programación de videojuegos o clases de JavaScript. Es indudable que YouTube es un poderoso espacio de aprendizaje para todo tipo de público; en esta plataforma se puede aprender desde cómo reparar una lavadora o cómo cocinar arroz, hasta diseñar modelos en 3D o páginas web. Sin embargo, la mayoría de su contenido está bajo constante crítica por la falta de calidad y control.[11]

Encuesta sobre acceso y uso de Internet en Paraguay

Esta encuesta se realizó por la Secretaría Nacional de Tecnologías de la Información y Comunicación (SENATICs) en el año 2017 en Paraguay, con una muestra representativa de personas de la Capital y de 14 Departamentos por la empresa First Análisis y Estudios. Los resultados tienen un margen de error de $\pm 3,5\%$ y un índice de confianza del 95 %. Entre otros puntos confirma que de las personas distribuidas entre hombres y mujeres del Paraguay, un 86,9 % tiene acceso a Internet a través de una computadora o un celular.[2]

Informalidad laboral desafiante en Paraguay

La situación de la República del Paraguay en cuanto a los indicadores de la población que realizan trabajos informales tomando como referencia el año 2010 al 2016. La informalidad laboral no es "propiedad exclusiva de Paraguay", sino un flagelo que se observa en todo el mundo. Sin embargo, en el país se registran ci-

fras alarmantes, especialmente en las pequeñas y medianas empresas, y en áreas como la construcción, el comercio, el servicio doméstico, entre otros.[12] Por ello es importante resaltar que la plataforma VIDUC puede aportar formación empírica a los usuarios en cuanto a los diversos oficios o intereses y a su vez ser una fuente importante de conocimiento que permita el desarrollo personal apuntando a mejores trabajos, entre ellos formales.

3. Propuesta de Solución

Se implementa un diseño descriptivo al ser una investigación social. Se ha optado por el desarrollo de una plataforma web para ejecutar el experimento aplicando la combinación de técnicas cuantitativas y cualitativas a través de una encuesta que permita reunir una serie de datos útiles que apoyarán el estudio. A continuación se explica en detalle el proceso llevado a cabo, tanto para la elección de la población a ser encuestada, seguidamente los instrumentos utilizados para fundamentar los resultados y el procedimiento aplicado, así como también la plataforma tecnológica utilizada en el desarrollo de la aplicación.

3.1. Arquitectura Tecnológica

La plataforma se elaboró con el Framework CAKE PHP [15]. Se utilizó como contribución al diseño web: Html5, JQuery y CSS. Se desplegó la aplicación en un contenedor Docker [17] sobre *OpenShift* [18] en la Nube. La aplicación se despliega sobre un Apache[19] y utiliza una base de datos MySQL[20]. El diseño utilizado es Modelo - Vista - Controlador (MVC), es orientado a objetos y es una aplicación web. El código fuente se encuentra en GitHub.[22]

Funcionalidades: La funcionalidad principal es llamar a la API de Youtube[21] para mostrarle al usuario videos filtrados de educación que le pueden ser de interés.

Calidad de Código: Se utilizó la plataforma de software libre SONARQUBE [13] para evaluar la calidad del código fuente del Proyecto VIDUC, realizando un análisis estático sobre dicho código, con el objetivo de advertir sobre diferentes puntos a mejorar. En el resultado salió un Ratio de «Deuda Técnica» menor al 10 %, por tal motivo se considera que el proyecto VIDUC está sano.

3.2. Población de la Muestra

La muestra está compuesta de 51 personas sin grado de formación universitaria de ambos géneros, la mayor cantidad 61 % femenino y 39 % masculino, cuyas edades comprenden entre los 15 y 44 años. En cuanto a nivel de formación académica el 78 % es nivel secundario y el 21 % con educación secundaria inconclusa.

Figura 1. Gráfico % Formación Académica de la Muestra



3.3. Instrumentos

Se elabora un prototipo de aplicación *online* denominado VIDUC (Vídeos Educativos) con el fin de que sea usado como herramienta para adquirir nuevos conocimientos y conocer el potencial de utilizar vídeos para aprender. Se puede acceder a través de la url: <http://viduc.net>. Complementando el trabajo realizado se seleccionó la metodología de encuestas que fue direccionado principalmente a los alumnos del programa Tenondera, se explica sobre el programa más adelante. En el cuestionario se analiza una serie de áreas consideradas de relevancia para la investigación (Ver Encuesta). La encuesta fue elegida para recopilar información sobre aspectos difíciles de observar directamente por el investigador. Es un instrumento muy adecuado para desarrollar el conocimiento de la conducta y los procesos educativos, así como para el estudio aplicado a las características de una población reducida.

3.4. Proceso experimental

El procedimiento en general del trabajo para completar el experimento es:

1. Probar y verificar las funcionalidades de la plataforma VIDUC antes de su puesta en producción, que la misma cumpla con los requerimientos pre-establecidos en su configuración.
2. Redactar la encuesta con el objetivo recolectar datos que permitan saber puntos de interés para la investigación. Verificación y corrección de la encuesta en base a los requerimientos finales de la plataforma.
3. Seleccionar la muestra, que fue los alumnos del programa Tenondera, atendiendo que los mismos son de escasos recursos y a través de la utilización de la plataforma VIDUC podrían adquirir conocimientos sobre temas de interés que les ayuden en su crecimiento personal y progreso económico.
4. Preparar el instructivo a seguir para realizar el experimento a fin de evitar la pérdida de información relevante para la investigación.

5. Realizar el experimento en un ambiente controlado (sala de informática de la Facultad Politécnica de la UNA) siguiendo la secuencia de tiempo establecido por los coordinadores del programa Tenonderá.
6. Verificar los datos recolectados, si se completó todos los campos pre-establecidos correctamente.
7. Analizar los resultados obtenidos de acuerdo a los parámetros establecidos en la plataforma VIDUC.

La recolección de datos se llevó a cabo a través del llenado del Formulario *Online Google Forms*, la encuesta se realizó en las instalaciones de la Facultad Politécnica de la Universidad Nacional de Asunción en el marco del Programa denominado TENONDERÁ, que tiene el fin de propiciar el crecimiento académico de los jóvenes de los últimos años de los cursos de la media, en instituciones que no cuentan con suficientes recursos de acceso a tecnología, proporcionándoles opciones de capacitación en las herramientas ofimáticas, demandadas en el mercado laboral[15]. Luego se procedió a la aplicación del experimento el cual tuvo una duración de 30 minutos. Se implementó la prueba, que incluyó algunas indicaciones, preguntas y respuestas y se mostró donde acceder a la encuesta. También se le compartió la aplicación y la encuesta a otras personas externas al grupo Tenonderá para tener más diversidad de personas y de edad. Se aclara que la aplicación y la encuesta está disponible en Internet. Para el análisis de datos se definió un cierre para no mezclar con nuevos datos.

4. Resultados

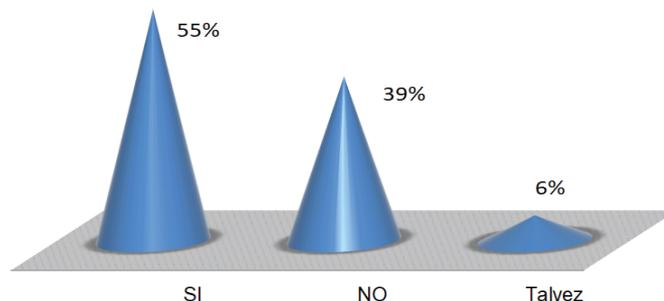
El trabajo se desarrolla en base a la utilización de la aplicación VIDUC (Vídeos Educativos) y mediante el llenado del Formulario *Online Google Form* en donde se toma como referencia preguntas direccionadas sobre la utilización de Internet, y la posibilidad de aprender diversos conocimientos, el cual arroja como resultado que actualmente en Paraguay hay un porcentaje elevado de personas que tienen acceso a Internet, a través de diversos dispositivos, como es el caso de las personas encuestadas mediante la investigación. De un total de 51 encuestados un 63 % representa mujeres y un 37 % varones.

A través de este trabajo se verifica que un importante número de personas le pareció de utilidad la aplicación. Por ende se resalta el hecho de que es útil potenciar un uso saludable de Internet, buscando la educación, el empoderamiento, el desarrollo y la alfabetización digital de las personas para disminuir en algo la brecha digital existente. La plataforma VIDUC es un caso básico que se enfoca en el contenido de vídeos, se podría aplicar mejoras para potenciar otras maneras de compartir contenido como por ejemplo: texto, audio, imágenes, entre otros.

Según la muestra indica que un 94 % de los encuestados tienen acceso a internet lo cual indica un crecimiento importante a nivel país en cuanto a la utilización de herramientas digitales. Se comprobó que un 93.1 % de los encuestados utilizan internet desde su teléfono móvil, y un 78.4 % afirmó que lo utiliza diariamente.

También se pudo comprobar que el uso de Internet se enfoca en la distracción ya que el 78.4 % utiliza en redes sociales y distracciones varias, y sólo el 22.1 %

Figura 2. Gráfico sobre el % referente a la encuesta realizada
Utilizaría VIDUC para aprender cosas nuevas



se enfoca en educación y adquirir nuevos conocimientos. Un razón desfavorable para el uso de Internet podría ser el hecho de que el 78.9% de los encuestados indicaron que no confían en la información encuentran en Internet.

No se puede dejar de mencionar que gran parte de las personas encuestadas tienen la opinión que pueden aprender cosas nuevas; de éstos el 84.3% opinan que sí puede aprender en Internet. Se confirma que el 57.7% afirma que no está seguro que le den un buen uso al Internet, pero mencionan que forma parte de un hábito diario.

En la figura 2 se observa que el 55% utilizaría VIDUC como herramienta para aprender nuevas habilidades. El 79.6% de los encuestados sienten que pueden aprender a través de videos y el 72.2% opta por recibir contenidos a través de videos, capaz por la facilidad de captar la atención y sencillez para comprender los contenidos.

Se confirma de esta manera que el uso de internet es algo que ya es un hábito que se practica en la sociedad del presente para cuestiones relacionadas con la vida diaria e incluso para aprender nuevas capacidades y adquirir nuevas competencias.

5. Conclusiones y Trabajos Futuros

Del análisis realizado, se afirma que el acceso a Internet forma parte de un hábito diario. La cuestión preocupante es que una gran mayoría no enfoca al máximo el aprovechamiento del Internet por distintas razones, desconocimiento de uso, fuente no confiable, desconocimiento de que existe en la web, entre otros. Y por ende no saben filtrar, diferenciar e interactuar con los contenidos que se encuentran disponibles en Internet. Se confirma que hace falta un mejor aprovechamiento de esta herramienta que es Internet, mover a las personas de su zona de confort, inculcar y motivarlas a la utilización más consciente de Internet para que puedan desarrollarse en sus propios temas de interés, problemas y dudas.

VIDUC, como parte de la investigación, es una herramienta que busca empoderar y motivar a las personas a aprender algún oficio, habilidad o técnica a través del uso de Internet. Su objetivo es filtrar contenidos de vídeo sin distracciones para motivar y mostrar que se puede aprender y encontrar contenidos útiles en internet. Las personas encuestadas se sorprendieron de los temas encontrados en la plataforma. Muchos expresaron que tienen intereses pero no saben como canalizar o filtrar estos temas en la Web. Cabe resaltar que la información ya se encontraba en Internet la herramienta solo las filtró y clasificó de Youtube para así captar la atención de los usuarios y mostrarles lo que pueden aprender y poner en práctica.

La proyección de la plataforma VIDUC tendría dos partes y funcionaría como un asistente. La primera le servirá al usuario para encontrar historias inspiradoras con las que se pueda identificar para avanzar e inclusive entre ellas encontrar algún mentor con el que se pueda comunicar. La segunda parte le mostrará cómo utilizar herramientas de Internet para avanzar hacia sus objetivos. A continuación, un ejemplo. El asistente le hace ciertas preguntas al usuario y aprende que esta persona tiene pocos recursos económicos y que le gustaría ser Chef. Entonces la plataforma le presentará la historia de una persona que pudo avanzar en la cocina con circunstancias parecidas al de esta persona para que se sienta inspirada y observe lo factible que son sus aspiraciones. Luego la plataforma le mostrará una serie de maneras de utilizar Internet para aprender habilidades de cocina, como canales de YouTube de cocina, blogs de Chefs, etc. La plataforma también podría tener la capacidad de que los mentores o personas que cuentan su historia, así como las personas que buscan aprender, puedan conectarse entre sí o por lo menos tener los enlaces para poder conectarse en otras plataformas externas.

Referencias

1. Secretaría Nacional de Tecnologías de la Información y Comunicación: DGEEC comparte datos sobre la población juvenil en Paraguay, <https://www.dgeec.gov.py/news/DGEEC-comparte-datos-sobre-la-poblacion-juvenil-en-Paraguay.php>, (2018) Accedido 14 Febrero 2019
2. Secretaría Nacional de Tecnologías de la Información y Comunicación: Encuesta sobre acceso y uso de Internet en Paraguay, <http://gestordocumental.senatics.gov.py/share/s/ntjnuNLeT8u3gbAHC6WeVw>, (2017) Accedido 20 May 2019
3. Bosch M., Pagés, C., Ripani, L.: El futuro del trabajo en América Latina y el Caribe. Banco Interamericano de Desarrollo, pp. 04-10 <http://dx.doi.org/10.18235/0001340> (2018).
4. Tumino, M., Bournissen, J., Forneron, F.: Validación de contenido de instrumentos para medir el nivel de integración tecnológica en el aula y el nivel de impacto en los estudiantes. Libro de Actas CACIC2018, pp. 146-154 <http://cacic2018.exa.unicen.edu.ar/> (2018).
5. World Economic Forum: Internet for All - A Framework for Accelerating Internet Access and Adoption. http://www3.weforum.org/docs/WEF_Internet_for_All_Framework_Accelerating_Internet_Access_Adoption_report_2016.pdf. (2016)

6. Jiménez, R.:Desafíos de la educación digital.Expansion , <https://hayderecho.expansion.com/2019/01/18/desafios-educacion-digital/>.(2019) Accedido 31 Jan 2019
7. Europa Press:Los vídeos que animan al estudio en YouTube crecen un 120 % en España durante 2018 <https://www.europapress.es/portaltic/internet/noticia-videos-animan-estudio-youtube-crecen-120-espana-2018-20190208/142641.html>.(2019) Accedido 30 April 2019
8. Delgado, P.:YouTube y su popularidad como plataforma educativa.Tecnológico de Monterrey <https://observatorio.tec.mx/edu-news/youtube-y-su-popularidad-como-plataforma-educativa>.(2019) Accedido 10 May 2019
9. Alcibar,M.,Monroy,A.,Jiménez,M.:Tecnologías digitales y educación para el desarrollo sostenible. Un análisis de la Producción Científica. Pixel-Bit: Revista de Medios y Educación 1(54), pp. 83-103(2019)
10. Soto, M.: Construcción de un instrumento para el aprendizaje en red de estudiantes universitarios. Revista Iberoamericana para la Investigación y el Desarrollo Educativo, 8(16)(2019) <https://doi.org/https://doi.org/10.23913/ride.v8i16.362>
11. Guijosa, C.:Invertirá YouTube 20 millones de dólaresen recursos para EduTubers.Tecnológico de Monterrey <https://observatorio.tec.mx/edu-news/invertira-youtube-20-millones-de-dolares-en-edutubers>.(2018) Accedido 29 May 2019
12. ABC Color: Informalidad laboral desafiante en el Paraguay, <http://www.abc.com.py/especiales/fin-de-semana/informalidad-laboral-desafiante-1643114.html>.(2017) Accedido 02 Jun 2019
13. SonarSource S.A: Code Quality and Security, <https://www.sonarqube.org/>.(2019) Accedido 15 Jun 2019
14. USERFOCUS: 247 web usability guidelines, <https://www.userfocus.co.uk/resources/guidelines.html>.(2014) Accedido 20 Jun 2019
15. Facultad Politecnica UNA:ProyectoTenonderaIII, <https://www.pol.una.py/sites/default/files/files/extension/ProyectoTenonderaIII.pdf>.(2009) Accedido 02 July 2019
16. Cake Software Foundation, Inc., <https://cakephp.org/>. (2019) Accedido 20 Mayo 2019.
17. Docker Inc., <https://www.docker.com/>. (2019) Accedido 25 Mayo 2019.
18. RedHat OpenShift, <https://www.openshift.com/>. (2019) Accedido 20 Mayo 2019.
19. The Apache Software Foundation, <https://www.apache.org/>. (2019). Accedido 20 Mayo 2019.
20. Oracle Corporation, <https://www.mysql.com/>. (2019). Accedido 20 Mayo 2019.
21. Youtube, <https://www.youtube.com/>. (2019). Accedido 20 Mayo 2019.
22. Código fuente de viduc-cakephp, <https://github.com/gcalcaterra/viduc-cakephp/>. (2019). Accedido 17 Julio 2019.

Estrategias didácticas para la enseñanza de la programación imperativa, utilizando el Robot Educativo Programable

Paula Lencina¹, Mariana Ado¹, Pedro Iglesias², Mónica Sarobe¹, Claudia Russo³

Instituto de Investigación y Transferencia en Tecnología (ITT)⁴
Comisión de Investigaciones Científicas de la Provincia de Buenos Aires (CIC)
Escuela de Tecnología (ET)
Universidad Nacional del Noroeste de la Provincia de Buenos Aires (UNNOBA)

{paula.lencina, mariana.ado, pedro.iglesias, monica.sarobe,
claudia.russo}@itt.unnoba.edu.ar

Abstract. El contexto tecnológico actual le impone permanentemente nuevos desafíos al sistema educativo, el cual debe reestructurar y reorientar sus propuestas pedagógicas a la luz de estos cambios. A la par, la sociedad contemporánea y los estudiantes en formación reclaman nuevas estrategias didácticas que sean acordes con los requerimientos de un mundo que brinda a los individuos una avalancha de información, que no es asimilable de forma sencilla.

Resulta indispensable que las instituciones educativas puedan adaptarse a los requerimientos del mundo actual, formando a las presentes generaciones en las nuevas formas de construcción del conocimiento, como complemento de los contenidos específicos que fueron el eje de la enseñanza hasta el momento.

Keywords: Estrategias didácticas - Interfaz - Robot - Programación Imperativa

1 Introducción

El presente trabajo se desarrolla en el marco de un proyecto de investigación denominado "Informática y Tecnologías Emergentes", aprobado y financiado por la Secretaría de Investigación, Desarrollo y Transferencia (SIDyT) de la UNNOBA en el marco de la convocatoria a Subsidios de Investigación Bianuales 2017 (SIB 2017). Este proyecto es ejecutado por un equipo constituido por docentes e investigadores,

¹ Docente-Investigador ITT

² Becario ITT

³ Investigador Adjunto sin Director CIC

⁴ Centro Asociado CIC

becarios y estudiantes, que se desempeñan en el ITT; y trabaja en busca de atender a la problemática sobre desgranamiento y deserción estudiantil que se presenta en la Escuela de Tecnología (unidad académica de UNNOBA). La ET tiene competencias en las carreras del área de Ingeniería, Diseño e Informática; estando inmersas en esta última área las carreras: Ingeniería en Informática, Licenciatura en Sistemas, Analista de Sistemas y Tecnicatura Universitaria en Soporte Informático.

En ese sentido, y con el objetivo identificar los factores causales de las problemáticas antes mencionadas, en la ET se lleva a cabo un relevamiento de datos de los estudiantes inscritos, matriculados y graduados, cuatrimestre por cuatrimestre, de acuerdo a cada cohorte. Es decir, se realiza un análisis longitudinal y transversal de la totalidad de los años de cursada para establecer así el índice de deserción por cada carrera y determinar su momento más crítico.

De acuerdo a los datos registrados durante el período 2005-2015, y a la indagación sobre los posibles factores causales de la deserción y el desgranamiento, desde el año 2016 en la ET se llevan a cabo diferentes estrategias y acciones con la intención de mejorar o revertir esta situación.

En el siguiente apartado se relatan brevemente estas acciones.

2 Deserción y desgranamiento, estrategias para mitigar sus causas

Existen diferentes acciones orientadas al apoyo y permanencia de los estudiantes durante los primeros años de ingreso a las carreras de grado. En el caso de la unidad académica de la que se está hablando, las mismas se pueden concentrar en los ejes que se describen a continuación:

Taller de ingreso.

Programa de tutorías.

Programas de voluntariados y extensión universitaria.

Investigación.

En el taller de ingreso se intenta articular con el nivel secundario, para facilitar el ingreso de los estudiantes a la universidad, como así también para atender sus necesidades de formación, de modo que puedan transitar con éxito el primer año de los estudios superiores.

Dicha propuesta educativa es de carácter semi-presencial y se divide en módulos que apuntan a trabajar el bagaje que el aprendiz trae consigo, así como también incorporar conceptos y habilidades nuevos. En ese sentido, en cada carrera se articula con alguna de las asignaturas que el estudiante tendrá durante el primer cuatrimestre de su año inicial.

Para el caso de los ingresantes a las carreras del área de informática la articulación se realiza con la asignatura Introducción a la Programación Imperativa (IPI). Es en esta asignatura en la que se interviene con el trabajo que relata el presente paper.

Vale mencionar que con tal articulación se aspira a promover el desarrollo del Pensamiento Computacional para favorecer las habilidades de los estudiantes, y de ese modo tener la capacidad de enfrentar y resolver los problemas de índole académicos, personales y sociales.

De esta manera, se espera que al momento de resolver un problema, puedan poner en juego habilidades de descomposición (descomponer un problema complejo en pequeños subproblemas, cuyas soluciones combinadas otorgan solución al problema general), de abstracción (centrar la atención en las características más relevantes del problema, de modo de captar su esencia y filtrar sus características secundarias para luego crear una representación simplificada del mismo) de reconocimiento de patrones (acudir a problemas similares resueltos con anterioridad) y algoritmo (conjunto de instrucciones claras y precisas, que se identifican y se planifican en un determinado orden para la resolución a un problema)

Por otra parte, en lo que a tutorías se refiere y a nivel unidad académica en la ET, se trabaja con el Programa de Apoyo y Fortalecimiento Académico (PAFA). El PAFA persigue el objetivo de implementar un sistema integral de asistencia a los estudiantes dividido en tres ejes. Estos ejes se denominan: Tutorías para estudiantes de los primeros años (TUPA), Cursos de apoyo para rendir finales de las Asignaturas de los primeros años de las Carreras de la Escuela (CARF) y Estrategias de apoyo para la graduación de los y las estudiantes de informática.

En el eje TUPA, que particularmente atañe al presente trabajo, se busca analizar la problemática de los primeros años de estudios universitarios, con el objetivo de lograr un acercamiento social con los estudiantes, donde se ofrezcan experiencias que enriquezcan el vínculo de éstos con la institución. De este modo, se pretende mejorar la permanencia y retención estudiantil en la ET, al mismo tiempo que se investiguen los factores condicionantes por las cuales los estudiantes universitarios desertan el primero y el segundo año.

En cuanto a los proyectos de voluntariados y extensión, se fomenta la participación de estudiantes y docentes en diferentes programas, capacitaciones y actualizaciones.

Tal es el caso de las capacitaciones en la Fundación Sadosky⁵. La misma reunió a un equipo de docentes universitarios para capacitarse en “La programación y su didáctica” parte uno (1) y dos (2), y propició el dictado de tres capacitaciones a docentes de primaria y secundaria en esta temática. El objetivo principal fue impulsar la enseñanza y el aprendizaje de las Ciencias de la Computación en las escuelas argentinas y, por ende, despertar la vocación en carreras afines en el nivel superior. Particularmente, esta capacitación aborda la estrategia de aprendizaje por indagación utilizando herramientas educativas, lo cual, está relacionado con el presente trabajo.

Otra estrategia ejecutada, concerniente al trabajo de investigación, y, pensando en que, en “la articulación” subyace el concepto de superar compartimentos estancos en educación, de manera que el sistema educativo mejore su calidad, en un marco de diversidad y equidad; es la muestra interactiva del saber y la tecnología, que se organiza con el fin de acercar la labor universitaria a los estudiantes del nivel secundario. En esa línea la ET presenta algunos espacios en los que se visibilizan algunas ofertas académicas

⁵ <http://www.fundacionsadosky.org.ar/presentacion-institucional/>

de la escuela. En el área de informática se ha presentado la actividad “Carrera de programadores”, en la que se invita a los participantes a programar en forma lúdica y con herramientas introductorias a la temática.

Por último, respecto a la investigación, y con el objetivo de continuar avanzando en cuanto al diseño y la evaluación de actividades para la inclusión de las TIC en las actividades dentro del aula, así como también en la búsqueda de disminuir el desgranamiento y la deserción estudiantil, aquí se relata cómo se ha avanzado en relación a estrategias educativas con respecto al paradigma imperativo de la programación de computadoras utilizando como herramienta una interfaz diseñada para el uso del Robot Educativo Programable (REP).

3 Contexto

Como ya se mencionó en este documento el presente trabajo se desarrolla en el marco del proyecto de investigación denominado “Informática y Tecnologías Emergentes”,

Allí mismo un grupo de investigadores ha trabajado en la construcción de un prototipo del REP [1]. Esta labor se llevó a cabo con el fin de contar con una herramienta propia de la universidad para trabajar a futuro en la enseñanza de la programación de computadoras, en el marco de una de las líneas de investigación del proyecto.

4 Desarrollando la herramienta para trabajar programación de computadoras

El proyecto de investigación “Informática y Tecnologías Emergentes”, abarca diferentes líneas de investigación, desarrollo e innovación. Puntualmente en la línea Hombre-Máquina (HCI) se incluye la investigación, diseño, construcción y programación de robots como solución a problemas multidisciplinarios y transversales.

Bajo esta línea, y ampliando el uso de la robótica en educación, se trabaja en busca de estrategias educativas con respecto al paradigma imperativo de la programación de computadoras. Se lo hace utilizando como herramienta una interfaz diseñada para el uso del REP, estudiando particularmente el caso de la asignatura IPI.

El plan de trabajo pertinente a la investigación incluyó varias tareas, entre la que se destaca el diseño de una interfaz para el uso del REP, lo cual se detalla en el siguiente apartado.

4.1 Características de la interfaz

Se ha diseñado una interfaz que favorece un trabajo introductorio en cuanto a conceptos y habilidades de programación.

Al iniciar la aplicación se visualiza una ventana principal dividida en seis paneles, como lo muestra la *Figura 1*:



Figura 1: captura de pantalla de la interfaz

La interfaz para REP trabaja con un pseudocódigo elemental que incluye las sentencias: Avanzar, Detenerse, Girar a la derecha, Girar a la Izquierda (órdenes para el robot); las estructuras de control de decisión y repetitivas de la siguiente manera: If, If-Else, While-Do, For; además facilita la creación y manipulación de variables, constantes y funciones. Con todos estos elementos se pueden abordar los conceptos básicos que deben desarrollarse en el inicio a la programación.

Luego, en el panel que se señala en la *Figura 2*, se visualizará el programa que se va armando del siguiente modo:

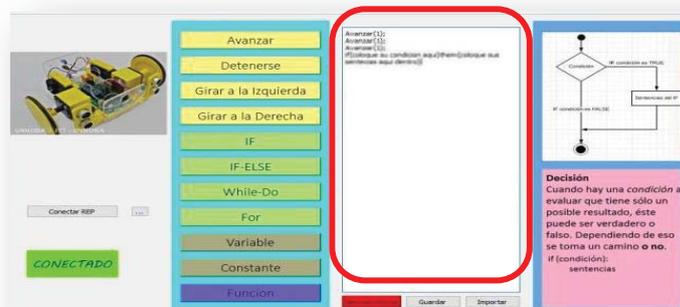


Figura 2: captura de pantalla de la interfaz señalando el panel del armado del programa

Los resultados de la ejecución del programa, se ven en los movimientos y comportamientos que el REP realice en el piso del aula.

La experiencia directa de visualizar cómo el REP responde a las órdenes que se le indican, transforma lo abstracto de las interfaces tradicionales de programación que simulan la ejecución de un algoritmo en pantalla, en un caso concreto.

En la *Figura 3* se señalan los paneles en los que se visualizan imágenes de ayuda en el lenguaje Python.

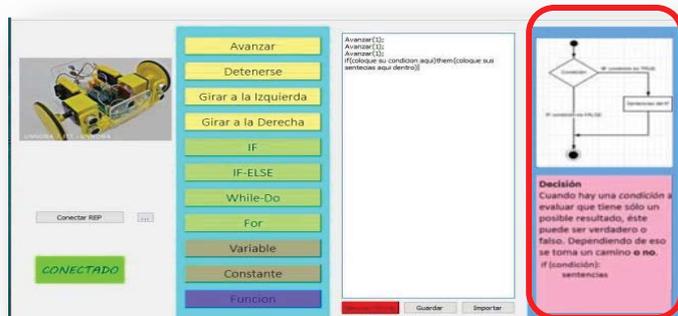


Figura 3: captura de pantalla de la interfaz señalando los paneles de ayuda

Los textos de ayuda se presentan en el lenguaje Python. Sin embargo, estas “ayudas” son sólo imágenes, pudiéndose incorporar otros lenguajes en función de las necesidades de los usuarios.

A través del uso de un pseudocódigo se pretende facilitar el aprendizaje de la programación de computadoras y solución de algoritmos, despojándose de las dificultades propias que implica introducir a un sujeto en cualquier lenguaje de programación.

Es importante destacar que el REP trabaja en una plataforma simulada de madera de 1,20 por 1,20 metros, sin embargo, podrá trabajar en cualquier otra plataforma que el docente que la utilice considere necesaria para su propuesta pedagógica.

Otro aspecto a destacar es que en principio se pensó en un panel en el que visualizará el enunciado de un ejercicio, pero, esto fue desestimado dado que limitaba el uso de la herramienta a un número preestablecido de ejercicios.

4.2 Detrás de escena de la interfaz

El REP está diseñado con Arduino [2], la programación de este micro-controlador está basada en C++, este es un punto a tener en cuenta para elegir el lenguaje de desarrollo de la interfaz, para poder hacer el traspaso de información entre la misma y el micro-controlador del REP de manera más fluida.

Otro ítem importante es el lenguaje que se utilizará en la interfaz, el cual en este caso es Python, ya que hay que tener en cuenta que esto también afecta a la complejidad del desarrollo a la hora de convertir las órdenes recibidas en Python en información para el REP.

5 El uso de tecnologías en propuestas educativas

En el presente trabajo no se concibe a la tecnología o mejor dicho, a su uso, como la panacea del éxito en una propuesta educativa. Por el contrario se asume que tal éxito está ligado a un conjunto de elementos que, combinados estratégicamente, devienen en una conquista educativa.

En ese sentido este trabajo se enfoca en la robótica en los procesos educativos, en el rol del docente, así como en el enfoque pedagógico que se aborda.

5.1 Robótica pedagógica

El interés por los aportes de la robótica en los procesos educativos, se evidencia desde los años setenta [3], abriendo paso a una nueva disciplina denominada “Robótica Pedagógica”. Los robots son sistemas compuestos por mecanismos que le permiten hacer movimientos y realizar tareas específicas, programables y eventualmente inteligentes, valiéndose de conceptos de distintas áreas del conocimiento como la electrónica, la mecánica, la física, la matemática, la electricidad y la informática, entre otras [4].

La Robótica Pedagógica utiliza los elementos multidisciplinares de la robótica, con fines didácticos, permitiendo la aplicación de ciertas herramientas tecnológicas, como apoyo en las diferentes metodologías de enseñanza y de aprendizaje, llevando la acción del lugar monopolizado del docente, al universo personal del estudiante [4]. Asimismo implica la creación de un contexto (robótica – Informática) que dinamiza y da significado al aprendizaje colaborativo en las diferentes áreas del conocimiento [5].

Se plantea entonces a la robótica educativa como una alternativa didáctica, que a la par de los métodos ya establecidos, propende por nuevos enfoques que promuevan en los estudiantes intereses que coadyuven en la creación de ambientes para el aprendizaje en el que se encuentren circunstancias favorables para la construcción de conceptos y de su interpretación personal de la realidad.

5.2 Rol docente

Es indudable que en todo proceso renovación en la enseñanza, los docentes son un componente decisivo, pues son ellos los que deben estar convencidos que se necesita de su innovación, de su creación y de su actitud hacia el cambio, para responder no sólo a los planteamientos y propósitos que se fijan en las propuestas didácticas, sino también para satisfacer a las exigencias de los contextos que envuelven a los estudiantes como sujetos sociales, históricos y culturales. [6]

Algunos paradigmas de modelos didácticos, valoran más los saberes previos de los estudiantes como aspecto fundamental para lograr mejores aprendizajes. Sin embargo, la enseñanza mediante el “conflicto cognitivo”, introduce un nuevo proceso con el objetivo de lograr un cambio conceptual. De acuerdo a este modelo el conocimiento científico es incompatible con el conocimiento cotidiano que tiene el educando, lo que plantea como meta, un cambio de sus saberes previos, al mostrarle los alcances y

limitaciones de estos, es decir, que se sienta insatisfecho con ellos y que infiera la necesidad de cambiarlos por otros más convincentes. [6]

5.3 Enfoque pedagógico

En este sentido se reconoce a un estudiante no sólo con una estructura cognitiva, sino también con ciertos saberes previos que hace del aprendizaje un proceso de confrontación constante, de inconformidad conceptual entre lo que se sabe y la nueva información. Es entonces, el educando, sujeto activo de su propio proceso de aprehensión y cambio conceptual, objeto y propósito de este modelo.

El docente planea situaciones o conflictos cognitivos, en donde se dé lugar a eventos como la insatisfacción por parte del educando con sus saberes, con la presentación de una concepción que reúna tres características para el educando: inteligible, creíble y mucho más potente que sus saberes previos.

Es entonces ese el enfoque pedagógico que se busca con este trabajo de investigación.

Para ilustrar lo anterior se cita a continuación una actividad que suele presentarse al comienzo de una clase, con el objetivo fomentar el trabajo en equipo dado que los estudiantes deben indicar en forma oral y conjunta qué sentencias debe seguir la docente para “salir del aula”; por otra parte se pretende trabajar la empatía dado que “se rompe el hielo” con una actividad algo informal y descontracturada. La comunicación también se trabaja ya que los estudiantes deben tener la capacidad de anunciar qué hacer a la docente. Y por último la confianza ya que se presenta como una actividad que los estudiantes saben resolver.

La actividad se divide en dos partes. En una primera etapa la docente se coloca estratégicamente en un lugar del aula y les pide a sus estudiantes que le indiquen en forma oral y conjunta qué pasos debe seguir para salir del aula. La docente exagerará algunas cuestiones para dejar notar que las órdenes deben ser concretas y no presentar ambigüedad. Por ejemplo si se le indica “caminar” la docente puede caminar sin parar chocándose un obstáculo, otro claro ejemplo sería que le indiquen a la docente girar y que lo haga en un sentido que no sea el indicado o bien que gire 360 °. Luego, la docente se coloca estratégicamente en un lugar diferente del aula quizá en otra posición o con diferentes obstáculos. Nuevamente se solicitará que le indiquen en forma oral y conjunta qué pasos debe seguir para salir del aula. Luego de esa segunda instancia la docente podrá abordar los conceptos que desea con ejemplos concretos de que lo le han indicado los estudiantes.

6 Pruebas piloto

Hasta el momento se han realizado dos pruebas en las que se expuso a estudiantes de primer año a trabajar con la interfaz y el REP.

Estas experiencias consistían en la resolución de una serie de ejercicios diseñados con la utilización del REP y su interfaz en la asignatura IPI. La primera de ellas tuvo lugar durante el primer mes de cursada del año 2018, y la segunda en el primer cuatrimestre del corriente año.

Los objetivos de IPI son: comprender y descomponer problemas; conocer y manejar, a través del desarrollo de algoritmos, soluciones a problemas; modularizar y parametrizar; comprender y aplicar el concepto de Programación Estructurada; manejar lenguaje Python -sintaxis y semántica-; manipular Estructuras de Control y maniobrar tipos predefinidos.

Para comprender el contexto se mencionarán brevemente los temas de la unidad temática en la que se realizó la experiencia: secuencia de instrucciones, procedimientos y repetición simple, alternativa condicional y repetición condicional.

La experiencia involucró a la totalidad de inscriptos en cada año, divididos en tres comisiones. Durante ese mes se abordaron cuatro (4) clases prácticas en las cuales las docentes plantearon diferentes ejercicios con un grado de dificultad creciente. Al finalizar estos encuentros se diseñó una evaluación en la que se pudieron observar algunos resultados que, luego de su análisis, permitieron la planificación de ciertos cambios en la interface y en las estrategias.

Luego de la última experiencia, y considerando que se pueden mejorar los resultados, se pensó en una tercera experiencia con estudiantes que ya han cursado IPI pero que aún no logran regularizar la asignatura. La misma se llevará a cabo durante el mes de agosto y parte de septiembre.

7 Conclusiones

Las experiencias en sí mismas no dan cuenta de un trabajo definitivo, sino que, por el contrario, propician el inicio de un camino en busca de estrategias para mejorar las propuestas educativas en cuanto a los inicios en la programación imperativa.

En ese sentido, se analizan los resultados finales de las cursadas de la asignatura para compararlos con los resultados que brindaron las pruebas piloto antes mencionadas.

Año	Cantidad de inscriptos	Asistentes		Regularizados sobre inscriptos		Regularizados sobre asistentes	
		Cantidad	Porcentaje	Cantidad	Porcentaje	Cantidad	Porcentaje
2019	216	122	56,48 %	48	22,22 %	48	39,34 %
2018	216	102	47,22 %	33	15,28 %	33	32,35 %
2017	169	93	55,03 %	34	20,12 %	34	36,56 %

Tabla 1: Resultados de las cursadas de los años: 2017, 2018 y 2019.

En la *Tabla 1* se pueden observar varias columnas. En primer lugar se encuentra la columna de los años que se analizan. Luego la cantidad de estudiantes inscriptos. La tercera cuarta y quinta columna muestran: los “Asistentes”, es decir los estudiantes que asistían regularmente a las clases; los “Regularizados sobre los inscriptos”, en donde se detallan los números que representan a los estudiantes que lograron aprobar la cursada de la asignatura en relación a los inscriptos; y por último los “Regularizados sobre asistentes”, citando los números de estudiantes que lograron aprobar la cursada de la asignatura en relación a los que efectivamente asistieron a clases. Las últimas tres columnas, se dividen en cantidad y porcentaje.

En función de los porcentajes que pueden apreciarse, tanto en la *Tabla 1*, se puede observar que la asistencia en el año 2019 creció, así como también la cantidad de estudiantes que lograron regularizar la asignatura. Inferimos que tiene relación directa con el uso de tecnología en el aula y que ello efectivamente los motiva a estudiar y trabajar en lo que se propone en IPI. Todo ello sin olvidar que, tal como se expresó en el apartado 5, la tecnología en el aula sin el sustento de una planificación que fundamente y acompañe su uso no asegura el éxito. Dicho de otro modo, la tecnología en el aula puede llegar a ser un gran motivador siempre y cuando esté inmersa en una propuesta educativa que sostenga su uso.

Creemos que esos ensayos son muestras de las percepciones, que permiten reflexionar acerca de las experiencias en sí mismas, el uso que se le dio a la interfaz y al REP, y a las posibilidades que pueden llegar a generarse a partir de su uso, pero principalmente al modo en que se los puede utilizar como herramientas educativas.

Es importante destacar que, dado que se trabajó durante el primer mes de cursada, la propuesta se basó en la didáctica de aprendizaje por indagación. La misma implica situar en un rol más activo al aprendiz y en uno más reactivo al docente. En ese contexto, el docente se limita a presentar diferentes situaciones a resolver por el alumno sin introducirlo en ningún concepto teórico. Luego, a medida que el docente observa necesaria su intervención, abordará conceptos o trabajará habilidades que guíen al estudiante en la construcción de su conocimiento y adquisición de habilidades.

Se observó que, si bien los estudiantes en principio esperaban en muchos casos la clase tradicional, en términos de recibir toda una explicación del docente para luego llevar a la práctica esos conocimientos para resolver, recibieron satisfactoriamente la dinámica de trabajo en la que ellos eran los protagonistas.

Se puede resumir esta forma de trabajo del siguiente modo: se planteaba un problema, se alentaba a los estudiantes a pensar una estrategia de solución a resolver con la interfaz, se reflexionaba sobre las soluciones planteadas (marcando aciertos, errores y las mejoras que pudieran existir), se brindan nuevas herramientas (de ser necesario), se probaban nuevas soluciones que pudieran llegar a surgir a partir de las intervenciones docentes, pudiendo finalmente volver a reflexionar sobre lo trabajado.

En busca de cumplir con los objetivos de la asignatura (mencionados en el punto 6) y evaluando los resultados satisfactorios y los comentarios propiciados por los alumnos, sumado a la observación de los docentes en las clases, así como en la evaluación; se decidió escribir nuevos enunciados en función de los conceptos futuros que se desean abarcar con el uso de la interfaz.

Referencias

1. Álvarez, E; Useglio, G; Osella Massa, G; Luengo, P; Russo, C; Sarobe, M; Llanos, E; Pérez, R; Serafino, S; Ramón, H.: Robótica: Aplicaciones en Educación y en Agricultura de Precisión. Evento: XVI Workshop de Investigadores en Ciencias de la Computación. 349-352 (2014). Disponible en: <http://sedici.unlp.edu.ar/handle/10915/41358>
2. Arduino: tecnología para todos. Disponible en: <http://arduinodhtics.weebly.com/iquestqueacute-es.html>
3. Ruiz, E.: La robótica pedagógica. Centro de Estudios sobre la Universidad CESU, Universidad Nacional Autónoma de México. (1987).

4. Pinto Salamanca, M.L., Barrera Lombana, N., Pérez Holguín, W.J. (2010). Uso de la robótica educativa como herramienta en los procesos de enseñanza. *Universidad Pedagógica y Tecnológica de Colombia*, 10 (1), 15 – 23
5. López Ramírez, P.A. (2013). Aprendizaje de y con robótica, algunas experiencias. *Revista Educación* 37(1), 43-63.
6. Ruiz Ortega, F. (2007). MODELOS DIDÁCTICOS PARA LA ENSEÑANZA DE LAS CIENCIAS NATURALES. *Revista Latinoamericana de Estudios Educativos (Colombia)*, 3 (2), 41-60.

Herramientas de Autor para Enriquecer Materiales de Lectura. Análisis Comparativo

Edith Lovos¹, Cecilia Sanz^{2,3},

¹ CIEDIS, UNRN – Sede Atlántica, 8500, Viedma, Argentina

² III-LIDI, Facultad de Informática -Universidad Nacional de La Plata

³ Investigador Asociado de la Comisión de Investigaciones Científicas de la Pcia. de Bs. As.

elovos@unrn.edu.ar, csanz@lidi.unlp.edu.ar

Abstract. La realidad aumentada se presenta como una tecnología disruptiva, que en el ámbito educativo posibilita entre otros aspectos, nuevas formas de acercamiento a los materiales de estudio. En el caso de los libros aumentados, no solo se enriquece el contenido del material de lectura, sino que da lugar a otras interacciones durante la lectura misma. En la práctica docente resulta interesante e innovadora la posibilidad de utilizar materiales de lectura aumentados, y más aún participar en su producción, a partir de libros ya pre-existentes o creados desde su inicio. Así en este trabajo, se presenta un análisis comparativo entre dos herramientas disponibles en la web, factibles de ser consideradas herramientas de autor que permiten la producción de materiales de lectura aumentados a partir de imágenes, y dónde la visualización e interacción se pueda realizar usando dispositivos móviles. El análisis se realiza a partir de una serie de criterios definidos por los autores y los resultados dan cuenta de la facilidad de uso de estas herramientas y el potencial para el trabajo de los docentes.

Keywords: realidad aumentada, herramientas de autor, materiales de lectura aumentados

1 Introducción

Diversos estudios e informes [1, 2, 3, 4, 5], destacan las posibilidades y beneficios que aporta la tecnología de realidad aumentada (RA) en los escenarios educativos de distintos niveles, entre ellos: la posibilidad de acercarse y comprender mejor conceptos abstractos [2, 3, 5], incrementar la motivación [1, 2], y posibilitar un aprendizaje activo [2, 5]. Sin embargo, uno de los problemas a los que se enfrentan los docentes que desean incorporar esta tecnología en el espacio de sus prácticas es encontrar materiales educativos que integren el uso de la RA y que se adapten a sus necesidades: idioma, temática, tipo de actividades educativas, tipo de material (libro,

e-book, guía, etc). Al mismo tiempo, contar con dispositivos (celular, *tablet*, etc) para el acceso a la RA en espacios áulicos, también puede ser un desafío. Ante este reto, y cuando los materiales disponibles en sitios especializados no se adaptan a las necesidades del contexto de aplicación, surge la posibilidad de utilizar herramientas informáticas, algunas de las cuales pueden ser consideradas herramientas de autor (HA), que posibilitan a través de interfaces de usuario amigables, la disponibilidad de plantillas predeterminadas y sistemas de guías o tutorías, el diseño y construcción de recursos o materiales educativos, con realidad aumentada, sin necesidad de tener conocimientos de programación [6, 7, 8, 9].

Algunos autores consideran diferentes niveles de RA según el tipo de reconocimiento que se utilice como disparador para aumentar el contexto físico y también según la forma de integración entre el escenario real y los objetos aumentados. Según Lens-Fitzgerald (2009) citado en [5], el nivel más bajo de RA, lo constituye el uso de códigos QR (considerados hiperenlaces entre el mundo físico y el digital), luego le siguen los marcadores de posición, el reconocimiento de objetos e imágenes, y por último, en el nivel más alto, en el que se considera la RA a través de dispositivos electrónicos especiales como lentes, auriculares, etc, dónde el escenario real y los objetos aumentados se encuentran plenamente integrados [4].

En este trabajo se presentan y discuten los resultados de un análisis comparativo entre dos herramientas disponibles en la web, factibles de ser consideradas HA (de acuerdo a los criterios establecidos en la literatura sobre el tema), que permiten la producción de materiales de lectura aumentados (o enriquecidos) a partir de imágenes (nivel 3 de RA, según lo antes descrito), y dónde la visualización e interacción con la escena aumentada, se pueda realizar usando dispositivos móviles (celulares y/o *tablets*). Esto último posibilitaría su inclusión en propuestas de enseñanza y aprendizaje tipo *e-learning* o *m-learning*.

A partir de aquí, este artículo se organiza de la siguiente manera: en la sección 1 se presenta la fundamentación teórica que sustenta el estudio, en la sección 2 se describe la metodología de trabajo. En la sección 3 se presentan y discuten los resultados del estudio y por último en la sección 4, se presentan las conclusiones.

Es importante resaltar que este trabajo ha sido parcialmente financiado a través de un proyecto de investigación acreditado por la Universidad Nacional de Río Negro (PI-40C-750), y el proyecto REFORTICCA - Recursos para el Empoderamiento de Formadores en TIC, Ciencias y Ambiente de la CICPBA (Ministerio de Ciencia, Tecnología e Innovación en Gobierno de la Provincia de Buenos Aires).

1.1 Realidad Aumentada

Azuma [10] definió la RA como una tecnología que posibilita agregar información sintética (modelos 3D, vídeos y gráficos en tiempo real) a un contexto o escenario real. En su definición, se destaca que la información virtual queda co-registrada en el espacio físico y esto se realiza en tiempo real. En este sentido, Cabero y Barroso [11], consideran que la RA no se limita a la combinación de elementos reales con agregados virtuales, sino que da lugar a una nueva escenografía comunicativa. Sin embargo, los beneficios de la RA no se limitan a la posibilidad de incrementar la información disponible en el contexto físico, sino también a la oportunidad de crear ciertas representaciones del mundo más afines al usuario, de allí que pueda ser utilizada en diferentes áreas como comercio, educación, entretenimiento, entre otras [12]. Estudios e informes de los últimos años sobre RA en educación, destacan que esta tecnología emergente es [1, 2, 3, 4, 5]: un aporte a la motivación de los estudiantes, posibilita presentar contenidos a través de dispositivos como los teléfonos celulares; aumenta la comprensión de conceptos abstractos y/o de difícil visualización y mejora en los resultados de aprendizaje. En el caso específico de libros aumentados, posibilitan enriquecer la lectura tradicional (sea ésta en medio impreso o digital) con la visualización de imágenes, vídeos, audios y/u objetos en 3D, entre otros elementos digitales. Estos se superponen al contenido del libro, y ofrecen nuevas experiencias en la que se fusionan la realidad con la virtualidad [13].

1.2 Herramientas de Autor

Las herramientas de autor (HA), se definen como aquellos aplicativos de software que posibilitan a través de interfaces de usuario amigables, y con el uso de plantillas predeterminadas y de sistemas de tutorías o guías, el diseño y construcción de software, como por ejemplo materiales educativos, sin necesidad de contar con conocimientos de programación para hacer uso de ellas [6, 7, 8, 9]. Los materiales construidos con estas herramientas pueden ir desde contenidos simples y sencillos hasta experiencias de aprendizajes más complejas [14]. En este sentido, las HA no solo pueden ser de utilidad al docente, para la producción de materiales educativos que posibiliten nuevas formas de acercamiento a un objeto de estudio, sino que se convierten en un recurso que posibilita la producción de contenido por parte de los estudiantes de una forma sencilla.

Moralejo [8] a partir de una revisión bibliográfica sobre el tema, propone un conjunto de características que permiten comparar HA, a saber: Licencia, Personalización, Plataforma sobre la que se ejecutan, Plantillas para la generación de distintos tipos de actividades educativas, Creación de paquetes de actividades, Posibilidades de navegación, Formatos de salida, Compatibilidad con estándares, y en

el caso específico de HA para la creación de contenidos que incluyan RA, la misma autora sostiene que resulta fundamental considerar características tales como: Funcionalidad de la herramienta y la Forma de interacción con el contenido aumentado, Mecanismos disponibles para que se produzca el aumento de la realidad, e Información aumentada soportada.

Por otra parte, Alonso Bartolomé [15], agrega como características deseables para una HA: Accesibilidad, en particular para usuarios con dificultades visuales o auditivas, Colaboración, de modo que se promueva el trabajo en equipo y la comunicación en sus diferentes formas. Cubillos [9] agrega como indispensables las características de reutilización del contenido producido y la de privacidad y control sobre éste, en el sentido de dotar al productor de facilidades que le permitan modificarlo o eliminarlo.

A partir de estos antecedentes, se presentan y describen en la Tabla 1, los criterios de análisis (CA) que serán utilizados en este trabajo. Se ha incluido también el criterio: funcionalidad de la aplicación, que permitirá luego de crear el contenido con la HA, la visualización e interacción con la escena aumentada. En varios casos, existe una aplicación para móviles que es la que posibilita la visualización del contenido creado con la HA.

Tabla 1 - Criterios de análisis propuestos para el análisis de HA. Basado en [8, 9, 15]

Criterios de Análisis (CA)		Detalle
CAx1	Licencia	Indica las condiciones bajo las cuales puede hacerse uso de la herramienta
CAx2	Funcionalidad de edición	Se refiere a la funcionalidad provista por la herramienta para la edición de contenido aumentado
CAx3	Funcionalidad de la aplicación de visualización e interacción	Hace referencia a la funcionalidad que permite visualizar e interactuar con el contenido aumentado.
CAx4	Información aumentada soportada	Hace referencia a los tipos de información aumentada que soporta la herramienta (imágenes, videos, objetos 3D, enlaces, etc.)
CAx5	Plantillas	Hace referencia a la disponibilidad o no de plantillas para la creación de contenido, provistas por la herramienta de edición.

CAX6	Distribución del contenido	Se refiere a la forma en la que el contenido producido puede distribuirse o publicarse
CAX7	Colaboración	Posibilidades para el trabajo en equipo y la comunicación
CAX8	Documentación	Se hace referencia a la documentación disponible que permite apoyar el uso de la herramienta

2 Metodología

En primera instancia se han establecido los CA para la evaluación de las herramientas de autor que posibilitan la producción de contenidos aumentados a partir de imágenes. Luego, se procedió a la selección de las herramientas a comparar, teniendo en cuenta: la disponibilidad y posibilidad de uso de éstas en la web (para su acceso desde cualquier lugar), la posibilidad de producción de libros aumentados con reutilización de materiales pre-existentes sean estos impresos o digitales, y el tipo de licencia (para fines educativos o gratuita sin restricciones). La muestra se compuso por las 4 herramientas descritas en la Tabla 2, teniendo en cuenta que han sido utilizadas en diferentes experiencias educativas encontradas en la literatura de referencia [16, 17, 18, 19].

Del total de herramientas presentadas, se han descartado ROAR y HP-Reveal. En el primer caso, porque no dispone de una licencia con fines educativos y la licencia de uso gratuito presenta una limitación en relación a la cantidad de veces que puede ser escaneada la imagen que activa la RA. Y en el caso de HP-Reveal se ha descartado ya que desde abril de 2019, la herramienta se encuentra en un proceso de transición y no es posible utilizarla para la creación de contenidos desde la web.

Tabla 2. Herramientas seleccionadas a partir de [16, 17, 18, 19]

Herramientas	Descripción
<i>HPReveal</i> (<i>ex-Aurasma</i>)	Ofrece una única licencia que permite usarla en forma gratuita, sin embargo para visualizar un contenido (aura) es necesario que el mismo haya sido compartido. (http://oa.upm.es/50686/1/Manual_HP_Reveal.pdf). En proceso de transición desde 2019
<i>Blippar</i>	Provee la plataforma <i>BlippBuilder</i> para la creación de escenarios aumentados e interactivos. Para su uso educativo, los contenidos aumentados deben compartirse a través de un código de acceso (configurable por el productor)

<i>Artuthor</i>	HA web gratuita que permite generar contenido para libros aumentados.
<i>ROAR</i>	Posibilita la creación de experiencias aumentadas, principalmente para la publicidad y marketing. No dispone de licencia para educación y la licencia gratuita solo permite que la imagen de activación de la RA se escanee una cantidad limitada de veces

A continuación se presentan y describen las dos HA seleccionadas y el análisis de cada una de ellas.

2.1 Blippbuilder

Blippbuilder y *Blippar* forman parte de una suite de herramientas creadas por la empresa *Blippar*, especializada en realidad aumentada, inteligencia artificial y visión por computadora, cuyos productos y servicios son factibles de ser utilizados en áreas como el marketing, los negocios y la educación, entre otros. En el área de RA, ofrece: a) una aplicación para la visualización de recursos aumentados como es el caso de la app para dispositivos móviles llamada *Blippar*, y b) la HA *Blippbuilder* que permite la creación de escenarios aumentados e interactivos -denominados *Blipps*- de una manera fácil e intuitiva usando una interface *drag-and-drop*. A continuación se describe *Blippbuilder*, bajo la consideración de los CA establecidos en el apartado anterior.

2.1.1 CAx1 - Licencia de distribución

Blippbuilder se ofrece a través de licencias gratuitas e ilimitadas en el tiempo y en cantidad de *blipps* construidos para el sector educativo, sin embargo el contenido sólo es accesible a través de códigos de acceso. De otro modo, es necesario contratar una licencia. No obstante, la aplicación móvil que permite visualizar e interactuar con el contenido producido- *Blippar*- es totalmente gratuita.

2.1.2 CAx2 - Funcionalidad de edición

Para utilizar *Blippbuilder*, es necesario registrarse con un usuario y contraseña, a través del sitio www.blippar.com. Los contenidos aumentados construidos con la herramienta se denominan *blipps*. La herramienta posibilita la construcción individual y/o armar un proyecto y dentro de éste integrar un conjunto de *blipps* bajo un mismo nombre y código de acceso, en el caso de la licencia educativa. De esta forma, el

usuario solo debe introducir el código una única vez, a través de la app *Blippar* en su dispositivo móvil y no por cada *blipp* escaneado.

El proceso de construcción comienza con la selección del marcador que se utilizará para aumentar la escena. Como marcador se puede utilizar una imagen del mundo real o cualquier objeto como ejemplo: envases, posters, remeras, etc. Por cada *blipp* es posible subir hasta 20 marcadores diferentes, y se recomienda para mejores resultados que estos cumplan con el formato JPEG y tamaño 300 x 800 píxeles, en ancho y altura. Cuenta además con la opción para autogenerar los marcadores. Un punto interesante, es que una misma imagen puede configurarse para activar diferentes experiencias de RA y se pueden reutilizar en con distintos objetos (ejemplo: una misma imagen se puede reusar en un poster, en un envase o en una camiseta); o también es posible configurar muchas imágenes diferentes para activar la misma experiencia de RA (por ejemplo, varios anuncios de prensa o variantes de empaquetado de productos). Por otra parte, *Blippbuilder* cuenta con una funcionalidad que permite la construcción de un *blipp* a partir de otro existente en la cuenta del usuario, así como también restaurar versiones anteriores de un *blipp*. Para ello el usuario dispone de un historial de versiones de cada *blipp*.

2.1.3 CAx3 - Funcionalidad de la aplicación para visualización e interacción

La aplicación móvil que permite visualizar e interactuar con un escenario aumentado producido con *Blippbuilder*, se denomina *Blippar*. Según sus desarrolladores, es el primer navegador a nivel mundial que permite la navegación visual, usando tecnología como RA, inteligencia artificial y visión por computadora.

Para escanear un *blipp* desde un dispositivo móvil, es necesario descargar la aplicación *Blippar*, desde la tienda de aplicaciones de Android o IOS. *Blippar* puede usarse en modo Explorar activado o desactivado. En el primer caso, intentará reconocer cualquier objeto al que se apunte con la cámara del dispositivo móvil.

Para acceder a un *blipp* educativo, es necesario contar con su código de acceso, que sólo puede ser compartido por su creador. Desde la aplicación móvil, y mientras se está ejecutando un *blipp*, es posible agregarlo a favoritos, compartirlo por diferentes medios (correo, red social, etc.) y/o tomar una instantánea de éste. En la Fig. 1, se puede visualizar el contenido aumentado de un material de lectura en formato papel.



Fig. 1 – Izquierda: material de lectura en formato impreso. Derecha: visualización del contenido aumentado usando *Blippar*

2.1.4 CAx4 - Información Aumentada Soportada

Blippbuilder denomina *assets* a los contenidos de un *blipp*, a saber: texto, imágenes (.jpg, .png opacas y transparentes, .gif), audio (.mp3), videos (.mp4) e iconos (*widjets*) y es a través de éstos, que permite agregar interacción. Para ello, dispone de un banco de íconos precargados (*widjets*) que permiten agregar interacciones sencillas como: acceso a videos (a través de la *url* de canales como *youtube* o subidos desde la computadora del usuario), audios que pueden ser reproducidos como sonido de fondo o a partir de cierta interacción. También es posible crear otras interacciones más complejas, como son la posibilidad de interactuar con funcionalidades propias del móvil (mensajería, llamadas telefónicas) y navegar entre diferentes escenas. Esto último, posibilitaría el diseño y construcción de actividades educativas en las que se plantea un recorrido y/o actividades tipo cuestionarios interactivos. Asimismo, es posible personalizar cada uno de los *assets* y agregar efectos de movimiento como: mover, rotar, escalar, desaparecer y rebotar. También se puede trabajar con modelos 3D en formato propio (BB3).

2.1.5 CAx5 - Disponibilidad de Plantillas

No cuenta con plantillas predefinidas para la generación de contenidos. Sin embargo, los *widjets* permiten incluir y configurar en el material editado funciones usadas frecuentemente (enlaces, redes sociales, *selfies*, etc).

2.1.6 CAx6 - Distribución del contenido

Blippbuilder permite compartir los *blipps* construidos, en las aplicaciones de redes sociales de cualquier usuario, así como también que los usuarios lo etiqueten como favorito dentro de la aplicación. Por defecto en el caso de los materiales producidos con la licencia educativa, éstos sólo se pueden publicar a través de un código. Al mismo tiempo, *Blippbuilder* permite que los contenidos puedan ser configurados para ser visualizados a nivel global o en determinadas regiones.

2.1.7 CAx7 - Documentación

Es posible acceder a un material de ayuda para el uso de *Blippbuilder*, tanto formato texto como videos, a través del siguiente enlace <https://support.blippar.com/hc/en-us/categories/200361418-BlippBuilder-Documentation>. El mismo se presenta solo en idioma inglés.

2.1.8 CAx8 - Colaboración

Blippbuilder no permite la edición colaborativa al menos en la versión educativa, que es la que se ha puesto a prueba para este trabajo.

2.2 ARTutor

ARTutor es una HA, desarrollada en el laboratorio AETMA del Departamento de Computación e Ingeniería Informática del Instituto de Tecnología de Macedonia del Este [17]. La plataforma destinada exclusivamente al ámbito educativo, dispone de una herramienta a través de la cual los usuarios pueden aumentar libros existentes, sean estos digitales o impresos, de forma fácil y sin necesidad de contar con conocimientos de programación. Ofrece además, una aplicación para dispositivos móviles que permite interactuar con el contenido aumentado que se superpone en los libros. Una de las funcionalidades que destaca a *ARTutor*, es la posibilidad de interactuar con los contenidos, a través de comandos de voz, de este modo, es posible realizar preguntas en forma verbal y recibir respuestas basadas en el contenido del libro aumentado que se ha creado. Por otra parte los recursos que se crean con *ARTutor* quedan disponibles y se comparten desde el sitio: <http://artutor.teiemt.gr/books/>.

2.2.1 CAx1 - Licencia

El uso de la plataforma es gratuito. Solo requiere el registro del usuario en el sitio <http://artutor.teiemt.gr/>. Una vez registrado, se accede a la herramienta a través de la siguiente url: <http://artutor.teiemt.gr/atoool/>. Desde aquí es posible editar y/o eliminar los libros creados por el usuario.

2.2.2 CAx2 - Funcionalidad de edición

La descripción que aquí se realiza, corresponde a la versión de *ARTutor* 2019. El proceso de edición comienza registrando información que permitirá catalogar el material, entre otras: título, autor, categoría del libro (manual, presentación, material de cátedra, etc.), idioma (por ahora solo inglés y griego), y luego se debe seleccionar el archivo en formato .pdf (menos de 20 MB) que será la base del libro aumentado. Para aumentar el documento base, se debe incluir un contenido aumentado (*asset*). Con este fin, *ARTutor* permite navegar el documento y seleccionar usando el *mouse*, el área (imagen o figura) que servirá de marcador para aumentar el libro. Luego es necesario asociarle *assets*, que se mostrarán cuando, con la aplicación móvil, se reconozca el área en el documento impreso o digital. Una de las características de *ARTutor*, es que permite indicar si el contenido aumentado es gratuito o de pago. Según sus autores, a diferencia de otras aplicaciones de realidad aumentada, el contenido accedido es dinámico y no existen limitaciones sobre visibilidad y acceso entre los usuarios. Por otra parte, la interacción con el contenido aumentado se puede realizar en forma háptica y verbal. Así cada contenido aumentado, se puede acompañar con comandos de voz que luego serán usados por la aplicación móvil para interactuar con estos. Los comandos posibles de incluir son: mover, parar (aplicable al formato video), estirar y achicar (aplicables a imágenes y modelos aumentados en 3D) y rotar (aplicable a modelos aumentados 3D), es importante resaltar que hasta aquí los comandos de voz que funcionan, solo reconocen los idiomas griego e inglés, que son los de la versión actual de *ARTutor*.

2.2.3 CAx3 - Funcionalidad de la aplicación para visualización e interacción

Para visualizar el contenido aumentado de un libro, es necesario descargar la aplicación para dispositivos móviles - *ARTutor2*- desde el almacén de IOS o Android. Cuando se inicia la aplicación, se despliega la lista de libros disponibles por categoría. Así es posible realizar una selección de libros por categoría. Por cada libro, se puede conocer su título, descripción, autor y el lenguaje del libro.

Una vez seleccionado el libro a visualizar, la *app* se conecta con el servidor y recupera y descarga al móvil todo el contenido aumentado que se ha incluido en las

páginas del libro. Una vez finalizada esta acción, se abre una nueva pantalla donde se presenta la vista previa de la cámara, y a partir de ese momento se pueden escanear todas las páginas del libro que estén aumentadas. En la esquina superior derecha de la pantalla, hay un ícono (libro) que al hacer clic sobre él, presenta en modo texto cuáles son las páginas que presentan contenido aumentado. A partir de que se reconoce con la cámara la imagen a aumentar dentro del libro, se pueden visualizar dos íconos en la parte inferior de la pantalla, uno corresponde a un botón de ayuda y el de la derecha es el ícono de *ARTutor*. En la Fig. 2, presenta una captura de pantalla del móvil, en el momento en la que se ha reconocido el área aumentada en una de las páginas del libro.



Fig. 2 - Visualización de contenido aumentado desde la app *ARTutor2*

Si el contenido aumentado es sonido, éste se reproduce en forma automática al detectar la imagen de activación, y en el borde superior izquierdo se podrá visualizar información del sonido activado. Cuando se hace ‘tap’ sobre el ícono *ARTutor* es posible interactuar con el sonido. En caso que la/s palabra/s o comando/s reconocido/s coincida/n con alguno de los *assets* que está activo, se producirá el aumento asociado. En cualquier otro caso, aparecerá un mensaje visual y en forma de audio que indica que el comando resulta desconocido. Esta funcionalidad permite interactuar con el contenido aumentado a través de comandos de voz o por gestos táctiles sobre la pantalla del dispositivo móvil.

ARTutor permite realizar consultas en forma de audio. Para diferenciar entre un comando y una consulta, si lo que se desea es esto último, entonces la grabación debe comenzar con la palabra “*Question*”. Al finalizar la grabación, se genera un patrón de

búsqueda quitando la palabra “*Question*” y se lo envía al servidor a través de un *web service*. Del lado del servidor, hay un *script* que se encarga de buscar ese patrón en el texto plano que se tomó como base para la creación del libro aumentado. La respuesta generada del lado del servidor, se devuelve al dispositivo y se muestra en la pantalla con una ventana emergente, que se puede escuchar usando la función de texto a voz. Un aspecto importante a resaltar, es que a diferencia de los comandos de voz, las preguntas pueden ser formuladas en el lenguaje del libro, que puede ser diferente del asociado a la configuración del dispositivo y/o de la aplicación. Esto posibilitaría aumentar libros en diferentes idiomas, sin necesidad de cambiar la configuración de idioma del dispositivo.

La funcionalidad de consultas a través del audio, promueve la creación de un escenario donde los estudiantes pueden formular consultas al tutor mientras exploran el libro, reforzando la idea de autoaprendizaje y brinda así posibilidades en el ámbito de la educación a distancia [17].

2.2.4 CAx4 - Información Aumentada Soportada

Hasta el momento, en la versión actual es posible agregar información aumentada de tipo imagen solo en formatos .png y .jpg; archivos de audio (.mp3) y archivos de video (.mp4) y objetos 3D en formato empaquetado en archivo.zip.

2.2.5 CAx5 Disponibilidad de Plantillas

La herramienta no dispone en la versión actual de plantillas. La creación se realiza a partir de la selección de partes del libro.

2.2.6 CAx6 - Distribución del contenido

Los contenidos producidos con *ARTutor* quedan disponibles a través del repositorio de esta herramienta, desde el momento de su creación. No dispone de un mecanismo que permita mantenerlo en forma privada.

2.2.7 CAx7 - Documentación

Provee un material de ayuda aumentado (http://artutor.teiemt.gr/wp-content/uploads/2018/09/ARTutor2_Manual_EN.pdf) para el uso de la herramienta de creación y de la app de visualización e interacción. Hasta el momento solo disponible en idioma inglés.

2.2.8 CAx8 - Colaboración

ARTutor no presenta características que permitan la edición colaborativa.

3. Resultados

Hasta aquí se han expuesto las características de las HA evaluadas en base a los criterios establecidos en el apartado 1.2. Aunque cada una, denomina a los materiales producidos de diferente forma, ambas permiten enriquecer materiales de lectura, usando RA, a partir de imágenes contenidas en estos.

En el caso de *ARTutor*, agrega la interacción con el contenido aumentado a través de comandos y preguntas. Aunque hasta aquí, esta funcionalidad esté limitada por el idioma de la herramienta, la misma resulta innovadora, dando lugar a la generación de un escenario donde los usuarios (estudiantes) pueden formular consultas al tutor virtual mientras exploran un material de lectura, reforzando como sostienen sus creadores, la idea de autoaprendizaje y/o el uso de la app en situaciones de educación a distancia. [17]

Aunque ambas herramientas incluyen como información aumentada: imágenes, videos, audios, objetos 3D. *Blippar* permite diferentes niveles de interacción con el contenido a través de los *widgets*, que abarcan desde el uso de funcionalidades propias del dispositivo móvil (llamadas, mensajerías, etc) hasta la posibilidad de trabajar con escenas. Esto último, abre posibilidades para el diseño y construcción de actividades educativas tipo cuestionarios interactivos y/o recorridos. De esta forma puede ser una alternativa válida para la generación de experiencias que incluyan aspectos lúdicos poniendo en juego metáforas como la búsqueda del tesoro.

Respecto a la interface de las HA, aunque ambas se presentan en inglés, *ARTutor* resulta visualmente menos sobrecargada de información, lo cual puede resultar ventajoso para un usuario novato y una opción viable para incluir a los estudiantes en la producción de contenidos educativos aumentados.

En relación al uso de la aplicación móvil que permite visualizar e interactuar con material de lectura enriquecido, *ARTutor* no permite el uso de enlaces a los contenidos. Así, si el material contiene videos y/o audios, el peso de los mismos afectará la velocidad del proceso de descarga.

4. Conclusiones

En este trabajo, se puso el foco en la posibilidad de enriquecer materiales de lectura a partir de la RA de nivel 3. Se analizaron HA disponibles en la web, y dónde el

contenido aumentado se pueda visualizar e interactuar a través de un dispositivo móvil, principalmente *tablets* y celulares. De esta forma, el tiempo de creación se destina fundamentalmente a la recolección del material digital que permitirá aumentar el contenido del material de lectura, y a la configuración de los marcadores (o imágenes en este caso) que activarán la RA en el libro.

Por lo expuesto, es posible concluir que las herramientas analizadas permiten avanzar con los objetivos establecidos, promoviendo la reutilización de materiales de lectura pre-existentes. Asimismo, abren posibilidades a diferentes formas de interacción con los contenidos de un material de lectura enriquecido a través de RA, y se constituyen herramientas de fácil uso y con amplias posibilidades para los docentes de cualquier área.

Como aspectos que aún no se han encontrado en estas herramientas y que serían deseables [15], se incluyen, la posibilidad de edición y construcción colaborativa del material de lectura aumentado, la posibilidad de personalizar el idioma (por ejemplo, español no es parte de las dos HA analizadas), así como también de disponer de más posibilidades para compartir las producciones finales (en forma privada, pública, para determinados grupos, etc.). A partir de este análisis se deja como línea futura de trabajo, la posibilidad de crear HA que contemplen estas vacancias.

Referencias

1. Johnson L, Levine A, Smith R, Stone S. (2010). The 2010 Horizon Report. Austin, Texas: The New Media Consortium.
2. Bacca, J., Baldiris, S., Fabregat, R., Graf, S., y Kinshuk, G. (2014). Augmented Reality Trends in Education: A Systematic Review of Research and Applications. *Educational Technology y Society*, 17(4), 133-149.
3. Ibáñez, M. B., Di Serio, Á., Villarán, D., & Kloos, C. D. (2014). Experimenting with electromagnetism using augmented reality: Impact on flow student experience and educational effectiveness. *Computers & Education*, 71, 1-13.
4. Prendes, C. (2015). Realidad aumentada y educación: análisis de experiencias prácticas. *Pixel-Bit. Revista de Medios y Educación*, 46, 187-203.
5. Huang, K. T., Ball, C., Francis, J., Ratan, R., Boumis, J., & Fordham, J. (2019). Augmented Versus Virtual Reality in Education: An Exploratory Study Examining Science Knowledge Retention When Using Augmented Reality/Virtual Reality Mobile Applications. *Cyberpsychology, Behavior, and Social Networking*, 22(2), 105-110.
6. Montero O'Farrill, J. L. y Herrero Tunis, E. (2008). *Las Herramientas de Autor en el proceso de producción de materiales educativos en formato digital*.
7. Camarda P., Minzi V. (2012). "Primaria Digital, Aulas digitales móviles, Manual general introductorio". Primera edición. Buenos Aires. Ministerio de Educación de la Nación, 2012. ISBN 978-950-00-0949-2.

8. Moralejo, L. (2014). Análisis comparativo de herramientas de autor para la creación de actividades de realidad aumentada (Doctoral dissertation, Facultad de Informática).
9. Cubillo, J. (2014). ARLE: una herramienta de autor para entornos de aprendizaje de realidad aumentada (Doctoral dissertation, UNED. Universidad Nacional de Educación a Distancia (España)).
10. Azuma, R., Baillot, Y., Behringer, S., Feiner, S., Julier, S y MacIntyre, B. (2001). Recent Advances in Augmented Reality. *IEEE Computer Graphics and Applications*. 3, 34-47.
11. Cabero Almenara, J., y Barroso Osuna, J. M. (2016). Posibilidades educativas de la Realidad Aumentada. *Journal of New Approaches in Educational Research*, 5 (1), 46-52.
12. G. Bhorkar, A. (2017) Survey of Augmented Reality Navigation. 8(2), 73–272, en Lytridis, C. y Tsinakos, A. *Smart Learn. Environ.* (2018) 5: 6. <https://doi.org/10.1186/s40561-018-0058-x>
13. Gazcón, N. F., y Castro, S. M. (2014). A novel authoring tool for augmented books. In *XX Congreso Argentino de Ciencias de la Computación* (Buenos Aires, 2014).
14. Almenara, J. C., Osuna, J. B., y Pérez, Ó. G. (2018). La producción de objetos de aprendizaje en realidad aumentada por los estudiantes. Los estudiantes como prosumidores de información. *Revista Tecnología, Ciencia y Educación*, (11).
15. Alonso, P. B. (2014). Las herramientas de autor en el aula de ELE. In *La enseñanza del español como LE/L2 en el siglo XXI*(pp. 141-150). Asociación para la Enseñanza del Español como Lengua Extranjera.
16. Carbonell-Carrera, C., Saorín, J. L., Meier, C., Melián-Díaz, D., y De-la-Torre-Cantero, J. (2016). Tecnologías para la incorporación de objetos 3D en libros de papel y libros digitales. *El profesional de la información (EPI)*, 25(4), 661-670.
17. Lytridis C., Tsinakos A., y Kazanidis I. (2018). [ARTutor-An Augmented Reality Platform for Interactive Distance Learning](#). *Education Sciences.*; 8(1):6
18. Gallego Pérez, Ó. (2018). Estudio y análisis sobre las posibilidades educativas de la realidad aumentada como herramienta de producción de experiencias formativas por parte del alumnado universitario.
19. Pérez, S. M., Robles, B. F., y Cabrerizo, R. A. F. (2018). Diseño de objetos de Realidad Aumentada: Experiencia con el alumnado de Educación. In *Conference Proceedings EDUNOVATIC 2017: 2nd Virtual International Conference on Education, Innovation and ICT* (p. 182). Adaya Press.

La Realidad Virtual y su impacto en el aprendizaje

Sebastián Calderón, Juan Manuel Bournissen y Marisa Cecilia Tumino
Universidad Adventista del Plata, Libertador San Martín, Entre Ríos, Argentina
{Sebastian.calderon, juan.bournissen, marisa.tumino}@uap.edu.ar

Resumen. El objetivo de este estudio fue determinar el impacto de la utilización de la Realidad Virtual (RV), como una técnica que permita reforzar las estrategias de enseñanza y aprendizaje utilizadas en la enseñanza de nivel superior. La RV es una herramienta que posibilita estudiar, en un entorno virtual, la realidad representada dinámicamente, facilitando el manejo de los componentes que conforman esa realidad. La RV permite al estudiante explorar ambientes y situaciones complejas, manipulando los elementos que componen esa realidad. Esta experiencia desarrollada mediante el estudio, con estudiantes de ciencias de salud, ha proporcionado datos suficientes que permiten estimar que el recurso utilizado tiene un impacto positivo en el proceso de enseñanza y aprendizaje de los estudiantes, ofreciendo una alternativa innovadora a las estrategias tradicionales utilizadas por las cátedras donde se realizó el experimento.

Palabras clave: Realidad virtual, TIC, Recursos educativos, Evaluación de impacto,

1. Introducción

Botella Arbona y otros [1], conceptúan a la RV como una tecnología que crea espacios tridimensionales, mediante la tecnología, simulando la realidad y permitiendo manipular los elementos y los eventos del ambiente virtual que se consideran útiles para lograr los objetivos propuestos. Los autores asocian a la RV con un cambio cualitativo respecto a otras tecnologías, puesto que “permite una inmersión total en una simulación de la realidad donde el usuario puede interactuar con el mundo virtual, de una forma similar a como interactúa con el mundo real” (Botella Arbona y otros, 2007, p. 18).

La realidad virtual es un término popularizado por varios autores a partir de los años 80 y 90 para definir los rasgos de los entornos inmersivos generados de manera artificial por medios digitales. Desde las primeras tentativas de desarrollo de entornos sensorialmente inmersivos como Sensorama (1962), los primeros simuladores de vuelo de los 60 o el experimento Aspen Movie Map, la inmersión en entornos virtuales se ha transformado, dando lugar a experiencias que exploran nuevas posibilidades en ámbitos de la comunicación y la narrativa. (Rubio-Tamayo y Gértrudix Barrio, p. 2) [2]

Con la recreación de un laboratorio de RV no solo se ahorra el espacio físico, sino también el riesgo que implica el trabajar en un laboratorio tradicional.

Las técnicas relacionadas con la RV resultan muy adecuadas para la formación en todas aquellas disciplinas y oficios que requieran destreza, pues facilitan la realización de prácticas en todo tipo de situaciones, Obrist Bertrand y Martínez Jara, p. 3. [3]

1.1. TIC y educación

El rápido progreso de las Tecnologías de la Información y Comunicación (TIC) transforma continuamente la forma de buscar, seleccionar, organizar, elaborar, producir y transmitir los conocimientos, por lo que los sistemas educativos con sus modelos, métodos y estrategias han experimentado la necesidad de adaptarse a una sociedad cada vez más integrada a las TIC, puesto que éstas tecnologías han impulsado la renovación del contenido de los cursos y de los métodos pedagógicos, Bautista Sánchez, Martínez Moreno y Hiracheta Torres, [4].

Por estrategia de enseñanza puede entenderse, según Díaz Barriga [5], a los procedimientos que los docentes utilizan con flexibilidad, adaptación, autorregulación y reflexión, a fin de promover el aprendizaje significativo de los estudiantes.

Los materiales y recursos en sentido amplio, y en particular los didácticos, son importantes, pero no tienen un especial valor por sí mismos. Su uso queda completamente justificado cuando son integrados, de forma adecuada, en el proceso educativo, el cual debe ser compatible, a su vez, con el entorno más amplio que lo rodea (escolar, regional, social, etc.). Blanco Sánchez, p. 3 [6].

Los recursos didácticos o medios de enseñanza permiten crear las condiciones materiales favorables para cumplir con las exigencias científicas del mundo contemporáneo durante el proceso de enseñanza-aprendizaje. Permiten hacer más objetivos los contenidos de cada asignatura del Plan de Estudios, y por tanto lograr mayor eficiencia en el proceso de asimilación del conocimiento por los estudiantes creando las condiciones para el desarrollo de habilidades, hábitos, capacidades, y la formación de convicciones. (Cañedo, Cáceres, p. 80) [7].

Bautista Sánchez y colaboradores [1], suponen que en el mundo educativo podemos encontrar infinidad de aplicaciones de las TIC, tales como portales o webs educativas, aulas virtuales de enseñanza y aprendizaje, videoconferencia o aplicativos educativos accesibles a través de Internet. Por ello la educación requiere replantear los objetivos, las metas, sus pedagogías y sus didácticas a fin de cumplir con la misión de responder a las necesidades del hombre.

En la actual era de la información es inminente la incorporación al aula de tecnologías de la información y comunicación (TIC), este proceso hace necesaria una revisión de su uso educativo y el sentido didáctico con el que se implementan.

Hay que tener claro que las TIC nos dan acceso a la información, pero no por eso se genera conocimiento, esta gran cantidad de información implica el desarrollo de procesos cognitivos que permitan al estudiante identificar, clasificar y priorizar el valor académico de las consultas, así como llegar a una construcción personal de la respuesta. Díaz-Barriga, p. 3 [8].

Por estas razones se considera pertinente explorar los beneficios de la implementación de tecnologías como recurso para robustecer las estrategias de enseñanza y aprendizaje. Las tendencias de la ciudadanía digital demandan, de las instituciones de educación superior, mayor flexibilización de procedimientos, y de infraestructura, a fin de adaptarse a las nuevas modalidades de formación.

De acuerdo con Salinas p. 2 [9], para que las instituciones educativas respondan a los desafíos actuales, deben revisar sus referentes y “promover experiencias innovadoras en los procesos de enseñanza-aprendizaje, apoyándose en las TIC y haciendo énfasis en la docencia, en los cambios de estrategias didácticas de los profesores y en los

sistemas de comunicación y distribución de los materiales de aprendizaje”. Es decir que debe prestarse especial atención a los procedimientos docentes orientados por las intenciones educativas.

Los objetivos o intenciones educativos son enunciados que describen con claridad las actividades de aprendizaje a propósito de determinados contenidos curriculares, así como los efectos esperados que se pretenden conseguir en el aprendizaje de los alumnos al finalizar una experiencia, sesión, episodio o ciclo escolar. Díaz, p. 7 [10].

Actualmente la curiosidad es una de las características que rigen la educación y el aprendizaje. Es por eso que se intenta presentar un recurso novedoso para implementar en el aula, utilizada como una herramienta que robustece el método de enseñanza y aprendizaje. “La realidad virtual [...] revolucionará las aulas. Los expertos en pedagogía coinciden en que los beneficios de estos avances en el aprendizaje serán suficientemente notorios como para abrirse paso en el sistema educativo.” Núñez, parr. 1 [11].

1.1.1. Realidad Virtual como recurso educativo

En los últimos años, la implementación de la realidad virtual ha ido en incremento, incluyendo todas las posibilidades que ofrece en el ámbito educativo. De hecho, el desarrollo educacional ofrece un ambiente propicio para el diseño y aplicación de la RV, gracias a la capacidad de introducir al estudiante en entornos inmersivos multisensoriales donde puede interactuar de modo que estimule su proceso de aprendizaje e impactar en su formación.

La calidad de la enseñanza demanda la incorporación de diversos recursos y estrategias con el propósito de crear en la clase un ambiente de aprendizaje participativo, colaborativo, práctico y ameno.

Tal como sostienen Hilera, Otón y Martínez, parr. 33, [12].

La Realidad Virtual es una tecnología especialmente adecuada para la enseñanza, debido a su facilidad para captar la atención de los estudiantes mediante su inmersión en mundos virtuales relacionados con las diferentes ramas del saber, lo cual puede ayudar en el aprendizaje de los contenidos de cualquier materia.

Como bien afirma Hilera, Otón y Martínez (1999, parr. 17) [12], “[...] la realidad virtual se utiliza para tratar sistemas que no pueden ser manejados en el mundo real”.

Así como lo afirma Vera Ocete et al., p. 16 [13] “La Realidad Virtual es una tecnología aplicable al terreno de la educación, debido principalmente a su capacidad de visualizar los procesos en estudio, independientemente de la disciplina a tratar.”

Como mayor antecedente se cuenta con la investigación de Obrist Bertrand y Martínez Jara [3], quienes describen la aplicación de la Realidad Virtual en una experiencia de aprendizaje:

El proyecto consiste en una aplicación que permite al usuario interactuar en un entorno virtual por medio de una interfaz web, en el cual se encuentran modelos en tres dimensiones que simulan distintas actividades. La aplicación se centra en la educación por medio de la utilización de la tecnología de la realidad virtual, enriqueciendo la percepción del alumno a través de la interacción con objetos en un

mundo artificial, facilitando su aprendizaje. (Obrist Bertrand y Martínez Jara, p. 1) [3].

El mayor problema que identifican los autores, en la formación profesional, es que en la gran mayoría de las universidades y en lugares de enseñanza profesional, no se cuenta con equipos adecuados para la educación de los futuros profesionales, quienes en pocas ocasiones pueden trabajar en situaciones cercanas a la realidad durante su cursado académico. Por lo que la idea de montar un “laboratorio virtual”, un entorno controlado, donde el estudiante o profesional se puedan desenvolver como si estuvieran en su lugar de trabajo, podría ser la respuesta a esta problemática.

1.2. La RV en la medicina

Vázquez-Mata, parr. [14] señala que es hora de provocar un cambio en las metodologías educativas que están siendo aplicadas en la medicina. Una estrategia adecuada, tal como afirma Gutiérrez Maldonado, p. 21 [15], podría ser desarrollar un grado avanzado de realismo, utilizando entornos virtuales ya sea en la formación o en las prácticas profesionales.

El campo de la salud permite recrear cuerpos y estructuras muy definidas y concretas a través de modelos virtuales en 3D, Diaz, parr. 4 [16]. Son variadas las facilidades en el aprendizaje que ofrece la RV en diferentes áreas de estudio, como por ejemplo el área de la salud. Disponer de una virtualización, donde el estudiante puede interactuar de una forma inmersiva, abre muchas ventanas a la hora de trabajar con el cuerpo humano.

Risk, parr. 2 [17] afirma que:

Los escenarios interactivos virtuales permiten planificar en detalle una cirugía antes de su ejecución. [...] Con este desarrollo, los cirujanos cuentan ahora con una herramienta de planificación interactiva y tridimensional. Este sistema de simulación les permite a los profesionales médicos combinar imágenes obtenidas en tomografías computadas y resonancia magnética, y con estos datos precisos ensayar en una computadora la cirugía: se pueden determinar las características del tumor, planificar los cortes más eficaces y seleccionar de un banco de huesos de donante para trasplante la opción más adecuada para la reconstrucción. Para esto último, se digitalizaron los datos de dimensiones y morfología de más de 50 huesos y con ellos crearon la versión virtual del banco.

La RV es una herramienta tecnológica con la que el estudiante o el profesional puede trabajar en un entorno controlado, haciendo pruebas sin correr ningún riesgo. Tal como lo dice Valeriy, parr. 2 [18] al utilizar esta herramienta, los estudiantes o profesionales pueden ejecutar pruebas funcionales e introducir cambios para modificar el entorno y registrar los parámetros y movimientos virtuales.

Es oportuno tener en cuenta el impacto que, la implementación de estas tecnologías, tienen sobre los sujetos que operan con ellas.

1.3. Impacto

Para definir el concepto de impacto de un proceso educativo, se toma como modelo la definición de Aguilar, p. 53 [19]:

El impacto de un proceso docente- educativo se traduce en sus efectos sobre una población amplia: comunidad, claustro, entorno, estudiante, administración, identificando efectos científico - tecnológicos, económico - social - cultural - institucional, centrado en el mejoramiento profesional y humano del hombre y su superación social.

El termino impacto se asocia a la revolución de la educación. El hecho de causar un impacto en los estudiantes crea una revolución en el entorno académico y produce en el estudiante un sentimiento de comodidad a la vez que se interesa por aprovechar cada vez las TIC disponibles Villa Múnera [20].

Luego de un estudio de corte cuantitativo, López de la Madrid [21] analizando el impacto de la integración de las TIC, concluye que “con la integración de las TIC en la educación superior, los docentes han venido generando nuevas estrategias y actividades que hasta hace pocos años no se tenían contempladas”.

Prieto Díaz y colaboradores, p 95) [22] reconocen los impactos del desarrollo y la implementación de las TIC sobre la universalización de la educación superior, como “fuente de información, canal de comunicación, instrumento cognitivo y de procesamiento de la información”.

En la investigación de Graells, p. 12 [23] se encuentra que “las TIC deben usarse tanto como recursos de apoyo para el aprendizaje académico de las distintas materias curriculares, como para la adquisición y desarrollo de competencias específicas en TIC.” De esta manera se incentiva el aprovechamiento de las TIC ya que no solo produciría un impacto positivo en los estudiantes sino también que los ayudaría en su desarrollo profesional.

Los educadores no debieran hacer oídos sordos a lo que las nuevas tecnologías ofrecen, sino investigar nuevas TIC e implementarlas en el aula como instrumentos útiles para la construcción del ambiente áulico.

El impacto que produce la implementación de las TIC permite mejorar la calidad de la enseñanza, tal como lo afirma Álvarez, parr. 6 [24]: “Las TIC están teniendo una gran repercusión en el mundo educativo. Adaptarse a estas nuevas tecnologías es un proceso que se está llevando a cabo lentamente, pero que está permitiendo a muchos docentes mejorar la calidad de la enseñanza.”

Se encuentra que la implementación de las TIC en el ámbito académico no solo impacta en el estudiante, sino también en los docentes y en las relaciones entre ellos. En un estudio realizado por Sallán, Ceacero y Juan, p. 26 [25] encontraron lo siguiente:

Las TIC facilitan el acceso y difusión de la información, la integración de recursos variados, impactan en la configuración de la relación con los estudiantes, mejoran el clima del aula y fomentan la función de guía del profesor; también ayudan a personalizar y enriquecer el proceso de evaluación. Resulta también interesante resaltar la potencialidad que tienen para facilitar la autogestión del aprendizaje por parte del alumnado [...].

“Por evaluación de impacto se entiende el proceso evaluativo orientado a medir los resultados de las intervenciones, en cantidad, calidad y extensión según las reglas preestablecidas.” Abdala, p. 28-29) [26].

A los fines de medir el impacto de la implementación de la RV en el aprendizaje de los estudiantes, se administró la escala elaborada y validada por Tumino y Bournissen [27], a dos grupos de estudiantes de las carreras de la Facultad de Ciencias de la Salud de una universidad privada del centro de Argentina. Los grupos estuvieron definidos por el tipo de herramienta tecnológica usada para fortalecer las estrategias de enseñanza en el aprendizaje de la Anatomía. Los estudiantes participantes de uno de los grupos utilizaron la RV mientras que en el segundo grupo se aplicaron herramientas tradicionales, tales como presentaciones y láminas, para el abordaje de los mismos contenidos.

2. Metodología

El objetivo principal del estudio fue evaluar el impacto de la implementación de RV, como recurso para reforzar las estrategias de enseñanza de anatomía, de las carreras de la Facultad de Ciencias de la Salud, en una universidad de noreste de Argentina. Se pretendió establecer un análisis comparativo que permitiera identificar el nivel de impacto de la implementación de la RV en el aprendizaje de los estudiantes, respecto del impacto causado en el aprendizaje por la utilización de herramientas tradicionales. La variable de estudio fue el impacto de las TIC en el aprendizaje de los estudiantes en sus dos dimensiones (aprendizaje y estrategias de aprendizaje).

La variable grupo distinguió a los estudiantes que experimentaron la implementación de la RV, como recursos de refuerzo de las estrategias de enseñanza y aprendizaje, de aquellos que utilizaron las tecnologías tradicionales como libros, maquetas, videos y presentaciones.

Es importante aclarar que, si bien el grupo que experimentó con RV tuvo la oportunidad de manipular el equipo en el entorno de los softwares propuestos para el estudio de la anatomía, el tiempo disponible para ello fue de aproximadamente treinta minutos, lo que implica una limitación, dados los tiempos de aprendizaje propios para la integración de las tecnologías.

El impacto fue medido mediante la encuesta de Nivel de Impacto de la Implementación de las TIC en el Aula de Tumino y Bournissen [27]. Esta variable presenta una estructura compuesta por ítems vinculados al aprendizaje (con 8 ítems) y por ítems relacionados con estrategias de aprendizaje (con 10 ítems). La escala utilizada en las dos dimensiones es de intervalo desde 1 = muy en desacuerdo, 2 = en desacuerdo, 3 = neutro, 4 = de acuerdo y 5 = muy de acuerdo.

La hipótesis planteada fue: H_1 : Existen diferencias estadísticamente significativas de medias de impacto en el aprendizaje entre los grupos de estudiantes definidos por el tipo de tecnologías implementadas como herramientas de refuerzo de las estrategias de enseñanza y aprendizaje (Realidad Virtual o tecnologías tradicionales).

El estudio fue de tipo explicativo, dado que se buscó conocer el nivel de impacto de la implementación de la Realidad Virtual, como recurso tecnológico que refuerza las estrategias de enseñanza y aprendizaje, y compararlo con las medias de impacto en el aprendizaje de los estudiantes que experimentaron con tecnologías tradicionales.

La muestra estuvo compuesta por estudiantes de distintas carreras que cursaban temáticas de Anatomía en la Facultad de Ciencias de la Salud de la Universidad Adventista del Plata. Se conformaron dos grupos, el primero, de 57 estudiantes que no

experimentaron la implementación de la RV en el aula y el segundo grupo estuvo conformado por 55 estudiantes que probaron la RV en sus experiencias de aprendizaje. Se aplicó, en ambos grupos, la escala de Impacto de Tumino y Bournissen [27] con el propósito de obtener los datos que intervinieron en el análisis estadístico.

A fin de crear el ambiente de aprendizaje con RV, se instaló el equipamiento en el Laboratorio de Anatomía de la Facultad de Ciencias de la Salud, con dos softwares especializados de anatomía: (a) el Sharecare VR, desarrollado y distribuido por la empresa Sharecare Inc. y (b) el 3D organon VR anatomy, desarrollado y distribuido por Medis Media. Ambos softwares ofrecen una amplia gama de posibilidades de visualizar cada órgano sano y operar a partir de sus diferentes patologías.

Una vez recolectados los datos, se procedió a aplicar la Prueba T para muestras independientes a fin de comparar las medias de impacto, en sus dos dimensiones (a) aprendizaje y (b) estrategia de aprendizaje, entre los dos grupos de estudiantes diferenciados por las tecnologías aplicadas en el ámbito de aprendizaje.

3. Resultados

Antes de presentar los resultados obtenidos del análisis de los datos, cabe describir brevemente las impresiones, de los investigadores, que observaron las reacciones satisfactorias de los estudiantes al contemplar las funcionalidades de los softwares vistos desde la RV. Los estudiantes que tuvieron la oportunidad de manipular el equipamiento manifestaron interés en investigar, dentro del campo virtual, los contenidos vistos en clases; lo que permite sugerir que la tecnología promueve el interés por la profundización de los temas tratados y facilita su aprendizaje.

Se encontró que los estudiantes comparaban el uso de la RV con la utilización de maquetas o dibujos, generando un impacto positivo al ver la anatomía humana desde un escenario virtual.

Se compararon las medias de impacto, en sus dos dimensiones (aprendizaje y estrategias de aprendizaje), entre los dos grupos participantes. Para contrastar la primera hipótesis nula: No existen diferencias estadísticamente significativas de medias de impacto en el aprendizaje, entre los grupos de estudiantes definidos por el tipo de tecnologías implementadas como herramientas de refuerzo de las estrategias de enseñanza y aprendizaje (Realidad Virtual o tecnologías tradicionales), se aplicó la Prueba T para muestras independientes y se encontró una diferencia significativa de medias de impacto entre los dos grupos ($t_{(110)}=7.78$, $r=0.59$). La media de impacto en aprendizaje resultó mayor en el grupo de estudiantes que experimentó la implementación de la RV ($M = 4.61$; $DE = 0.37$), en contraste con el grupo que no utilizó la RV ($M=3.69$; $DE=0.79$).

El tamaño del efecto es grande, explicando más del 25% de la varianza total de los niveles de impacto ($r=0.59$), lo que mostró que los equipos de RV pueden actuar como refuerzo de las estrategias de enseñanza y aprendizaje.

La segunda hipótesis nula: H_{02} : No existen diferencias estadísticamente significativas de medias de impacto de estrategia de aprendizaje entre los grupos de estudiantes definidos por el tipo de tecnologías implementadas como herramientas de refuerzo de las estrategias de enseñanza y aprendizaje (Realidad Virtual o tecnologías tradicionales).

Se aplicó la prueba T para muestras independientes y se encontró una diferencia estadísticamente significativa de media de estrategia de aprendizaje entre los grupos ($t_{(110)}=9.50$, $r=0.67$ $p<.05$). La mayor media de impacto en estrategia de aprendizaje se obtuvo en el grupo de estudiantes que experimentaron con RV ($M = 4.58$, $DE = 0.41$), en contraste con el grupo que nunca utilizó la RV ($M=3.45$; $DE=0.78$).

El tamaño del efecto revela que el modelo explica más del 25% de la varianza total del impacto ($r=0.67$), lo que permite inferir que la implementación de la RV puede fortalecer las estrategias de enseñanza y aprendizaje implementada en el ámbito educativo.

4. Discusión y conclusiones

Del análisis de los datos obtenidos, se halló que existe una diferencia estadísticamente significativa de medias de impacto, tanto en el aprendizaje como en las estrategias de aprendizaje, entre los dos grupos estudiados. El grupo de 57 estudiantes que nunca tuvieron contacto con equipos de RV mostró una media en el factor de aprendizaje de 3.69, con una desviación estándar de .79; mientras que el grupo de 55 estudiantes que tuvieron la posibilidad de usar el equipo de RV, obtuvo una media en el mismo factor de 4.61, con una desviación estándar de .37.

Respecto del factor de estrategias de aprendizaje, se encontró que en el primer grupo de 57 estudiantes se obtuvo una media de 3.45 con una desviación estándar de .78; mientras que el segundo grupo de 55 estudiantes mostró una media de 4.58 con una desviación estándar de .40.

Si bien en los datos recabados, de la aplicación de la escala, se encontraron algunas puntuaciones en torno a la neutralidad o al desacuerdo, se supone que la restricción del tiempo destinado a la manipulación del equipo significó una limitación en la percepción del potencial que ofrecen estas herramientas, como oportunidades para crear los escenarios de aprendizaje.

De las observaciones de los investigadores, resulta relevante destacar el entusiasmo que los estudiantes manifestaron por explorar el entorno virtual con propósitos educativos. La reacción más recurrente recogida en las observaciones fue la de asombro durante la navegación y exploración de las funcionalidades en el entorno virtual, donde los estudiantes destacaban la calidad de las representaciones anatómicas visualizadas. Estas manifestaciones fueron atesoradas como un estímulo para continuar mejorando los ámbitos de enseñanza, y las estrategias ofrecidas a los estudiantes, con el objetivo principal de brindarles las mejores oportunidades de formación profesional, en armonía con Levis [28]:

Sin negar las evidentes ventajas que ofrece el uso de las técnicas de simulación y comunicación digital en la enseñanza, consideramos que en demasiadas ocasiones se suelen ignorar tanto las necesidades como los comportamientos de los alumnos, que son en última instancia quienes han de sacar provecho de estas herramientas. (Levis, p. 17) [28].

Fue este el objetivo que impulsó el estudio, con resultados que permiten declarar a las nuevas tecnologías, tal como la RV, como más que recomendables a la hora de diseñar ambientes favorecedores del aprendizaje.

A partir de las observaciones y resultados obtenidos en el estudio, es posible inferir que la RV se presenta como un recurso recomendable para reforzar las estrategias de enseñanza y aprendizaje en el aula y en las prácticas profesionales.

Se encontró que el impacto causado por la RV en los estudiantes, desde sus dos dimensiones, es mayor ($M= 4.61$ y $M= 4.58$) en comparación con los observados por las tecnologías tradicionales ($M= 3.69$ y $M= 3.45$), efecto que explica más del 25% de la varianza total.

Los resultados se encuentran en concordancia con Vera Ocete et al., p. 16) [13], quienes afirman que la implementación de la RV facilita a los profesores su tarea de impartir clases mostrando de una forma más interactiva los contenidos de sus asignaturas.

Si bien los resultados evidencian las ventajas de la implementación de la RV en el aula, se reconoce la necesidad de dedicar tiempo en la integración de esta tecnología por parte de docentes y estudiantes, dado que el aprovechamiento de sus múltiples funcionalidades demanda un período de aprendizaje que bien merece invertir.

Como proyección a futuro, la institución donde se desarrolló el trabajo se encuentra diseñando un espacio dedicado al equipamiento que hará posible convertirlo en un laboratorio de realidad virtual.

5. Referencias

1. Botella Arbona, C., García-Palacios, A., Baños Rivera, R. M. y Quero Castellano, S. (2007). Realidad Virtual y Tratamientos Psicológicos. *Cuadernos de medicina psicosomática y psiquiatría de enlace*, 82, 17-31.
2. Paz e Silva, A. (2007). *Realidad Virtual*. Panamá.
3. Obrist Bertrand, V. U. y Martínez J. (2015). Aplicación de la Realidad Virtual en una experiencia de aprendizaje. Paraguay: facultad Politécnica – U.N.E.
4. Bautista Sánchez, M. G., Martínez Moreno, A. R. y Hiracheta Torres, R. (2014). El uso de material didáctico y las tecnologías de información y comunicación (TIC's) para mejorar el alcance académico. *Ciencia y Tecnología*, 14, 183-194. Recuperado de <https://dSPACE.palermo.edu/ojs/index.php/cyt/article/view/217/111>
5. Díaz Barriga, F. (2003). Cognición situada y estrategias para el aprendizaje significativo. *Revista Electrónica de Investigación Educativa*, 5 (2). Consultado el 17 de julio de 2009 en: <http://redie.ens.uabc.mx/vol5no2/contenido-arceo.html>
6. Blanco Sánchez, M. (2012). *Recursos didácticos para fortalecer la enseñanza-aprendizaje de la economía*. Universidad de Valladolid, Valladolid, España.
7. Cañedo Iglesias, C. y Cáceres Mesa, M. (2011). Fundamentos teóricos para la implementación de la didáctica en el proceso enseñanza-aprendizaje. Cuba: Universidad Cienfuegos, Carlos Rafael Rodríguez.
8. Diaz-Barriga A. (2013). *TIC en el trabajo del aula. Impacto en la planeación didáctica*. Revista Iberoamericana de Educación Superior.
9. Salinas, Jesús (2004). Innovación docente y uso de las TIC en la enseñanza universitaria. *Revista de Universidad y Sociedad del Conocimiento (RUSC)*, 1(1). Recuperado de <http://www.uoc.edu/rusc/dt/esp/salinas1104.pdf>.
10. Díaz, F. (1999). Estrategias Docentes para un Aprendizaje Significativo. Una interpretación constructiva. McGraw-Hill, México, cap. 4 y 5.

11. Núñez, P. (2016). Realidad virtual: así transformará nuestro sistema educativo. El Mundo. España.
12. Hilera, J. R., Otón, S., Martínez, J. (1999). Aplicación de la Realidad Virtual en la enseñanza a través de Internet. Universidad Complutense Madrid.
13. Vera Ocete, G., Ortega Carrillo, J. A. y Burgos González, A. (2003). La realidad virtual y sus posibilidades didácticas. *Etic@net*, 2. ISSN-e 1695-324X, Recuperado de <https://dialnet.unirioja.es/servlet/articulo?codigo=6871642>
14. Vázquez-Mata, G. (2008). *Realidad virtual y simulación en el entrenamiento de los estudiantes de medicina*. Centro de Simulación Médica Avanzada e Innovación Tecnológica. Parque Tecnológico de Ciencias de la Salud. Fundación IAVANTE. Junta de Andalucía, España.
15. Gutiérrez Maldonado, J. (2002). Aplicaciones de la Realidad Virtual en psicología clínica. España: Universidad de Barcelona.
16. Diaz, M. M. (2016). *Realidad virtual y educación, un futuro prometedor*. ojuLearning, España.
17. Risk, M. (2015). Realidad virtual al servicio de la medicina. Conicet
18. Valeriy, K. (2016). Realidad virtual para simplificar el diagnóstico del Parkinson y la esclerosis múltiple. EUROPA PRESS, Madrid, España.
19. Aguilar M. A. (2009). El impacto de la carrera de economía de la BUAP en el mercado laboral: la visión de los egresados de la generación 1995-2000. Benemérita Universidad Autónoma de Puebla. México.
20. Villa Múnera D. S. (2017). Tic y formación virtual, el impacto de la tecnología en la educación. ElMundo.com
21. López de la Madrid, M. C. (2013). Impacto de las tecnologías de la información y la comunicación (TIC) en el docente universitario: El caso de la Universidad de Guadalajara. *Perspectiva Educativa*, 52(2), 4-34. Recuperado de: <http://www.redalyc.org/pdf/3333/333328170002.pdf>
22. Prieto Díaz, V., Quiñones La Rosa, I., Ramírez Durán, G., Fuentes Gil, Z., Labrada Pavón, T., Pérez Hechavarría, O. y Montero Valdés, M. (2010). Impacto de las tecnologías de la información y las comunicaciones en la educación y nuevos paradigmas del enfoque educativo. *Educación Médica Superior*, 25(1),95-102. Recuperado de: http://scielo.sld.cu/scielo.php?script=sci_arttext&pid=S0864-21412011000100009
23. Graells P. M. (2012). *Impacto de las TIC en la educación: funciones y limitaciones*. 3Ciencias. Revista de investigación.
24. Álvarez A. F. (2017). *Ventajas y riesgos de las TIC en educación*. Nubemia.
25. Sallán J. G., Ceacero D. C., y Juan C. M. (2017). El impacto de las TIC en el aula desde la perspectiva del profesorado. Fundación Mapfre, Universitat Autònoma de Barcelona.
26. Abdala, E. (2004). *Manual para la evaluación de impacto en programas de formación para jóvenes*. Montevideo: CINTERFOR.
27. Tumino M., Bournissen J. y Forneron F. (2018). *Validación de contenido de instrumento para medir el nivel de integración tecnológica en el aula y el nivel de impacto en los estudiantes*. XXIV Congreso Argentino de Ciencias de la Computación 2018. Facultad de Ciencias Exactas de la Universidad Nacional del Centro de la Provincia de Buenos Aires, Tandil.

28. Levis, D. (1997). Realidad virtual y educación. Documento on line. Recuperado de:
<https://docplayer.es/33253453-Diego-levis-realidad-virtual-y-educacion.html>

MoLE: A web authoring tool for building mobile learning experiences

Pedro A. Dal Bianco¹; Federico R. Mozzon Corporaal¹; Alejandra B. Lliteras^{1,2};
Julián Grigera^{1,2}; Silvia E. Gordillo^{1,2}

¹ UNLP, Facultad de Informática, LIFIA, Calle 50 &, Av. 120, La Plata, Buenos Aires

² CICPBA, Buenos Aires, Argentina

{pdalbianco, fmozzon, lliteras, jgrigera, gordillo}@lifia.info.unlp.edu.ar

Abstract. This paper presents a web authoring tool that allows teachers to build their own mobile learning experiences, so that their students can carry out a learning activity of the teachers authorship. In this work, a learning activity consists of a set of tasks organized through a workflow, which are given to students in relevant places (within a physical space). The workflow defines the way in which students access each of the tasks proposed in the activity. A preliminary case study and two technological implementations are presented. Although the present is a work in progress, it is based on previous works on the subject and proposes an evolution of them, incorporating technological and usability improvements both in the author's web tool and in the mobile learning application that it generates.

Keywords: authoring tool, mobile learning, mobile learning application, web tool, workflow, learning activity, physical interaction, digital interaction

1 Introduction

Mobile devices, especially smartphones, are heavily present in various activities of daily life. These do not only work as cell phones, but they are also powerful tools that integrate functionalities for which, many years ago, it was necessary to have different physical devices. For example, they integrate the possibility of taking photographs, recording audio, accessing online games, knowing the current geographical position and being connected through social networks.

The presence of these devices went through the walls of schools occupying a place in the classroom. It is very important for the teacher to capitalize on its use for educational activities, guiding the student to a safe and appropriate use of the educational context. Some proposals for its use in the classroom, for example, involve using search engines like Google or applications like the calculator to solve mathematical activities. In other cases, technology experts design and implement educational proposals mediated by mobile technology so that teachers can use them in the classroom, but without the teachers being able to participate in that process.

However, it should be possible for teachers to create their own mobile learning activities to be used by their students as part of the classroom activity, without requiring the assistance of technology experts. Different approaches to this problem are given by existing proposals [1], [2], which although they allow the creation of a whole mobile application, still require the teacher to have some knowledge about

concepts such as block-based programming, and aim to the user learning those skills and creating a generic application, rather than being used to design a learning activity.

Because of this, in this work, we highlight the need of providing teachers with tools that allow them, in an intuitive and independent way, to design, implement and carry out mobile learning experiences, to be used by their students through mobile devices. As this tool is aimed at users with little or no knowledge about application development, one of the first aspects to consider is the fact that it should be as intuitive as possible, so its usability will be a fundamental pillar at design time. This is one of the main points that differences this tool with other works in this same line, since the user has more freedom when developing an activity, this implies (as will be seen in Section 2) that he will have to learn certain programming concepts to be able to carry out the development of an application.

The present work aims at constituting an element of empowerment for teachers when designing and proposing their own solutions mediated by mobile technology.

This paper is structured as follows: Section 2 presents related works, then Section 3 describes the work that is adopted as the basis for the proposal introduced in Section 4. Section 5 presents the implementation of the proposed work and in Section 6, a first case study is presented where the web authoring tool is used to generate two technological implementations. Finally, in Section 7, the conclusions are presented and future works are proposed.

2 Related Work

Some similar tools already exist in the field. App Inventor [1], developed by MIT, allows users to generate their own mobile applications with the simplicity provided by block programming, allowing to use all the previously mentioned tools such as GPS, camera, storage and other functionalities provided by the mobile devices. Although it is precisely this freedom that users have when creating their tool which, as mentioned in [3], does not allow exploited all functionalities. Many users of this tool, however, cannot fully exploit its functionalities as they are not familiar with programming. As mentioned in [3], users do not use something that is not intuitive.

Vedils [2], is built on top of App Inventor, incorporating the integration of augmented reality and learning analytics features. This tool maintains most of the features of the aforementioned App Inventor [1], and therefore many of the problems already introduced at the beginning of this section. In [4] Vedils is studied in depth and a concrete case of use where is presented for teachers developed their own educational activities, but for this, the teachers first had to carry out workshops where they were offered knowledge about programming in blocks assisted by experts. In [5] there is another case of using Vedils for the creation of an educational tool, this time to assist in the teaching of a mathematics course, in this case, the tool was developed entirely by experts. The above suggests that Vedils does not allow people with no programming background to design and implement learning activities with mobile devices on their own.

Another relevant work is described in [6], where a web authoring tool is developed so that teachers in the area of mathematics can design learning activities based on

positioning, in which students must go through a series of locations on a map, guided by mobile devices. Although to design an educational activity using this tool does not require any specific knowledge in the area of programming, the tool aims exclusively to be used in the area of mathematics, and the locations on the map are only used to build a route, the student does not receive specific content in each of them.

3 Basis of the proposed work

We have previously presented a web authoring tool for configuring Android learning applications. Using it, a teacher defines a learning activity divided into tasks to be performed in relevant places within a physical space. In particular, this tool considers three tasks for collecting elements of the real world and one task for depositing them.

When the teacher creates the learning activity, they define the physical space where the activity will take place, between 2 and 3 criteria for depositing elements and 3 tasks for collecting elements. For each of these tasks, the teacher defines: i) a statement that the student will receive in their mobile device, ii) the elements of the physical world with which they can interact, indicating which are correct to collect to answer the statement and which are not, iii) to which criterion of the defined to deposit each element corresponds and iv) optionally, additional scaffolding material.

The above described defines a particular scheme of work. This implies a particular way to design and plan the activity, that is, the amount and type of tasks to be performed is predetermined, as well as the order in which the tasks should be performed. The collection tasks can only be carried out sequentially. The sequence is determined by the order of creation. The tasks that involve depositing elements can be accessed after performing any collection task. The above can be visualized in Fig. 1.

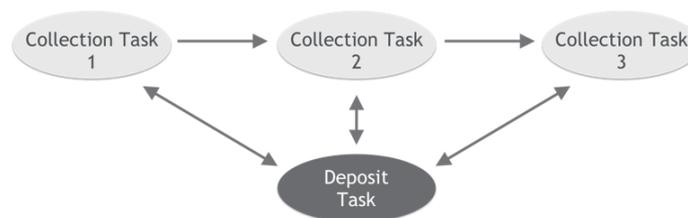


Fig. 1. Scheme of work

From the technological aspects, the mobile application is presented as an independent and configurable tool. The one that considers the positioning of the users through the use of QR codes doesn't require an internet connection or the use of GPS.

The configuration of the mobile application is done through a JSON file and a set of images of the physical space. These images are used to assist the user in their mobility. Fig. 2 shows the architecture of the solution presented in [7].

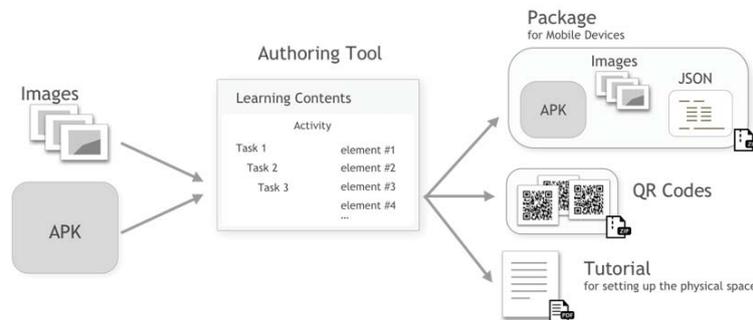


Fig. 2. Image (translated) extracted from [7] showing the architecture of solution presented.

Using this tool, the teacher can generate everything necessary for the configuration of the learning activity to be done with mobile devices, namely:

- The QR codes to identify each task, the elements of each task and the deposits defined for the activity.
- An instruction on how to configure the physical space where the activity will take place. In this instruction, each of the steps to be followed is detailed.
- Also, a folder containing:
 - The generic mobile application that must be installed on the mobile device
 - A JSON file that contains the configuration of the activity
 - A set of auxiliary images for the configuration of the activity

The tool presented in [7], however lacks the necessary flexibility for the teachers to implement their own ideas, since it doesn't allow creating tasks that are not for collecting or depositing elements, and that the only interaction considered is physical. Additionally, "Resuelvo Explorando" allows only access to a single configuration.

4 A Web Authoring Tool to Build Mobile Learning Experiences

Based on the need to have a flexible authoring tool that allows the teacher to create mobile learning experiences, we designed "MoLE". MoLE is a web authoring tool that allows teachers without knowledge about mobile development or programming to create their own learning activities mediated by mobile technology

The construction of mobile learning activities using MoLE follows the conceptual framework proposed in [8]. This means that the activity is defined in terms of the following aspects: the educational content (tasks), the physical space where it will take place, the identification of relevant places within that space, the structuring of the content (linear, free or graph) and finally indicating in which place of the physical space will be carried out each component of the chosen structure. Defining the activity in this way favors the possibility of reusing the aspects involved [8].

MoLE allows creating different types of tasks:

- **Multiple choice tasks**, where the person who performs the activity must choose the correct option from a set of options created by the teacher.
- **Open question tasks**, where a text should be written in answer to a question selected by the teacher.

- **Collection tasks**, where it should recollect real-world objects based on a criterion defined by the teacher.
- **Deposit tasks**, where deposit the elements recollected in the previous tasks in an appropriate deposit.
- **Tasks for the generation of content**, either through images, audio or video that respond to a given statement.

In addition to the described above, the possibility of extending the different types of tasks in the future is contemplated in a simple way in order to provide more flexibility to the teacher when it comes to diagramming the tasks.

The content is structured by a workflow. Each step indicates the task and the place within the physical space where it will be provided. Finally, the teacher sets the order. For this, they can choose between predetermined workflows: sequential (lineal), without pre-established order (free or set), or customized (graph). In addition, each activity designed by the teacher can be reused on any mobile device that has the application for mobile devices.

The type of each task can be multiple choice question, single-choice question, open question, image captures, video or sounds according to a slogan, collection or deposit of pre-established elements as part of the task.

The concept of workflow has been implemented considering some of the aspects described in [9] and [10] where different types of workflows are seen and how each one can be convenient for different activities and for different user experiences. According to the authors, an important aspect when designing a workflow is that the users, in this case the teachers, get constant feedback of what they are doing.

5 MoLE Implementation

The implementation of MoLE consists of three components that interact with each other: a web authoring tool, a REST API and a mobile learning application. Below is a brief description of each one.

5.1 Web authoring tool

This tool guides the teacher in the creation of location-based learning activities. Each activity can be composed by tasks of different types. Each task can be done in a relevant place in the physical space. The tool proposes stages of creation following the conceptual framework proposed in [8] and implements the activities approach presented in [11]. The front-end of this tool was developed using the libraries ReactJS¹ and Redux², and for the backend it connects with an API (see Section 5.2).

¹ <https://reactjs.org>

² <https://redux.js.org>

5.2 REST API

This component provides an interface to a database designed with MongoDB³ through methods to save the generated activities and then retrieve them and be able to consult information about them. This mechanism offers the possibility of retrieving information from both the web tool (described in Section 5.1) and the mobile learning application (described in Section 5.3) and if a new type of device is added eventually, do so through these same methods.

This API was implemented using NodeJS⁴ and Express⁵ library.

5.3 Mobile learning application

The mobile learning application downloads the activities from the API described in Section 5.2, allowing in one hand to have multiple activities in the mobile application and in other hand, to execute each one, independently of each other, repeatedly. This application was implemented considering the aspects of usability and the lessons learned in [12]. React Native⁶ was used for its development, which allowed to have a native and multiplatform application. This application is available for any mobile device with an Android version greater than 5.0, or a version of IOs greater than 10.0. Fig. 3 shows the architecture of the implementation.

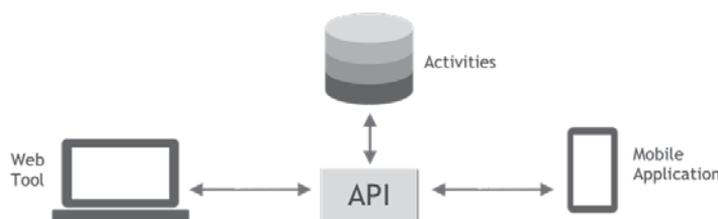


Fig. 3. Architecture of the proposed implementation

6 Case Study

In this section we present a case study to use the proposed web tool to obtain a functional mobile application. In the first place, the learning activity is described from the conceptual perspective and then, two possible forms of implementation are presented, on the one hand involving physical interaction with elements of the real-world and on the other, with digital elements (digital interaction).

³ <https://www.mongodb.com>

⁴ <https://nodejs.org>

⁵ <https://expressjs.com>

⁶ <http://www.reactnative.com>

6.1 Conceptual learning activity

Following the conceptual framework proposed in [8], a learning activity was designed to raise awareness regarding the recycling of Waste Electrical and Electronic Equipment (WEEE).

In the first place, a multidisciplinary team was formed with members of the “E-Basura” program of the UNLP⁷, members of the Environmental Awareness Board of the Informatics School of the UNLP and some of the authors of this work. Then, the learning activity was conceptually designed. For this, three tasks were defined whose statements reflected the need for the participant to identify WEEE elements according to the use that could be given to them, and then determine if said element could be delivered to the user.

This activity was designed to be carried out after members of the “E-Basura” program presented the necessary concepts to the participants, in order to know which of them were not acquired by the participants and reinforce them

In a next instance, aspects related to the interaction that the participant was expected to have with the WEEE elements (that were a priori conceptual) were determined. From this, two technological implementations were proposed, one that involved elements of the real-world, so that the participant could manipulate by means of a physical interaction and, on the other hand, one based on digital elements through digital interaction. Each of them is described below.

6.1.1 Implementation considering physical interaction

The implementation is based on the fact that the participants could see and "touch" (manipulate) electrical and electronic equipment, which was considered important so that these could later recognize in their usual spaces (e.g. study, work, housing), elements that could be taken to the recycling center of the “E-Basura” program.

In this implementation of the case study, to carry out the physical interaction, three tasks of the type of collection and two of the deposit types (both types described in section 4) were instantiated. A collection or deposit criteria was specified for each one, and then a series of elements was defined, that correspond to the elements of the real-world, indicating the criteria under which they had to be deposited or collected. We used different electrical and electronic elements and statements related to their utility.

From the perspective of the web tool proposed in this work, it was necessary to create the leaning activity, the tasks and set up the workflow with the tasks and indicate where each of the tasks of the workflow will be carried out. Upon completion of the creation from the web authoring tool, the user has the following files:

- A configuration file indicating how to configure the physical space.
- A file with QRs identifying the tasks.
- A file with QRs that identify each element of each task.
- A code to download from the mobile application the configuration (activity) created from the tool for the WEEE domain.

⁷ <https://e-basura.unlp.edu.ar>

To use the mobile application, users must:

- Configure the physical space of use, locating the corresponding QR codes to the tasks according to the configuration file of the space.
- Place each corresponding QR element.
- Physically arrange each element with your QR, in the relevant place of your task.
- Access to the download code from each device, to make available the configuration for WEEE.

Fig. 4 shows WEEE elements used for the three collecting tasks.



Fig. 4. WEEE elements used for the three collecting tasks

Next, Fig. 5 shows the statements of the three tasks that involve collecting elements for the use that can be given and that it implies physical interaction, since the real-world elements must be manipulated and its QR read from the mobile device.

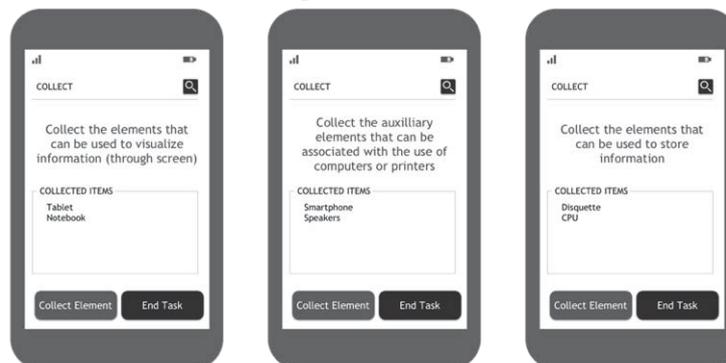


Fig. 5. Visualization of the statements to collect WEEE elements with physical interaction

6.1.2 Implementation considering digital interaction

In this implementation, the end user does not have to manipulate WEEE elements from the real-world or use the QR reader to collect them. For each setpoint of a task that it receives in a certain place (by reading a task QR), in its device it visualizes a list with the names of the elements of the task and it will have to select each one that consider correct by touching the screen.

From the perspective of the tool proposed in this work to create digital interaction, it was necessary to create the learning activity, the tasks and the workflow. This does not vary from the implementation made to be used with physical interaction.

After the creation from the web authoring tool, the user obtains the following files:

- A configuration file indicating how to configure the physical space.
- A file with QRs identifying the tasks.
- A code to download from the mobile application the configuration (activity) created from the web tool for the WEEE domain

To use the mobile application, users must:

- Configure the physical space of use, locating the QR codes corresponding to the tasks according to the configuration file of the space
- Access the downloaded code from each device to make available the configuration for WEEE

Fig. 6, shows the statements of the three tasks considering digital interaction.

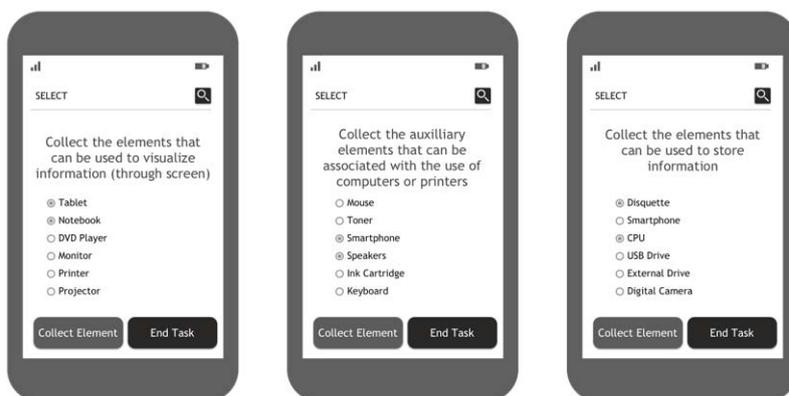


Fig. 6. Visualization of the statements to select WEEE elements with digital interaction

7 Conclusions and Future Work

This article presented the work in progress of a web authoring tool for the construction of mobile learning experiences that consider the position of the student to provide tasks consistent with it. Given that the development is at an early stage, there have not yet been tests with end users that allows us to compare the students' preference for both types of interaction implemented. It is expected that this tool will give the teacher the possibility of including technology in the class through activities of his authorship carried out on mobile devices, in an intuitive way and without the need to have knowledge about application development.

As future work we plan to complete the development of the tool presented in order to be evaluated by end users, considering aspects of learnability and usability of both the tool and the resulting mobile application. We also plan to work in the management of a community of teachers who use this tool to design their own learning activities, in order to get real feedback regarding the tool's functionality and usefulness. Another proposal consists in using the modeling approach that implements our tool and the lessons learned to perform activities that involve data or samples collection (e.g. images, audio, video, guided observations) to be used in Citizen Science projects.

Acknowledgments

The authors would like to thank Patricia Bazán director of "Recicla tu Compu-Recicla tu Mundo" project of the "Facultad de Informática , UNLP, Viviana Ambrosi director to "E-Basura" program of de UNLP, and Edgar Vega member of this program for their contribution on the configuration using the proposed tool.

References

1. App Inventor, <http://appinventor.mit.edu>
2. Vedils, <http://vedils.uca.es>
3. Xie,B., Shabir,I., Abelson, H.: Measuring the usability and capability of App inventor to create mobile Applications. In: 3rd International Workshop on Programming for Mobile and Touch (PROMOTO 2015), pp. 1--8. ACM, New York, NY, USA, (2015). DOI=<http://dx.doi.org/10.1145/2824823.2824824>
4. Mota., J.M., Ruiz-Rube, J., Doderó, J.M., Arnedillo-Sánchez, J.: Augmented reality mobile app development for all. *Computers & Electrical Engineering*. 65, 250--260 (2018). DOI: 10.1016/j.compeleceng.2017.08.025.
5. Person,T., Mota, J.M., Listán, M.C., Ruiz-Rube, I., Doderó, J.M., Rambla Barreno,F., Patino, C.M., Serván, A.R., Vidal Pérez, J.M.: Authoring of educational mobile apps for the mathematics-learning analysis. In: Sixth International Conference on Technological Ecosystems for Enhancing Multiculturality (TEEM'18), Francisco José García-Peñalvo (Ed.) pp. 299--305 ACM, New York, NY, USA. (2018). DOI: 10.1145/3284179.3284234
6. Sollervall, H., Gil de la Iglesia, D., Zbick, J.: Supporting Teachers' Orchestration of Mobile Learning Activities. *Mathematics Education in the Digital Era*, 91--111. (2018). DOI:10.1007/978-3-319-90179-4_6
7. Mozzon Corporaal., F.R., Dal Bianco, P.A., Lliteras, A.B., Grigera, J., Gordillo, S.E.: Una herramienta de autor web para generar aplicaciones educativas basadas en posicionamiento con interacción física. XXIV Congreso Argentino de Ciencias de la Computación. 253-262. ISBN: 978-950-658-472-6. (2018).
8. Lliteras, A.B., Challiol, C., Gordillo, S.E.: Location-based mobile learning applications: A conceptual framework for co-design," Twelfth Latin American Conference on Learning Technologies (LACLO), La Plata, pp. 1-8. (2017) DOI: 10.1109/LACLO.2017.8120946
9. Sprinks, J., Wardlaw, J., Houghton, R., Bamford, S., & Morley, J.: Task Workflow Design and its impact on performance and volunteers' subjective preference in Virtual Citizen Science. *International Journal of Human-Computer Studies*, 104, 50--63 (2017) DOI: 10.1016/j.ijhcs.2017.03.003
10. Crowston, K., Kitchell, E.M., Østerlund, C.: Coordinating Advanced Crowd Work: Extending Citizen Science. In: 51st Hawaii International Conference on System Sciences. Vol. 4, no. 1, pp. 1681--1690. (2018). DOI: 10.24251/HICSS.2018.212
11. Lliteras, A.B., Gordillo, S.E., DalBianco, P.A., Mozzon, F.R.: A customizable location-based Mobile Learning Prototype: A case of study, XIII Latin American Conference on Learning Technologies (LACLO), in press, (2018)
12. Lliteras, A.B., Grigera, J., Dal Bianco, P.A., Mozzon Corporaal, F.R., Gordillo, S.E.: Challenges in the design of a customized location-based mobile learning application. XIII Latin American Conference on Learning Technologies (LACLO), in press, (2018).

Perfiles de tesistas de la disciplina informática y su vinculación con los proyectos de investigación de la UM

Iris Sattolo¹, Marisa Panizzi¹, Javier Lafont¹, Nicolás Armilla¹

¹ Universidad de Morón. Facultad de Informática, Ciencias de la Comunicación y Técnicas Especiales. Cabildo 134. Pcia de Buenos Aires. Morón. Argentina.

iris.sattolo@gmail.com; marisapanizzi@outlook.com; lafontjavier@hotmail.com; nicolasarmilla@hotmail.com

Resumen. La participación de estudiantes de carreras de grado en proyectos de investigación permite detectar vocaciones para que desarrollen su carrera como investigadores en el ámbito académico y/o reconozcan la importancia de la investigación en el contexto industrial. En este trabajo, el objetivo es descubrir la relación entre las características de los tesistas y su vinculación con los proyectos de investigación ejecutados en la UM, como así también la identificación de los tesistas que han experimentado de otras actividades vinculadas a la investigación. Para alcanzar el objetivo, se realizó un conjunto de experimentos utilizando los algoritmos Kmeans, Expectation Maximization, J4.8, NaivesBayes, y BayesNet. Se utilizaron los entornos WEKA y Elvira. Los resultados de los experimentos permitieron descubrir que el motivo de la selección de la tesis es el atributo de mayor incidencia hacia la participación en proyectos de investigación como así también en otras actividades de investigación propuestas por la cátedra.

Palabras clave: Tesistas de informática, Perfiles, Proyectos de investigación, Minería de datos.

1 Introducción

La aplicación de la minería de datos en el ámbito de Educación ha adquirido un gran auge en la última década para resolver problemas de diferentes tipos. La Minería de Datos Educativos (Educational Data Mining - EDM) es una disciplina relacionada con el desarrollo de métodos para extraer información útil a partir de los datos que se generan en los entornos educativos, y utilizarla para mejorar dicho entorno. La información así obtenida se convierte en el insumo indispensable para la toma de decisiones [1].

En el contexto internacional, existen publicaciones previas, las cuales han demostrado que la minería de datos se puede utilizar para detectar estudiantes en riesgo de deserción. En su trabajo, Luan [2] aplicó exitosamente técnicas de minería de datos para predecir qué grupos de estudiantes podrían abandonar los estudios. En su trabajo, Lin [3] usó técnicas de minería de datos para optimizar los esfuerzos para retener estudiantes. Investigadores de la Universidad Estatal de Bowie, desarrollaron

un sistema basado en minería de datos que permite a dicha institución, identificar y atender a estudiantes en riesgo de abandono de estudios [4].

En los últimos años, la Universidad de Morón mediante la Secretaria de Ciencia y Tecnología (SeCyT-UM) ha priorizado la investigación dentro de su plan estratégico. La presentación y el desarrollo de los proyectos de investigación están orientados a estimular y afianzar en la UM actividades sistemáticas que están estrechamente relacionadas con la generación, el perfeccionamiento y la aplicación de los conocimientos en Ciencia y Tecnología. Se espera con esta acción promover dentro del campo científico y tecnológico la investigación e innovación, la formación de recursos humanos, las actividades de difusión, el desarrollo de servicios tecnológicos (bases de datos, biblioteca especializada, etc.) y la transferencia de conocimientos a la Comunidad [5].

Este trabajo tiene como objetivo definir los perfiles de los estudiantes que han participado en proyectos de investigación para la realización de su trabajo de fin de carrera. También se pretende lograr caracterizar a estudiantes que hayan vivenciado la escritura de artículos y realizado su presentación en eventos científicos, aunque no hayan formado parte de un proyecto de investigación de la UM.

En la sección 2 se describen los trabajos relacionados, la sección 3 presenta el desarrollo del proceso de minería de datos, se describe cada uno de los pasos realizados y finalmente en la sección 4, se presentan las conclusiones y futuros trabajos.

2 Trabajos relacionados

En trabajos anteriores [6], realizamos un estudio de mapeo sistemático siguiendo el proceso propuesto en [7]. El objetivo del SMS (en Inglés, Systematic Mapping Studies), las preguntas de investigación, el formulario de extracción utilizado, el esquema de clasificación de estudios primarios, la distribución de los estudios según la clasificación propuesta y el listado de los estudios primarios utilizados para el SMS, se encuentran en el Apéndice: <https://doi.org/10.6084/m9.figshare.9202445>

De un total de 110 estudios primarios, existen trabajos que definen perfiles de estudiantes; pero en la caracterización de este, no se considera la relación con proyectos de investigación de su universidad. En estos trabajos, tampoco se discute si los estudiantes vivenciaron la escritura de artículos y/o la participación en eventos científicos. Si bien reconocemos que nuestro SMS no es absoluto, nos ha permitido descubrir las características consideradas en otras investigaciones para la definición de perfiles de estudiantes.

3 Desarrollo

Para el descubrimiento de la información se aplican las fases del proceso KDD (en Inglés, Knowledge Discovery in Databases, KDD) [8] en correspondencia con trabajos anteriores [9] y [10]. A continuación, se presentan las fases y lo realizado en cada una de ellas.

3.1 Fase de Integración y recopilación

Para la realización de esta fase, se utilizó la planilla de cálculo que la cátedra posee, la cual permitió un registro sistemático de las tesis defendidas y el instrumento de recolección de datos construido denominado “TESISTAS-UM” [9]. En este trabajo, se utilizaron las encuestas de los estudiantes recibidos durante los últimos 10 años ya que, desde esa fecha se cuenta con un registro de los proyectos de investigación ejecutados en la UM., resultando una muestra de 97 registros compuesta por 47 graduados de la carrera Licenciatura en Sistemas y 50 graduados de la carrera Ingeniería en Informática.

3.2 Fase de selección, limpieza y transformación

En esta fase se identifican las características adecuadas que deben participar en la construcción del modelo. El objetivo propuesto en este trabajo fue determinar si existe relación entre los perfiles obtenidos de los tesis de las carreras de informática de la Universidad de Morón con su participación en la investigación, tanto a nivel de proyectos como propuestas de cátedra. Por tal motivo se trabajó con los perfiles derivados de nuestros trabajos anteriores [9][10], donde los atributos son: carrera, edad, grupo familiar, área de trabajo (laboral), área de trabajo de la tesis. Se incorporaron los atributos: motivo de selección de tesis y, si resultó alguna publicación de su trabajo de tesis (ambos recuperados de la encuesta realizada).

Para el atributo clase, se utilizó la pregunta de la encuesta “De su trabajo de tesis ¿surgió alguna publicación?” y de las mismas se comprobó su veracidad con los registros de los proyectos de investigación.

La tabla 1, presenta los atributos a utilizar en la fase de minería de datos con sus valores asociados, luego de la transformación y limpieza de los datos.

Tabla 1. Atributos significativos para la fase de minería de datos.

Atributos	Valores
Área de tesis	Agentes y sistemas inteligentes/ Ingeniería de Software/ Base de Datos y Minería de datos/ Innovación en Sistemas de Software/ Arquitectura, Redes y Sistemas Operativos/ Seguridad Informática/ Tecnología y Educación/ Procesamiento de señales y sistemas en tiempo real.
Carrera	Licenciatura en Sistemas/ Ingeniería en Informática.
Área de trabajo	Análisis funcional y requerimientos/ Bases de datos y minería de datos/ Desarrollo/Infraestructura/ Procesos de negocio/ Seguridad informática/Testing/Varios/ No trabaja.
Edad	Menores de 25 años/ Entre 25 y 30 años/ Mayores a 30 años.
Grupo familiar	Con compromiso/ Sin compromiso.
Motivo de elección del tema	Vinculación con su trabajo/Interés personal/Propuesto por la cátedra/Líneas de investigación/Otros. “Propuesto por la cátedra”: significa que el estudiante puede seleccionar las áreas de trabajo que ofrece la cátedra en la primera reunión de clases. Para la definición de las áreas de trabajo, se consideraron las áreas

	propuestas en el Congreso Argentino de Computación (CACIC), evento científico organizado por la Red de Universidades con carreras en informática (RedUNCI) [11]. “ <i>Líneas de investigación</i> ”: son las líneas de investigación de los proyectos que se encuentran en ejecución cuando comienza el año lectivo; y se les ofrece a los estudiantes su incorporación para desarrollar su trabajo de fin de carrera en el marco del proyecto.
Tiene Publicación	No, Si-con proyecto de investigación, Si-sin proyecto de investigación

En la tabla 2, se presentan los proyectos de informática de los últimos 10 años ejecutados en la UM y sus correspondientes áreas de investigación.

Tabla 2 proyectos de investigación del área de informática de la UM.

Área de investigación	Nombre del proyecto de investigación
Ingeniería de Software	Ingeniería de software en el desarrollo de aplicaciones sensibles al contexto. Validación de las herramientas desarrolladas para la implementación de Sistemas de Información Automatizados. Aproximación a una metodología para la Implementación de Sistemas de Información Basados en Computadoras. Análisis y fortalecimiento de la etapa de Implementación del Proceso Software.
Agentes y Sistemas Inteligentes	Aprendizaje de Robots por Refuerzo con supervisión y Control a Través de Bioseñales. Interfase Cerebro-máquina aplicada a la domótica y a la Robótica. Control de artefactos y robots centrado en emociones. Estudio exploratorio para el desarrollo de un framework en el contexto de la ciberdefensa y la ciberseguridad. Influencia en el estado biométrico emocional de personas. Explotación de datos EEG y parámetros fisiológicos de usuarios interactuando en contextos virtuales.
Tecnología para la Educación	Ambientes virtuales inmersivos de aprendizaje orientados a la capacitación y al desarrollo de simulaciones en situaciones de riesgo. Ambientes virtuales inmersivos- elementos interactivos.
Minería de Datos	Aplicación de tecnologías inteligentes de explotación de información para el análisis de perfiles de tesis de grado de carreras informáticas de la UM.

3.3 Fase de minería de datos

La selección de características o atributos es una técnica necesaria en el pre procesamiento cuando se desea realizar tareas de minería de datos. Los algoritmos de selección de características tienen dos objetivos principales:

Reducir el costo computacional asociado tanto al aprendizaje como al propio modelo de conocimiento generado (eliminando atributos irrelevantes o redundantes).

Aumentar la precisión de dicho modelo (eliminando atributos perjudiciales para el aprendizaje) [12].

En general, las técnicas de selección de atributos pueden clasificarse en dos grandes grupos:

Técnicas de filtrado. Evalúan los atributos de acuerdo con heurísticas basadas en características generales de los datos e independientes del algoritmo de aprendizaje.

Técnicas envolventes. Trabajan juntamente con el algoritmo que va a usarse para el aprendizaje para determinar que atributos son más relevantes.

En esta instancia, se aplicaron distintos evaluadores para determinar con que atributos se trabajarán los nuevos experimentos. En la tabla 3 se presentan los resultados obtenidos con los evaluadores utilizados [12].

De los evaluadores propuestos, todos seleccionan como atributo relevante a: motivo de selección de tesis con un 100% de correspondencia. Se observa que Grupo familiar, es el atributo menos relevante. Por este motivo se decidió eliminar el atributo grupo familiar para las siguientes fases.

Tabla 3. Resultados al aplicar los evaluadores propuestos.

Evaluadores	Modo de evaluación	Método	Evaluación de atributos
cfsSubSet Eval (filtro)	Cross-validation	Best first	Motivo de selección 100%
ChiSquareAttribute Eval (filtro)	Cross-validation	Ranker :	rank atributo 1 +- 0 6 motivo-selección- 2 +- 0 5 área-tesis 3.1 +- 0.3 4 Trabajaba 3.9 +- 0.3 2 Edad 5.1 +- 0.3 1 Carrera 5.9 +- 0.3 3 Grupo-fam.
ReliefAttributeEval (filtro)	Cross validation	Ranker	rank atributo 1 +- 0 6 motivo-selección 2.3 +- 0.64 5 área-tesis 3.1 +- 0.54 2 Edad 3.6 +- 0.66 4 Trabajaba- 5.4 +- 0.49 1 Carrera 5.6+- 0.49 3 Grupo-fam.
WrapperSubsetEval weka.classifiers.bayes. BayesNet 5 padres envoltura	-B Cross-validation	Greedy-Stepwise	Motivo de selección de tesis 100%
WrapperSubsetEval weka.classifiers.bayes. BayesNet 5 padres envoltura	-B Cross validation	Best-first	1 Carrera 20 % 2 edad 10 % 3 grupo-familiar 0 % 4 trabajaba-en 10 % 5 área-tesis 0% 6 motivo-selección 100%

Experimento 1.

La pregunta de investigación que surgió en una primera instancia fue:

¿Existe correlación entre los perfiles obtenidos de los tesis de informática, con el atributo planteado como clase: tiene -publicación?

Se trabajó con el software WEKA [13], del cual se obtuvo un modelo de clasificación con técnicas de clustering. Estas tratan de encontrar grupos dentro de un conjunto de individuos, donde el principal objetivo reside en agrupar con características similares entre sí a los sujetos, y a su vez diferenciadas de los individuos pertenecientes al resto de los grupos. [8] Los clusters fueron observados en el SMS como tareas muy frecuentes en la minería de datos.

Para este trabajo se utilizaron los algoritmos de clustering: K-means o k-medias, basado en particiones y EM o Expectation Maximization basado en probabilidades. K-means necesita como dato de entrada el número de grupos en los que se segmentará la población. A partir de este número k de clusters, el algoritmo coloca primero k puntos aleatorios (centroides). Luego asigna a cualquiera de esos puntos todas las muestras con las distancias más pequeñas. A continuación, el punto se desplaza a la media de las muestras más cercanas. Esto generará una nueva asignación de muestras, ya que algunas muestras están ahora más cerca de otro centroide. Este proceso se repite de forma iterativa y los grupos se van ajustando hasta que la asignación no cambia más moviendo los puntos. El algoritmo EM, empieza prediciendo, al azar, los parámetros de las distribuciones y los usa para calcular las probabilidades de que cada instancia pertenezca a un cluster y usa esas probabilidades para estimar nuevamente los parámetros de las probabilidades [13].

Se inicializó para ambos algoritmos el número k (cantidad de clusters) igual a 3 y se corrieron estos, utilizando la clase para su evaluación. En la tabla 4 se muestran los resultados obtenidos para ambos algoritmos, donde cluster 0 = “No tiene publicación”, cluster 1 = “Si tiene publicación, (sin proyecto de investigación)” y Cluster 2 = “Si tiene publicación (con proyecto de investigación)”.

Tabla 4. Comparación entre los algoritmos K-means y EM.

Algoritmo utilizado	Cantidad de instancias que se obtienen en cada clase			Incorrectas con respecto a la clase
	Cluster 0	Cluster 1	Cluster 2	
	64	13	20	
K-means	43	4	8	43,29%
EM	45	3	12	38,14%

De estos resultados, se interpreta que existe una baja correlación entre los clusters obtenidos y el atributo clase, *tiene-publicación*.

La tabla 5 muestra la distribución de los centroides obtenidos por K-means, donde el número, en la primera fila de cada cluster (59, 23,15), representa la cantidad de instancias que considera para el grupo.

Tabla 5. Centroides obtenidos por K-means.

Atributo	Cluster 0 (59 instancias)	Cluster 1 (23 instancias)	Cluster 2 (15 instancias)
Carrera	Ing-I	Lic-S	Lic-S
Edad	25-30	Mayor-30	Mayor-30
Trabaja-en	Varias	An-fun-req	Varias
Área-de-tesis	Ag.Sis-Int	Ing-Sof	Ag-Sis-Int
Motivo-selección	In-per	In-per	L-investigación

En este experimento se evidencia que, al utilizar algoritmos de agrupamiento, no existe correlación entre los perfiles obtenidos por el cluster y, si el tesista logró o no, una publicación científica.

Experimento 2

¿Cómo reaccionan los algoritmos de clasificación ante el set de datos con los nuevos atributos: “*motivo de selección de tema y si tiene-publicación*”?

Entre los algoritmos de clasificación, se utilizó el J4.8, (algoritmo supervisado), en el cual se tiene una variable dependiente o clase y el objetivo del clasificador es determinar el valor de dicha clase para nuevos casos.

Se trabajó con los algoritmos J4.8, Naïves Bayes, y Bayes Net (con 5 padres y 3 padres). En todos los casos se utilizó cross validation con 10 carpetas. Los resultados al comparar dichos algoritmos con la herramienta WEKA, en modo experimenter, se muestran en la tabla 6, en porcentajes de atributos clasificados correctamente.

Tabla 6 Porcentaje de atributos clasificados correctamente.

Algoritmo	J4.8	Naives Bayes	Bayes Net (5 padres)	Bayes Net (3 padres)
Porcentaje de correctos	82,52%	78,07	76,30	76,30

Se observa que, al aplicar algoritmos probabilísticos, como Naïves Bayes o Bayes Net, en ningún caso mejora el rendimiento en la clasificación obtenida por el algoritmo J4.8

A continuación, se presenta el árbol obtenido por el algoritmo J4.8, donde se clasificaron correctamente el 82,48% instancias de un total de 97 instancias.

motivo-selección-tesis = In-per: No (50.0/9.0)

motivo-selección-tesis = Prop-cátedra

| Edad = 25-30: No (6.0/1.0)

| Edad = menor-25: Sí-SIN-PID (2.0)

| Edad = mayor-30: No (6.0/1.0)

motivo-selección-tesis = L-investigación: Sí-CON-PID (20.0/2.0)

motivo-selección-tesis = Vinc-trabajo: No (10.0/1.0)

motivo-selección-tesis = Otros: No (3.0/1.0)

De este experimento, se interpreta que el atributo “*motivo de selección de tesis*” es el que tiene mayor incidencia en la clase.

Experimento 3.

La pregunta de investigación fue: *¿es posible obtener un modelo predictivo confiable para nuestros datos?*

Uno de los problemas a los que se enfrenta la minería de datos es ocuparse de la incertidumbre. El que se soluciona al trabajar con métodos y técnicas bayesianas, ya que una de sus principales características es el uso explícito de la teoría de probabilidad.

La idea de usar el teorema de Bayes, en cualquier problema de aprendizaje automático, es que se puede estimar la probabilidad a posteriori de cualquier hipótesis consistente con el conjunto de datos. El algoritmo utilizado fue el clasificador Naïve Bayes realizado por Duda & Hart en el año 1973 [14], donde supone que todos los atributos son independientes entre sí, conocido el valor. Se utilizó para tal fin, el software Elvira [14]. Se obtuvo la red que se muestra en la Figura 1, donde la variable clase es: *tiene-publicación* (publicación) y los nodos hijos son las variables independientes: carrera, edad, grupo familiar, área de trabajo y motivo de selección de tesis. La red muestra la distribución conjunta de probabilidades obtenida por el teorema de Bayes.

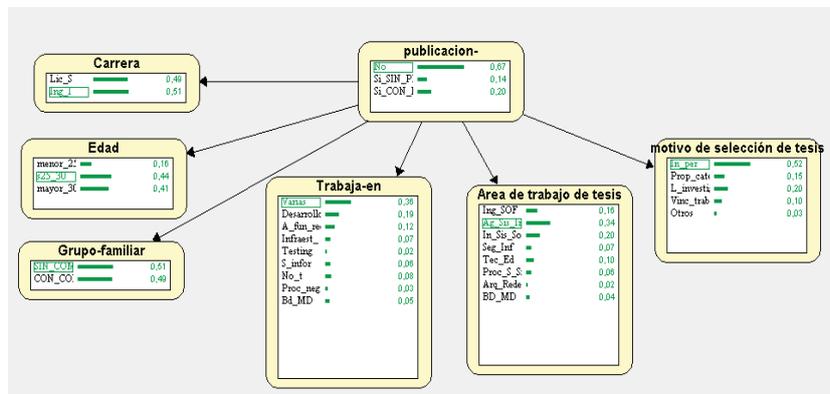


Fig. 1. Red obtenida con el software Elvira aplicando Naïves Bayes. Distribución conjunta de probabilidades.

En la Figura 2, se muestra un nuevo caso en el cual la evidencia es: Carrera = Ingeniería, Edad = mayor a 30, trabaja en = Varios, Área de trabajo de tesis = Agentes y sistemas inteligentes, motivo de elección de tesis = interés personal (todos los nodos en gris) donde el nodo clase publicación tiene un 90 % de probabilidad que el tesista no realice una publicación.

Si a este caso se modifica sólo el motivo de selección de tesis por líneas de investigación, los valores de la clase cambian a “si- publica con proyectos en un 75 %” (ver Figura 3).

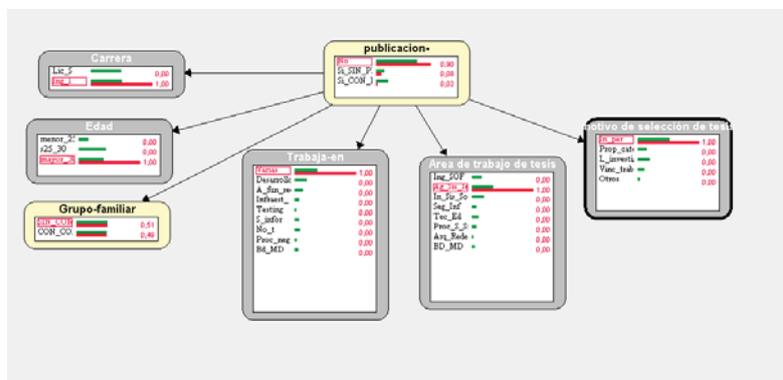


Fig. 2. Evidencia 1: Carrera. Ingeniería, Edad: mayor a 30, trabaja en Varios, Área de trabajo de tesis. Agentes y sistemas inteligentes, motivo de elección de tesis: interés personal.

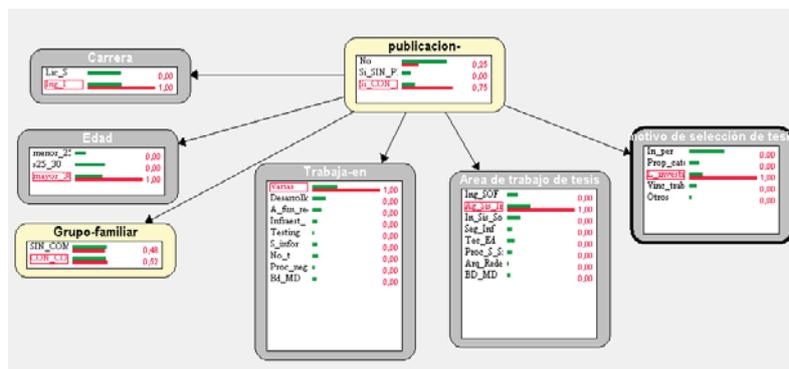


Fig. 3. Evidencia 2: Carrera. Ingeniería, Edad: mayor a 30, trabaja en Varios, Área de trabajo de tesis. Agentes y sistemas inteligentes, motivo de elección de tesis: líneas de investigación.

4 Conclusiones y futuros trabajos

Se han presentado los trabajos relacionados que fueron revisados mediante un método de investigación, estudio de mapeo sistemático (SMS). Este permitió identificar que problemas se resuelven con la minería de datos educacional. Logramos evidenciar que los estudios primarios enfocados en la definición de perfiles de estudiantes, ninguno considera, en sus características, el acercamiento del estudiante a las actividades de investigación.

Los perfiles hallados de los tesisistas, utilizando clusters, no permitieron evidenciar su participación en un proyecto de investigación.

Se descubrió el grado de interés de los tesisistas por determinadas áreas de investigación, mediante el atributo “motivo de selección de tesis”. En el caso de interés personal o propuesta de cátedra, existe una tendencia hacia la escritura de

artículos y participación en eventos científicos. Si el motivo de selección es líneas de investigación, los tesisistas desarrollan su trabajo de fin de carrera en el marco de un proyecto de investigación, esto permite motivar a los alumnos hacia la escritura de artículos y luego la presentación de estos en eventos científicos.

Los resultados de este trabajo permitieron generar una serie de propuestas a los graduados, como la posibilidad de obtener un subsidio para realizar su doctorado y/o la incorporación, como investigador-graduado UM, en un proyecto de investigación.

Como futuros trabajos se identifican: (a) continuar la experimentación con otros algoritmos (b) analizar las áreas de interés de aquellos alumnos que se han interesado por la escritura de artículos y/o participación en eventos científicos con el objetivo de viabilizar la presentación de proyectos de investigación en dichas áreas.

Referencias

1. Romero, C., Ventura, S., Pechenizkiy, M. & Baker, R. S. Handbook of educational data mining. CRC Press. Eds.(2010).
2. Luan, J. Data mining and its applications in higher education. New directions for institutional research, (113), 17-36 (2002).
3. Lin, S. H. Data mining for student retention management. Journal of Computing Sciences in Colleges, 27(4), 92-99 (2012).
4. Chacon, F., Spicer, D., & Valbuena, A. Analytics in support of student retention and success. Research Bulletin, 3, 1-9. (2012).
5. Universidad de Morón. Secretaria de Ciencia y Tecnología (SeCyT-UM). Recuperado el 20 de julio de 2019, página web: <https://www.unimoron.edu.ar/area/cienciaytecnologia>
6. Marisa Daniela Panizzi. "Establecimiento del estado del arte sobre la Minería de Datos Educativa en el Nivel Superior: Un Estudio de Mapeo Sistemático". Revista de Investigaciones Científicas del Universidad de Morón. Vol. 3 nro. 1, 2019.
7. Kitchenham, B. and Charters, S. Guidelines for performing Systematic Literature Reviews in Software Engineering. Versión 2.3 EBSE-2007-01 (2007).
8. Hernández Oraldo José, Ramírez Quintana María José, Ferri Ramírez César. Introducción a la Minería de Datos. Ed. Pearson Education S.A. Madrid. (2004).
9. Iris Sattolo, Gastón Alvarez, Nicolás Armilla, Oscar Bravo, Matias García, Javier Lafont, Gabriel Mariuz, Lucila Mira, Marisa Panizzi. Hacia la caracterización de perfiles de tesisistas de Carreras de Informática de la Universidad de Morón. XIII Congreso Nacional de Tecnología en Educación y Educación en Tecnología (TE&ET 2018). Universidad Nacional de Misiones. Posadas, Misiones. Argentina. 14 y 15 de junio 2018.
10. Iris Sattolo, Gastón Alvarez, Matias Garcia, Javier Lafont, Lucila Mira, Gabriel Mariuz, Nicolás Armilla, Marisa Panizzi. Descubrimiento de las áreas de investigación seleccionadas por los tesisistas de las carreras de informática de la UM mediante árboles de decisión. XXIV Congreso Argentino de Ciencias de la computación (CACIC 2018) Tandil Universidad Nacional del Centro de la Pcia. De Bs.As. ISBN 978-950-658-472-6
11. RedUNCI (Red de Universidades con carreras de Informática). Recuperado el 18 de julio de 2019, página web: <http://reduinci.info.unlp.edu.ar/>
12. Elvira soft. Recuperado:16-07-2019 web: <http://www.ia.uned.es/investig/proyectos/elvira/>
13. Weka. University of Waikato. Machine Learning Group. Página web:www.cs.waikato.ac.nz/ml/Weka/downloading.html. Recuperado: 20 -06- 2019.
14. Castillo, Gutiérrez y Hadi. Sistemas Expertos y Modelos de Redes Probabilísticas. Recuperado:10-07-2019 web: <https://personales.unican.es/gutierjm/papers/BookCGH.pdf>

Procedimiento para instruir en la programación de dispositivos inteligentes

Mg. José Luis Filippi¹, Lic. Guillermo Lafuente², Mg. Carlos Ballesteros³, Mg. Rodolfo Bertone⁴

Facultad de Ingeniería, UNLPam, General Pico, La Pampa, Rep.Arg.

{filippi¹, lafuente², charlyballe³}@ing.unlpam.edu.ar

pbertone@ada.info.unlp.edu.ar⁴

Resumen

La programación de dispositivos de última generación, es una tarea compleja que demanda en los estudiantes la incorporación de conocimientos derivados desde distintas disciplinas como la matemática, física, entre otras ciencias; al mismo tiempo que se deben incorporar las nociones correspondientes al propio campo de estudio.

Desarrollar un algoritmo que resuelve un problema específico requiere un esfuerzo intelectual significativo que los estudiantes deben afrontar.

Con el objetivo de mejorar y optimizar las técnicas y/o métodos de estudio aplicados en la programación, se propone un nuevo procedimiento, que posibilita a los estudiantes iniciarse en el desarrollo de programas orientados a dispositivos inteligentes fundamentado en la Teoría del pensamiento creativo mencionado por Howard Gardner. [1]

Esta experiencia se instituye en la Facultad de Ingeniería de la UNLPam en las asignaturas del primer año de las carreras Analista Programador e Ingeniería en Sistemas.

Palabras claves.

Programación. Dispositivos móviles. Pensamiento creativo. Procedimiento pedagógico.

1. Introducción.

Se entiende por procedimiento al conjunto de acciones u operaciones que se realizan de la misma forma, para obtener siempre el mismo resultado bajo las mismas circunstancias. [2]

En la teoría de la computación, un procedimiento efectivo es una secuencia de pasos repetible y determinista; es decir, una secuencia de pasos en que siempre se irán obteniendo los mismos conjuntos de valores de salida, para los mismos conjuntos de valores de entrada. [3]

Un procedimiento desde el punto de vista pedagógico es un conjunto de etapas que el estudiante ejecuta como instrumento para aprender.

Estudios realizados por Ken State University por John Dunlovsky, los dos procedimientos más utilizados por los estudiantes para rendir un examen son:

Preparar el examen con mucho tiempo de antelación y en segundo lugar el estudiante simula una situación de examen y practica con exámenes realizados por educandos de años anteriores. Además, se destacan otras acciones por ejemplo auto interrogatorio, auto explicación, resúmenes, técnicas mnemotécnicas, releer, intercalar estudios diferentes, mapas mentales, fichas de estudio, test, incorporar recursos visuales, etc. [4]

Nuevas tendencias educativas requieren de un nuevo perfil de estudiante, que utilice estrategias de aprendizajes acorde al momento actual. El Instituto de la Economía Digital (ICEMD) de ESIC Business & Marketing School destaca tendencias que se aplican a las instituciones educativas; por ejemplo, una visión de los estudiantes como sujetos activos que demanda una educación personalizada, que le ayude a encontrar su pasión, promueva el auto conocimiento, se realice de forma permanente a lo largo de la vida, que incorpore la programación de dispositivos inteligentes, donde se traten contenidos conectados con la realidad, estimule el aprender haciendo, fomente la colaboración y democratice los medios de comunicación. [5]

En este sentido el docente en su rol de gestionar el proceso formativo, debe adaptar su práctica educativa a las tendencias que la sociedad demanda. Por consecuencia se propone un procedimiento que posibilita a los estudiantes iniciarse en el desarrollo de programas orientados a dispositivos inteligentes fundamentado en la Teoría del pensamiento creativo mencionado por Howard Gardner.

Se realiza la experiencia en la asignatura Introducción a la Informática perteneciente al primer año de las carreras Analista Programador e Ingeniería en Sistemas perteneciente a la Facultad de Ingeniería de la UNLPam.

2. Metodología

Los primeros pasos que el estudiante incursiona en la programación de dispositivos inteligentes, notebook, tabletas digitales, celulares de última generación, etc., conlleva a aplicar diferentes técnicas para desarrollar el pensamiento creativo que va a permitir encontrar el algoritmo que responde al problema en cuestión.

Entiéndase el pensamiento creativo a partir de la concepción de Howard Gardner. Supone que el individuo creativo es alguien que regularmente es capaz de resolver un problema. Para saber si una persona es creativa Gardner propone observar a una persona durante un tiempo en su lugar de trabajo y verificar como resuelve las diferentes situaciones que se le presentan. Entiende el pensamiento creativo como la adquisición de conocimiento de un modo muy particular que se caracteriza por su flexibilidad, fluidez, originalidad y funciona como estrategia para resolver situaciones problemáticas en un contexto de aprendizaje. [1]

El aprendizaje basado en problemas permite a los estudiantes desarrollar el razonamiento y pensamiento crítico de manera más eficiente que el modelo educativo tradicional; donde se despliegan habilidades profesionales como la toma de decisiones, auto aprendizaje, trabajo en equipo, participación en discusiones relevantes. [6]

En el año 2006 la doctora Jeannette Marie Wing, profesora de Ciencias de la Computación en la Universidad Carnegie Mellon en Estados Unidos, publica el artículo “Computational Thinking”, pensamiento computacional, donde acuña la concepción de un proceso a través del cual los problemas y sus soluciones se representan de tal forma que puedan ser resueltos de manera efectiva por un agente de procesamiento de información. Usando métodos, estrategias y modelos que automatizan la solución mediante un planteamiento algorítmico. Considera que el pensamiento computacional es una habilidad y una actitud de aplicación universal para todas las personas, y que será fundamental para mediados del siglo XXI. [7]

Pensamiento creativo y pensamiento computacional dan origen a un campo multidisciplinar, la creatividad computacional cuyo objetivo es modelar, simular o replicar la creatividad utilizando una computadora. [8]

Gardner menciona bajo la concepción del pensamiento creativo los elementos que lo conforman.

- a. Producir una forma de pensar nueva.
- b. Darle solución a una problemática.
- c. Desarrollar la idea original en su totalidad.
- d. Prever las consecuencias que traerá la nueva idea consigo.

A partir de ésta noción que plantea Gardner, se propone un procedimiento que va a permitir desarrollar algoritmos para solucionar diferentes problemáticas de programación.

El procedimiento está conformado por diferentes etapas:

- a. Etapa 1.

Objetivo: Comprender el problema.

Acción:

Esta tarea conlleva en el estudiante un proceso que envuelve varias acciones: leer la consigna, interpretar el vocabulario científico, reconocer patrones, analizar el contenido en su conjunto, entre otras opciones.

La especialista Marta Stone Whiskey indica que comprender es poder realizar una presentación flexible del tema, justificarlo, extrapolarlo y explicarlo de manera que vaya más allá de la repetición rutinaria. Indica además que comprender vincula 4 dimensiones: conocer conceptos importantes, métodos de razonamiento disciplinados, propósitos y límites de las distintas esferas de comprensión, formas de expresar ante los demás lo comprendido. [9]

Pensando la comprensión a partir de lo manifestado, se propone a los estudiantes en esta etapa, redefinir el problema que plantea la actividad propuesta por el docente. Se aborda la actividad en grupos reducidos de no más de 3 personas, con el propósito de generar un ámbito de trabajo donde fluya el intercambio de información, la colaboración entre pares y la generación de nuevas ideas, en donde los estudiantes asumen responsabilidades y son participantes activos de su propio aprendizaje.

[10]

- b. Etapa 2.

Objetivo: Búsqueda de la solución algorítmica al problema planteado.

Acción:

Definimos algoritmo desde el punto de vista de la informática como una secuencia de instrucciones que se le brinda a la computadora para desarrollar una determinada tarea. [11]

En ésta etapa los estudiantes realizan diferentes algoritmos que surgen de las propuestas que emanan de cada uno de los integrantes del grupo de trabajo. El lenguaje que se utiliza en ésta instancia es pseudocódigo o lenguaje de descripción algorítmica. Un lenguaje que utiliza convenciones estructurales de programación, pero diseñado para la lectura humana en lugar de la lectura de una computadora. [12]

c. Etapa 3.

Objetivo: Seleccionar el algoritmo que mejor responde al problema planteado.

Acción: Analizar los diferentes algoritmos examinando las ventajas y desventajas de cada propuesta.

La característica fundamental que todo algoritmo debe cumplir se basa en su desempeño, es decir, que su comportamiento sea el adecuado. Cumplido ésta premisa debemos considerar la cantidad de recursos que cada algoritmo consume, en relación al tiempo que demanda su ejecución y al espacio o cantidad de memoria que utiliza.

El tiempo se calcula a partir de dos variables, el tamaño de la entrada de datos y de la complejidad de las operaciones que realiza. Para ello se aplican diferentes fórmulas preestablecidas que permiten efectuar éstos cálculos.

A modo de ejemplo se puede considerar que ante dos posibles algoritmos llamados A y B, y ambos responden de forma óptima a la consigna solicitada, debemos seleccionar aquel que consume menos recursos. [13]

En caso que, del total de algoritmos propuestos como solución al problema dado, no cumplan con los requisitos mínimos indispensables para un buen algoritmo; el procedimiento cicla a la Etapa 1 reiniciando el procedimiento; caso contrario pasa a la Etapa 4.

d. Etapa 4.

Objetivo: Codificar la solución algorítmica en el lenguaje de programación solicitado por la cátedra.

Acción: Desarrollar la aplicación utilizando las técnicas de programación adecuadas. Efectuar las pruebas necesarias para optimizar el funcionamiento.

En esta instancia se utiliza un entorno de desarrollo de programación simple e intuitivo, que utiliza un pseudolenguaje en español, complementado con un editor de diagramas de flujo; de esta forma facilita al educando, centrar su atención en los conceptos fundamentales de la algoritmia computacional, minimizando las dificultades propias de los lenguajes de programación. La plataforma que se utiliza brinda además un entorno de trabajo con ayudas en línea y recursos didácticos para el aprendizaje.

El entorno de desarrollo que facilita el proceso de codificación, es PSeInt. Un entorno de programación potente, liviano, ideal para dar los primeros pasos en la programación de computadoras y/o dispositivos inteligentes. [14]

El procedimiento que se propone aborda la problemática que afronta el estudiante que da sus primeros pasos en la programación de computadoras y/o dispositivos de última generación.

A modo de ejemplo y a partir de la experiencia implementada en el aula, se detalla el uso del procedimiento por parte de estudiantes del primer año de las carreras Analista Programador e Ingeniería en Sistemas.

Caso de Uso:

El docente propone a los estudiantes desarrollar un código que permita resolver un problema determinado, que para un mejor entendimiento le denominamos $P(x)$. Se debe aplicar para su resolución el procedimiento aquí planteado. Antes de comenzar se conforman grupos de no más de tres o cuatro estudiantes.

Etapas 1.

Tiene por objetivo interpretar el problema planteado por el docente. Para ello los estudiantes de cada grupo reformulan la consigna, generando nuevas imágenes sobre el problema a resolver, descubriendo posibles recorridos que en etapas sucesivas derivan en la solución buscada. Dependiendo de la complejidad de la problemática se obtienen 3 o más reformulaciones - $R(x)$ - por cada grupo de trabajo. Este proceso favorece el entendimiento de $P(x)$.

Etapas 2.

Tiene por objetivo la búsqueda de la solución algorítmica de $P(x)$.

Los estudiantes proponen alternativas algorítmicas - $A1, A2, \dots, An$ - que surgen a partir de las diferentes reformulaciones - $R(x)$ - que se desarrollaron en la Etapa 1.

Etapa 3.

Se procede a analizar las ventajas y desventajas que poseen los diferentes algoritmos planteados en la etapa 2. Los integrantes del grupo analizan cada algoritmo, teniendo en cuenta las siguientes pautas: claridad del código, uso apropiado de las estructuras de datos, cantidad de sentencias, agregación de los casos especiales, incorporación de funciones, inserción adecuada de variables, entre otras posibilidades.

Se evalúa el desempeño general del algoritmo y en caso de aceptación, se calculan los recursos que cada algoritmo necesita durante su ejecución.

A partir del análisis efectuado, los integrantes deciden que algoritmo cumple con la consigna del problema $P(x)$. Ante la situación que ningún algoritmo cumpla los requerimientos de un buen algoritmo, el procedimiento me envía a la Etapa 1, en el caso contrario continua en la Etapa 4.

Etapa 4.

A partir del algoritmo seleccionado en la etapa 3, los estudiantes desarrollan la aplicación utilizando las técnicas de programación adecuadas.

Para un mejor entendimiento, se grafica el procedimiento en todas sus etapas. Grafico 1.

El ciclo que se observa con mayor claridad en el Grafico 1, y que va desde la Etapa 3 a la Etapa 1, tiene por objetivo reiniciar todo el procedimiento, cuando el algoritmo propuesto por los estudiantes no cumple los fundamentos mínimos de un buen algoritmo.

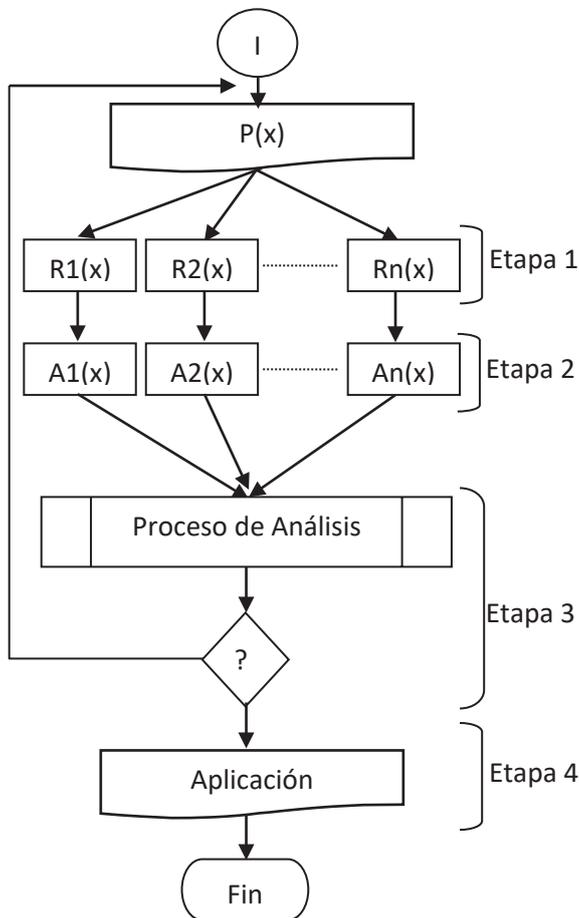


Gráfico 1.

3. Conclusión

¿Porque es importante contar con un procedimiento que planee el trayecto que los estudiantes deben recorrer para el desarrollo de un buen algoritmo informático?

La importancia reside en disponer de diferentes estrategias que ayuden al estudiante a resolver problemas informáticos.

Un procedimiento con pautas claras hace posible: ahorrar tiempo durante el desarrollo de las aplicaciones, ordenar las tareas dando prioridades según los requerimientos

solicitados, crea buenos hábitos laborales, elimina la improvisación, orienta la actividad cotidiana y optimiza recursos disponibles, entre otras ventajas.

Contar con un procedimiento claro y preciso incrementa el porcentaje de éxito que posee cada estudiante que decide introducirse en la programación de computadoras o dispositivos de última generación.

La primera dificultad que debe afrontar al tratar de resolver un problema, es lograr interpretar de forma correcta la consigna que el docente provee. El procedimiento propuesto conformado por diferentes etapas pone énfasis en su uso en forma grupal con el objetivo de lograr resolver las tareas encomendadas por el docente a partir del trabajo grupal y colaborativo.

Sin embargo, un procedimiento, método o técnica como el propuesto no asegura el éxito del proceso educativo. Es imprescindible que el procedimiento se desarrolle en el marco de una planificación en el cual se indique como potenciar el proceso formativo tradicional con diferentes propuestas pedagógicas.

Aquí reside el desafío para el docente, formular nuevos métodos, procedimientos y/o técnicas que permitan a los estudiantes incorporar nuevos saberes a partir del uso de variadas alternativas educativas.

4. Bibliografía

1. Gardner, Howard (1995). *Mentes creativas. Una anatomía de la creatividad vista a través de las vidas de S. Freud, A.Einstein, P. Picasso, I. Stravinsky, T.S. Elliot, M. Graham, M. Gandhi.* Paidós, Barcelona.
https://es.wikipedia.org/wiki/Pensamiento_creativo
2. Stair, Ralph M., et al. (2003). *Principles of Information Systems, Sixth Edition* (en inglés). Thomson Learning, Inc. p. 132. ISBN 0-619-06489-7.
<https://es.wikipedia.org/wiki/Procedimiento>
3. Stallings, William (2005). *Sistemas operativos: aspectos internos y principios de diseño* (5ª edición). Pearson Prentice Hall. p. 109.
https://es.wikipedia.org/wiki/Procedimiento_efectivo
4. John Dunlosky (2013). *Which Study Strategies Make the Grade?*
<https://www.psychologicalscience.org/news/releases/which-study-strategies-make-the-grade.html>

5. Instituto de la Economía Digital (ICEMD) de ESIC Business & Marketing School. (2015). 10 Tendencias de la educación del siglo XXI. <http://noticias.universia.es/en-portada/noticia/2015/02/06/1119646/10-tendencias-educacion-siglo-xxi.html>
6. Hector David Martínez Villacrez. (2013). Aprendizaje Basado en Problemas como estrategia para potenciar el aprendizaje académico. <http://repositorio.uta.edu.ec/jspui/handle/123456789/7459>
7. Nelson Monzón López. (2012). La Informática como Herramienta Docente: Otro Medio para Aprender a Pensar https://www.researchgate.net/profile/Nelson_Monzon/publication/266291733_La_Informatica_como_Herramienta_Docente_Otro_Medio_para_Aprender_a_Pensar/links/542bfb400cf27e39fa91c676.pdf
8. Creatividad computacional. https://es.wikipedia.org/wiki/Creatividad_computacional
9. Stone Wiske, Marta (2006). Enseñar para la comprensión. Paidòs. Buenos Aires.
10. Juan Carlos Sandí Delgado Mainor, Alberto Cruz Alvarado (2016). Propuesta Metodológica de Enseñanza y Aprendizaje para Innovar la Educación Superior. Revista InterSedes. VOL. 17 NÚM. 36. <https://www.scielo.sa.cr/pdf/is/v17n36/2215-2458-is-17-36-00153.pdf>
11. Brassard, Gilles; Bratley, Paul (1997). *Fundamentos de Algoritmia*. Madrid: PRENTICE HALL. <https://definicion.de/algoritmo/>
12. Pseudocódigo. <https://es.wikipedia.org/wiki/Pseudoc%C3%B3digo>
13. Amalia Duch. (2007). Análisis de Recursos. <https://www.cs.upc.edu/~duch/home/duch/analisis.pdf>
14. PSeInt. <http://pseint.sourceforge.net/>

Propuesta de automatización de auto-percepción de capacidades visuales de estudiantes universitarios

Maria Laura Gronda^{1,2}, Sonia I. Mariño¹

¹Facultad de Ciencias Exactas y Naturales y Agrimensura, Universidad Nacional del Nordeste, Corrientes, Argentina

^{1,2}Facultad de Derecho, Universidad Nacional del Nordeste, Corrientes, Argentina
{lauragronda@hotmail.com, simarinio@yahoo.com}

Resumen. La accesibilidad web es un tema actual dado la diversidad de soluciones disponibles en Internet. En este estudio se presenta un procedimiento diseñado para identificar las capacidades visuales de estudiantes universitarios, dado que estas inciden en el acceso a los contenidos. Particularmente, se diseñó una encuesta en SIU-KOLLA y se integró al sistema SIU-GUARANÍ. Con fines de validación se aplicó el instrumento de recolección de datos en el primer semestre del año 2019 a los estudiantes. El procesamiento de los datos relevados muestra que existe una relación inversamente proporcional entre los problemas en la capacidad visual auto percibida y la percepción de calidad del sitio WEB evaluado. Además, esta información permitió proponer una taxonomía en torno a las capacidades visuales compuesta de las siguientes categorías: a- Capacidad visual normal, b- Problemas de visión moderados, c- Problemas de visión severos, d- ceguera al color (daltonismo) y e- ceguera.

Palabras clave: accesibilidad web, educación superior, auto-percepción, discapacidad visual

1 Introducción

El presente trabajo aporta a la línea de I+D expuesta en [1, 2, 3]. En particular, se propone la automatización de la automatización de la auto-percepción de capacidades visuales de estudiantes universitarios a través del diseño e implementación de una encuesta. La encuesta se diseñó utilizando SIU-KOLLA y se implementó a través del SIU-GUARANÍ. Para verificar la propuesta se aplicó a los estudiantes en el primer semestre de 2019 de una unidad académica de la UNNE. El artículo se compone de cuatro secciones, además de la introducción, la segunda, describe el método desarrollado, la tercera describe la propuesta y su verificación. Finalmente, se establecen las conclusiones y líneas de trabajo futuras.

1.1 Inclusión en la educación superior

La masividad de la educación superior viene acompañada por la democratización del acceso a las tecnologías y en especial a las relacionadas con la información y la

comunicación. La transición “enseñar con TIC” a “integrar las TIC al trabajo pedagógico” [4] es un proceso lento y dispar en las instituciones educativas argentinas de todos los niveles [5]. Resumiendo, a pesar de las declaraciones de derechos y garantías, los nuevos marcos teóricos, los programas de formación docente masiva gratuita, la disponibilidad de contenidos en formatos digitales y dispositivos asequibles, persisten barreras a la inclusión educativa.

En 2011 el Consejo Interuniversitario Nacional (CIN) aprueba el “Programa Integral de Accesibilidad para las Universidades Públicas”. En 2017 la Comisión de Discapacidad y Derechos Humanos se transforma en la Red. En la mayoría de las universidades cuenta con un espacio institucional desde donde diseñar e implementar proyectos y acciones que favorezcan la inclusión educativa de estudiantes con discapacidad. Buena parte de estos esfuerzos se dedican a la capacitación y concientización de la comunidad educativa, convenios de colaboración con otras universidades y organizaciones de la sociedad civil para adaptar recursos educativos.

1.2 Acceso a los contenidos. La accesibilidad web

En el marco de la accesibilidad para las universidades un aspecto fundamental a incluir es lo relativo al acceso a los contenidos debido a la transformación de las TIC en actividades administrativas y de aprendizaje. Habitualmente la accesibilidad web (AW) se asocia al diseño de sitios destinados a usuarios con algún tipo de discapacidad, visual, motora, cognitiva o limitada por la edad, limitaciones derivadas por utilización de dispositivos anticuados u otras dificultades asociadas al entorno [6].

Un enfoque particular de la AW en los sitios web de las instituciones públicas de educación superior, es considerarla como una herramienta para dar cumplimiento a un derecho, el derecho humano a la igualdad de acceso a la información [7]. Es decir, el derecho a la educación superior. La aplicación práctica de este derecho implica asegurar el acceso integral a los contenidos del sitio web institucional, desde el inicio de la oferta académica, transitando todas las instancias de cursado y evaluación, hasta la finalización de la carrera de grado.

Lo expuesto precedentemente, implica reducir significativamente las barreras para una persona con algún tipo de discapacidad. Se debe, por lo tanto, proporcionar a los sitios, las alternativas para superar las barreras que impiden a los estudiantes acceder a los contenidos

Entre algunos antecedentes legales se mencionan: Normas UNE 139803:2004, Ley 26/2011, Real Decreto Legislativo 1/2013. En la Argentina la Ley 26.653 entró en vigencia en 2010, sin embargo, aún en estos días es un tema de relevancia y al cual se puede aportar desde espacios de educación superior. Algunos estudios que abordan la indagación en torno a la AW en educación superior son los descriptos en [8,9,10 y 11].

Las WCAG 2.0 se organizan en 4 principios fundamentales para la accesibilidad del contenido:

- **PERCEPTIBLE:** La información y los componentes de la interfaz de usuario deben ser presentados a los usuarios de modo que ellos puedan percibirlos.
- **OPERABLE:** Los componentes de la interfaz de usuario y la navegación deben ser operables.
- **COMPENSIBLE:** La información y el manejo de la interfaz de usuario deben ser comprensibles.
- **ROBUSTO:** El contenido debe ser suficientemente robusto como para ser interpretado de forma fiable por una amplia variedad de aplicaciones de usuario, incluyendo los productos de apoyo.

A su vez, cada uno de estos grandes principios tiene asociadas pautas. En total son 12 pautas: los dos primeros principios presentan 4 pautas, el tercero 3 y el último 1. Estas pautas no son verificables en sí, sino que proporcionan las metas básicas para hacer el contenido accesible, y comprender los criterios de conformidad e implementarlos.

Cada una de estas pautas tiene asociados ciertos criterios de conformidad (61 en total) que se deberían cumplir y que sí son verificables. Los criterios de conformidad se ordenan según su nivel de cumplimiento asociado (A, AA y AAA).

El máximo organismo en la jerarquía de Internet responsable de promover la accesibilidad es el W3C, en especial su grupo de trabajo WAI (Web Accessibility Initiative) [referencia]. La Web Accessibility Initiative (WAI) es una Iniciativa para la Accesibilidad Web y una rama del World Wide Web Consortium. Se ocupa de la accesibilidad de la Web, siendo su idea general crear una serie de reglas claras. Publica las Guías de Accesibilidad al Contenido Web [12].

2 Metodología

El método aplicado para la elaboración del presente trabajo constó de las siguientes fases:

- Fase 1. Indagación de estándares, normas y métodos referentes a la AW.
- Fase 2. Diseño de la propuesta para automatizar la auto-percepción de capacidades visuales de los estudiantes universitarios.
- Fase 3. Verificación de la propuesta.
- Fase 4. Exposición de los resultados.

3 Resultados

En espacios de educación superior, es relevante asegurar la accesibilidad de un sitio Web con la finalidad de garantizar el derecho al acceso igualitario a la información y así contribuir al cumplimiento de la Ley 26.653. Por ello, se describe una propuesta

de automatización de la auto-percepción en torno a sus capacidades visuales de los estudiantes utilizando dos sistemas/módulos SIU. La propuesta se valida en una facultad de la UNNE.

3.1 Definición de la propuesta

En la propuesta se consideró:

3.1.1 Definición de una encuesta para relevar la auto percepción de los estudiantes

Esta fase constó de las siguientes actividades

- Entrevista a un experto en discapacidades visuales. Los datos relevados y trabajos previos [17], permitieron diseñar un cuestionario a fin de conocer el uso de herramientas de software que utilizan los docentes de la Instituto para ciegos Valentín Haüy de la ciudad de Corrientes.
- Diseño de una encuesta en línea orientada a relevar/identificar las capacidades visuales en estudiantes universitarios de una facultad. Se contempla la siguiente clasificación:
 - Capacidad visual normal
 - Problemas de visión moderados
 - Problemas de visión severa o ceguera

Es importante destacar que esta clasificación se basó en la auto percepción subjetiva de la capacidad visual, es decir que dicha auto-percepción podría no coincidir con la opinión de un profesional oftalmólogo. Para conocer la tipificación de un profesional se entrevistó al director médico del Instituto para ciegos Valentín Haüy de la ciudad de Corrientes, quien instruyó acerca de la evaluación clínica que se realiza en cada paciente.

La Fig.1 presenta la encuesta propuesta identificada como Evaluación de la accesibilidad Visual del sitio Web institucional. En su diseño se atendió al anonimato con el objetivo de obtener las respuestas lo más sinceras posible. Se consultó al encuestado la capacidad visual auto percibida, y luego utilizando la escala de Likert se indaga sobre el diseño de cada página Web de la muestra. Finalmente, se incluyen cuestiones referentes a los dispositivos y software más utilizados durante la navegación en el sitio de estudio. Cabe aclarar que la muestra de las páginas web se basó en una estrategia orientada a lograr la concordancia con las directrices de la WAI. Es decir, una es la página principal por ser la primera impresión que se tiene del sitio visitado, una página con tabla simple, una con tablas más complejas, una con contenidos recomendados para material de estudio, una con video, como así también dos páginas seleccionadas como control de la muestra.

3.1.2 Definición de un consentimiento informado

Se elaboró un consentimiento informado (CI) para su inclusión junto a la encuesta administrada por el sistema SIU-KOLLA. En el CI, se especificó el siguiente texto: "Esta encuesta es voluntaria y agradecemos tu colaboración. El objetivo es la mejora de la calidad de la accesibilidad visual del sitio Web de una Facultad. Tu identidad y los datos recabados serán mantenidos por el responsable del estudio bajo condiciones

de confidencialidad mediante códigos máscaras, la información se obtiene bajo protocolos de investigación. Tanto en informes como comunicaciones científicas, los resultados se publicarán siempre de manera agregada, a fin de impedir la identificación de sujetos particulares.

3.1.3 Verificación preliminar del instrumento

- Se realizaron pruebas de la encuesta con un estudiante voluntario a fin de asegurar la correcta comprensión de las preguntas.
- Se aplicó la encuesta personalmente a tres estudiantes uno por cada capacidad visual estudiada, se grabaron las respuestas preservando el anonimato de los entrevistados y subidas a la nube para la consulta posterior.

3.1.4. Implementación para verificar la encuesta en línea.

Validada la comprensión de la encuesta diseñada, se:

- Solicitó al área TIC de la Facultad la integración al SIU-GUARANÍ.
- Difundió el lanzamiento de la encuesta en el sitio Web de la Facultad y en la cuenta de Facebook desde el área de prensa institucional.

3.2 Verificación de la propuesta de automatización

De una población de más de 12.000 estudiantes activos. Se encuestaron a 374 alumnos en el periodo de un mes, de forma presencial y virtual.

Para la modalidad presencial se seleccionaron al azar tres estudiantes representativos de la tipificación más característica de las capacidades visuales estudiadas. Tanto para la modalidad virtual como presencial, se preguntó respecto al diseño de las páginas del sitio Web de Facultad mediante el modelo de encuesta de la Fig.1

<ul style="list-style-type: none"> • Ante la presencia de una respuesta con los símbolos <input type="radio"/> seleccione una y solo una opción. • Ante la presencia de una respuesta con los símbolos <input type="checkbox"/> podrá seleccionar más de una opción. 			
Evaluación de la accesibilidad VISUAL del sitio Web de la Facultad de Derecho de la UNNE			
Esta encuesta es voluntaria y agradecemos tu colaboración. Tu identidad y los datos recabados serán mantenidos por el responsable del estudio bajo condiciones de CONFIDENCIALIDAD mediante Códigos Máscaras. CONSETIMIENTO INFORMADO	<input type="radio"/> Si	<input type="radio"/> No	
¿Cómo describiría Usted su visión?	<input type="radio"/> Mi visión es normal	<input type="radio"/> Tengo problemas de visión moderados	<input type="radio"/> Tengo problemas de visión severos
	<input type="radio"/> Tengo ceguera al color (daltonismo)	<input type="radio"/> Soy No vidente	<input type="radio"/> otro
¿Su problema de visión es permanente?	<input type="radio"/> Si	<input type="radio"/> No	
El diseño de la PÁGINA PRINCIPAL del sitio oficial de la Facultad de Derecho UNNE, le resulta adecuado a su capacidad visual.	<input type="radio"/> Totalmente de acuerdo	<input type="radio"/> De acuerdo	<input type="radio"/> Soy neutral
	<input type="radio"/> En desacuerdo	<input type="radio"/> Totalmente en desacuerdo	
¿Los contenidos textuales de MATERIAL DE ESTUDIO provistos en el sitio web del sitio de la Facultad de Derecho, les resultan adecuados a su capacidad visual.?	<input type="radio"/> Totalmente de acuerdo	<input type="radio"/> De acuerdo	<input type="radio"/> Soy neutral
	<input type="radio"/> En desacuerdo	<input type="radio"/> Totalmente en desacuerdo	
¿La presentación de las NOVEDADES: noticias variadas, jornadas, congresos, charlas, etc. del sitio Web del sitio de la Facultad de Derecho le resultan adecuados a su capacidad visual?	<input type="radio"/> Totalmente de acuerdo	<input type="radio"/> De acuerdo	<input type="radio"/> Soy neutral
	<input type="radio"/> En desacuerdo	<input type="radio"/> Totalmente en desacuerdo	
Las páginas Web con el CALENDARIO DE EXAMENES LIBRES (Tabla de Contenidos simples) le resultan adecuados a su capacidad visual.	<input type="radio"/> Totalmente de acuerdo	<input type="radio"/> De acuerdo	<input type="radio"/> Soy neutral
	<input type="radio"/> En desacuerdo	<input type="radio"/> Totalmente en desacuerdo	
¿Las páginas Web con OFERTA DE CURSOS (Tabla compleja) le resultan adecuados a su capacidad visual?	<input type="radio"/> Totalmente de acuerdo	<input type="radio"/> De acuerdo	<input type="radio"/> Soy neutral
	<input type="radio"/> En desacuerdo	<input type="radio"/> Totalmente en desacuerdo	
¿Los VIDEOS del sitio de la Facultad de Derecho le resultan adecuados a su capacidad visual?	<input type="radio"/> Totalmente de acuerdo	<input type="radio"/> De acuerdo	<input type="radio"/> Soy neutral
	<input type="radio"/> En desacuerdo	<input type="radio"/> Totalmente en desacuerdo	
Los ENLACES a otras Páginas Web del sitio de la Facultad de Derecho les resultan adecuados a su capacidad visual	<input type="radio"/> Totalmente de acuerdo	<input type="radio"/> De acuerdo	<input type="radio"/> Soy neutral
	<input type="radio"/> En desacuerdo	<input type="radio"/> Totalmente en desacuerdo	
¿Las IMAGENES del sitio de la Facultad de Derecho le resultan adecuados a su capacidad visual?	<input type="radio"/> Totalmente de acuerdo	<input type="radio"/> De acuerdo	<input type="radio"/> Soy neutral
	<input type="radio"/> En desacuerdo	<input type="radio"/> Totalmente en desacuerdo	
El software (programa) que usted utiliza habitualmente para acceder al sitio de web de la Facultad es	<input type="radio"/> Navegador habitual Google Chrome, Mozilla FireFox, Internet Explorer	<input type="radio"/> Facebook	<input type="radio"/> Twiter
	<input type="radio"/> Instagram	<input type="radio"/> otro	
El Hardware (dispositivo) que usted utiliza habitualmente para acceder al sitio de web de la Facultad es	<input type="radio"/> PC de escritorio	<input type="radio"/> Notebook/Netbook	<input type="radio"/> Tablet
	<input type="radio"/> SmartPhone (celular o teléfono móvil)	<input type="radio"/> otro	

Fig. 1 – Modelo de encuesta: Evaluación de la accesibilidad visual del sitio web institucional.

La muestra constó de 370 personas quienes respondieron en el periodo comprendido desde 1 al 31 de mayo de 2019. Los datos relevados se procesaron y analizaron, generándose la siguiente información.

El mayor porcentaje de los estudiantes censados tienen una visión normal (51%), seguidos muy de cerca por estudiantes con problemas de visión moderados (45%) un 4% tiene problemas visuales severos y el 1,07 % otros problemas no especificados. El análisis de las respuestas de los 134 participantes en un total de 8 páginas Web, dan cuenta de la necesidad de proponer estrategias superadoras en torno a la accesibilidad de los contenidos dispuestos en la web institucional.



Fig. 2 - Distribución de las respuestas por auto-percepción

Se examinó la relación de la autopercepción en las tres categorías encontradas entre las respuestas a la encuesta, estas son: a- Capacidad visual normal, b- Problemas de visión moderados, c- Problemas de visión severos, a fin de medir el impacto de la autopercepción en la página Web más compleja de la muestra, identificada como *oferta de cursos*.

La Tabla 1 detalla las respuestas a la pregunta 7. ¿Las páginas Web con *oferta de cursos* (Tabla compleja) le resultan adecuados a su capacidad visual? de la Fig. 1, de los estudiantes encuestados, filtradas por capacidad visual normal.

Tabla 1 – Auto-percepción “Mi visión es normal” y cómo aprecian Tablas complejas en la página web

Respuesta	Cantidad	Porcentaje
Totalmente de acuerdo	69	37%
De acuerdo	91	49%
Soy Neutral	17	9%
En desacuerdo	9	5%
Totalmente en desacuerdo	1	1%
Total	187	100%

Tabla 2 Auto-percepción “Tengo problemas de visión moderados” y la apreciación de Tablas complejas

Respuesta	Cantidad	Porcentaje
Totalmente de acuerdo	38	22%
De acuerdo	87	51%
Soy neutral	28	17%
En desacuerdo	15	9%
Totalmente en desacuerdo	1	1%
Total	169	100%

La Tabla 2 muestra un detalle en referencia a las Respuestas a la pregunta 7. ¿Las páginas Web con *oferta de cursos* (Tabla compleja) le resultan adecuados a su capacidad visual?. de la Fig. 1, de los estudiantes encuestados, filtradas por problemas moderados en la capacidad visual.

Tabla 3 - Auto-percepción “Tengo problemas de visión severos” y la apreciación de Tablas complejas

Respuesta	Cantidad	Porcentaje
Totalmente de acuerdo	4	29%
De acuerdo	4	29%
Soy neutral	2	14%
En desacuerdo	4	29%
Totalmente en desacuerdo	0	0%
Total	14	100%

La Tabla 3 muestra un detalle en referencia a las Respuestas a la pregunta 7.¿ Las páginas Web con *oferta de cursos* (Tabla compleja) le resultan adecuados a su

capacidad visual?. de la Fig. 1, de los estudiantes encuestados, filtradas por problemas severos en la capacidad visual.

Se analizaron las tablas donde se cuantifican la apreciación del usuario de las páginas Web con tablas complejas. Se destaca el marcado descenso del acuerdo con el diseño de la misma, es decir la apreciación positiva tiende a disminuir, lo que indicaría la necesidad de aplicar medidas correctivas en las paginas evaluadas.

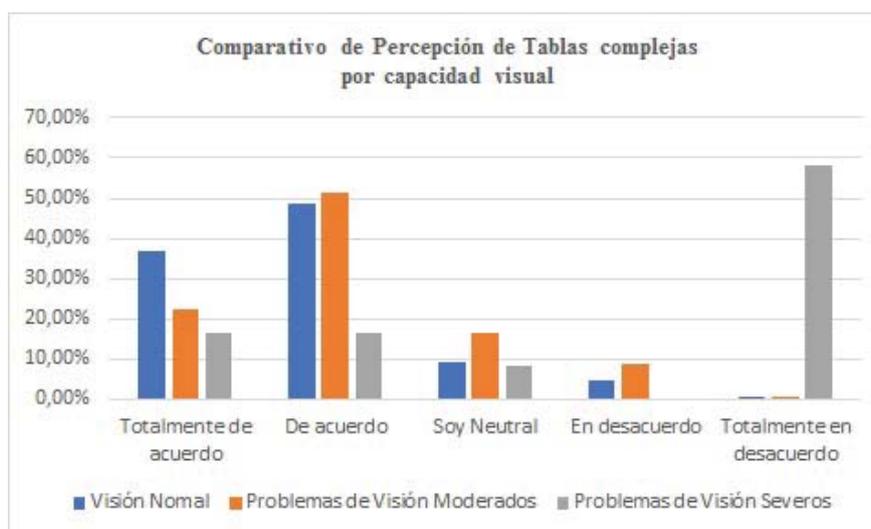


Fig. 3- Comparativa percepción de tablas complejas por capacidad visual

En la **Fig. 3** se visualiza la marcada tendencia a percibir las páginas web con tablas complejas como mal diseñadas y esto se pondera si se tiene en cuenta la información de vital importancia organizada en ella, dado que despliega la oferta de cursos/ asignaturas para la cursada del cuatrimestre. Por lo expuesto, ante esta situación aquel estudiante con problemas severos de visión o ceguera se encuentra ante una barrera significativamente difícil de superar.

Los resultados obtenidos a través de la indagación propuesta, induce a preguntar cuántos estudiantes no han contestado la encuesta aprovechando que no es obligatoria y por considerarla una tarea complicada para su capacidad visual. El modo de calcular ese valor es materia de estudio en un trabajo futuro.

4 Conclusiones

En el artículo se expuso una propuesta de automatización de la auto-percepción de capacidades visuales de estudiantes universitarios. La encuesta desarrollada en el

sistema SIU-KOLLA, se integró al sistema SIU-GUARANÍ y así se logró la vinculación directa con el estudiante, usuario habitual del sistema mencionado.

Se seleccionó como objeto de estudio la auto-percepción de la capacidad visual para proponer estudios equivalentes a la registrada en la preinscripción vía SIU-GUARANÍ, con la salvedad que en la preinscripción es claramente identificable quien responde y se incorporan una variedad de datos personales. Lo expuesto indica que puede conducir a completar todos los campos obligatorios con datos tipo comodín u observaciones tales como “no tengo discapacidad pero completo el campo porque si no, no tengo comprobante de preinscripción” u otro de similar tenor. En referencia a la verificación de la propuesta, la cantidad de respuestas recibidas es alentadora y se podría asociar al anonimato de la encuesta en conjunto con la condición de voluntaria.

Se infiere la necesidad de evaluar las páginas web utilizando herramientas y métodos adecuados y mejorar el acceso a la información aplicando aquellos recursos de programación, herramientas y técnicas disponibles en el proceso de desarrollo de recursos educativos.

En esta exposición se destaca la posibilidad de diseñar y ejecutar trabajos de vinculación entre unidades académicas, especialmente en torno a temas de educación superior y acceso con calidad a los contenidos en entornos virtuales.

Para trabajos futuros se propone analizar la auto-percepción y su vinculación con otras páginas de interés académico tales como los contenidos del material de estudio disponibles en el sitio Web de la Facultad, como así también la evaluación de los hipervínculos sugeridos por los docentes como material de estudio.

Agradecimientos

Se agradece a las autoridades de la Facultad de Derecho y Ciencias Sociales y Políticas, Universidad Nacional del Nordeste, por su colaboración e interés en la verificación de la propuesta expuesta orientada a relevar datos en el ámbito de la facultad. También a los expertos en la temática abordada, el director médico, profesionales y a los docentes del Instituto para ciegos “Valentín Haüy”, Corrientes, Argentina, y al equipo de profesionales técnicos de SIU-GUARANÍ y SIU-KOLLA.

Referencias

1. Mariño, S. I., Godoy, M. V. y Alfonzo, P. L.: Avances en torno a la formación en accesibilidad web. XIX Workshop de Investigadores en Ciencias de la Computación. 2017.
2. Mariño, S. I.; Alfonzo, P. L.: Evaluación de la accesibilidad web. Una mirada para asegurar la formación en la temática. *Campus Virtuales*, 62, pp. 21-30, 2017.
3. Mariño S. I., Alfonzo P.L. *Inclusión Social: TIC y Accesibilidad Web*. Primer Congreso de Educación y Tecnologías del Mercosur, Corrientes, Argentina, 2017.
4. Sánchez, J.: Integración curricular de TIC. Concepto e ideas. [En línea]. *Revista Enfoques educativos*. 6º Congreso Iberoamericano, 4º Simposio Internacional de Informática Educativa, 7º Taller Internacional de Software Educativo: IE-2002: Vigo, 20, 21, 22 de noviembre de 2002. Consultado el 17 de septiembre de 2009. Disponible en: <http://lsm.dei.uc.pt/ribie/docfiles/txt2003729191130paper-325.pdf>

5. Lugo, M. T., & Brito, A.: Las Políticas TIC en la educación de América Latina. Una oportunidad para saldar deudas pendientes. *Archivos de Ciencias de la Educación*, 2015. Recuperado <http://www.archivosdeciencias.fahce.unlp.edu.ar/article/view/Archivos09a03>
6. Hassan-Montero, Y: Diseño Web Centrado en el Usuario: Usabilidad y Arquitectura de la Información, JO - Hipertext.net; Núm.: 2, 2004
7. Cárdenas Tapia, M. et al.: Las TIC en las instituciones de educación superior en México. Caso Escuela Superior de Comercio y Administración, Unidad Tepepan. 2009
8. Alahmadi, T., & Drew, S.: An evaluation of the accessibility of top-ranking university websites: Accessibility rates from 2005 to 2015. There and back: Charting flexible pathways in open, mobile and distance education pp. 224–233. Hamilton. 2016.
9. Comeaux, D., & Schmetzke, A.: Web accessibility trends in university libraries and library schools. A. Schmetzke, Ed. *Library Hi Tech*, 254, 457–477. Emerald Group Publishing Limited. 2017, REVCOM; año 3, no. 5p. 191-205 2007.
10. Acosta-Vargas, P. et al: Challenges to Assess Accessibility in Higher Education Websites: A Comparative Study of Latin America Universities, 2018 https://rua.ua.es/dspace/bitstream/10045/77893/1/2018_Acosta-Vargas_et_al_IEEEAccess.pdf
11. Bayan Abu Shwar: Evaluating Web Accessibility of Educational Websites, 2015 <http://dx.doi.org/10.3991/ijet.v10i4.4518>
12. Information Technology—W3C Web Content Accessibility Guidelines WCAG 2.0, document ISO/IEC 40500:2012, 2012.
13. Ocampo Chaparro J. M Auto-percepción de salud: importancia de su uso en adultos mayores, *Colomb Med* 2010; 413, p: 275-289
14. Gallegos-Carrillo K et al: Autopercepción del estado de salud: una aproximación a los ancianos en México, *Rev Saude Pública* 2006 ; 405:792-801
15. SIU-GUARANI, <https://www.siu.edu.ar/siu-guarani/>
16. SIU-KOLLA, <https://www.siu.edu.ar/siu-kolla/>
17. Villafañe Hormazábal G. P.: Estudiantes con discapacidad en una universidad chilena: desafíos de la inclusión, Universidad de Concepción Chile 2016, *Revista Complutense de Educación* 353 ISSN: 1130-2496 Vol. 27 Núm. 1 353-372 http://dx.doi.org/10.5209/rev_RCED.2016.v27.n1.46509

Propuesta de automatización para proyectos de minería de datos educativa

Santiago Bianco, Sebastian Martins, Hernán Amatriain, Hernán Merlino

Laboratorio de Investigación y Desarrollo en Ingeniería de Explotación de Información
Grupo de Investigación en Sistemas de Información. Universidad Nacional de Lanús
Remedios de Escalada, Buenos Aires, Argentina.
santiago.bianco.sb@gmail.com, smartins089@gmail.com, hamatriain@gmail.com,
hmerlino@gmail.com

Resumen. En los últimos años es cada vez más frecuente que establecimientos educativos cuenten con diversos sistemas de información. La aplicación de técnicas de minería de datos permite hacer uso de la información generada por estos sistemas para mejorar la calidad en la enseñanza de las instituciones. Tanto es así que existe una disciplina dedicada exclusivamente a esto denominada Minería de Datos Educativa. No obstante, esta práctica requiere de una experticia en el campo de la explotación de información que no muchos miembros de la comunidad educativa poseen, lo que dificulta su aplicación. Para subsanar este problema, en este trabajo se propone un marco de trabajo automatizado, demostrando que se pueden conseguir resultados iguales o mejores a los que se obtendría aplicando procesos de minería de datos tradicionales y sin necesidad de conocer en detalle el funcionamiento de los algoritmos que se aplican.

Palabras clave: Tecnología Informática Aplicada en Educación, Automated Machine Learning, Educational Data Mining.

1. Introducción

La Minería de Datos Educativa (EDM, por sus siglas en inglés) se define como “una disciplina emergente, relacionada con el desarrollo de métodos para explorar los tipos únicos de datos que provienen del entorno educativo y el uso de esos métodos para entender mejor a los estudiantes y al entorno en el que aprenden” [1]. Las principales categorías en las cuales las líneas de investigación se han centrado son [2]: análisis y visualización de los datos, detección de comportamientos no deseados, modelado del estudiante, proveer recomendaciones a los docentes, administrativos y/o responsables académicos, predicción del rendimiento de los estudiantes, entre otros. Se señala, además, que las investigaciones en los últimos años se han centrado principalmente en el estudio del comportamiento de los estudiantes en sistemas educativos. Existe, además, una incipiente tendencia hacia el análisis de la información para ayudar a dichos sistemas, y la potencial mejora de algunos aspectos de la calidad de la educación y de los procesos de aprendizaje y el análisis del comportamiento de los estudiantes en cursos y carreras universitarios [3].

Cada vez existen más algoritmos y herramientas para implementar las distintas técnicas de minería de datos, aunque generalmente estas se centran en la eficiencia y precisión más que en la facilidad de uso. En este contexto, Romero y Ventura señalan como líneas de investigación de interés, el diseño de procesos o métodos que faciliten a educadores y/o usuarios no expertos en el área de minería de datos la implementación de las técnicas de extracción de conocimiento [2].

El objetivo de los algoritmos de automatización de aprendizaje automático (del inglés Automated Machine Learning, AML o AutoML) es generar un modelo predictivo evitando realizar manualmente algunas tareas iterativas que requieren un conocimiento específico del área de la ciencia de datos, como la selección del modelo apropiado de acuerdo al problema, la optimización de sus hiperparámetros y la selección de atributos relevantes [4].

En base a estos dos últimos aspectos, en el presente trabajo se explora la aplicación de AutoML en proyectos de EDM, de manera que se pueda automatizar el proceso de selección de algoritmos y su hiperparametrización, así como también gran parte del pre-procesamiento de datos. De esta manera se busca que usuarios no expertos en el área de la minería de datos y ciencia de datos sean capaces de entrenar un modelo en base a sus datos y puedan extraer conclusiones que sirvan de soporte para la toma de decisiones.

2. Estado de la cuestión

En las últimas décadas se han desarrollado numerosas investigaciones aplicando minería de datos para la resolución de problemas en el dominio de la educación: reglas de asociación [5], agrupamiento [6], clasificación [7], entre otras. Estas pueden utilizarse, por ejemplo, para describir el comportamiento de poblaciones estudiantiles, como apoyo a la toma de decisiones, identificación de causales de deserción y abandono y gestión de aulas virtuales [8]. Incluso también han empezado a utilizarse técnicas de minería de procesos y minería de datos no convencionales, como minería de grafos, para resolver problemas de las instituciones educativas [9] [10].

El inconveniente es que en todas las investigaciones antes mencionadas se cuenta con la inclusión en el equipo de un experto en el área de minería de datos o alguna disciplina afín, algo que encarece el proceso y dificulta su aplicación a las instituciones educativas que se interesen en este tipo de soluciones. En un proceso de EDM tradicional como el que se muestra en la figura 1, es necesario realizar tareas que involucran un conocimiento del dominio de la educación, así como también otras que son propias del área de minería de datos y requieren un cierto grado de experticia para ser ejecutadas correctamente. Este subconjunto de tareas incluye el pre-procesamiento de datos, selección de algoritmos y refinación iterativa del modelo obtenido, como puede observarse en detalle en la figura 2.

Si bien existen herramientas que facilitan la manipulación de datos, análisis algorítmico, visualización de datos y ejecución de procesos de explotación de información [11], estas siguen requiriendo tener un alto grado de experticia en el área de la ciencia de datos.

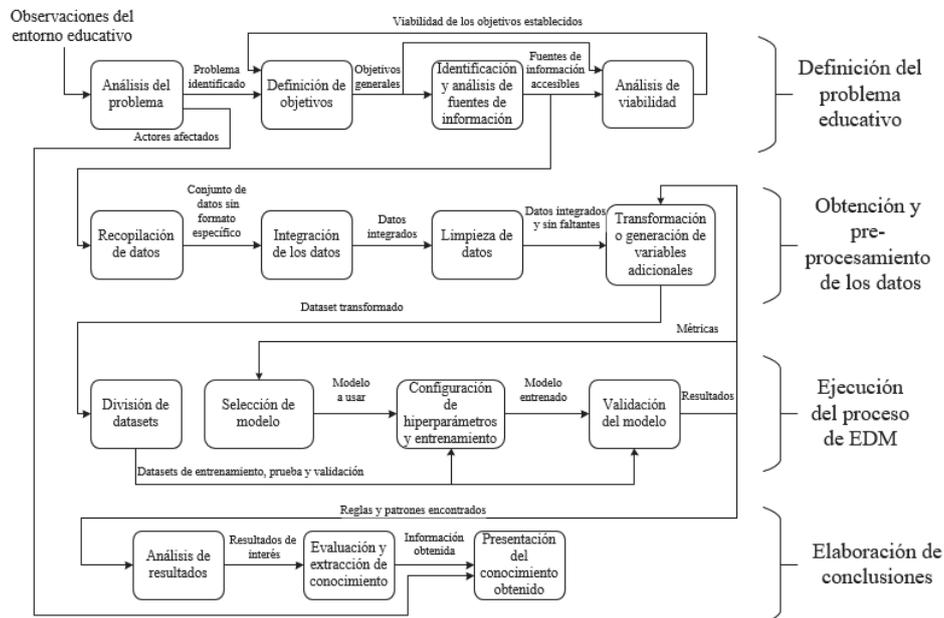


Fig. 1. Actividades de un proceso de EDM tradicional.

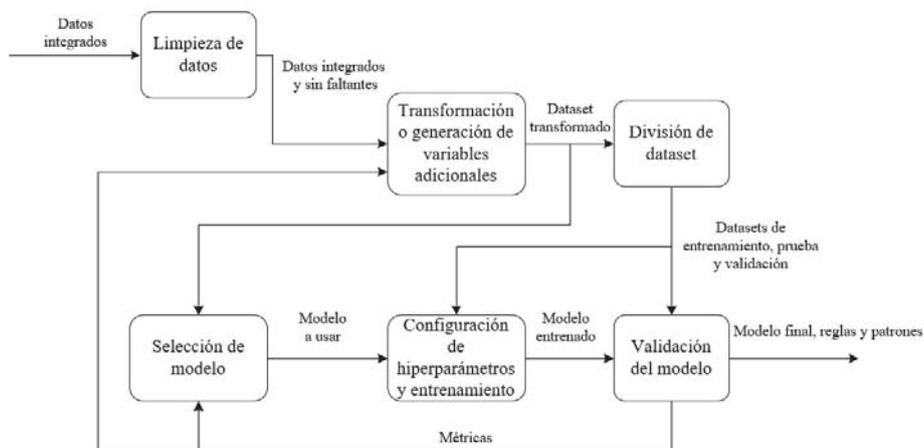


Fig. 2. Detalle de actividades propias de un proceso de minería de datos.

En otros dominios, como respuesta a esta situación se comenzó a trabajar con herramientas de Automated Machine Learning para la búsqueda de patrones en grandes masas de información [12] [13]. AutoML es un campo emergente que estudia los métodos por los cuales se pueden optimizar y automatizar los procesos de entrenamiento de los algoritmos de aprendizaje automatizado. Esta idea puede ser representada como se muestra en la figura 3, un proceso que recibe un conjunto de datos integrados y como salida provee la lista de transformaciones que se deben aplicar a los

datos, el modelo más óptimo que puede aplicarse y sus hiperparámetros, todo agrupado en un pipeline que puede ser ejecutado con una línea de código o mediante consola. Lo único que hay que hacer es proveerle un nuevo conjunto de datos y realizará la transformación y predicción. Se pueden utilizar tanto para clasificación como para regresión y existen varias herramientas desarrolladas que utilizan diversos enfoques para lograrlo, como redes bayesianas [14] [15] o algoritmos genéticos [16], lo que evita probar a fuerza bruta las distintas combinaciones de transformaciones y configuraciones de los modelos. Esto último también evita el refinamiento iterativo que debe realizarse en un proceso de EDM tradicional, en el cuál se valida el modelo y, en caso de no lograr una buena métrica, se lo debe modificar, elegir otro distinto o cambiar alguna transformación de los datos.

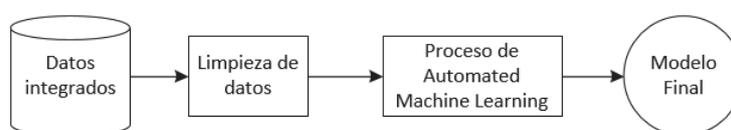


Fig. 3. Ejemplo AutoML pensado como un proceso.

Si bien el uso de AutoML simplifica la obtención de modelos predictivos de calidad, también genera que estos modelos sean más complejos y difíciles de interpretar. Para problemas como la clasificación de imágenes o texto, generalmente no es necesario saber las características del modelo implementado sino que solamente interesa la salida del mismo. Es decir, importa más que el algoritmo haga una clasificación correcta antes que saber el por qué decidió realizar esa clasificación, por lo que la generación de modelos complejos no es un inconveniente. No ocurre lo mismo en dominios en los cuales lo que se busca es, más allá de una correcta clasificación o predicción, encontrar una explicación del porqué de esas situaciones. El área de la educación se incluiría dentro de este segundo grupo. En los procesos de EDM es necesario explicar los modelos obtenidos para conseguir un análisis más profundo del problema educativo. Esto puede verse, por ejemplo, a través del estudio de la deserción universitaria. No solo se busca predecir tempranamente si un estudiante es propenso a abandonar una carrera o no, sino que también se pretende determinar las causas por las que los estudiantes abandonan sus estudios. De esta manera se pueden determinar en dónde asignar recursos para prevenir estas situaciones en un futuro y mejorar la calidad de la institución educativa. Por lo tanto, para poder implementar técnicas Automated Machine Learning en la educación es necesario, además, agregar una etapa de interpretación del modelo obtenido, para extraer reglas, identificar cómo influyen los distintos atributos en el resultado y definir la interrelación entre los mismos.

3. Descripción del problema

Según lo expuesto previamente se identifica la necesidad de aplicar un proceso de minería de datos en entornos educativos que no requiera conocimientos específicos en el área de explotación de información y afines. Esto podría subsanarse con la aplicación de Automated Machine Learning para la etapa de extracción de conocimiento, pero

estos procesos generalmente proveen, no sólo modelos generalmente difíciles de interpretar, sino también distintas transformaciones a los datos que pueden derivar en la creación de nuevos atributos, los cuáles deben poder rastrearse y relacionarse con los atributos originales para generar conclusiones valiosas.

La interpretación de modelos complejos de caja negra es un área de creciente interés [17], por lo que se han generado algunas técnicas que, combinadas adecuadamente con AutoML, serían capaces de generar un proceso de EDM aplicable por cualquier usuario no experto. En base a esto surgen las siguientes preguntas de investigación:

- ¿Es posible aplicar Automated Machine Learning en entornos educativos y obtener modelos predictivos con una efectividad similar a los de un proceso tradicional de EDM?
- De poder obtener esos modelos, ¿es posible agregar una etapa posterior de interpretabilidad que posibilite la comprensión de los patrones generados para que sirvan como apoyo a la toma de decisiones?
- De resultar ambas respuestas positivas, ¿Se puede generalizar el proceso descrito en un marco de trabajo que permita automatizar los proyectos de EDM?

4. Solución propuesta

Para resolver el problema antes descrito se propone utilizar AutoML junto con herramientas de interpretación de la forma en la que se muestra en la figura 4.

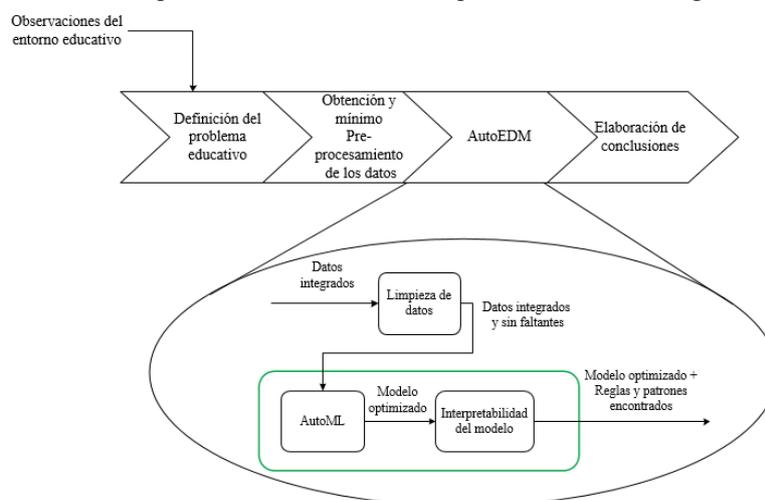


Fig. 4. Proceso de AutoEDM propuesto.

Las reglas y patrones encontrados pueden ser utilizados para la elaboración de informes y conclusiones con respecto a los datos que ayuden al proceso de toma de decisiones en la institución o aula. El modelo optimizado que se obtiene puede ser usado posteriormente para predecir el atributo clase en nuevos casos.

Este framework de trabajo posibilitaría la simplificación de cualquier proceso de EDM, evitando los problemas inherentes a los procesos de minería de datos tradicionales,

como selección del algoritmo apropiado, optimización de hiperparámetros y transformación de los datos. De esta manera, se puede concentrar el trabajo en las áreas de análisis del dominio y elaboración de conclusiones, aspectos que pueden ser realizados por un usuario con conocimientos en el área educativa.

5. Prueba de Concepto

Para probar qué tan factible es la utilización de AutoML en la educación, se contrasta un proyecto aplicando esta tecnología sobre los mismos datos utilizados en un proyecto de EDM tradicional, desarrollado por este grupo de investigación [18]. La idea es comparar los resultados obtenidos para demostrar que no sólo se simplifica el proceso sino que el modelo final presenta mejores métricas para el conjunto de datos de prueba. Se analizará puntualmente el caso de la deserción universitaria, describiendo las causas por las que los alumnos pierden la regularidad en la carrera. La base de datos cuenta con los atributos descritos en la tabla 1. El atributo clase a analizar será “Es Regular” y se utilizarán todo el resto de los atributos para su descripción, tal como se hizo en el estudio original.

Tabla 1. Descripción de los atributos de la base de datos utilizada.

Variable	Tipo	Valores	Distribución
Edad al primer año de cursada	Discreto	18 a 64	$\mu=25,31$ $\sigma=4,64$
Discapacidad	Booleano	Sí	97,78% (1409)
		No	2,22% (32)
Diferencia entre el egreso del secundario e ingreso a la carrera	Discreto	0 a 37	$\mu=3,54$ $\sigma=4,16$
Tipo de colegio secundario	Nominal	Técnico	12,77% (184)
		Bachiller	55,59% (801)
		Comercial	31,37% (452)
		Sin datos	0,28% (4)
Trabaja	Booleano	Sí	35,39% (510)
		No	64,61% (931)
Categoría últimos estudios del padre	Ordinal	0	27,34% (394)
		1	38,03% (548)
		2	27,62% (398)
		3	6,18% (89)
		4	0,83% (12)
Categoría últimos estudios de la madre	Ordinal	0	24,64% (355)
		1	29,49% (425)
		2	30,19% (435)
		3	14,84% (185)
		4	2,85% (41)
Es Regular	Booleano	Sí	16,38% (236)
		No	83,62% (1205)

Para la implementación de la etapa de AutoML se propone el uso de una librería denominada TPOT [16], la cual está implementada en el lenguaje de programación Python. Dicha librería fue elegida por su simplicidad de uso y porque utiliza algoritmos

genéticos para optimizar los distintos modelos que prueba, los cuales presentan mejores resultados con respecto a las otras estrategias [19].

Como se mencionó anteriormente, existe una creciente demanda para poder interpretar modelos de Machine Learning de caja negra, por los cuales se han desarrollado varias alternativas [20]. Por la naturaleza de los casos de EDM, y en base a trabajos ya realizados por este grupo de investigación [21] [22], las alternativas más viables que se pueden aplicar son SHAP [23], por su aplicabilidad en cualquier tipo de modelo, y el método de interpretación propuesto por Satoshi Hara [24] ya que con él se logran reglas interpretables simples tanto para clasificación como para regresión y no requiere configuración extra. Para la prueba se utilizaron solamente los estimadores de TPOT cuyos modelos finales incluyeran árboles de decisión, simples o ensamblados. Esto puede realizarse a través del diccionario de configuración. Para más detalles, el código para la ejecución de este experimento puede encontrarse en un repositorio abierto [25].

6. Resultados obtenidos

Como primera medida para comparar los resultados se utilizó la métrica de exactitud. Sin embargo, como se trata de una muestra desbalanceada (16% regulares vs. 84% no regulares), se aplicaron varias métricas para medir la calidad del modelo generado. El resumen comparativo puede observarse en la figura 5.

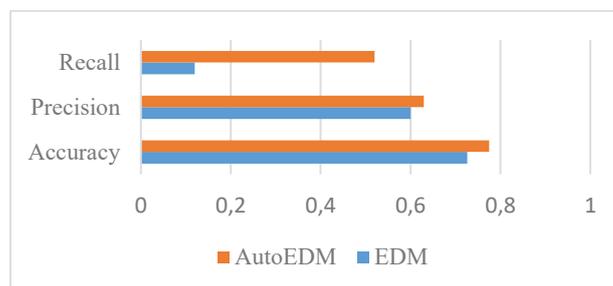


Fig. 5. Métricas para la clase positiva

Puede verse que para el mismo conjunto de datos y para el mismo problema de minería de datos, el modelo de AutoEDM presenta mejores resultados que utilizando el enfoque tradicional. Sin embargo, a esto hay que agregarle algo de interpretabilidad, como se mencionó anteriormente, para poder utilizar la información que brinda el modelo, además de predicciones. Esto se consiguió de distintas formas. Primero, utilizando una implementación del algoritmo defragTrees [24], obteniéndose reglas simples que explican el patrón identificado por el modelo obtenido. En la figura 6 se ejemplifican las reglas generadas para la clase con menor cantidad de ejemplos.

En dicha figura puede observarse que las variables influyentes para que un alumno mantenga la regularidad son la edad al primer año de cursada, Xt6 y Xt2. Estas dos últimas tienen esos nombres ya que son variables derivadas de los atributos originales.

```

[Regla 5]
y = 1 when
  Edad primer año < 0.333236
  Xt6 < 0.000355

[Regla 6]
y = 1 when
  0.333330 <= Edad primer año < 0.334549
  Xt2 < 0.000301

```

Fig. 6. Extracto de reglas obtenidas con defragTrees.

Para poder identificar cuáles son, se utiliza la matriz de correlación identificando la variable original con mayor relación (figura 7). Observando los colores más altos y más bajos de la escala de la derecha se identifica que se corresponden con las variables “Categoría último estudio madre” y “Trabaja”.

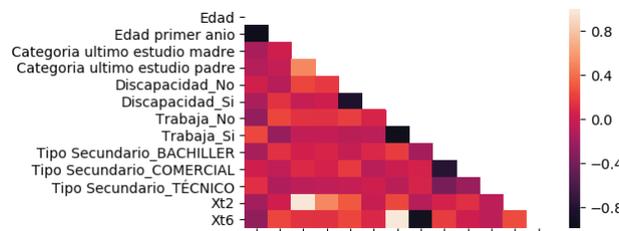


Fig. 7. Matriz de correlación

Es importante señalar que luego de la ejecución del módulo de AML, además de generarse los dos atributos antes mencionados (Xt2 y Xt6), se estandarizaron los valores de los atributos para optimizar el comportamiento de los modelos, perdiendo interpretabilidad. La solución propuesta permite convertir los valores transformados a su valor original obteniéndose el resultado que se muestra en la figura 8. Aquí se ve como la edad, los estudios de la madre y el trabajo influyen en el atributo clase.

```

[Regla 5]
Regular = SI when
  Edad primer año < 27
  Categoría último estudio madre < 1

[Regla 6]
REGULAR = SI when
  25 <= Edad primer año < 27
  Trabaja = NO

```

Fig. 8. Reglas reinterpretadas del modelo

Finalmente, el algoritmo SHAP permite identificar los atributos con más incidencia en el modelo. En el ejemplo propuesto, se obtuvo como resultado que las características más influyentes para determinar la regularidad de un estudiante son, en orden: a) la edad al primer año de la cursada, b) si es estudiante trabaja y c) los estudios de la madre, coincidiendo con lo mostrado en las reglas antes mencionadas.

De los resultados obtenidos en la prueba de concepto no solo se verifica la funcionalidad de la propuesta, permitiendo ejecutar de manera más simple un proceso

de EDM, sino que además se consiguen mejores resultados. También se consigue mostrar que, a pesar de generar modelos complejos, puede agregársele técnicas de interpretación de modelos de caja negra para permitir una posterior elaboración de conclusiones que brinden mayor conocimiento sobre las variables relevantes para el proceso de toma de decisiones.

7. Conclusiones y Trabajo Futuro

En el presente trabajo se identificó la necesidad de contar con mecanismos de abstracción para los procesos de EDM que permitan a los usuarios no expertos en el área de minería de datos llevar a cabo un proyecto de este estilo. Como solución se presentó autoEDM, un marco de trabajo que involucra Automated Machine Learning en combinación con técnicas de interpretabilidad para modelos de caja negra.

Para validar este framework, se ejecutó con un conjunto de datos utilizado en un proyecto de EDM tradicional como prueba de concepto, comparando los resultados obtenidos en ambos proyectos. A raíz de esta prueba se verificó la correcta integración del proceso propuesto combinando herramientas de AML y de interpretación permitiendo generar patrones de conocimiento que sirvan como apoyo a la toma de decisiones en entornos educativos.

En este sentido, la presente investigación presenta evidencias iniciales sobre la posibilidad de diseñar pipelines genéricas para cualquier proyecto de minería de datos educativa que puedan ser utilizadas por usuarios no expertos.

Como futuras líneas de investigación se propone:

- Desarrollar nuevos casos de validación con otros proyectos de EDM.
- Ampliar la aplicación en distintos tipos de problemas (por ejemplo aquellos que requieren modelos de regresión).
- Evaluar otros algoritmos de interpretabilidad no analizados en esta investigación, como LIME [26].
- Desarrollar un ambiente integrado mediante una interfaz que permita la ejecución íntegra del proceso de autoEDM.

8. Referencias

- [1] IEDMS (2019). International Educational Data Mining Society. www.educationaldatamining.org. Página vigente al 29/07/2019.
- [2] Romero, C., & Ventura, S. (2010). Educational data mining: a review of the state of the art. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, 40(6), 601-618.
- [3] Baker, R. S., & Yacef, K. (2009). The state of educational data mining in 2009: A review and future visions. *JEDM| Journal of Educational Data Mining*, 1(1), 3-17.
- [4] Guyon, I., Bennett, K., Cawley, G., Escalante, H. J., Escalera, S., Tin Kam Ho, Viegas, E. (2015). Design of the 2015 ChaLearn AutoML challenge. 2015 International Joint Conference on Neural Networks (IJCNN), 1-8.
- [5] Bose, R., & Sugumaran, V. (1999). Application of intelligent agent technology for managerial data analysis and mining. *ACM SIGMIS Database: the DATABASE for Advances in Information Systems*, 30(1), 77-94.

- [6] Chrysostomou, K., Chen, S. Y., & Liu, X. (2009). Investigation of users' preferences in interactive multimedia learning systems: a data mining approach. *Interactive Learning Environments*, 17(2), 151-163.
- [7] Su, J. M., Tseng, S. S., Lin, H. Y., & Chen, C. H. (2011). A personalized learning content adaptation mechanism to meet diverse user needs in mobile learning environments. *User modeling and user-adapted interaction*, 21(1-2), 5-49.
- [8] Prince Sattam Bin Abdulaziz University, Osman Hegazi, M., & Abugroon, M. A. (2016). The State of the Art on Educational Data Mining in Higher Education. *International Journal of Computer Trends and Technology*, 31(1), 46-56.
- [9] Bogarín, A., Cerezo, R., & Romero, C. (2018). A survey on educational process mining: Survey on educational process mining. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 8(1).
- [10] Calders, T., & Pechenizkiy, M. (2012). Introduction to the special section on educational data mining. *Acm Sigkdd Explorations Newsletter*, 13(2), 3-6.
- [11] Slater, S., Joksimović, S., Kovanovic, V., Baker, R. S., & Gasevic, D. (2017). Tools for Educational Data Mining: A Review. *Journal of Educational and Behavioral Statistics*, 42(1), 85-106.
- [12] Sebastiani, F. (2002). Machine learning in automated text categorization. *ACM computing surveys (CSUR)*, 34(1), 1-47.
- [13] Avasarala, B. R., Day, J. C., & Steiner, D. (2016). System and method for automated machine-learning, zero-day malware detection U.S. Patent No. 9,292,688. Washington, DC: U.S. Patent and Trademark Office.
- [14] Thornton, C., Hutter, F., Hoos, H. H., & Leyton-Brown, K. (2013, August). Auto-WEKA: Combined selection and hyperparameter optimization of classification algorithms. In *Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining* (pp. 847-855). ACM.
- [15] Feurer, M., Klein, A., Eggersperger, K., Springenberg, J. T., Blum, M., & Hutter, F. (2019). Auto-sklearn: Efficient and Robust Automated Machine Learning. In *Automated Machine Learning* (pp. 113-134). Springer, Cham.
- [16] Olson, R. S., & Moore, J. H. (2019). TPOT: A tree-based pipeline optimization tool for automating machine learning. In *Automated Machine Learning* (pp. 151-160). Springer, Cham.
- [17] B. Kim, R. Khanna, and O. Koyejo. Examples are not enough, learn to criticize! Criticism for interpretability. *Advances In Neural Information Processing Systems*, pages 2280–2288, 2016b.
- [18] Bianco, S., Martins, S., Rodríguez, D., & García Martínez, R. (2017). Ingeniería de explotación de información aplicada a la gestión universitaria: caso licenciatura en sistema Universidad Nacional de Lanús. In *XII Congreso de Tecnología en Educación y Educación en Tecnología (TE&ET)*.
- [19] Zutty, J., Long, D., Adams, H., Bennett, G., & Baxter, C. (2015, July). Multiple objective vector-based genetic programming using human-derived primitives. In *Proceedings of the 2015 Annual Conference on Genetic and Evolutionary Computation* (pp. 1127-1134). ACM.
- [20] Hall, P., & Gill, N. (2018). *An Introduction to Machine Learning Interpretability-Dataiku Version*. O'Reilly Media, Incorporated.
- [21] Kuna, H., García Martínez, R., Villatoro, F. (2009). Identificación de Causales de Abandono de Estudios Universitarios. Uso de Procesos de Explotación de Información. *Revista Iberoamericana de Tecnología en Educación y Educación en Tecnología* 5: 39-44.
- [22] Díaz, L., Martins, S., Garcia-Martinez, R. 2015. Descubrimiento de Patrones Socio-económicos de Población Estudiantil de Carreras de Ingeniería Basado en Tecnologías de Explotación de Información. *Proceedings X Congreso de Tecnología en Educación y Educación en Tecnología*. Pág. 306-315. ISBN 978-950-656-154-3.
- [23] Lundberg, S. M., & Lee, S.-I. (2017). A Unified Approach to Interpreting Model Predictions. En I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, & R. Garnett (Eds.), *Advances in Neural Information Processing Systems 30* (pp. 4765–4774).
- [24] Hara, S., & Hayashi, K. (2016). Making Tree Ensembles Interpretable: A Bayesian Model Selection Approach. *arXiv:1606.09066*
- [25] Repositorio del framework autoEDM (2019). <https://github.com/santibianco/autoEDM/> Página vigente al 29/07/2019.
- [26] Ribeiro, M. T., Singh, S., & Guestrin, C. (2016). Why should i trust you?: Explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining* (pp. 1135-1144). ACM.

Propuesta de Metodología de Gestión de Procesos para la Evaluación de la Calidad en los Entornos Virtuales de Enseñanza y Aprendizaje

Roberto García, Hernán Amatriain, Santiago Bianco y Dario Rodriguez

Laboratorio de Investigación y Desarrollo en Ingeniería de Software
Grupo de Investigación en Sistemas de Información.
Departamento de Desarrollo Productivo y Tecnológico. Universidad Nacional de Lanús
licrg@yahoo.com.ar, hamatriain@gmail.com, santiago.bianco.sb@gmail.com,
dariorodriguez1977@gmail.com

Resumen. La evaluación de un EVEA debe obtener y analizar la información para evaluar el grado de motivación de los alumnos en la utilización del ambiente. Esta motivación está relacionada con la calidad de los contenidos, la facilidad de uso, los servicios y apoyo que recibe de parte de los docentes y del personal de administración y servicio del ambiente. La Metodología que se propone brinda un marco de referencia para evaluar la calidad de un EVEA.

Palabras clave. EVEA, Calidad, Evaluación EVEA, Evaluación Calidad, Metodología Evaluación.

1. Introducción

La utilización de un EVEA es un proceso ya consolidado y extendido en la mediación del aprendizaje, pero existen muchos desarrollos que en su proceso de análisis, diseño y puesta en producción no ha estado presente el concepto de cadenas de valor. [Roldan D. y otros, 2011 Gestión de Proyectos de E-Learning]

En la formación educativa mediada a través de los EVEA intervienen aspectos de negocio (en un sentido amplio), pedagógicos y el uso de la tecnología; los que coordinados generan nuevos estilos y formas de aprendizaje. En virtud de lo mencionado es que se hace evidente orientar la evaluación de la calidad poniendo como eje central, lo que se le propone como servicio a los usuarios. [Roldan D. y otros, 2011 Gestión de Proyectos de E-Learning]

Es importante remarcar que el EVEA no solo le brinda servicio al estudiante, también a los docentes en sus roles de generadores de contenidos, seguimiento del proceso de aprendizaje de los alumnos y evaluadores del alcance de los objetivos en cuanto a la transferencia deseada. [García-Martínez, R., Servente, M. y Pasquini, D. 2003. Sistemas Inteligentes.]

El objetivo de la evaluación de un EVEA debe obtener y analizar la información para evaluar el grado de motivación de los alumnos en la utilización del ambiente [Diez, E., 2013. Aseguramiento de la Calidad en la Construcción de Sistemas Basados en el Conocimiento Un Enfoque Práctico]. Esta motivación está relacionada con la calidad de los contenidos, la

facilidad de uso, los servicios y apoyo que recibe de parte de los docentes y del personal de administración y servicio del ambiente.

En el ámbito de la educación se comprobaba evaluando los índices de repetición, abandonos, aseguramiento de conocimientos previos para determinada asignatura, mejoramiento en el resultado de evaluaciones, etc.[Diez, E.. 2013. Aseguramiento de la Calidad en la Construcción de Sistemas Basados en el Conocimiento Un Enfoque Práctico.]

La Metodología que se propone brinda un marco de referencia o guía para los analistas y grupos de trabajo que se dedican a evaluar la calidad de un EVEA.

2. Descripción del Problema

Algunos EVEAS no siempre cumplen con los objetivos de mediación efectiva en el proceso de enseñanza aprendizaje. Se ha visto que el mayor “servicio” que aporta es de repositorios de documentos en algún formato rígido (PDF, PPT, DOC, etc.) los cuales no siempre están disponibles a los alumnos en tiempo y forma. Otro “servicio” es el de comunicación de eventos más o menos importantes o recepción de peticiones de los alumnos.

El concepto de cadena de valor en algunos EVEAS no se ha tenido en cuenta en el ciclo de diseño y desarrollo. Es necesario que estos sean sometidos a una exhaustiva evaluación tanto en el eje pedagógico como en el informático a fin de alcanzar la mejora continua de los servicios y productos que lo componen.

En la formación educativa mediada a través de los EVEA intervienen aspectos de negocio (en un sentido amplio), pedagógicos y el uso de la tecnología; los que coordinados generan nuevos estilos y formas de aprendizaje. En virtud de lo mencionado es que se hace evidente orientar la evaluación de la calidad poniendo como eje central, lo que se le propone como servicio a los usuarios.

Es importante remarcar que el EVEA no solo le brinda servicio al estudiante, también a los docentes en sus roles de generadores de contenidos, seguimiento del proceso de aprendizaje de los alumnos y evaluadores del alcance de los objetivos en cuanto a la transferencia deseada.

El objetivo del presente trabajo es describir una Metodología de Gestión de Procesos (MGP-EVEA) que brinde una guía para la evaluación de la calidad en los Entornos Virtuales de Enseñanza y Aprendizaje (EVEA) tanto en el ámbito de instituciones educativas de nivel superior como en empresas que requieran capacitación a sus recursos humanos.

3. Solución Propuesta

El objetivo de validar la calidad está orientada a ajustar/adaptar rápidamente los procesos de servicios que se brindan al estudiante y docentes, la generación de contenidos y verificación de resultados en términos medición de la eficacia y eficiencia de la mediación del aprendizaje utilizando un EVEA.

Por tanto, se propone generar una metodología concebida como un conjunto de herramientas que sirva de guía alcanzar el objetivo mencionado anteriormente.

La Metodología de Gestión de Procesos (MGP-EVEA) propuesta brinda un marco de referencia o guía para los analistas y grupos de trabajo que se dedican a evaluar la calidad de un EVEA.

Las FASES del MGP-EVEA son las siguientes:

- Fase de Preparación.
- Fase de Análisis y Evaluación.
- Fase de Implementación de Mejoras.

3.1 Descripción de Fases

A continuación se describen en detalle las fases de la metodología propuesta.

3.1.1 Fase de Preparación

El objetivo de esta fase es la preparación de las condiciones necesarias para garantizar la Gestión de Calidad en los Procesos Educativos (GCPE) y los EVEA.

Para planificar y realizar la GCPE, se debe asegurar que todos los elementos necesarios se encuentren preparados y listos para brindar soporte al trabajo continuo a ejecutar sobre los procesos de la institución educativa.

Para cumplir con los objetivos de esta fase se debe definir un modelo general de procesos de referencia “a nivel de la institución educativa”, y para ello se debe: asegurar que la organización esté preparada adecuadamente para ejecutar la GCPE; y garantizar que los procesos a gestionar estén formalizados y que sea posible ejecutar un control sobre ellos. Se entiende por formalizados a la existencia de procedimientos y normativas escritas con la correspondiente descripción de funciones de la estructura de la institución educativa.

Esta fase tiene una importancia muy alta para el ámbito educativo, no sólo tiene como objeto la definición y formalización de los procesos de capacitación en las instituciones educativas y empresas, sino que además representan el primer paso de definición y formalización de los mismos.

Esto no significa que los resultados de la ejecución de esta fase sean definitivos, ya que siempre será posible intervenir para modificarlos. Sin embargo, como el costo de estas modificaciones puede ser importante, es aconsejable dedicar el tiempo suficiente a esta fase para asegurar un alto nivel de calidad y no tener que intervenir para su corrección en un momento posterior.

Para garantizar este alto nivel de calidad, quienes actúen en esta fase de Preparación, deben poner mucha atención y precisión en los resultados que se quieren conseguir (modelo, definiciones, etc.) y en cómo se consiguen los resultados (metodologías y herramientas aplicadas).

Es necesario cumplir todos los pasos que se planifiquen de la fase y también ejecutarlos en orden secuencial sólo en el primer ciclo de introducción de la GCPE. Este ciclo prevé que después de la definición del modelo general, se deben elegir los procesos que serán gestionados, para los cuales se ejecutan todos los restantes pasos de la fase.

Para cada proceso a gestionar, la fase se concluye cuando después de la elección y la designación del equipo, la documentación sea formalizada, los indicadores sean

definidos y el plan de las actividades sea actualizado. La fase es completada “en general” sólo cuando todos los procesos están en esta condición.

Una vez preparado y puesto en práctica el sistema de GCPE y el EVEA, pueden surgir situaciones en las cuales se evidencia la necesidad de modificar algunos de los elementos definidos en esta fase (como la estructura del modelo o la asignación de las responsabilidades).

3.1.2 Fase de Análisis y Evaluación

Esta fase representa el núcleo fundamental del MGP-EVEA es decir toda la gestión de los procesos. Incluye el trabajo cotidiano y también el análisis específico sobre cada proceso que lo compone.

Una de las tareas fundamentales de esta fase es la de evaluar en forma permanente el estado de un proceso (análisis ordinario) con la verificación de los valores medidos en el tiempo; cuando algunos de ellos aparecen fuera del alcance de los umbrales se deben activar los pasos para decidir si hay que intervenir y cuáles acciones son necesarias ejecutar. Esta actividad, en la mayoría de los casos, es ejecutada por parte del equipo técnico permanente que administra el campus.

En esta fase está incluida la reacción a todos los estímulos externos (problemas, oportunidades, cambios impuestos, etc.) que tengan impacto en la gestión de los procesos y la activación de los pasos (análisis extraordinario) necesarios para cumplir lo que el estímulo requiere. Esta actividad, en la mayoría de los casos, es ejecutada por un equipo ad hoc (especialistas en educación, medios, administradores y técnicos) establecido según el tipo de análisis.

En el ámbito de esta fase, algunos pasos se deben ejecutar en modo continuo (por ejemplo analizar los indicadores medidos) mientras otros se ejecutan sólo si existe una necesidad específica.

No existe un orden preestablecido y secuencial de ejecución de todos los pasos. El orden de ejecución depende de las necesidades y oportunidades que se presentan a la institución / empresa. También los resultados de un paso pueden afectar los pasos siguientes de análisis.

Algunos pasos pueden ser ejecutados en una secuencia principal. La secuencia puede ser interrumpida por otros pasos, ejecutados individualmente o en secuencias colaterales. Los pasos componentes de la secuencia principal son:

- Analizar el proceso y detectar causas.
- Proponer evolución del proceso.
- Obtener aceptación de una propuesta de mejora.
- Formalizar la propuesta aprobada.
- Proyectar la actividad de implementación.

3.1.3 Fase de Implementación de mejoras

Esta Fase permite poner en práctica las mejoras de procesos individualizadas en las fases previas. Su complejidad depende de la intencionalidad didáctica que se quiere imponer al servicio del EVEA, incluye:

- La complejidad del proceso,
- La importancia de las modificaciones,
- La dificultad del cambio,

- La sensibilidad del proceso.

Una de las actividades que se deben cumplir en esta fase, si es necesario, es el desarrollo de la capacitación necesaria para operar el proceso según las modificaciones individualizadas previamente. El desarrollo eficaz de la capacitación puede ser un aspecto clave para conseguir éxito en todo el proyecto de mejora.

Otra actividad de esta Fase es el paso de ejecución de la Prueba Piloto, que es necesario sólo si la complejidad de la mejora que se debe implementar lo requiere.

3.2. Factores generales de la evaluación de calidad

Para poder realizar una evaluación de la calidad en el modelo propuesto se basa en las recomendaciones asociadas a los principios de calidad [Becta 2006], a la norma DIN PAS 1032-1 de la Agencia Alemana de Normalización que plantea el modelo de calidad en el ciclo de vida de un e-learning y a las Normas ISO 9000, deben tenerse en cuenta los factores que se detallan en la Tabla 1.

Factores generales de evaluación.	Evidencia
Apariencia	Los alumnos y docentes deben percibir orden y practicidad en la composición del entorno (proporción de elementos dentro de la pantalla, distribución y reparto de los elementos, usos de puntos de atención y formatos de tipografía, etc.)
Economía	La formación debe proporcionarse a un coste razonable referidos a los medios que se proporciona en el ámbito físico de la entidad / institución debe ser irrestricto y eficiente. Esto quiere decir que en todo caso el ingreso al entorno debe ser al menor costo posible.
Idiomas	La variedad de idiomas depende de la comunidad de alumnos que se desea atender, lo cual hace al entorno más atractivo para sus participantes.
Imágenes, video Audio	Las imágenes, videos y audios deben ser pertinentes con la propuesta educativa, la utilización de estos y fáciles de entender, el recurso de animación debe ser correctamente usado y no debe competir negativamente con el objeto de enseñanza, la correcta mezcla de música y locución debe contemplar la calidad de la sincronización de los diferentes elementos utilizados. (participación de especialistas en la utilización de estos recursos)
Agilidad	La navegación por las diferentes aulas virtuales debe ser de diseño homogéneo y de acuerdo a un patrón. La navegación debe ser intuitiva.
Facilidad de Lectura	En el desarrollo de los contenidos además del experto en el área (docentes) es importante que la escritura sea entendible y la información que se quiere transferir a los alumnos se logre de manera estructurada permitiendo el objetivo primordial que es la intencionalidad de aprendizaje.
Versatilidad	El entorno debe estar orientado a diferentes necesidades de los usuarios objetivos. Alumnos con asistencia virtual, aulas extendidas, autoaprendizaje y aprendizaje guiado, la comunicación entre los miembros de la comunidad educativa debe ser a veces en tiempo real y en ocasiones en forma asíncrona. El entorno debe tener toda esta flexibilidad instrumentada.
Conectividad	La interconexión en la red proporciona beneficios imprescindibles en el funcionamiento eficiente para la transmisión de multimedia, información, hardware y software, los cuales son procesos importantes para incrementar la productividad

	del aprendizaje de un alumno.
Accesibilidad	Permitirle al alumno tener acceso desde cualquier ubicación espacio-temporal le proporciona una autonomía en el uso del entorno virtual, de igual manera es importante la seguridad con claves de acceso para que el alumno este seguro que la información de sus actividades del curso, su página personal, y la información que posee no la pueda obtener otra persona
Disponibilidad del Servicio	La flexibilidad de su modelo personaliza el aprendizaje a las exigencias del usuario y posibilita elegir el horario. Se adapta a las necesidades pedagógicas y metodológicas de cada curso. Favorece la desaparición de barreras físicas al ser irrelevante la ubicación del usuario y el momento que el alumno haga uso de ella.
Facilidad de Navegación	La interfaz debe ser intuitiva y accesible, lo cual debe facilitar la adaptación a la plataforma. Debe posibilitar el seguimiento completo de las actividades de los alumnos por parte de los docentes; debe en definitiva asegurar la eficiencia y eficacia de la transferencia de los contenidos objeto de aprendizaje y la evaluación para la acreditación de conocimientos.
Mapa de navegación	Los alumnos deben tener la posibilidad de ver el mapa de un módulo o del campus virtual para encontrar un contenido rápidamente dentro del entorno.
Índice de contenido	En el entorno debe existir un índice de contenido para que los alumnos y docentes puedan consultar bibliografía de los contenidos propuestos.
Descarga de Aplicaciones	Si en el entorno se ponen a disposición de los alumnos contenidos que utilizan un software especial, se debe indicar o proveer la descarga del mismo desde el entorno.
Velocidad de Transferencia	El ambiente tecnológico debe asegurar que la velocidad de transferencia sea el conveniente de acuerdo a la usabilidad de lo publicado para el alumno.
Manual de Usuario	El entorno debe existir un link o archivo que permita conocer la funcionalidad del entorno.
Foros de Discusión	Deben existir espacios donde los alumnos tengan la posibilidad de discutir sus opiniones y resolver sus dudas acerca de los temas relacionados.
Página personal	Esta función debe existir en el entorno para permitir que los alumnos personalice el entorno personal de manera que se sienta parte del entorno, potenciando que interacción en con sus partes y promuevan un mayor acercamiento y evitando el aislamiento individual. Se debe promover el aprendizaje de pares.
Programación Académica	El entorno debe asegurar que el alumno al iniciar un curso tenga toda la información necesaria para planificar y organizar sus tiempos de estudio.
Calendario	Debe existir un espacio compartido entre alumnos, tutores y profesores donde se publiquen y se tengan presentes eventos de importancia en la enseñanza / aprendizaje en el plano general de la institución, como así también en los diferentes cursos / materias del entorno.
Agenda	Debe existir un espacio confidencial y personal del alumno que permita recordar notas y eventos tanto del curso como personales.
Guías de Apoyo por Asignatura	Debe existir una funcionalidad que asegure una ayuda al alumno respecto de la utilización de los contenidos, facilitando la interacción con los profesores y tutores a cargo de la asignatura / curso.
Mail Interno	Los alumnos deben disponer de un correo electrónico que permitan una comunicación privada entre alumnos, profesores y tutores.
Apuntes On line	Debe existir un repositorio de apuntes On line por asignatura / curso solo accesible por los alumnos inscriptos.
Editor de texto	Formato Debe existir la funcionalidad de generar textos, inserción de imágenes, inserción de tablas, inserción de enlaces que permita que el alumno genere textos o apuntes para su propio uso o compartirlos con sus pares.

Chat en tiempo real	Debe existir la función de comunicar por escrito a los actores del entorno que se encuentren en línea. Puede utilizarse para que el alumno y el profesor establezcan horarios de tutoría para intercambiar mensajes que ayuden en la resolución de dudas o problemas. Toda comunicación por este medio debe estar resguardado mientras dure el curso.
Pizarra Compartida	Debe existir un espacio gráfico que posibilite que docentes y alumnos compartan gráficos para hacer demostraciones, ejemplificar teorías, etc. y a permita escribir sus aportes o apreciaciones en un curso / asignatura.
Links a espacios virtuales de investigación académica	Se debe tener la funcionalidad apropiada para promover el acceso a espacios virtuales orientados a la investigación para que los alumnos tengan acceso a estos círculos de investigación con el objetivo de promover la investigación, brindar y recibir información y promover fundamentalmente el desarrollo y evolución de un aprendizaje en constante movimiento.
Biblioteca Digital	Debe existir la funcionalidad que permita ingresar desde cualquier lugar y mediante los medios de comunicación apropiados, tanto para los profesores como para los alumnos y a facilitar la búsqueda y variedad en los recursos educacionales que están utilizando como parte de su formación académica en el nuevo modelo pedagógico.
Ayuda/orientación	Se debe tener una funcionalidad efectiva de help desk del entorno y en forma separada por curso / asignatura para resolver todas las dudas y consultas de los alumnos en el uso de todas las herramientas y contenidos publicados.
Búsqueda dentro de un curso	Se debe poseer una herramienta que permita reducir la velocidad para encontrar un documento, o concepto, a fin de asegurar que el alumno tenga una transferencia de conocimientos efectiva de un contenido de aprendizaje.
Laboratorios on line	Se debe poseer un ámbito donde los alumnos puedan aplicar los conocimientos adquiridos mediante la práctica de los conceptos.
Evaluaciones	Se debe poseer una facilidad que permita la evaluación en sentido amplio del desempeño del alumno que incluya los contenidos específicos del curso / asignatura, participación en el entorno, aportes y resolución de TP. El entorno debe aportar métricas de participación del alumno en el entorno.
Espacio para tutorías	Se debe contar con un espacio definido para que el tutor en estado on line y el alumno se reúnan en forma virtual para resolver sus dudas y pueda alcanzar el aprendizaje de los contenidos del curso / asignatura.
Ejercicios Prácticos	Se debe contar con un espacio que se le brinda al alumno para que practique lo aprendido y ser evaluado en su producción en tiempo y forma.
Contenidos	Los contenidos de un curso / asignatura deben estar elaborados de manera que permitan ser autoinstructivos, autoevaluativos, adaptados a las situaciones de aprendizaje pretendido (apoyo, aula virtual, aula extendida, etc.) y al perfil del alumno para asegurar al máximo la transferencia del aprendizaje objetivo.
Combinación de contenido y actividades	El entorno debe poseer el uso de multimedia para enriquecer el aprendizaje. La utilización correcta de la multimedia debe ser diseñada por especialistas de contenidos y especialistas en el desarrollo de las herramientas utilizadas.
Simulaciones	El entorno debe poseer un espacio que permita, en aquellos contenidos que así lo ameriten, la simulación de casos que permita maximizar el aprendizaje mediante un ambiente no real.
Evaluaciones de ensayo	Poseer un espacio en el entorno que permita a los alumnos efectuar exámenes en línea que orienten a practicar previamente al examen formal de acreditación. El objeto de estos exámenes es que los alumnos tengan evidencia de los temas que tiene debilidades conceptuales.

Tabla 1. Factores para la evaluación de la calidad

4. Validación

Para validar la propuesta se comenzó a realizar el proceso de evaluación de la metodología con los indicadores y métricas propuestas en los campus de dos Universidades Nacionales: Lanús (UNLa) y Lomas de Zamora (UNLZ).

Si bien la tarea no finalizó, los resultados parciales son alentadores en cuanto a reunión de información para evaluación y poder aplicar medidas correctivas y realimentar el proceso mismo en una práctica de calidad basada en la mejora continua.

5. Conclusiones y futuras líneas de investigación

La metodología que se desarrolla constituye un soporte que contribuye a:

- Alcanzar estándares comunes;
- Alcanzar un contexto de trabajo estable, maduro, repetible;
- Alcanzar resultados y documentos homogéneos;
- Ordenar, evitar redundancia y/o faltantes;
- Identificar los ajustes y modificaciones necesarias para la mejora continua del EVEA;
- Organizar los controles y/o las verificaciones de consistencia.

La estructura de la metodología posee las siguientes características:

- Se organizara en fases y cada una de éstas en actividades.
- Se detallaran las fases y actividades en forma correlacionadas.
- Se detallaran las herramientas (sistemas y pautas) de soporte.

Se debe continuar con la evaluación en los campus UNLa y UNLZ, para tener resultados más sólidos que avalen por completo el modelo propuesto en el presente trabajo.

6. Bibliografía

- AENOR (1995) UNE-EN-ISO 8402: Gestión de la calidad y aseguramiento de calidad. Vocabulario (ISO 8402:1994).
- Alonso, F., Juristo, N., Maté, J., Pazos, J. 1996. Software engineering and knowledge engineering: Towards a common life cycle. *Journal of Systems and Software*, 33(1): 65-79.
- Becta (2006) Quality Principles for digital learning resources http://partners.becta.org.uk/upload-dir/downloads/page-documents/quality_principles.pdf
- Borrajó, D. 1993. Inteligencia artificial: Métodos y técnicas. Ed. Centro de Estudios Ramón Areces.

- Britos, P., García-Martínez, R., 2009. Propuesta de Procesos de Explotación de Información. Proceedings XV Congreso Argentino de Ciencias de la Computación. Workshop de Base de Datos y Minería de Datos. Págs. 1041-1050. ISBN 978-897-24068-4-1.
- Britos, P., Hossian, A., García Martínez, R., Sierra, E. 2005. Minería de Datos Basada en Sistemas Inteligentes. Nueva Librería.
- Brulé, J., Blount, A. 1989. Knowledge acquisition. McGraw-Hill, Inc.
- Carrillo Verdun, J. 1987. Metodología para el Desarrollo de Sistemas Expertos. Tesis Doctoral. Universidad Politécnica de Madrid.
- Club-BPM. Business Process Management. Centro Oficial del BPM. España y Latinoamérica. (<http://www.club-bpm.com/>, último acceso 30-05-2016).
- Diez, E.. 2013. Aseguramiento de la Calidad en la Construcción de Sistemas Basados en el Conocimiento Un Enfoque Práctico. Revista Latinoamericana de Ingeniería de Software, 1(5) 167-206, ISSN 2314-2642.
- Dramis, L., Britos, P., Rossi, B., García-Martínez, R. 2000 Verificación de Base de Conocimiento basada en Algebra de Grafos. Proceedings del VI Congreso Internacional de Ingeniería Informática. Pag. 729-738. Editado
- García-Martínez, R., Britos, P. 2004 Ingeniería de Sistemas Expertos. Editorial Nueva Librería. ISBN 987-1104-15-4.
- García-Martínez, R., Servente, M. y Pasquini, D. 2003. Sistemas Inteligentes. Editorial Nueva Librería. ISBN 987-1104-05-7.
- Grover, M. 1983. A Pragmatic Knowledge Acquisition Methodology. IJCAI Vol. 83: 436-438.
- Gómez, A., Juristo, N., Montes, C., Pazos, J. 1997. Ingeniería del conocimiento. Editorial Centro de Estudios Ramón Areces.
- Hauge, O., Britos, P., García-Martínez, R. 2006. Conceptualization Maturity Metrics for Expert Systems. IFIP International Federation for Information Processing, Volume 217, ed. M. Bramer, (BostonSpringer), Pag. 435-444.
- Ibarra, G., Bazán, P. 2013. Análisis y comparación de plataformas BRMS a través de una prueba de concepto. XV Workshop de Investigadores en Ciencias de la Computación.
- IEEE, 1998. IEEE Std 1074-1997. IEEE Standard 1074 for Developing Software Life Cycle Processes.
- ISO; 2008. ISO/IEC 12.207, Information Technology / Software Life Cycle Processes. International Organization for Standardization
- Kliegr, T., Kuchar, J., Sottara, D., Y Vojr, S. 2014. Learning business rules with association rule classifiers. In RuleML14 The 8th International Web Rule Symposium. Springer.
- Nguyen, T., Perkins, W., Laffey, T., Pecora, D. 1987. Knowledge-base verification. AI Magazine, 8(2) 69-75.
- PMBOK, 2013. A Guide to the Project Management Body of Knowledge. Project Management Institute.
- Pressman, R., 2010. Ingeniería del Software. Un Enfoque Práctico. Séptima edición. Capítulo 1. McGraw-Hill, Inc. ISBN: 978-607-15-0314-5.
- Roldan D. y otros, 2011 Gestión de Proyectos de E-Learning Segunda edición ISBN 978-607-707-095-5

- Rossi, B. 2001. Sistema Experto de Ayuda para la Selección del Modelo de Ciclo de Vida. Tesis de Master en Ingeniería del Conocimiento. Universidad Politécnica de Madrid.
- Schreiber, G. 2000. Knowledge engineering and management: the CommonKADS methodology. MIT press.
- Sommerville, I., 2005. Ingeniería del Software. Séptima edición. Capítulo 1. Pearson Addison. ISBN: 84-7829-074-5.
- Staab, S., Studer, R., Schnurr, H. P., Sure, Y. 2001. Knowledge processes and ontologies. IEEE Intelligent systems, (1): 26-34

Simulador en línea para capacitación de individuos hipoacúsicos adultos

Quintana Nelba¹, González Alejandro Héctor⁴, Vallejo Alcira², Pereyra José María³

Facultad de Humanidades y Ciencias de la Educación. Universidad Nacional de la Plata

Instituto de investigación en informática III-LIDI. Facultad de Informática de la Universidad Nacional de la Plata

Comisión de Investigaciones Científicas Pcia. Buenos Aires

¹n.quintana@fahce.unlp.edu.ar, ²agonzalez@lidi.info.unlp.edu.ar, ³alcira.vallejo@cyt.cic.gba.gob.ar, ⁴jose.pereyra@presi.unlp.edu.ar

Resumen. Este trabajo describe el diseño y desarrollo de un prototipo de simulador educativo en línea destinado a la capacitación de personas hipoacúsicas que utilizan dispositivos auditivos (audífonos o implantes cocleares). Particularmente la estrategia de simulación consiste en la recreación de un escenario donde la persona hipoacúsica se enfrenta a una situación de entrevista laboral. El prototipo es la instancia conclusiva de un conjunto de actividades formativas complementarias para resolver en línea, diseñadas en HTML5 y JavaScript. El prototipo de simulador, de características multimedia, se desarrolló a partir de la herramienta “Escenarios de ramificación” de la serie de código abierto que ofrece el portal H5P, embebido en un sitio web desarrollado en HTML5 y el lenguaje de programación PHP. El sitio tiene la capacidad de almacenar los datos de cada usuario, incluidas las actividades educativas que realiza, en una base de datos relacional MySQL.

Palabras clave: Tecnología aplicada en Educación, Simuladores educativos, Hipoacusia, H5P.

1 Introducción

El desarrollo del presente proyecto, de carácter multidisciplinar, fue motivado en la idea que la tecnología informática aplicada a la educación es una herramienta altamente favorable para el diseño de estrategias educativas de personas hipoacúsicas, por su alto carácter visual.

Por lo tanto, podemos encontrar múltiples dispositivos, sistemas y recursos que favorecen la interacción comunicativa cuando la persona presenta dificultades para que pueda llevarse a cabo.

Actualmente existe un amplio abanico de instrumentos tecnológicos que se han desarrollado para mejorar la calidad de vida de este colectivo favoreciendo la accesibilidad total y potenciando la autonomía personal. Es el caso de los audífonos e implantes cocleares que permiten, a las personas con discapacidad auditiva, acceder con mayor efectividad a la información oral que reciben de su entorno más próximo. Estos instrumentos disminuyen las barreras de comunicación que impiden el acceso a la información y a la comunicación.

El uso de las estrategias visuales y texto escrito es una estrategia de compensación muy eficaz. La utilización de la tecnología informática como procedimiento para incrementar, mantener o mejorar las capacidades funcionales de estos individuos, se ha convertido en un recurso esencial. En este marco se incluyen acciones de formación y capacitación, como las que se describirán en el presente trabajo.

El proyecto de investigación, se desarrolla como base para la tesis de Maestría en Tecnología Informática Aplicada en Educación de la Facultad de Informática de la Universidad Nacional de la Plata, contempla la necesidad de capacitación de adultos hipoacúsicos que usan prótesis auditivas, por ser un grupo etario poco considerado en situaciones de aprendizaje. El diseño de la capacitación está orientado a reproducir situaciones de entrevistas laborales, mediante la aplicación de un simulador desarrollado en HTML5 y JavaScript, embebido en un sitio web específicamente desarrollado en PHP para la aportación de contenidos y la realización de actividades interactivas previas a la utilización del simulador propiamente dicho.

1.1 El diseño como aporte a la adquisición del conocimiento

El proceso de la comunicación ha sido enriquecido, en las últimas décadas, por los medios digitales interactivos. Estos medios permiten presentar la información en una forma no lineal, en forma de estructuras entrelazadas compuestas por una red de nodos semánticos que permiten al usuario elegir cómo moverse dentro de esta red [1].

Tratar con éxito con estos medios multicanales - sonido, música, voz, tipografía, imágenes, películas, movimiento - requiere diferentes competencias, que en general son aportadas por equipos multidisciplinarios.

El diseño comunicacional actualmente representa un rol cognoscitivo esencial en el aprendizaje y el conocimiento; que se va incrementando con la continua expansión de la tecnología de información. Con la intervención del diseño, la presentación del conocimiento y la comunicación pueden ser mediatizados a través de una interfaz que puede ser percibida y asimilada. De allí que el diseño comunicacional de la información es una herramienta indispensable en el proceso de comunicar y aportar conocimientos [2]. Para ello requiere estructurar un gran conjunto de datos y luego traducirlos a un ámbito visual y auditivo con un patrón en forma de red para la navegación.

A la hora de diseñar material educativo, por lo tanto, deben aplicarse nociones de jerarquía, estructura, y lo que podría llamarse "gestión sensorial", o sea la experta selección de estímulos que guían la atención de los usuarios, reduciendo así la complejidad del conocimiento y contribuyendo a la claridad y a la comprensión.

1.2 La adquisición de conocimientos mediante el aprendizaje significativo

David Ausubel [3] desarrolla una teoría cognitiva del aprendizaje verbal significativo en contraposición al aprendizaje verbal memorista. Para ello los nuevos conocimientos deben relacionarse con los saberes previos que posea el aprendiz.

El aprendizaje significativo es un proceso activo, integrador e interactivo que tiene como producto la adquisición y retención de un conocimiento significativo a partir de la interacción entre significados potenciales pertenecientes a un dominio en cuestión y las ideas pertinentes (de anclaje) de la estructura cognitiva de quien aprende. Este proceso es significativo porque produce nuevos significados (conocimientos) que se estructuran en un sistema organizado.

Ausubel propone tres tipos de aprendizajes significativos: el aprendizaje representacional, el de conceptos y el proposicional.

En el *aprendizaje representacional* el significado de los símbolos surge de equiparar esos símbolos con sus referentes.

El *aprendizaje de conceptos* se produce por formación, con la experiencia directa en los niños, y por asimilación en los escolares y adultos, por la combinación de referentes ya existentes. Los conceptos son importantes en el aprendizaje significativo porque la resolución significativa de problemas depende de la disponibilidad, en la estructura cognitiva, de conceptos de orden superior y subordinados a los que los nuevos conceptos se puedan adaptar.

Hay tres tipos de *aprendizaje proposicional*: en el subsumidor, las nuevas proposiciones se relacionan con proposiciones de orden superior; en el de orden superior, la proposición nueva ordena proposiciones subordinadas; y en el combinatorio, se da una combinación de contenidos pertinentes y no pertinentes.

En el aprendizaje significativo se produce una modificación de las ideas de anclaje cuando se produce el vínculo entre las ideas nuevas y las ideas preexistentes.

El diseño del prototipo de simulador como recurso educativo involucra la consideración de los procesos y habilidades cognitivas que deben ponerse en juego [4]. Se fijan objetivos, pero, de tal modo que puedan modificarse en base al análisis que se va realizando durante el proceso de aprendizaje de los participantes. La estrategia del simulador es imitar situaciones problemáticas que permitan a los alumnos interactuar de forma flexible y obtener una retroalimentación consecuente con dicha actividad, la cual tiene como objetivo crear nuevos conocimientos y dar lugar a aprendizajes significativos.

1.3 El rol de la inteligencia emocional en el aprendizaje de individuos hipoacúsicos

Daniel Goleman [5] ha investigado los efectos que una adecuada inteligencia emocional ejerce sobre las personas. En el caso de las personas con hipoacusia, su estado emocional es altamente sensible. Al ser un tipo de discapacidad que no se ve, las personas oyentes desconocen a priori la condición del hipoacúsico viéndose éste en la obligación de estar continuamente informando de su situación. La reiteración constante de su condición limitante comunicacional disminuye su autoestima y tiende a aislarlo socialmente. Por ello es fundamental impulsar y fortalecer la formación de los individuos con discapacidad auditiva con dispositivos auditivos para facilitar su

incorporación a su entorno sociocultural, en el cual predomina la oralidad. De este modo, se busca lograr que la simulación se constituya en experiencia previa del aprendiz, para así facilitar su desempeño y favorecer su autonomía en la situación de entrevista y toma de decisiones en la vida real.

1.4 Los simuladores como estrategia pedagógica

El uso de la simulación o el juego como estrategia pedagógica fue consolidándose a lo largo de las últimas décadas, pero es el resultado de una historia milenaria, donde el carácter inicial puramente lúdico, fue adoptando características funcionales, que actualmente revisten múltiples aplicaciones, tanto en la investigación científica como en la educación. Este largo camino condujo, en nuestros días, al desarrollo de nuevas tecnologías educativas como los modernos y sofisticados simuladores que reproducen escenarios reales y los sistemas expertos basados en inteligencia artificial.

En este desarrollo entendemos como simulación al proceso en el cual se sustituyen las situaciones reales por otras creadas artificialmente, pero donde el modelo reproduce la apariencia, la estructura y la dinámica del sistema [6]. Es un entorno experimental simulado, una situación simplificada y artificial que contiene suficiente verosimilitud, o ilusión de realidad, para inducir respuestas reales por parte de los participantes, ya que está diseñado para instruirlos sobre alguna situación del mundo real. Su propósito es provocar en los participantes situaciones experienciales que puedan transferirse, luego, al sistema real.

La idea de aprender haciendo desde la concepción de Dewey en la actualidad ya se encuentra naturalizada en todas las instancias educativas. La estrecha vinculación de la teoría con la práctica permite aplicar y obtener nuevos conocimientos por descubrimiento a partir de la práctica, así como la comprobación de la validez de los conceptos teóricos [7]. La base de este enfoque es que el estudiante tenga una experiencia directa a través del planteo de un problema auténtico que motive su interés, y que además pueda construir los conocimientos necesarios para resolverlo [8]. Este enfoque del aprender haciendo requiere que los estudiantes puedan implicarse personalmente con el problema a resolver, que para ello deberá poseer ciertas características que lo asemejen a los problemas “reales” y que no sean problemas de tipo “académicos”, sin vinculación directa con las prácticas concretas en situaciones de la vida real.

En este marco, se ha identificado el posible potencial de la simulación computarizada como instrumento del método en la construcción significativa del conocimiento; es decir, que pueden ser la solución a la descontextualización del aprendizaje, predominante en la educación institucionalizada. El problema de la descontextualización comienza a visualizarse más claramente a partir del surgimiento del concepto de aprendizaje situado. Brown, J. y sus colaboradores, plantean que las actividades educativas y la adquisición de conceptos debe tener lugar en los ambientes en los cuales el aprendizaje se desarrolla y se aplica el conocimiento y por lo tanto tiene pertenencia. La construcción del conocimiento se genera de forma dinámica, mediante la interacción con la situación. Los autores declaran que “el conocimiento está situado, siendo en parte un producto de la actividad, del contexto y de la cultura en la cual se desarrolla y se utiliza” [9].

Los entornos de simulación aplicados a la educación reemplazan en gran medida el contexto de realidad, generando un entorno propicio para el aprendizaje situado, siempre y cuando presenten una interfaz que posibilite al aprendiz la representación de un evento con las características que presentaría un sistema real [6].

Otro apoyo teórico significativo para el aprendizaje a través de simuladores lo constituye el modelo de “Entornos de aprendizaje constructivista, propuesto por Jonassen [10], que plantea el involucramiento y compromiso del aprendiz en la construcción del conocimiento. Al presentar un problema mediante una simulación, se incorporan tres componentes del modelo de Jonassen: el contexto del problema, la simulación propiamente dicha y un espacio de manipulación que permite al alumno interactuar con el problema sintiéndolo como propio. Esta interacción permite que el alumno pueda influir y modificar el entorno, logrando así un aprendizaje significativo [11].

2 La propuesta educativa

La propuesta educativa en desarrollo está adaptada fundamentalmente a las posibilidades de individuos que utilizan dispositivos auditivos, que poseen formación secundaria o terciaria y pretenden aspirar a un puesto de trabajo. Para ello se concibió un diseño basado en los presupuestos teóricos de D. Ausubel, ya que aportan una visión integradora para comprender la adquisición de conocimientos.

Desde el punto de vista del usuario de este prototipo de simulador educativo, se pretende generar un proceso de cambio conceptual, a través del anclaje de los conocimientos nuevos en los ya disponibles: el usuario podrá aprovechar de una manera eficaz su propio conocimiento ya existente, como una matriz ideacional (de ideas) y organizativa para la incorporación, comprensión, retención y organización de ideas nuevas.

En el recurso de simulación de este trabajo se distinguen dos tipos de procesos de aprendizaje verbal significativo: los procesos perceptivos y los cognitivos (la diferencia entre ambos es de inmediatez y de complejidad).

Por ejemplo la percepción de los videos supone un contenido inmediato de la conciencia. Por otro lado, para la resolución de problemas se apelará a la cognición, que supone procesos como relacionar el nuevo material con aspectos pertinentes de la estructura cognitiva ya existente, determinar cómo se puede conciliar el nuevo significado resultante con el conocimiento establecido y recodificarlo en un lenguaje más familiar e idiosincrásico.

Las personas apelan de esta manera a distintas maneras jerárquicas de relacionar la nueva información con sus ideas de anclaje en la estructura cognitiva.

2.1 Los objetivos de la propuesta

La propuesta educativa se desarrolla considerando como *objetivo general* la capacitación para resolver con éxito las instancias de una situación de entrevista laboral a la que opten presentarse personas con distintos grados de pérdida auditiva que utilizan dispositivos auditivos (audífonos e implantes cocleares). Esta propuesta incluye como

objetivos específicos: desarrollar un prototipo de simulación de entrevista laboral, familiarizar a los participantes con el uso de un simulador como recurso de aprendizaje, reunir experiencia en la toma de decisiones que promuevan respuestas acertadas en una entrevista laboral, juntamente con el entrenamiento en la aplicación de técnicas comunicacionales eficaces durante la supuesta entrevista.

2.2 Desarrollo de la propuesta

Se llevó a cabo una investigación en la que se buscó identificar y describir desde la psicología cognitiva cómo los avances tecnológicos y digitales pueden enriquecer el proceso de aprendizaje significativo de hipoacúsicos con prótesis auditivas.

Para el trabajo se realizó una revisión bibliográfica sobre simulación educativa para personas hipoacúsicas. Se buscaron simuladores en línea relacionados con la hipoacusia y el desarrollo de entrevistas laborales.

Se lleva actualmente un trabajo experimental basado en la elaboración de un prototipo evolutivo de un simulador en línea para poder indagar las estrategias comunicacionales para hipoacúsicos postlocutivos adultos. Se puede acceder desde el sitio: <https://simulador.aulalibre.com.ar/>

Para la evaluación del prototipo se realiza una presentación del simulador denominado SETH (Simulador de entrevistas de trabajo para hipoacúsicos) a los participantes y se desarrolla una observación no participante de cómo se recorre el simulador por parte de los interesados. Las reflexiones de los usuarios a lo largo del recorrido del simulador se registran en fichas evaluativas en línea enviadas a través de correo electrónico. Una vez finalizado el recorrido del simulador por todos los participantes, se concreta un encuentro final de cierre para obtener una reflexión conjunta sobre los beneficios del recurso por cada etapa del recorrido del simulador.

Con el fin de validar el simulador seleccionado se busca que reúna las siguientes características: los audios deben estar acompañados por textos de apoyo los cuales deben ser considerados como elementos indispensables, tipografía legible, textos de fácil comprensión, variedad de las imágenes dado que las personas con hipoacusia desarrollan la memoria visual, interfaz amigable e intuitiva compuesta de elementos de acción. Este trabajo tiene en cuenta lo conforme a la Ley 26.653 de Accesibilidad de la Información en las páginas Web.

En una etapa posterior se obtendrán datos cuanti-cualitativos. Se analiza si la interface del simulador propuesto favorece la atención, memoria, estrategia comunicacional e inteligencia emocional al momento de la entrevista laboral de personas. En particular interesa estimar la posible aplicación, utilización y consecuencias cognitivas en el desarrollo de estrategias comunicacionales propuestas por un simulador en línea.

Para la parte cualitativa se utiliza un “estudio explicativo” con el fin de conducir a un sentido de comprensión o entendimiento de un determinado fenómeno. Apunta a las causas de los eventos físicos o sociales. Pretende responder a preguntas como: ¿por qué ocurre? ¿en qué condiciones ocurre?

2.3 Diseño del material educativo

El diseño comunicacional e instruccional del recurso multimedia contempla la inclusión de material en formato de texto, imágenes y video como instancia de entrenamiento previo a la experiencia del simulador.

Se incluyeron recursos en formato de texto por ser considerados como un elemento indispensable en el caso de las personas con hipoacusia [12]. También se incluyó imágenes, infografías, presentaciones y videos subtítulados, dado que las personas con hipoacusia desarrollan preponderantemente la memoria visual [13].

Los contenidos son complementados con instancias de evaluación formativa presentadas mediante preguntas automatizadas del tipo de opción múltiple, con retroalimentación específica en cada ítem y actividades colaborativas planteadas como tareas grupales, de manera de promover la interacción comunicacional entre los participantes. En la Fig.1 puede verse el prototipo de la interfaz para la etapa de autoanálisis.



Fig. 1. Interface del prototipo del simulador SETH

El diseño del prototipo de simulador incluyó la recreación de escenarios y la reproducción de la interacción en los mismos, partiendo de la guionización de patrones de comportamiento, así como de las competencias relacionadas con el conocimiento de los procesos de enseñanza-aprendizaje [14, 15].

Las funciones cognitivas que pretende estimular este simulador son:

Atención y concentración, para escuchar y estar atentos.

Memoria, para recordar y aprender con ello, lenguaje apropiado a una entrevista para comunicarse efectivamente.

Ejecutiva, para planificar actividades

Lenguaje, para una mejor comunicación.

Percepción y reconocimiento o inteligencia emocional, entendiéndose por la habilidad cognitiva de una persona para comprender el universo emocional de otra. El propio historial emocional de las personas hipoacúsicas puede afectar o distorsionar qué emociones se perciben en los demás.

2.4 Desarrollo técnico

El material se presenta como un sitio web desarrollado en HTML5 y el lenguaje de programación PHP, bajo la versión 7.

La información relevante de cada usuario (datos personales y las actividades educativas que realiza) es almacenada en una base de datos relacional MySQL, administrada mediante la herramienta PhpMyAdmin.

Para facilitar la modificación y mantenimiento del sitio, se utiliza el patrón MVC (Modelo-Vista-Controlador). Este patrón de arquitectura de software permite separar la lógica de la interfaz de usuario, organizando el sitio en tres capas completamente definidas. Con la finalidad de respetar este patrón, se optó por instalar la versión 2.0 del motor de plantillas Twig, utilizado por el framework Symfony. Entre sus características, Twig permite definir el esqueleto de un sitio, haciendo que todas las páginas se acoplen a él. Con este motor se logró independizar el contenido de la estructura, como también eliminar código repetitivo.

En cuanto al diseño se utilizó, del lado del cliente, el framework Materialize basado en Material Design (normativa de diseño creada por Google). Materialize se enfoca en proporcionar los elementos necesarios para mejorar el aspecto estético del sitio y su capacidad de adaptación a diferente formato de dispositivos.

En cuanto a la interacción con el usuario, se desarrolló el código necesario para captar los eventos que ocurren en la totalidad del sitio, por medio de la biblioteca de JavaScript jQuery. Se registra la información de acceso, tiempos, cantidad de intentos, grado de avance, y revisiones que realizan los participantes. Esta información será luego descargada y analizada.

Se incorporó también la biblioteca PHPMailer para el envío seguro de correos. Esta biblioteca permite, mediante PHP, disponer correos automáticos tanto para validar una cuenta, como para recuperar un usuario o cambiar una contraseña, a través de un servidor SMTP.

El desarrollo y construcción del prototipo del simulador se basó en la utilización herramientas en línea, a partir de interfaces intuitivas que no requieren el conocimiento de lenguajes de programación. Para ello se utilizó una nueva herramienta denominada “Escenario de decisión”, lanzada en 2019 y todavía en fase beta, basada en HTML5 y JavaScript ofrecida en el portal H5P (<http://h5p.org>). H5P es un portal de desarrollo comunitario diseñado con licencia del MIT (Massachusetts Institute of Technology). Es completamente libre y de código abierto [16].

Para la selección de esta herramienta de código abierto se consideró su facilidad de uso y su amplia potencialidad.

La construcción de un sistema de simulación implica varias etapas [8]:

- a. La definición del problema a resolver, la delimitación del sistema a ser simulado
- b. El diseño del modelo, partiendo de diagramas de flujo o bloques hasta el diseño experimental preliminar
- c. La traducción del modelo al lenguaje computacional
- d. La verificación del funcionamiento y comprobación de la validez del modelo
- e. La experimentación y puesta en práctica

Las simulaciones basadas en historias ramificadas poseen en general bajos requerimientos de elaboración. En ellas los estudiantes tienen múltiples opciones para su elección en una secuencia de sucesos que representan una situación dada o un fenómeno observable. Estos modelos suelen denominarse escenarios de decisión.

La herramienta H5P, posee una gran versatilidad en cuanto al tipo de medios y recursos que admite en su contenido (texto, imágenes y video) y ofrece dos tipos de interfaz intercambiables, que facilitan el diseño del material: una interfaz de vista previa que muestra al autor el resultado de su producción, tal como la verán luego los estudiantes y una interfaz específica que permite ir construyendo el esquema de navegación en forma de diagrama.

El escenario de decisión consta de una serie ramificada de situaciones de entrevista laboral representadas mediante videos subtítulos, filmados con actores reales, donde el supuesto entrevistador realiza preguntas que exigirán a la persona hipoacúsica entrevistada la elección de diferentes alternativas de respuesta, obteniendo una puntuación que se irá sumando de acuerdo con la valoración previamente asignada de cada respuesta.

De esta forma, se busca capacitar a las personas hipoacúsicas para la consecución de una entrevista laboral con resultados exitosos en los aspectos relacionados con las habilidades comunicacionales, el manejo de la información adecuada que deben aportar al entrevistador respecto a su discapacidad y la gestión de los aspectos emocionales.

3. Primeros resultados y Conclusiones

Se ha presentado la fundamentación, el diseño y las funcionalidades de un simulador orientado a entrevistas laborales con personas hipoacúsicas, si bien es adaptable a otras funcionalidades educativas para personas con discapacidades auditivas.

En el diseño se ha puesto énfasis en la comunicación, la interactividad, la simplicidad para el usuario y las actividades colaborativas.

Se diseñó el simulador y se señala la gran potencialidad que ofrece la herramienta online utilizada para trabajar con contenido multimedia de forma intuitiva y simple, además de permitir el diseño del simulador a partir de una interfaz de diagrama, indispensable para la correcta configuración de las rutas de decisión.

Se presentó la propuesta en Abril de 2019 ante el colegio de fonoaudiólogos de la provincia de Buenos Aires con una muy buena recepción. En ese contexto se trabajaron algunos lineamientos del prototipo. Actualmente se trabaja con el grupo de hipoacúsicos e implantados de la Plata quienes están revisando y probando la interfaz.

4. Trabajo futuro

Se trabaja en la validación del simulador para poder obtener los datos cuantitativos y cualitativos que requiere la tesis.

Se realizará la conexión de los datos registrados en H5P con el prototipo del simulador en línea de manera de tener una información integral de cada participante.

Una vez probado y validado el simulador sobre una plataforma de PC convencional, se trabajará sobre una versión para móviles, multiplataforma. Este simulador quedará abierto a modificaciones conforme a los resultados de satisfacción obtenidos de los usuarios a través de un cuestionario interactivo. Las mejoras podrán ser tanto educativas como técnicas conforme a los avances que a lo largo surjan con el tiempo en ambas áreas. Para lo cual se reunirá un equipo de especialistas.

Referencias

1. Sangrà, A., & Wheeler, S.: Nuevas formas informales de aprendizaje: ¿O estamos formalizando lo informal? RUSC. Universities and Knowledge Society Journal, 10(1), 107-115. (2013)
2. Bonsiepe, G.: Una Tecnología Cognoscitiva. 2000-08 (2009). Recuperado de: <http://guibonsiepe.com.ar/guiblog/text/>
3. Ausubel, D. P.: Adquisición y retención del conocimiento una perspectiva cognitiva No. 370.15 A9. (2002)
4. García-Carbonelli, A. y Watts F.: Perspectiva histórica de simulación y juego como estrategia docente: de la guerra al aula de lenguas para fines específicos. Ibérica 13:65-84 (2007)
5. Goleman, D., & Senge, P.: The triple focus: A new approach to education. Florence, MA: More Than Sound. (2014)
6. Amaya Franky, G.: La simulación computarizada como instrumento del método en el proceso de enseñanza y aprendizaje de la física, desde la cognición situada: ley de Ohm. Revista Electrónica "Actualidades Investigativas en Educación", vol. 8 No. 1 (2008)
7. Cabero-Almenara, J., Costas J.: La utilización de simuladores para la formación de los alumnos. Prisma Social, núm. 17, pp. 343-372 (2016)
8. Cataldi, Z.; Lage, F. y Dominighini, C.: Fundamentos para el uso de simulaciones en la enseñanza, Revista de Informática Educativa y Medios Audiovisuales Vol. 10(17), págs.8-16 (2013)
9. Brown, J., Collins, A. & Duguid, P.: Situated Cognition and the culture of learning. Educational Researcher, 18 (1), 33-42. (1989)
10. Jonassen, D.H.: El diseño de entornos constructivistas de aprendizaje. En Reigeluth, Ch., Diseño de la instrucción. Teoría y modelos. Madrid, Aula XXI Santillana, pp. 225-249. (2000)
11. Casanovas, I.: La utilización de indicadores didácticos en el diseño de simuladores para la formación universitaria en la toma de decisiones, TE&ET, Revista Iberoamericana de Tecnología en Educación y Educación en Tecnología, No. 2 (2007)
12. ONU: "Convención sobre los Derechos de las Personas con Discapacidad y Protocolo facultativo"; Fundación AEquitas y Colegio de Escribanos de la Provincia de Buenos Aires; FEN Editora Notarial. (2008)
13. Sacco, Antonio.: Estrategias para la utilización de tecnología en educación especial. Análisis de la implementación de las TICs en la atención a la diversidad. Propuestas para su eficaz aprovechamiento. (2009) Recuperado en marzo 2017 desde: <http://sedici.unlp.edu.ar/handle/10915/4165>
14. Fernández, C. A.: Diseño pedagógico del simulador SIPAD Trabajo fin de grado. Facultad de ciencias de la educación. Curso 2013/2014. (2013). Recuperado de: <http://www.terras.edu.ar/biblioteca/31/31IDE-LA-TORRE-saturnino-Cap3-Parte1-exito-error.pdf>

15. Ainciburu, María C.: Simulación en la Web2 y los lenguajes específicos, la comunicación profesional a través de la red Universidad de Siena (2009). Recuperado de: http://cvc.cervantes.es/ensenanza/biblioteca_ele/asele/pdf/20/20_0103.pdf
16. Buhu, A., Buhu L. Developing interactive elearning courses based on HTML5 for students in textile engineering, 9th International Conference on Education and New Learning Technologies, Barcelona, Spain. (2017)

Un modelo para la incorporación de computadoras livianas extensible en ambientes educativos

Nonino Fabian Alberto¹
Lic. Alejandro Sartorio²

Desarrollador e investigador del proyecto¹

fabian.nonino@gmail.com

Docente Coordinador²

alejandro.sartorio@uai.edu.ar

Universidad Abierta Interamericana - Rosario, Santa Fe, Argentina

Resumen. El objetivo del presente trabajo, es brindar a instituciones educativas, un modelo de infraestructura computacional que además de ser lo más económico posible, responda de manera eficiente a los requerimientos para el normal desempeño de las clases.

Esto se logró partiendo de una solución tecnológica ya existente en el mercado, como es el cliente delgado. Sobre esta se diseñó y construyó un modelo para el uso en ambientes educativos

Usando como cliente computadoras de placa simple Raspberry Pi y administrando las sesiones de los usuarios mediante un LDAP logramos crear un modelo económico y fácilmente adaptable a cada

Para mejorar la experiencia del usuario, realizaremos una selección del protocolo de comunicación ideal, adaptaremos la mejor solución para 2 escenarios planteados y evaluaremos su desempeño

Palabras Clave: cliente delgado, sbc, infraestructura económica, inclusión educativa, TICs en la educación.

1 Contexto

Desde el año 2004, la Municipalidad de la ciudad de Rosario con su proyecto *Munich*, promueve el uso de Software adaptado en todo el ámbito de la administración municipal “...En el 2004 la Municipalidad de Rosario comenzó la transición hacia el uso de Software Libre a nivel de escritorio, en lo que ha denominado Proyecto *Munich*. Éste fue planificado y gestionado internamente y contó con el aval del poder ejecutivo y el marco legal dispuesto a tal fin, utilizando los instrumentos legales provistos por la ordenanza N° 7787/2004 y el decreto N° 2833/2005...”¹.

Inspirados en *Munich*, nuestra idea nace en respuesta a la necesidad de promover el uso de las **TICs** en el ámbito educativo no sólo para enseñar, sino también para aprender y mejorar la experiencia de los usuarios. Favoreciendo además la

¹ <https://www.rosario.gov.ar/mr/softwarelibre/infraestructura-sl>

accesibilidad. Es por esto que comenzamos a desarrollar un modelo cliente-servidor (Thin Client/Cliente Delgado) lo más económico posible. Para lograrlo, utilizamos como dispositivos clientes "Single Board Computers" **SBC Raspberry Pi 3 model B** (Fig. 1) conectados a un servidor central, el cual por medio de sesiones de escritorio adaptadas intenta darle al usuario el acceso a los medios necesarios para el desarrollo de las clases.



Fig. 1. SBC Raspberry Pi 3 model B

2 Introducción

El ámbito de la educación viene redefiniendo sus prácticas dentro del aula en un contexto de avance de las nuevas tecnologías. Estas herramientas se han ido popularizando, pero muchos sectores ya sea por su situación socioeconómica o geográfica no pueden acceder aún. El presente trabajo trata de acercar a instituciones educativas que se encuentran en este contexto, un modelo de infraestructura computacional accesible desde el punto de vista económico, que pueda ser de utilidad a la hora de ser utilizado en las clases. Intentamos que tanto alumnos como docentes puedan acceder al uso de la tecnología y de esta forma reducir la **"brecha digital"**². Haremos foco también en la accesibilidad de alumnos con capacidades especiales y su inserción al mundo de las TIC.

Este modelo, reemplaza la típica configuración de clientes autónomos, donde cada uno de los terminales utilizan para el procesamiento de la información sus propios recursos tanto de Hardware; memoria RAM, disco rígido, microprocesador, etc; como también de Software. Este tipo de clientes "gruesos" o "fat", en ciertos casos puntuales continúan siendo recomendables, pero nuestro proyecto apunta a reemplazarlos por **Cientes Delgados** (Véase esquema básico de configuración en Fig. 2) apoyados en la tecnología **SBC** o Single Board Computer para luego personalizar el Sistema Operativo.

Un cliente liviano es básicamente una computadora que se configura de forma que pueda conectar con un servidor, el cual puede estar en un lugar remoto.

El servidor es en este caso el que realizará las tareas que los usuarios soliciten desde sus terminales:

² <http://www.labrechadigital.org>

- Procesamiento de datos
- Ejecución de aplicaciones
- Almacenamiento de información[4]

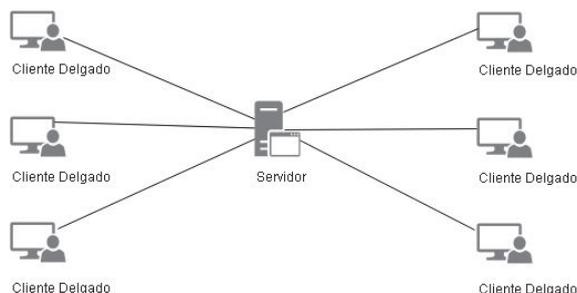


Fig. 2. Esquema básico de un modelo de Cliente Delgado

En el desarrollo del trabajo haremos referencia al conjunto de equipos de hardware que integran nuestro modelo como “infraestructura” y así también las posibles variantes que lo componen para poder adaptarlo a distintos escenarios.

Con el fin de montar una infraestructura que minimice la inversión monetaria en la compra de hardware, que pueda acercarse al desempeño que tiene un conjunto de hosts autónomos o gruesos, utilizaremos una SBC "Single Board Computer" como terminal. Para esta ocasión el modelo seleccionado será: Raspberry Pi 3 Model B. A ella se le conectará una PC-Clon que funcionará como servidor, en el mismo se ejecutarán todas las sesiones de los usuarios, las cuales previamente se personalizaron y gestionaron mediante un LDAP teniendo en cuenta que la solución debe aportar a los usuarios una experiencia similar a la que se tiene al utilizar computadoras autónomas.

3 Estado del Arte

A continuación se hace referencia a diferentes proyectos que fueron tomados como punto de partida a la hora de llevar adelante la realización del presente trabajo.

El trabajo número uno trata sobre la performance de los Clientes Delgados y las técnicas existentes para analizarla. Se enfoca principalmente en una técnica para el análisis de rendimiento llamada “Slow Motion Benchmarking” [1].

El trabajo número dos hace referencia al impacto de las TIC en el ámbito educativo y en la sociedad [2].

En el artículo número tres, se analiza la performance de gran cantidad de Clientes Delgados distribuidos en distintos países y funcionando a través de internet [3].

El trabajo número cuatro basa sus esfuerzos en la realización de un modelo al que denominaron “SRIDesk” el cual por medio de la virtualización de escritorios

adaptados y la aplicación de técnicas de optimización en la performance de la arquitectura, intenta brindar al usuario la mejor experiencia posible [4].

Tomando como referencia los trabajos anteriores, se presenta en la Tabla 1. una comparativa entre los diferentes trabajos y las características más importantes del presente trabajo de investigación.

Tabla 1. Comparativa de trabajos anteriores con el proyecto actual.

Trabajo	C1	C2	C3	C4	C5	C6	C7
Measuring ThinClient performance Using Slow-Motion Benchmarking	X				X		
TIC: Su uso como Herramienta para el Fortalecimiento y el Desarrollo de la Educación Virtual			X				
On the performance of wide-area thin-client computing	X				X	X	
SRIDesk: A Streaming based Remote Interactivity architecture for desktop virtualization system	X			X	X		
Trabajo Actual	X	X	X	X	X	X	X

C1: ThinClient C2: Entornos adaptados C3: TICS en la educación

C4: Single Board Computers (SBC) C5: Rendimiento C6: Eficiencia Energética C7: LDAP

De acuerdo a los estudios relevados, pueden apreciarse análisis enfocados tanto a clientes delgados como a las tecnologías de la información. Ninguna de estas investigaciones relaciona el modelo Cliente Delgado con las TICs en el ámbito educativo.

Nuestro trabajo trata de integrar los conceptos exhibidos en la Tabla 1. para conseguir un modelo conceptual que se ajuste a los requerimientos para fomentar el uso de las tecnologías de la información en la educación.

4 Análisis e Implementación

Tanto el análisis como la posterior implementación del modelo propuesto (ver Fig.3.) se realizaron por etapas. La primera etapa corresponde a la Selección del protocolo de comunicación entre el cliente/servidor. Luego se procedió a seleccionar el hardware involucrado, al hacerlo tuvimos en cuenta algunos puntos fundamentales:

- Minimizar la inversión en la compra de hardware.
- Equipos fácilmente reemplazables.
- Incorporar equipos que puedan cumplir con los requerimientos del usuario.

En la tercera etapa se comenzó con la configuración de la solución, luego en la etapa cuatro se personalizaron las sesiones de los usuarios, agregando software para facilitar la enseñanza/accesibilidad.

Luego en la etapa cinco, hacemos referencia al rendimiento de la solución y a las técnicas que se utilizan comúnmente para optimizarla.

Al finalizar, analizamos los beneficios de la solución propuesta.

Modelo "SBC Para la Inclusión"

Referencias

-  **Cliente:** Raspberry Pi 3 Model B Operado por Usuario + Monitor y periféricos.
-  Cliente genera información desde su puesto de trabajo. La misma es enviada al servidor, pasando en primera instancia por un Switch.
-  La información ya procesada por el servidor se devuelve al usuario en pantalla.
-  Switch de Acceso para la comunicación entre los clientes y el servidor.
-  PC Clon: utilizada como servidor de sesiones que por medio del protocolo RDP son accedidas por los usuarios

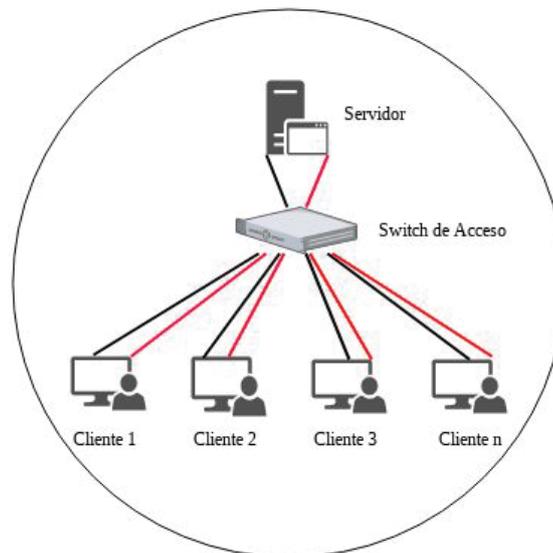


Fig. 3. Diagrama del modelo propuesto.

4.1 Protocolos para la comunicación entre cliente y servidor

El software utilizado por los clientes delgados consiste en un GUI (Interfase gráfica de usuario), un agente de acceso (por ejemplo RDP, PCoIP, ICA, etc), un web browser local, un set básico de utilidades locales y en algunos casos cuenta con emulaciones por terminal.

Existen una gran cantidad de protocolos para la comunicación entre el terminal y el servidor principal, entre ellos los más destacados son:

- RDP
- ICA
- RFB
- NX
- X Windows System

Seleccionamos para nuestra solución RPD “Protocolo de Escritorio Remoto” el cual está basado en una extensión de los estándares de la familia T-120. Es un protocolo multicanal que permite separar canales virtuales para llevar datos de presentación, comunicación de dispositivos seriales, información de licencias, datos fuertemente encriptados (actividad del mouse y teclado) y más.

RDP es una extensión del protocolo central "T.Share", tiene muchas otras capacidades, como las características arquitectónicas necesarias para admitir multipunto (sesiones múltiples). Es un protocolo que tuvo varias revisiones y que actualmente se encuentra en vigencia; su estabilidad es sin dudas una de las características por lo que lo elegimos. Otro punto que nos pareció atractivo es la posibilidad que tiene el protocolo para admitir sesiones múltiples o multipunto.

4.2 Gestionar de usuarios

A la hora de realizar la gestión de los usuarios, se decidió por la implementación dentro del protocolo LDAP

El protocolo LDAP está diseñado para gestionar usuarios y privilegios de accesos de gran cantidad de Computadoras y sistemas operativos de manera centralizada, remota y sencilla.

Se optó por OpenLDAP, ya que es una alternativa de código abierto, la cual se ajustó a la necesidad del sistema, de manera que podamos gestionar accesos y usuarios en esta arquitectura, pudiendo incluso, nuclear la administración varios centros de cómputos, que residan en el mismo establecimiento o geográficamente distantes.

4.3 Selección de Hardware

El equipo más importante dentro de nuestro proyecto y hablando en términos de eficiencia, es la SBC Raspberry Pi 3 modelo B.

Los motivos principales por los que seleccionamos este equipo fueron: gran potencial en muy poco espacio, su costo es ínfimo, lleva años en el mercado; lo cual demuestra que conoce el nicho en el cual ofrece sus productos. La comunidad que desarrolla software para la misma es sumamente grande y se encuentra documentación de todo tipo/para cualquier tipo de proyecto.

Para el servidor principal se utilizará una pc clon.

4.4 Funcionamiento

Hablaremos en primer lugar del core de la arquitectura, es decir; la PC Clon. En ella se configuró Windows Server 2016 educacional con el protocolo RDP habilitado para que los usuarios puedan loguearse remotamente. Para esta primera implementación, se lanza desde un Raspbian³ (el cual estará presente en las SD de las terminales) un aplicativo como rdesktop, que permita por medio del protocolo RDP iniciar sesión al usuario en el servidor remoto.

Dentro del servidor Windows Server, se instala un servidor DHCP para asignar direcciones IP a los terminales clientes. Decidimos hacerlo de esta forma para facilitar la administración y el posterior escalamiento de la red. Además se agrega en el mismo, un servidor DNS para que todos los recursos locales puedan ser accedidos por nombre.

³ <https://www.raspberrypi.org/downloads/raspbian>

4.5 Personalización de los entornos

Para demostrar la adaptabilidad del proyecto, a continuación se plantean 2 escenarios diferentes donde implementar la solución.

Escenario 1

Aula informática con 10 puestos de trabajo que consten de acceso a internet, recursos compartidos, paquete office, herramientas de desarrollo de software y diseño.

Escenario 2

Aula informática con 25 puestos de trabajo, que consten de acceso a internet, recursos compartidos, paquete office y herramientas para personas invidentes.

Solución:

Software. A la hora de instalar el servidor, donde los clientes delgados van a ir a buscar y procesar las aplicaciones necesarias, se prepara una imagen ya desde su instalación para ambos escenarios, teniendo todos los recursos de software ya integrados, solo resta volcar esa imagen en el servidor escogido en cada escenario.

Se creó un repositorio de software Libre, en el que incluimos software para facilitar la accesibilidad a la información, entre ellas: **NVDA**⁴ el cual tiene la capacidad de leer la información de la pantalla y es de gran ayuda para las personas no videntes, como característica adicional permite transformar texto a Braille si se le conecta un “Braille Display”. **Dasher**⁵ permite escribir a través de un sistema predictivo basado en el movimiento del puntero del ratón. **Lumisonic**⁶, aplicación que “visualiza” el sonido en tiempo real de una forma en que las personas con discapacidad auditiva pueden interpretar.

Creamos también una interfaz gráfica para que los docentes puedan incluir cualquier tipo de software que favorezca la accesibilidad. En la misma se incluyen gran parte de los que presenta la **Athena free AT Software Inventory**⁷

Hardware. Los puestos de trabajo irán equipados con raspberry Pi 3, no importa el escenario, solo hay que abordar la cantidad requerida en cada caso.

El punto a evaluar son las capacidades computacionales del servidor requeridas en cada escenario.

⁴ <https://www.nvaccess.org/>

⁵ <http://www.inference.org.uk>

⁶ <http://soundandmusic.org/projects/lumisonic/>

⁷ <http://access.uoa.gr/ATHENA/eng/pages/home>

Tabla 1. A continuación se presenta una tabla comparativa entre el modelo tradicional y el propuesto en este trabajo, en ambos escenarios, evaluando costos, tanto monetario como en función del tiempo que insume cada tarea.

Tareas	Escenario 1				Escenario 2			
	Modelo Tradicional		Modelo Propuesto		Modelo Tradicional		Modelo Propuesto	
	C1	C2	C1	C2	C1	C2	C1	C2
T1	150.000	0	100.000	0	375.000	0	250.000	0
T2	0	0	17.000	0	0	0	40.000	0
T3	0	10	0	1.5	0	25	0	4
T4	0	0	0	0.25	0	0	0	0.25
T5	800	5	800	5	800	12	800	12
Totales	150.800	15	117.800	6.75	375.000	37	290.800	16.25

C1: Costo Monetario (\$) **C2:** Tiempo que insume la tarea (Hs) **T1:** Hardware puestos de trabajo **T2:** Hardware servidor **T3:** Instalación software de puestos de trabajo **T4:** Instalación software servidor **T5:** Montaje de infraestructura

4.5 Rendimiento

La gran mayoría de las herramientas que existen para la realización de análisis/medición de rendimiento, fueron creadas para probar tanto PCs autónomas o Servidores. En el modelo propuesto, el procesamiento de la información no se realiza en el cliente, sino que se lleva a cabo en el servidor central; en consecuencia, bajo condiciones normales de funcionamiento total de ambos escenarios, se realizaron mediciones tanto de uso de cpu como de consumo de memoria ram en dichos servidores

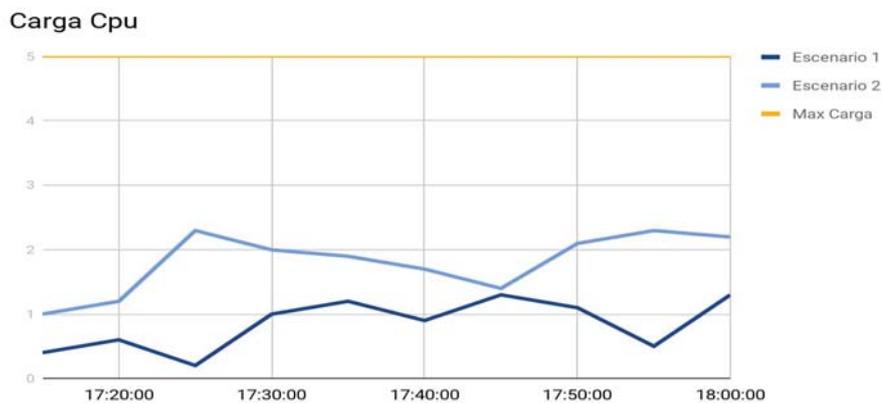


Fig. 4. Gráfico comparativo de carga de Cpu de los servidores de ambos escenarios, con todos sus puestos de trabajo funcionando

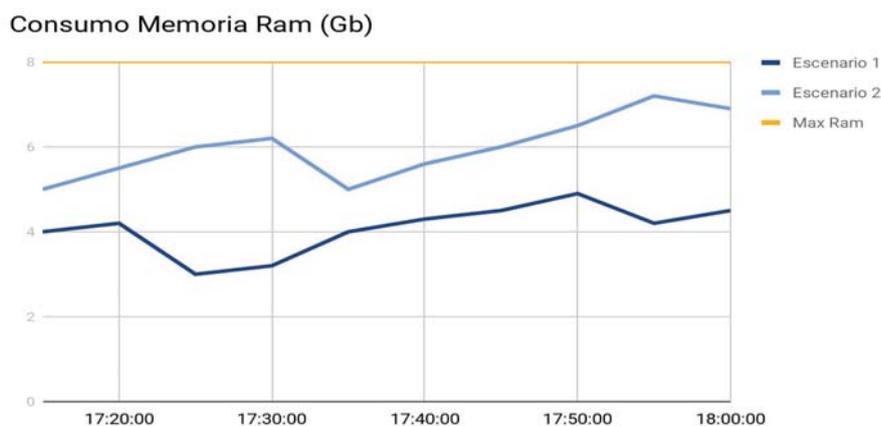


Fig. 5. Gráfico comparativo de uso de memoria Ram de los servidores de ambos escenarios, con todos sus puestos de trabajo funcionando

5 Conclusión

La infraestructura propuesta, es sumamente flexible y permite incorporar gran cantidad de nuevas terminales con un esfuerzo mínimo (fácilmente **escalable**). Esto va acompañado al uso eficiente de los recursos informáticos con los que se cuenta.

Este modelo reduce considerablemente los costos y tiempos de implementación, sea el escenario que sea (ver Fig.3.)

Partiendo de los escenarios planteados anteriormente, vemos que esta propuesta reduce un 21% el costo monetario del escenario 1 y un 28% el costo monetario del escenario 2 y a razón de un 55% el tiempo de implementación en ambos.

Esto demuestra la adaptabilidad del modelo frente a 2 escenarios diferentes, manteniendo sus virtudes

Teniendo en cuenta los resultados de las pruebas que realizamos durante el desarrollo de nuestro modelo, podemos concluir en que la solución propuesta es una alternativa sumamente interesante con múltiples beneficios ya sea en lo **económico** por el valor del hardware involucrado, en la **seguridad de la información** al tener un solo equipo que contenga la información a securizar y por sobre todo adaptativa a cualquier escenario posible sin que sus beneficios se vean afectados

BIBLIOGRAFÍA

- [1] Nieh, Jason, S. Jae Yang, and Naomi Novik. "Measuring thin-client performance using slow-motion benchmarking." *ACM Transactions on Computer Systems (TOCS)* 21.1 (2003): 87-115.
- [2] Rosario, Jimmy. "TIC: Su uso como Herramienta para el Fortalecimiento y el Desarrollo de la Educación Virtual." *DIM: Didáctica, Innovación y Multimedia* 8 (2006).
- [3] Lai, Albert M., and Jason Nieh. "On the performance of wide-area thin-client computing." *ACM Transactions on Computer Systems (TOCS)* 24.2 (2006): 175-209.
- [4] Wu, Jiewei, et al. "Sridesk: A streaming based remote interactivity architecture for desktop virtualization system." *Computers and Communications (ISCC), 2013 IEEE Symposium on. IEEE, 2013.*
- [5] Schmidt, Brian K., Monica S. Lam, and J. Duane Northcutt. "The interactive performance of SLIM: a stateless, thin-client architecture." *ACM SIGOPS Operating Systems Review. Vol. 33. No. 5. ACM, 1999.*
- [6] Jain, Vipul K., Bhaskar Peddinti, and James Molenda. "Power optimization on a thin client device." U.S. Patent Application No. 13/071,622.
- [7] Bertolín, Javier Areitio. *Seguridad de la información. Redes, informática y sistemas de información.* Editorial Paraninfo, 2008.
- [8] Tanenbaum, Andrew S. *Redes de computadoras.* Pearson Educación, 2003.
- [9] Forouzan, Behrouz A. *Behrouz A. Transmisión de datos y redes de comunicaciones.* McGraw-Hill, 2007.
- [10] Tso, Fung Po, et al. "The glasgow raspberry pi cloud: A scale model for cloud computing infrastructures." *Distributed Computing Systems Workshops (ICDCSW), 2013 IEEE 33rd International Conference on. IEEE, 2013.*
- [11] Partridge, Craig, and Robert M. Hinden. *Version 2 of the reliable data protocol (RDP). No. RFC 1151. 1990.*
- [12] <http://caeti.uai.edu.ar/lineas.aspx>
- [13] Bustamante, Fernando Prieto, Carlos Andrés Peña Guzman, and Juan Diego Lopez Vargas. "ANÁLISIS DE LA APLICACIÓN DEL GREEN IT EN LAS ORGANIZACIONES."
- [14] Petrov, Nikola, et al. "Examples of Raspberry Pi usage in Internet of Things." (2016): 112-119. <http://eprints.fikt.edu.mk>

Una Solución Tecnológica para el uso de TIC en escuelas rurales aisladas

Sergio Rocabado¹, Diego Rodríguez², Nicolas Lera², Carlos Cadena³

¹CIUNSa – Consejo de Investigación de la Universidad Nacional de Salta

²Facultad de Ciencias Exactas de la Universidad Nacional de Salta

³INENCO – Instituto de Investigaciones en Energía No Convencional (UNSa - CONICET)

¹srocabado@di.unsa.edu.ar, ²droduro@di.unsa.edu.ar, ³cadencinenco@gmail.com

Resumen. La brecha digital puede ser definida en términos de la desigualdad de posibilidades que existen para acceder a la información, al conocimiento y la educación mediante las TIC. El desafío es poder llevarlas a escuelas rurales aisladas que no disponen de acceso a la electricidad.

Se presenta una solución tecnológica para brindar acceso a información digital a estas mediante el uso de dispositivos móviles abastecidos con energía solar fotovoltaica.

La propuesta se sustenta en la reducción del consumo energético en los dispositivos móviles y en las comunicaciones y en un correcto aprovechamiento de la energía solar fotovoltaica disponible para la recarga de baterías. Fue utilizada para realizar experiencias educativas en escuelas rurales aisladas del Noroeste Argentino, haciendo posible que los alumnos se nutran de estrategias de aprendizaje mediadas por TIC y contribuyendo a reducir la brecha digital existente entre los alumnos de los centros urbanos y rurales.

Palabras Clave: Eficiencia energética, TIC, Energía solar, Dispositivos móviles, Escuelas rurales

1. Introducción

La integración de las TIC (Tecnologías de Información y Comunicación) en el sistema educativo argentino no es un fenómeno nuevo. Con distintas modalidades, en las dos últimas décadas se han incorporado dispositivos y recursos tecnológicos tanto para el uso pedagógico como para la gestión y la administración del sistema educativo. Al respecto, es posible sostener que actualmente existe un fuerte consenso acerca de la necesidad de universalizar el acceso a las nuevas tecnologías para promover la inclusión en la cultura digital y modificar las pautas tradicionales que rigen los procesos de enseñanza y aprendizaje.

En los últimos años y cada vez con más fuerza aparece en la agenda el tema de las TIC en educación. El espacio escolar puede constituir un ámbito de reducción de brechas digitales. Cada vez con mayor rapidez se abren distancias entre las culturas de comunicación de la escuela y las culturas de comunicación que utilizan los niños y los adolescentes. Estas brechas se profundizan en escuelas de ámbitos urbanos respecto a escuelas rurales, al mismo tiempo que se han reforzado distancias en los resultados de aprendizaje de los sectores más pobres respecto de la población escolar de las capas medias y altas. Las posibilidades de acceso y de uso de TIC abren otro tipo de brechas

que pueden tener efectos en la integración y cohesión social. Es por ello que las decisiones en torno a TIC hoy están relacionadas a cuestiones que hacen a la equidad y a la justicia social [1].

Ante estos cambios, decidir políticas de TIC con criterios de justicia implica priorizar los ámbitos rurales y suburbanos desfavorecidos, más desprotegidos en términos educativos y más desconectados.

En relación con la situación de TIC en escuelas rurales, se puede constatar con evidencia directa e indirecta falencias que hay que superar. Si bien hubo crecimiento en la dotación de TIC en zonas rurales, el mero equipamiento no alcanza para promover cambios. Además, estos avances aún son insuficientes e inequitativos en la comparación con la situación de escuelas urbanas. Otra carencia es la poca evidencia en torno a aquellas acciones exitosas que muchas escuelas aplican en relación con TIC, insumos necesarios para la construcción de políticas más efectivas [2].

Queda clara la necesidad de la implementación de las TIC en el sistema educativo nacional teniendo presente la desventajosa situación de las escuelas rurales.

En este contexto, en la Argentina se ejecutó el Proyecto de Mejoramiento de la Educación Rural (PROMER I). Las propuestas técnicas son las descritas en el "Documento para la Educación Rural 2006-2010". Incluye a todas las escuelas rurales del país y se ejecuta a partir de Convenios Bilaterales celebrados entre el Ministerio de Educación y las autoridades provinciales. Posteriormente, se aprobó el PROMER II [3] que apunta entre sus principales objetivos a reducir las tasas de repitencia en aulas de zonas rurales y aumentar la inscripción y las tasas de terminación de los estudios en la educación secundaria en las zonas rurales de Argentina.

Como resultado de la política educativa para las zonas rurales de la Argentina, en los últimos diez años y con la aplicación del PROMER I y II, se observa un achicamiento de la brecha entre las zonas urbanas y rurales si se comparan los porcentajes de asistencia de niños y niñas a la escuela.

El proyecto da prioridad a aquellas escuelas ubicadas en zonas rurales dispersas o remotas, donde casi el 40 por ciento de la población tiene al menos una de las necesidades básicas insatisfechas.

Si bien los proyectos PROMER I (finalizado) y PROMER II (en ejecución) tienen objetivos claros para mejorar la educación en las áreas rurales, debe mencionarse que en ningún punto se plantea la incorporación de las TIC.

Las zonas rurales y en particular las zonas rurales aisladas se caracterizan, entre otros aspectos, por su baja densidad demográfica, cobertura de red celular limitada y carencia de servicio de distribución de energía. En este contexto, el proceso enseñanza aprendizaje mediado por TIC es sumamente escaso debido al elevado consumo de energía de los equipos computacionales.

En este artículo se presenta una solución diseñada para posibilitar la enseñanza mediada por TIC en escuelas rurales recurriendo a una alternativa viable sustentada en que los dispositivos móviles presentan un bajo consumo energético y permiten el aprovechamiento de energías renovables (solar fotovoltaica).

2. Solución tecnológica

En su revisión literaria sobre el impacto de los ambientes educativos, el profesor de la Universidad de Durham Steve Higgins destaca una evidencia clara de que

"ambientes de aprendizaje extremadamente pobres tienen un efecto negativo en los estudiantes y el personal docente", así como que, al mejorarlo, "se obtienen beneficios significativos". La calidad interna del aire, ventilación y confort térmico, iluminación, acústica y el tamaño de la escuela y de las aulas son algunos de los atributos espaciales que pueden incidir sobre el rendimiento de alumnos y profesores [4]. El concepto por el cual la telefonía celular puede fortalecer la enseñanza cabe aquí perfectamente y es por ello que se debe poner mucho esfuerzo en hacer eficiente su uso. Ciertas zonas rurales de la región que nos ocupa (fundamentalmente las provincias de Salta y Jujuy) son carentes de energía de red, pero además es poco probable que en un futuro cercano obtengan el acceso. Este hecho preocupante, tiene su atenuante porque disponen de un abundante recurso solar: días despejados, muchos días de sol por año, buena intensidad, claridad atmosférica, etc. Dicho de otro modo, muchas de las escuelas en las regiones de altura poseen niveles de radiación solar (expresado en horas pico de sol) entre los mayores del mundo. Esto tiene una consecuencia directa, ya que se puede obtener energía, no sólo para la recarga de los móviles tanto de docentes como de alumnos, sino también para mejorar las señales de la red de telefonía celular, con el empleo de amplificadores. Para aprovechar la energía solar disponible en estas zonas se utilizan sistemas fotovoltaicos, que generalmente son pequeños y entregan una cantidad reducida de energía, por lo que el uso de esta energía debe ser optimizado, esto se consigue reduciendo el consumo energético en los dispositivos móviles y en las comunicaciones.

2.1 Reducción del consumo energético en los dispositivos móviles

Se plantea una metodología para reducir el consumo de energía de los dispositivos móviles y facilitar su utilización en zonas rurales aisladas con limitaciones de energía. La metodología, desarrollada por nuestro grupo de investigación y publicada en [5], permite seleccionar las aplicaciones móviles de menor consumo energético para realizar tareas de uso frecuente en zonas rurales (eficiencia energética). A continuación se resumen los pasos de la metodología desarrollada:

1. Relevar las tareas de uso frecuente en zonas rurales, a modo de ejemplo se pueden mencionar: Navegación Web, Mensajería instantánea, Acceso a contenidos educativos.
2. Asignar aplicaciones móviles para ejecutar las tareas, a continuación se mencionan ejemplos de tareas y aplicaciones: Navegación Web (Google Chrome, Mozilla Firefox, Opera Mini), Mensajería instantánea (Whatsapp, Facebook Messenger, Google Hangout), Acceso a contenidos educativos (Apprendiendo!).
3. Determinar el consumo energético de las aplicaciones para realizar la tarea asignada. La medición se realizó utilizando "Trepn Profiler" siguiendo el procedimiento publicado en [6].
4. Establecer el consumo energético de las tareas. Para cada tarea se selecciona la aplicación que consuma menos energía para ejecutarla (eficiencia energética).
5. Agrupar las tareas en perfiles en función de los requerimientos informáticos de los pobladores y/o visitantes de la zona.
6. Determinar el consumo energético de cada perfil, en función del consumo de las tareas pertenecientes al perfil.

En la Figura 1 se ilustra un ejemplo que agrupa tres tareas: Navegación, Mensajería y Acceso a contenidos educativos. Se observa que las aplicaciones de menor consumo energético son: Chrome para la navegación (1549mWh), Whatsapp para mensajería (1612 mWh) y Aprendiendo! para acceso a contenidos educativos (950mWh). Además, el total de consumo para el perfil es de 4111 mWh.



Figura 1. Perfil de consumo energético

2.2 Reducción del consumo energético en las comunicaciones

Las zonas rurales aisladas del NOA disponen de servicios de telefonía celular con cobertura limitada y ancho de banda reducido. En estas condiciones, las aplicaciones que hacen uso de Internet se vuelven inestables, generando un gran número de retransmisiones que degradan el rendimiento e incrementan drásticamente el consumo de energía en el dispositivo móvil.

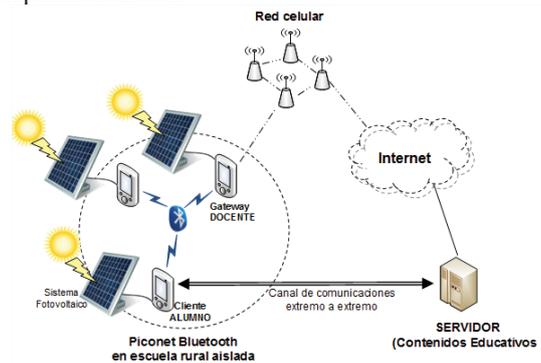


Figura 2. Escenario de comunicaciones

Para subsanar estos inconvenientes, se puede disponer de un escenario de comunicaciones (Figura 2) que aprovecha mejor el ancho de banda disponible y reduce el consumo de energía sin afectar el correcto funcionamiento de las aplicaciones móviles [7]. En este escenario se conecta una Piconet Bluetooth Low Energy [8] desplegada en zona rural a un servidor a través de la red celular. Los dispositivos móviles (nodos de la Piconet) se conectan a un servidor utilizando un

canal lógico extremo a extremo. El tráfico entre el nodo móvil y el servidor se gestiona a través de uno de los nodos que actúa como pasarela de comunicaciones entre la Piconet y la red celular. Este nodo es el encargado de enviar los paquetes de datos hacia los routers de la red celular, desde donde y a través de Internet son direccionados y entregados al servidor.

2.3 Aprovechamiento de la Energía Solar Fotovoltaica

La energía solar es un recurso limpio y renovable que puede ser aprovechado mediante el uso de paneles fotovoltaicos para transformar la radiación solar en electricidad.

Si bien los sistemas fotovoltaicos actualmente son utilizados en muchas zonas aisladas [9], generalmente en comunidades rurales donde tuvo intervención el PERMER (Proyecto de Energías Renovables en Mercados Rurales) [10], el aprovechamiento de la energía solar para la carga de dispositivos móviles no está lo suficientemente difundido. En este trabajo se introduce la aplicación de esta tecnología en escuelas rurales, contribuyendo a mejorar las posibilidades de comunicación de la comunidad educativa de las mismas.

Como parte de la solución se presenta un cargador solar de tamaño y peso reducido que interactúa con el dispositivo móvil. Los datos de radiación solar y temperatura que entrega el cargador permiten estimar la energía disponible para el dispositivo. Esta energía se administra planificando la ejecución de aplicaciones en función de los requerimientos energéticos.

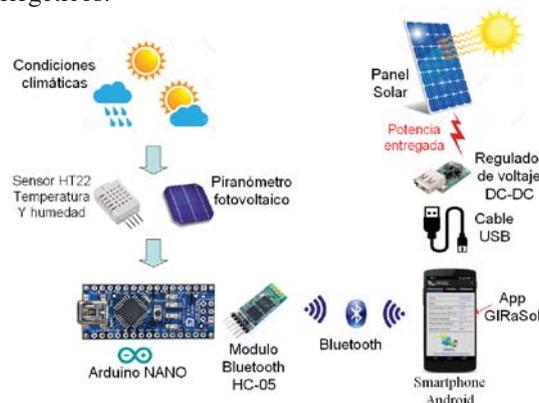


Figura 3. Prototipo de Sistema Fotovoltaico

El prototipo (Figura 3) se construyó a partir de un panel fotovoltaico de tamaño y peso reducido al cual se le agregaron sensores de radiación solar, temperatura y humedad y una placa de control (Arduino). En esta placa se ejecuta un programa de control que periódicamente recolecta los valores de los sensores que tiene conectados (temperatura y radiación solar) y los almacena en una memoria flash para ser enviados al dispositivo móvil cuando sean requeridos.

Para comunicar el dispositivo móvil con el cargador se desarrolló la aplicación móvil GIRaSol (Gestión Inteligente de Recarga Solar), esta aplicación se comunica a través de Bluetooth con la placa Arduino instalada en el cargador, obtiene y utiliza los datos almacenados en la memoria flash para estimar la cantidad de energía que el cargador solar puede suministrar al dispositivo móvil.

Antes de utilizar Girasol se deben definir perfiles de aplicaciones y caracterizar su consumo energético utilizando la metodología presentada en el apartado anterior. Los perfiles a utilizar en el dispositivo se cargan en Girasol mediante un formulario de carga de datos.

Para explicar el funcionamiento de la herramienta realizamos una prueba del prototipo de cargador solar y de la aplicación Girasol:

- En primer lugar, se caracterizaron dos perfiles: el PERFIL 1 con 3 aplicaciones (Chrome, Whastapp y Aprendiendo!) con un consumo de 4,11Wh y el PERFIL 2 con 2 aplicaciones (Chrome, Whastapp) con un consumo 3,16 Wh. Estos datos fueron cargados en Girasol (Figura 4b).
- A continuación se conectó el dispositivo móvil con la placa Arduino del sistema fotovoltaico. Girasol detectó la conexión, recolectó los datos almacenados en la memoria flash de Arduino y estimó la potencia que el panel fotovoltaico podía entregar. En la Figura 4a se muestra una captura de pantalla con la información mostrada por Girasol, además de la Potencia se observa información meteorológica recolectada por los sensores (Radiación, temperatura y humedad).
- Finalmente, se verificaron en girasol las franjas horarias en las que se puede ejecutar un perfil de aplicaciones (Figura 4c). En la prueba realizada se observa que el perfil de aplicaciones 1 se puede ejecutar de 11 a 16



Figura 4a. GIRA SOL – Información

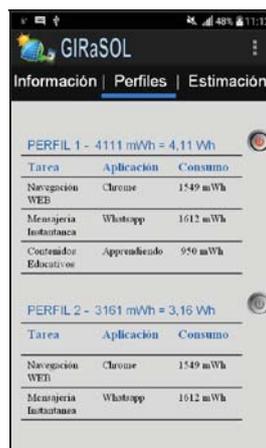


Figura 4b. GIRA SOL – Perfiles



Figura 4c. GIRA SOL – Estimación

Aspectos que se pueden destacar del prototipo:

- Girasol posibilita obtener la siguiente información: Potencia entregada por el sistema fotovoltaico, Franjas horarias en las que se pueden ejecutar los perfiles de aplicaciones, Tiempo de carga de la batería, Temperatura y humedad ambiente.
- No se utilizan baterías externas, el sistema fotovoltaico se conecta en forma directa al celular; de esta forma se evita la contaminación producida por las baterías y se eliminan tareas de mantenimiento que son costosas y difíciles de realizar en zonas aisladas.
- El voltaje que entrega un sistema fotovoltaico, generalmente, es superior al voltaje requerido por la batería del celular. En nuestro prototipo el panel fotovoltaico fue modificado para conectar las celdas en paralelo, esto genera un incremento de la corriente y la disminución del voltaje a la salida del panel, y en consecuencia se

consigue un incremento de potencia entregada al celular [11]. El sistema fotovoltaico modificado optimiza la potencia entregada al celular y posibilita la ejecución de perfiles de aplicaciones que serían descartados con el panel original.

3. Implementación de la Solución tecnológica

Equipamiento necesario:

- Para el docente: Un celular con Chipset Snapdragon y Sistema Operativo Android y un sistema fotovoltaico con placa de control Arduino (prototipo presentado en el apartado 2.3).
- Para cada alumno: Un celular con Chipset Snapdragon y Sistema Operativo Android y un sistema fotovoltaico simple (sin placa de control ni sensores). El panel solar del sistema fotovoltaico simple tendrá las mismas dimensiones del panel utilizado por el docente.

La implementación se divide en 5 etapas que se resumen a continuación:

1. Relevamiento de aplicaciones necesarias para desarrollar la clase en la escuela rural. Las aplicaciones libres se pueden instalar desde el Marketplace de Google. Si se requiere acceder a contenidos educativos alojados en un servidor remoto se recomienda desarrollar una aplicación específica.
2. Definición de perfiles. Agrupar en perfiles las aplicaciones relevadas y caracterizar el consumo energético para cada perfil. Esta tarea será desarrollada por personal técnico utilizando la metodología presentada en 2.1.
3. Capacitación al docente. El personal técnico debe capacitar al docente de la escuela en los siguientes temas: Principios básicos de la energía solar fotovoltaica, uso del sistema fotovoltaico, uso de la aplicación GIRA Sol, consumo energético de los perfiles de aplicaciones, configuración de una piconet Bluetooth y conexión del dispositivo móvil a la red celular.
4. Planificación de la clase. El docente se traslada a la escuela rural y procede a:
 - Instalar el sistema fotovoltaico y direccionar el mismo para optimizar la ganancia.
 - Conectar (por Bluetooth) el celular a la placa Arduino del cargador solar.
 - Ejecutar girasol y determinar las bandas horarias para la ejecución de perfiles.
 - Definir horarios y contenidos de su clase en función de las bandas horarias en las que se puede ejecutar las aplicaciones que necesita utilizar.
5. Desarrollo de la clase. El docente realiza las siguientes acciones:
 - Verificar la carga de los equipos de alumnos. En caso de ser necesario conecta el dispositivo del alumno a un panel fotovoltaico simple y orienta el mismo para mejorar la ganancia.
 - Configurar la piconet Bluetooth para comunicar los dispositivos: El celular del docente actúa como maestro y los dispositivos de los alumnos como esclavos (hasta 7), de ser necesario se arman grupos de alumnos por dispositivo.
 - Dictar la clase con acceso mediado por dispositivos móviles, a contenidos educativos en Internet y/o recursos m-learning almacenados en servidores remotos.

La solución tecnológica fue utilizada exitosamente para realizar experiencias educativas, mediadas por TIC, en escuelas ubicadas en zonas rurales aisladas del Noroeste de Argentina (NOA). En la Figura 5 se ubica geográficamente a las escuelas, marcándolas con un círculo rojo.

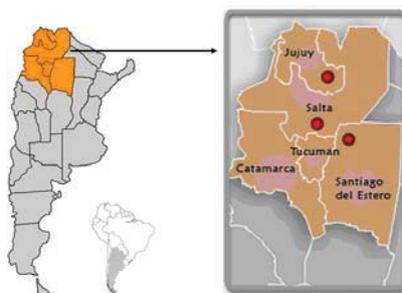


Figura 5. Ubicación geográfica de las escuelas rurales

En todas las experiencias, al finalizar se realizaron encuestas a docentes y alumnos, además de una entrevista personal al coordinador/director. Los resultados alcanzados, medidos a partir de las encuestas y entrevistas, fueron positivos y altamente significativos desde el punto de vista educativo.

La descripción de las experiencias realizadas y sus correspondientes resultados fueron presentados en las siguientes comunicaciones:

- Aprendizaje mediado por dispositivos móviles en escuelas rurales aisladas [12] y [13].
- Uso de TIC en comunidades educativas rurales [14].

4. Conclusiones

En las zonas rurales aisladas del país la instalación de redes eléctricas tiene un elevado costo debido a las distancias, a las dificultades de acceso y al escaso número de pobladores. El prototipo presentado apunta a solucionar esta carencia con una propuesta sustentable de bajo costo, haciendo uso de componentes económicos y energía renovable.

El uso de fuentes de energías renovables permite no solo llegar a escuelas rurales sin acceso a la energía eléctrica, sino que de forma indirecta introduce las energías renovables en la vida de estos pobladores, pudiendo concientizar a los mismos de las ventajas de su uso.

El equipamiento involucrado no utiliza baterías de plomo ácido, las cuales además de contaminantes, necesitan de mantenimiento, y de recambio luego del período de vida útil, cuestiones no menores debido a las distancias que existen a los centros urbanos próximos.

Las principales contribuciones de la propuesta son:

- Disminución de la brecha digital existente entre establecimientos educativos rurales y establecimientos urbanos
- Uso de una fuente de energía renovable (solar fotovoltaica) para la recarga de dispositivos móviles en escuelas rurales que no dispongan de servicio de energía

eléctrica. Docentes y alumnos de estas escuelas no tendrán necesidad de desplazarse a centros urbanos para recargar sus equipos.

- Mejoras en el uso de la tecnología disponible en zonas aisladas, teniendo en cuenta que algunos alumnos de estas escuelas son propietarios de equipos celulares que utilizan como reproductores de música o cámaras fotográficas y no como dispositivos de comunicación.
- Incremento de las posibilidades de comunicación de docentes y alumnos de escuelas rurales, permitiendo el acceso a aplicaciones de Internet tales como correo electrónico, mensajería y redes sociales.
- Mejoras en el proceso de enseñanza aprendizaje en comunidades escolares rurales aisladas, posibilitando a los docentes el empleo de las TIC y acercando las tecnologías de la información y comunicación a niños de estas comunidades.

Con base en los resultados alcanzados en experiencias realizadas en escuelas rurales del NOA, se puede concluir que el uso de dispositivos móviles abastecidos con energía solar fotovoltaica, se constituye como una alternativa tecnológica que posibilita el acceso a información digital a docentes y alumnos de escuelas ubicadas en zonas rurales aisladas. La solución tecnológica estudiada puede ser utilizada no sólo en zonas rurales aisladas del NOA sino también en otras regiones del país que se encuentren fuera del alcance de los centros de distribución de energía.

5. Trabajo futuro

Para continuar con esta línea de investigación se tiene previsto la implementación de Intranets educativas con recarga basada en energía solar, para posibilitar el acceso a contenidos educativos digitales a comunidades educativas rurales aisladas que no dispongan de acceso a redes celulares. Esta solución se sustenta en tecnologías de bajo consumo energético que permiten el aprovechamiento de energías renovables. Se utilizarán computadoras de placa simple (Raspberry) para montar los Servidores, las aplicaciones y contenidos educativos se instalarán en los servidores que serán accedidos desde dispositivos móviles utilizando una Piconet Bluetooth Low Energy (Figura 6).

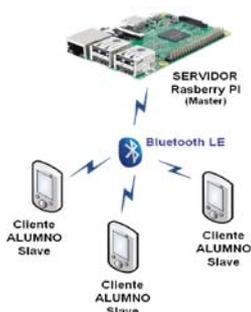


Figura 6. Piconet Bluetooth

Referencias

1. J. C. Tedesco, C. Steinberg, A. Tófaló. (2015). *Principales resultados de la Encuesta Nacional sobre Integración de TIC en la Educación Básica Argentina*. Programa TIC y Educación Básica. Informe general. Fondo de las Naciones Unidas para la Infancia (UNICEF).
2. E. Duro. (2008). *TIC y Justicia educativa, Las TIC: del aula a la agenda política*. Ponencias del Seminario internacional Cómo las TIC transforman las escuelas. IPE-UNESCO, Sede Regional Buenos Aires, UNICEF.
3. Ministerio de Educación, Cultura, Ciencia y Tecnología. (2015 - 2019). Segundo Proyecto de Mejoramiento de Educación Rural. <https://www.argentina.gob.ar/promer-ii>.
4. Higgins, Steven, Hall, Elaine, Wall, Kate, Woolner, Pam, & McCaughey, C. (2005). The Impact of School Environments: A Literature Review. *The Centre for Learning and Teaching-School Education, Communication and Language Science. University of Newcastle*.
5. Rocabado, Sergio; Cadena, Carlos. (2015). *Cargadores solares portátiles para el uso de dispositivos móviles en zonas rurales aisladas del NOA*. Paper presented at the ASADES 2015.
6. Rocabado, Sergio; Cadena, Carlos. (2015). *Uso de dispositivos móviles inteligentes en zonas rurales aisladas abastecidas con energía solar fotovoltaica*. Paper presented at the CACIC 2015.
7. Rocabado, Sergio; Sanchez, Ernesto; Diaz, Javier y Arias, Daniel. (2012). *Integración Segura de MANETs, desplegadas en zonas de recursos limitados, a Redes de Infraestructura*. Paper presented at the CACIC 2012, Bahía Blanca - Buenos Aires - Argentina. <http://sedici.unlp.edu.ar/handle/10915/23762>
8. SPECIAL INTEREST GROUP Bluetooth. (2010). Bluetooth v4.0 low energy (LE). from <http://www.bluetooth.com/Pages/Low-Energy.aspx>
9. Ottavianelli, Emilce; Ibarra Marcelo; Cadena Carlos. (2013). *Uso de indicadores sociales en estudios de factibilidad de instalación de sistemas solares para generación de electricidad en localidades rurales*. Paper presented at the XX Simposio peruano de Energía solar, Tacna – Perú.
10. Secretaria de Energía - Ministerio de Planificación. (2015). PERMER, Proyecto de Energías Renovables en Mercados Rurales. from <https://www.se.gob.ar/permer/>
11. Rocabado, Sergio; Cadena, Carlos. (2016). *Mini sistemas fotovoltaicos para el uso de dispositivos móviles en zonas rurales: Optimización de la potencia entregada y consumida*. Paper presented at the Congreso Brasileiro de Energía Solar - CBENS 2016.
12. Rocabado, Sergio; Cadena, Carlos; Herrerra, Susana. (2016). *M-learning en escuelas rurales*. Paper presented at the TE&ET 2016, Morón - Provincia de Buenos Aires - Argentina. ISBN: 978-987-3977-30-5. <http://sedici.unlp.edu.ar/handle/10915/53517>
13. Rocabado, Sergio; Herrerra, Susana. (2013). *M-learning en zonas de recursos limitados*. Paper presented at the TE&ET 2013, Santiago del Estero - Argentina. <http://sedici.unlp.edu.ar/handle/10915/27585>
14. Rocabado, Sergio; Cadena, Carlos; Ottavianelli, Emilce. (2017). *Propuesta para el uso de TDICs en comunidades educativas rurales. Una experiencia en escuelas albergue*. Paper presented at the Congresso Brasileiro de Informática na Educação, Recife - Brasil.

Valoración del grado de atención en contextos áulicos con el empleo de interface cerebro-computadora en el marco de la computación afectiva.

Jorge Ierache , Gonzalo Ponce, Ramiro Nicolosi, Iris Sattolo, Gabriela Chapperón

Instituto de Sistemas Inteligentes y Enseñanza Experimental de la Robótica (ISIER) Secretaria de Ciencia y Tecnología -Universidad de Morón
Cabildo 134, Buenos Aires, Argentina
jierache@unimoron.edu.ar

Abstract. Se presenta en este trabajo, una experiencia áulica que contempla el registro del grado de atención de un alumno durante una clase, en particular de algoritmos de programación. Se describe la arquitectura desarrollada durante la experimentación y se discuten los resultados obtenidos a través del registro de electroencefalograma (EEG) del alumno con BCI de NeuroSky, encuestas (SAM) y el registro de imágenes del pizarrón de la clase y del rostro del alumno, determinándose los momentos de mayor atención del alumno, contrastando estos resultados con el esperado por el docente.

Keywords: Computación Afectiva, Interfase Cerebro Computadora, Nivel de Atención, Tecnología en Educación, Tutoría Inteligente.

1 Introducción

El rápido crecimiento de la computación afectiva [1], como área de investigación emergente, se ha visto reflejado en numerosos trabajos y en distintas aplicaciones tales como seguridad, salud, marketing, robótica y educación. Estos sistemas afectivos deben ser capaces de reconocer y luego capturar los estados emocionales, procesarlos, para luego generar respuestas sobre estas emociones. Sobre el *reconocimiento* de los estados emocionales existen teorías fisiológicas, neurológicas y cognitivas de las cuales surgen diversos modelos para clasificarlos. En cuanto a *la captura* de estados emocionales, las fuentes de datos son diversas. Para ello se utilizan cámaras, dispositivos conocidos como interfases cerebro computadora (BCI), señales fisiológicas (medidas por electromiograma, electro-oculograma, actividad electrodérmica y electrocardiograma, detectores de ritmo cardíaco, entre otros). El uso de cámaras se sustenta en la postura en la que el rostro de la persona refleja la emoción. En educación trabajos como [2],[3] lo demuestran. En este sentido Ekman y Friesen [4] desarrollaron el sistema de codificación de acción facial (FACS) en el cual los movimientos individuales de los

músculos faciales son codificados, convirtiéndose en un estándar de clasificación de emociones. Otra manera de capturar estados emocionales es a través del registro del comportamiento en el uso del mouse, en la cadencia del tecleo, el seguimiento del movimiento ocular, y test a los participantes. En educación trabajos como [5] y [3] registran movimiento de mouse y teclado. Se puede diferenciar entre los sistemas unimodales -los que exploran una sola fuente de datos-, y multimodales -los que combinan dos o más fuentes de datos- en el trabajo [6] se realiza una revisión de estas dos modalidades. La neurociencia ha realizado avances recientes en imágenes cerebrales en un intento por mapear los circuitos neuronales que subyacen a la experiencia emocional [7], y está aportando evidencia a las discusiones sobre modelos dimensionales[8] donde la valencia y la excitación podrían ser apoyadas por distintas vías neuronales [9].Particularmente en el contexto de educación, las investigaciones recientes han proporcionado evidencia sólida de que las emociones afectan fuertemente a la motivación y el compromiso, y por lo tanto juegan un importante rol en el aprendizaje. En tal sentido, los trabajos se orientan tanto en la educación presencial como en la educación a distancia. Se presenta en la primera sección los modelos de emociones, en la segunda sección se discute el problema en orden a evaluar el nivel de atención de un alumno, en la tercera sección se presenta la solución propuesta acompañada de una arquitectura del sistema, en la cuarta sección se presenta una prueba realizada como ejemplo y se discuten los resultados obtenidos de una experiencia áulica que contempla el registro del grado de atención de un alumno durante una clase en particular de algoritmos de programación. Del alumno se registran: electroencefalograma (EEG) con BCI de NeuroSky, imágenes del rostro para ser empleadas en determinar su estado emocional, encuesta para capturar emociones SAM (Self-Assessment Manikin) una al comienzo de la experiencia, otra con imágenes IAPS, previo a la clase, y una más al finalizar clase. También se registra, para complementar el análisis, imágenes del pizarrón de la clase, y se graba el audio de esta. Al finalizar la clase se solicita al docente indique los momentos que requieren mayor atención por parte del alumno en la explicación del tema. Luego estos se contrastan con la información registrada por el sistema, para verificar si los puntos obtenidos por análisis del mayor nivel de atención determinado por el sistema en las capturas concuerdan con los rangos de momentos indicados por el docente. Finalmente, en la quinta sección se presentan las conclusiones y futuras líneas de trabajo.

1.1 Emoción

Desde el último medio siglo, se propusieron diferentes formas de modelar las emociones, representándolas desde diferentes perspectivas. Dos de ellas se utilizan ampliamente, una es el enfoque categórico y otra es el enfoque dimensional [10]. El enfoque categórico, desarrollado inicialmente por el psicólogo Paul Ekman [11], afirmó que había un conjunto de seis emociones básicas y universales que no están determinadas por las culturas, estas son: la alegría, el miedo, la tristeza, la ira, el disgusto y la sorpresa. Por otro lado, el enfoque dimensional establece que los estados afectivos se distribuyen en un espacio continuo cuyos ejes dimensionales indican una cuantificación de una característica. Uno de los modelos más aceptados es el Circumplex Model of Affectivity, o Modelo Circumplex de Afectividad, de James

Russell [8], [12], también conocido como circunplejo o modelo Arousal–Valence. Este es un modelo bidimensional, sus ejes son Excitación o Arousal (relajado vs. excitado) y Valencia o Valence (placer vs. disgusto).

1.2 Modelo de Russell

A pesar de que actualmente no existe un consenso acerca de lo que es la definición de emoción, si existe un acuerdo en que las emociones pueden clasificarse en tres dimensiones continuas: 1) **Valencia** (Valoración): un constructo bipolar, que va de agradable a desagradable, 2) **Excitación** (Arousal): que va de calmado a activado (o excitado), y 3) **Dominancia** (Control): valor que aparece con menos consistencia interna y se caracteriza por ser fuerte-débil, dominante-sumiso, control-controlado.

Existen distintos modelos que intentan reflejar estas emociones, el más utilizado es el presentado por Russell en el año 1980, el cual después de 30 años se sigue utilizando [8]. Este permite entender cómo un individuo puede regular sus emociones, y a esta actividad la representa como el resultado de dos vectores de fuerza, que empujan al sujeto de una situación emocional a otra más placentera y mejor valorada. La idea fundamental que transmite este modelo es que los estados emocionales están mejor representados en un círculo con dimensiones bipolares que en dimensiones independientes. Russell representa las emociones mediante un modelo espacial en el que los conceptos afectivos caen en un círculo en el siguiente orden, en el cual el número representa el vector o radial: placer o agrado (0), excitación (45), activación (90), angustia (135), disgusto (180), depresión (225), somnolencia (270) y relajación (315). La siguiente imagen muestra el modelo mencionado.

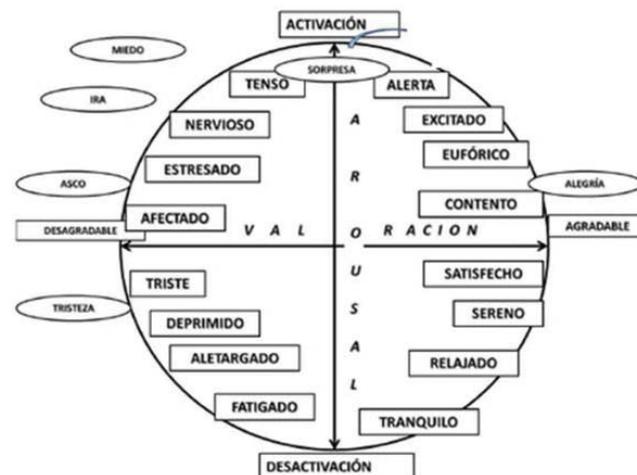


Fig. 1. Modelo de Russell

1.3 Imágenes IAPS

Uno de los conjuntos de estímulos estandarizados más utilizados en la investigación experimental es el International Affective Picture System (IAPS) [13]. El IAPS es una colección de más de 1000 fotografías en color -agrupadas en 20 conjuntos de aproximadamente 60 imágenes- que representan objetos, personas, paisajes y situaciones de la vida humana cotidiana. La decisión de estandarizar las imágenes del IAPS en función de las evaluaciones de los sujetos en las dimensiones afectivas de valencia, activación y dominancia está basada tanto en datos teóricos como experimentales [13].

1.4 Encuestas SAM

Otra herramienta que se utiliza para capturar emociones es SAM (por sus siglas en inglés Self-Assessment Manikin) propuesto por Lang en 1985. Tal como lo describen sus autores [14], SAM es un método no verbal fácil de administrar para evaluar rápidamente el placer (valencia), la excitación (arousal) y la dominancia asociadas con la reacción emocional de una persona ante un evento. SAM se extiende desde una figura sonriente y feliz hacia una figura con ceño fruncido e infeliz cuando se representa la dimensión de placer; y se extiende desde una figura excitada con los ojos abiertos hacia una figura relajada con sueño para la dimensión de excitación. La dimensión de dominancia representa cambios en el control, con variaciones en el tamaño de SAM, una gran figura indica el máximo control de la situación. Los valores dimensionales provistos por SAM no necesitan ser transformados. Existen dos escalas, una en el rango 1–5, que coincide con las cinco representaciones del personaje de izquierda (1) a derecha (5). La segunda escala permite agregar cuatro valores intermedios, de esta forma se obtiene un rango posible de 1–9; en este caso, no se suele dibujar al personaje nueve veces, sino que se dibuja el punto posible en el medio. Ver figura 2

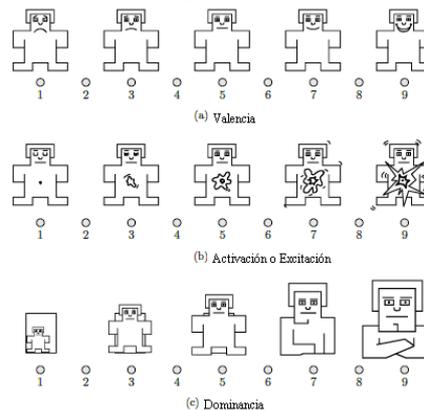


Fig. 2. SAM (Self-Assessment Manikin)

1.5 Interfases Cerebro Computadora (BCI)

En el campo de las interfases cerebro computadora se desarrollan desde hace tiempo dispositivos que traducen información neuronal de áreas determinadas del cerebro a datos que pueden controlar, software o hardware externo [15]. Los BCI se usan a menudo como dispositivos para asistir a personas con discapacidades motoras o sensoriales, video juegos, control de robots entre otros. En Trabajos anteriores [16], [17], [18] se exploró el potencial del equipo obteniendo las bio-señales eléctricas de los usuarios, generadas por los patrones de actividad de músculos faciales, los ojos y el cerebro. En trabajos recientes, se aprovecha su potencial para capturar distintas lecturas emocionales en contextos virtuales, [19], [20]. Se presentan trabajos relacionados en educación. En [21] se utilizó esta interfase para registrar la actividad cerebral de una clase de doce estudiantes de secundaria (simultáneamente) a lo largo de un semestre, durante las actividades regulares del aula para obtener patrones de la dinámica del grupo. En [22] se propone y evalúa un sistema multiagente, denominado NORA, que predice los atributos emocionales de los alumnos a través de las ondas cerebrales, dentro de un sistema de tutoría inteligente. Este sistema utiliza un EEG inalámbrico portátil. En [23] se utiliza, para la detección de las ondas en cuestión, el Emotiv EPOC, en un estudio de usuarios trabajando en un Sistema de Tutorías Inteligente (ITS). Este último estudio intenta demostrar la exactitud y validez de los valores del EPOC, relacionándolo con informes verbales de los alumnos, sobre sus estados emocionales. No se encontraron relaciones significativas entre los estados emocionales informados por los participantes y los valores emocionales del EPOC.

2 Problema

Evaluar el nivel de atención relacionado al comportamiento biométrico de alumnos, en una actividad de aprendizaje, en un contexto áulico, de modo que se pueda lograr un perfil de atención y su historial comparativo en distintas sesiones áulicas. Es necesario entonces obtener información del nivel de atención de un alumno -mediante el empleo de una interfaz Cerebro-Máquina (BCI)-, y observar su evolución durante el desarrollo de la clase, y comparar sus niveles de atención en función de los niveles esperados -en especial en los momentos (indicados por el docente en relación con los contenidos que imparte), que requieren mayor atención durante el desarrollo de la clase -. Luego de observar el estado del arte y considerar los datos necesarios para analizar el nivel de atención de los alumnos dentro de un contexto de áulico, se plantea la necesidad de una aplicación que tenga una captura multimodal de la información biométrica-emocional del alumno en cada instante, que permita el registro de imágenes de su rostro, y el contenido que se desarrolla en el pizarrón.

3 Descripción de la solución

Se diseña una arquitectura de sistema que tiene como entradas principales por un lado la captura de las ondas cerebrales -con el empleo del BCI-, y por el otro la toma de

capturas del pizarrón y rostro del alumno. Esta información se guarda en una base de datos dedicada de la aplicación, que identifica cada instante con el “Time” sincronizado, y que actúa como valor único de identificación. La aplicación permite etiquetar los niveles de las ondas para luego poder analizarlos más fácilmente, y así poder ubicar los momentos en los que cada una de las ondas o niveles supera o es menor que ciertos valores o umbrales de señal configurables. Se representa en la figura 3 el modelo conceptual del sistema.

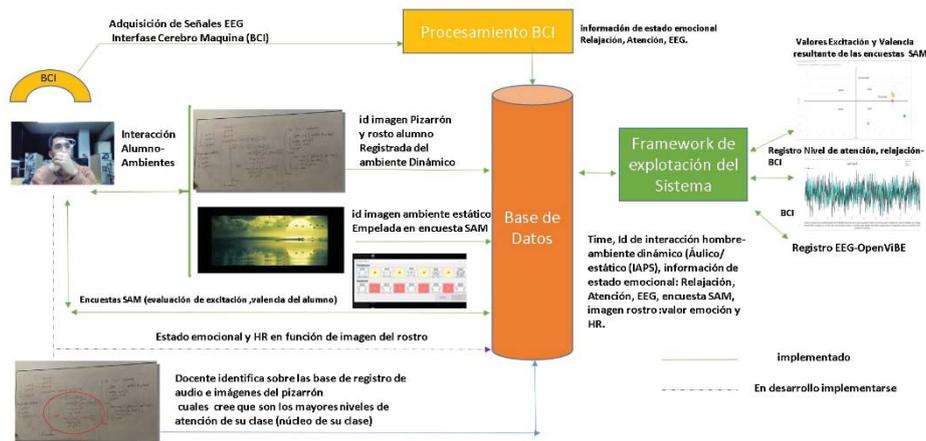


Fig. 3. Modelo Conceptual de la arquitectura de Valoración del grado de atención en contextos áulicos con el empleo de interfase cerebro – maquina

En este orden, la aplicación de captura de bioseñales cuenta con una interfaz la cual manipula los datos necesarios para adquisición de los datos del EEG que entrega el BCI Neurosky [24] para esto cuenta con las variables locales donde van a ser almacenadas cada una de estas. [25]. También almacena la fecha y hora de ejecución precisa para que pueda hacerse la integración con los datos de la aplicación de captura de imágenes de pizarrón y rostro del alumno. El framework de registro de bioseñales correspondiente al BCI se implementó en el lenguaje C#, utilizando la librería ThinkGear.dll la cual posee las funciones necesarias para interpretar las señales. Esta aplicación registra los metadatos de atención y relajación en un rango de 0 a 100. También registra los datos en hz de las ondas cerebrales. La estructura de los datos que registra es la siguiente: HORARIO_DE_PC, TEST_NAME, TEST_COMMENT, PORT, ROW_TIME, POORSIGNAL, ATTENTION, RELAJACION (también llamada Meditación), EEGPOWER, EegPowerDelta, EegPowerTheta, EegPowerAlpha1, EegPowerAlpha2, EegPowerBeta1, EegPowerBeta2, EegPowerGamma1, EegPowerGamma2. La figura 4 muestra un registro de explotación de valores de atención y relajación empleando la herramienta Power BI [26]. Con el empleo de OpenViBE [27], se registran y visualizan las señales de EEG.

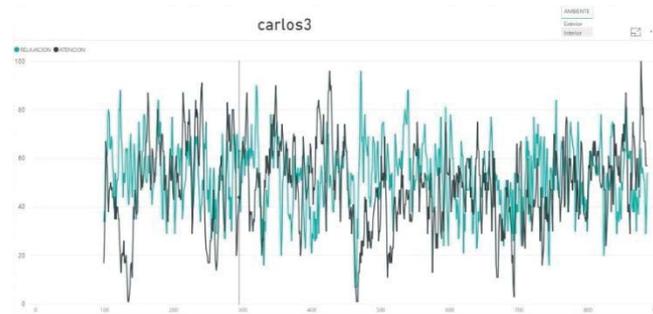


Fig.4 Registro de explotación de valores de atención y relajación

El sistema de captura de imágenes del entorno áulico (pizarrón) se implementó en el lenguaje C#, y registra en la base de datos secciones, identificadores, fecha, la ubicación de la imagen y una descripción de esta. La estructura de los datos que se registran es la siguiente: Sección, Identificador, Date, Imagen, Descripción. La base de datos se implementa en SQL Server, para su explotación se emplea Power BI y OpenViBE.

4 Pruebas y resultados obtenidos

Para el desarrollo de la prueba, y el análisis de los resultados alcanzados, se pide al docente que indique qué partes de su clase considera que requieren mayor atención por parte del alumno. Luego se comparan los mayores niveles de atención registrados por el sistema con el empleo del BCI, sumando la imagen de pizarrón que acompaña a este temporalmente y el rostro del alumno. También se valora en que partes el alumno muestra menor nivel de atención, y que tan lejos se encuentra temporalmente el alumno del nivel de atención esperado. La secuencia del protocolo experimental (Fig 5), que se conforma de las siguientes fases: Primera Fase -registro SAM sin imágenes con el objetivo de valorar el estado emocional inicial del alumno previo al inicio de la clase-, Segunda Fase -presentación de imágenes IAPS correspondientes al primer cuadrante del modelo de Russell, se muestran durante 40 segundos un total de seis imágenes IAPS (códigos 2154, 2340, 7281, 7289, y 7481), y luego se realiza encuesta SAM para valorar el estado emocional del alumno, Tercer Fase -inicio y desarrollo de la clase, se registra el EEG, rostro del alumno, e imágenes del pizarrón durante toda la clase-, Cuarta Fase -finaliza la clase y se realiza al alumno un nuevo test de SAM sin imágenes.



Fig. 5 Resumen de la secuencia del protocolo experimental

La docente destaca como punto de principal foco de atención, a la explicación del manejo de recorrida por los vectores con un índice de manejo independiente para cada uno, según se muestra en la mitad izquierda de la Figura 6, imagen capturada del pizarrón por el sistema e identifica como rango temporal desde 4:17 (+130seg de carrey inicial: 387seg) hasta 5:38 (+130seg de carrey inicial: 468seg), estos se cotejan con el audio de la clase, para ser comparados con los obtenidos con el empleo del BCI: registro de EEG, nivel de atención y relajación del alumno. Se muestra en la mitad derecha de la Figura 6, el rostro del alumno y el conjunto de valores y su relación con el valor máximo de atención y de relajación (96-48 respectivamente) registrado durante el desarrollo de la clase (a los 426 segundos), el que se corresponde con el intervalo temporal (387 seg- 468 seg) indicado por el docente en relación con el foco de máxima atención esperada. En la Figura 7 se muestran la imagen del registro de la encuesta SAM (fase 4) que completa el alumno en el sistema, y los valores de relajación y atención que tenía en ese instante correspondiente al fin de la clase, junto la imagen de su rostro. En la Figura 8 se muestran los valores obtenidos de excitación y valencia del alumno en función de la encuesta de SAM para cada fase del experimento.



Fig. 6 A la derecha imagen del núcleo de atención de la clase. A la izquierda imagen correspondiente a los valores de atención registrados del alumno durante el desarrollo del núcleo de atención de la clase

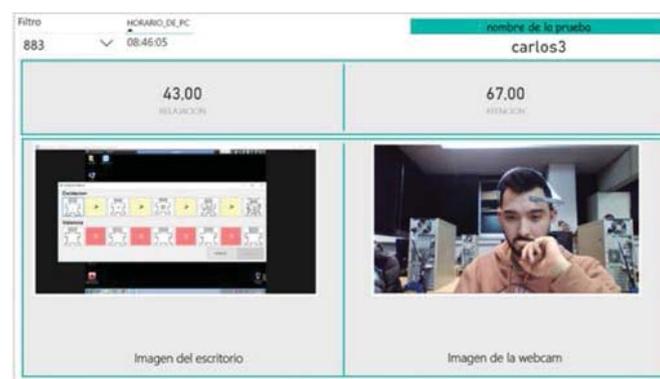


Fig. 7 Registro encuesta SAM

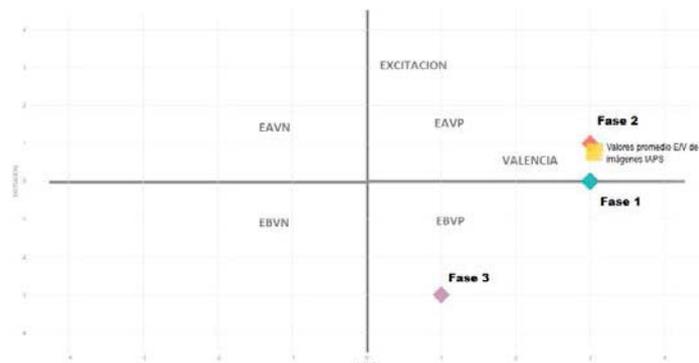


Fig. 8 Valores de Excitación/Valencia por cada fase

Conclusiones y futuras líneas de trabajo

Los resultados correspondientes a las pruebas basales fueron exitosos, el sistema registró el EEG, atención y relajación del alumno, en correspondencia con el contexto del ambiente áulico, acompañado de la captura de imágenes de pizarrón y rostro del alumno en correlación con sus valores biométricos. Estos resultados contribuyen al desarrollo de ambientes de computación afectiva en función del perfil emocional del alumno, en particular su nivel de atención y relajación, lo cual resulta valioso para los docentes. Como futuras líneas de trabajo se encuentra en desarrollo inicial la integración de parámetros fisiológicos del alumno con el empleo de una pulsera inteligente. En el marco de las futuras líneas de investigación se explorará la integración de aplicaciones, que capturan imágenes de rostros, de la cara del alumno, y permitan inferir el estado emocional, como así también otras que determinan la variación del ritmo cardiaco.

Referencias

1. R. W. Picard. "Affective Computing" MIT Press. (1997)
2. Arroyo, D.G. Cooper, W. Bursleson, B.P. Woolf, K. Muldner, and R. Christopherson, "Emotion Sensors Go to School," Proc. 14th Conf. Artificial Intelligence in Education, pp. 17-24, 2009
3. Mar Saneiro, Olga C. Santos, Sergio Salmerón-Majadas, and Jesús G. Boticario. 2014. Towards emotion detection in educational scenarios from facial expressions and body movements through multimodal approaches. Scientific World Journal 2014 (2014).
4. P. Ekman and W. Friesen, Facial Action Coding System: A Technique for the Measurement of Facial Movement: Investigator's Guide 2 Parts. Consulting Psychologists Press, 1978
5. Boticario Jesús, Santos Olga, Salmerón-Majadas, Uria-Rivas, Cabestreros Raul Quirós Pilar, Arealillo Miguel, Ferri Francesco BIG-AFF: Exploring Low Cost and Low Intrusive Infrastructures for Affective Computing in Secondary Schools- UMAP 2017, July 9-12, 2017, Bratislava, Slovakia.

6. Sidney K D'Mello, Jackeline Kory A Review and Meta-Analysis of Multimodal Affect Detection Systems ACM Computing Surveys (CSUR), 2015
7. T. Dalgleish, B. Dunn, and D. Mobbs, "Affective Neuroscience: Past, Present, and Future," *Emotion Rev.*, vol. 1, pp. 355-368, 2009.
8. Russell, J. A. A circumplex model of affect. *Journal of Personality and Social Psychology*, 39(6), 1161-1178. (1980)
9. Calvo Rafael Affect Detection: An Interdisciplinary Review of Models, Methods, and Their Applications *IEEE Transactions on Affective Computing*, vol. 1, no. 1, january-june 2010.
- 10 H. Gunes, B. Schuller, M. Pantic et al., "Emotion representation, analysis and synthesis in continuous space: A survey", en *Face and Gesture 2011, IEEE*, 2011, págs. 827-834, ISBN 978-1-4244-9140-7.
- 11 P. Ekman, "Basic Emotions", *Handbook of Cognition and Emotion*, T.Dalgleish y M.J.Power, eds., Chichester, UK: John Wiley & Sons, Ltd, 2005, cap.3, págs.45-60, ISBN: 978-0-47197-836-7.
- 12 J.Posner, J.A. Russell y B.S. Peterson, "The circumplex model of affect: an integrative approach to affective neuroscience, cognitive development, and psychopathology.", *Development and psychopathology*, vol. 17, n° 3, págs. 715-34, 2005, ISSN: 0954-5794.
13. Lang, P.J., Bradley, M.M., & Cuthbert, B.N. *International Affective Picture System (IAPS): Technical Manual and Affective Ratings*. NIMH 1997.
14. Lang, P. J. The cognitive psychophysiology of emotion: Fear and anxiety. In A. H. Tuma & J. D. Maser (Eds.), *Anxiety and the anxiety disorders* (pp. 131-170). Hillsdale, NJ, US. (1985).
15. Hamadicharef, "Brain Computer Interface Literature- A bibliometric study", in 10th *International Conference on Information Science, Signal Processing and their Applications*, Kuala Lumpur, 2010, pp. 626-629.
16. Ierache, J., Dittler M., Pereira G., García Martínez R, "Robot Control on the basis of Bio-electrical signals" XV CACIC 2009, UNJu, ISBN 978-897-24068-3-9
17. Ierache J, Pereira G, Iribarren J, Sattolo I, "Robot Control on the Basis of Bioelectrical Signals": "International Conference on Robot Intelligence Technology and Applications" (RiTA 2012). *Series Advances in Intelligent and Soft Computing of Springer*.
18. Ierache, J, Pereira.,G, Sattolo,I ,Guerrero, A, D'Alto J, Iribarren, J. Control vía Internet de un Robot ubicado en un sitio remoto aplicando una Interfase Cerebro-Máquina ".CACIC 2011, UNLP, ISBN 978-950-34-0756-1.p 1373-1382.
19. Ierache J, Cervino C, Eszter E, Fortin D, Castro Menna, Alvaro. "Registro emocional de personas interactuando en contextos de entornos virtuales", WICC 2018, UNNE , ISBN: 978-987- 3619-27-4, pp 785-789.
- 20 Ierache J, Ponce G, Nicolosi R, Cervino C, Eszter E, Sattolo I, Chapperon G "Explotación de datos EEG y parámetros fisiológicos de usuarios interactuando en contextos virtuales",WICC 2019, Universidad Nacional de San Juan, ISBN: 78-987-3984-85-3, pp 709 -713.
21. Silas Formunyuy Verkijika,Lizette De Wet, Using a brain-computer interface (BCI) in reducing math anxiety: Evidence from South Africa, *Computer And Education* Febrero 2015-
22. Heraz A, Frasson C, Towards a brain-sensitive intelligent tutoring system: detecting emotions from brainwaves, *Journal Advance in Artificial Intelligence*, Volume 2011, January 2011.Article No. 1
23. Harrison Tegan The Emotiv mind: Investigating the accuracy of the Emotiv EPOC in identifying emotions and its use in an Intelligent Tutoring System UC University of Canterbury (2013)
24. <http://neurosky.com> vigente julio 2019
25. Ierache J, Nicolosi R, Ponce G, Cervino C, Eszter E."Registro emocional de personas interactuando en contextos de entornos virtuales", CACI 2018, UNICEN, 2018, ISBN: 978-950-658-472-6, pp 877-886,
26. <https://powerbi.microsoft.com/es-es/> vigente julio 2019
27. <http://openvibe.inria.fr/> vigente julio 2019



CACIC 2019

Congreso Argentino de Ciencias de la Computación

XVII Workshop Computación Gráfica, Imágenes y Visualización

Coordinadores

Castro, Silvia (UNS)
Guerrero, Roberto (UNSL)
Bría, Oscar (UNLP / INVAP)

Comité Científico

Abásolo, María José (Argentina)	Gröller, Eduard (Austria)	Meschino, Gustavo (Argentina)
André Da Costa, Cristiano (Brasil)	Grötsch, Eberhard (Alemania)	Pardo, Alvaro (Uruguay)
Baldasari, Sandra (España)	Guerrero, Roberto (Argentina)	Perales, Francisco (España)
Bria, Oscar (Argentina)	Hernández Peñaloza, José Tiberio (Colombia)	Pina, Alfredo (España)
Castro, Silvia (Argentina)	Kozlíková, Barbora (República Checa)	Puppo, Enrico (Italia)
Collazos Ordóñez, César Alberto (Colombia)	Larese, Mónica G. (Argentina)	Rodríguez, Rojas Jorge (Venezuela)
D'Amato, Juan Pablo (Argentina)	Larrea, Martín (Argentina)	Selzer, Matías (Argentina)
Deinzer, Frank (Alemania)	Larregui, Juan Ignacio (Argentina)	Serón Arbeloa, Francisco (España)
Ganuzza, María Luján (Argentina)	Lecumberry, Federico (Uruguay)	Simari, Patricio (USA)
García Bauzá, Cristian (Argentina)	Manresa-Yee, Cristina (España)	Urribarri, Dana K. (Argentina)
Gazcón, Nicolás Fernando (Argentina)	Más Sansó, Ramón (España)	Varona Gómez, Xavier (España)
Giacomantone, Javier (Argentina)	Matković, Krešimir (Austria)	Vaz de Carvalho, Carlos (Portugal)
González Hidalgo, Manuel (España)		

Temas

Modelado Geométrico	Visualización de Software
Modelado de Fenómenos Naturales	Visualización de Algoritmos
Iluminación y Rendering	Visualización de Terrenos
Animación	Visualización de Flujos
Realidad Virtual y Aumentada	Visual Analytics
Interacción Humano-Computadora	Generación, Adquisición, Registración,
Diseño Asistido por Computadora	Manipulación, Segmentación, Restablecimiento y
Interfaces Gráficas	Compresión de Imágenes
Interfaces con Dispositivos No Convencionales	Correcciones de Color
Juegos y Tecnologías del Entretenimiento	Codificación y Encriptación de Imágenes
GPU y Sistemas Gráficos	Recuperación de Imágenes
Visualización de Información	Clasificación y Clusterizado
Visualización Científica	

Plataforma para Repositorios Digitales 3D de Colecciones Biológicas

Nicolás Jofré, Graciela Rodríguez, Yoselie Alvarado,
Jacqueline Fernandez, and Roberto Guerrero

Laboratorio de Computación Gráfica (LCG)
Universidad Nacional de San Luis,
Ejército de los Andes 950
Tel: 02664 420823, San Luis, Argentina
{npasinetti, gbrodriguez, ymalvarado, jmfer, rag}@unsl.edu.ar

Resumen Las colecciones institucionales son repositorios especializados, siendo esenciales para estudios científicos sobre la biodiversidad de una región y su conservación. Estas requieren el desarrollo de un plan a corto y largo plazo para prevenir daños en la colección. Por lo tanto, es necesario proporcionar mecanismos innovadores para salvaguardar la valiosa información proporcionada por las colecciones y, al mismo tiempo, evitar cualquier posible pérdida de información. En este trabajo se presenta una plataforma informática que incluye un marco de trabajo y una arquitectura, donde a través de la combinación de hardware y software, es posible conformar una colección digital estructurada y organizada de muestras científicas. La plataforma propuesta es validada a través de un caso de estudio surgido entre la *Unidad de Herpetología* y el *Laboratorio de Computación Gráfica*, ambas instituciones pertenecientes a la *Universidad Nacional de San Luis*.

Palabras Claves: Digitalización 3D, Repositorio 3D, Colección Biológica.

1. Introducción

Toda colección es una puerta hacia la investigación y la inspiración de nuevas ideas, fundamentada en una continua investigación de las variables que lo hacen diferente y singular.

Las colecciones universitarias son repositorios especializados de fuerte matriz científica, con recursos humanos altamente formados y compuesto por colecciones de objetos patrimoniales producto de prácticas científicas sistemáticas, obtenidas en trabajos de campo y de laboratorio que constituyen un conjunto de materiales de alto valor científico y cultural. Las colecciones biológicas, en particular, conforman un acervo de información valiosa y un recurso indispensable de conservación de la propia biodiversidad [1].

Una colección puede ser pensada como un modelo de museo o un espacio integral, que tiene entre sus prioridades el servicio público y, en consecuencia, la construcción de su carácter educativo, para lo cual es fundamental la nueva

forma de entender el patrimonio que sirve a la construcción del conocimiento en las instituciones.

Dichas colecciones son de gran utilidad para estudiantes, profesores, investigadores y demás profesionales, convirtiéndolas en centros de estudio científico y social. Estas premisas quedan formalizadas en la declaración de los alcances u objetivos establecidos para todo repositorio [2].

La inmediatez de la valoración social encuentra respuesta en nuestra capacidad de comunicar e informar. En consecuencia, no debe establecerse distinción de carácter electivo entre conservación y comunicación, por el contrario, sólo se conserva en cuanto se comunica. El desarrollo de las nuevas tecnologías de la comunicación y la información (TIC), además de las diferentes iniciativas gubernamentales, están creando nuevos modelos de accesibilidad y usos de los diferentes centros de divulgación científica, sean estos museos, colecciones biológicas o cualquier institución dedicada a aumentar el grado de percepción de la ciencia y la tecnología [3,4,5].

En la actualidad, los avances en el campo de adquisición de información tridimensional son capaces de recrear de manera fidedigna, mediante datos informáticos, una representación fiel de un objeto real. De la misma manera, los avances en el campo de la visualización e interacción de la información tridimensional, utilizando técnicas de Realidad Virtual, Realidad Aumentada o prototipado rápido, están reemplazando la manipulación directa de objetos reales, que pueden ser escasos, valiosos o frágiles [6,7]. Dichos avances asisten y enriquecen la información que, los museos en general y las colecciones biológicas en particular, transmiten a las personas, principales consumidoras de las TIC y centro de la experiencia [8].

Es de importancia, entonces, proteger y conservar el patrimonio científico mediante la conformación de una colección digital estructurada y organizada de muestras científicas, desarrollada según un esquema conceptual predefinido, a los fines de ampliar el potencial didáctico-científico de estos recursos. En este trabajo se describe una plataforma informática para crear repositorios digitales 3D de colecciones científicas.

Este documento está organizado de la siguiente forma: en la sección 2 se introduce de forma breve el concepto de colecciones biológicas. En la sección 3 se detalla cómo está conformada la plataforma. Además de explicar en la sección 4 un caso de estudio en el cual fue utilizada. Por último, en la sección 5 concluimos nuestro trabajo y proponemos algunos trabajos futuros.

2. Colecciones Biológicas

Recolectar objetos es una característica natural de la raza humana. Probablemente, también lo es el impulso de organizar colecciones de una manera “sistemática”. En particular, dentro de la biología la sistematización es uno de los procesos que se utilizan principalmente para describir una especie. Los científicos se basan en lo que se conoce como ejemplares tipo, los cuales constituyen un patrón de referencia. Estos son un instrumento esencial para los taxónomos y un

patrimonio de la ciencia, por lo que deben guardarse en instituciones que garanticen su conservación y sean accesibles a la comunidad científica. Sin embargo, en ocasiones resulta difícil reconocer una especie porque no existe material tipo con el que hacer una comparación. En estos casos, lo aconsejable es rastrear los probables tipos en colecciones históricas, anteriores al siglo XX.

En este contexto, existen los curadores y auxiliares cuyas tareas están relacionadas con las acciones de preservar, conocer y difundir el patrimonio integral, natural y cultural (en todas sus manifestaciones especialmente aquellas que, de manera tangible o intangible, identifican a la institución o a la región entre otros) con los parámetros sobre los que se fundamenta la técnica del coleccionismo, intentando convertirlo en un espacio vivo para la acción cultural [9,10].

Es así que se disponen de *colecciones biológicas*, las cuales son repositorios sistematizados (bien identificados, clasificados y ordenados) que funcionan como representación del patrimonio natural de un país o región y constituyen un archivo histórico de utilidad múltiple, donde la preservación de especímenes y su información asociada sirven como base para estudios taxonómicos, sistemáticos, ecológicos, filogenéticos, biogeográficos, de genética de poblaciones y conservación. La mayoría de estos repositorios se encuentran depositados en museos de ciencia, pero también en universidades, centros de investigación e, incluso, total o parcialmente en colecciones privadas.

El manejo de la información biológica en una colección se representa de tres formas según el método de conservación [11]:

1. *Ejemplares en seco*, divididos en cuatro categorías:
 - Categoría 1. Piel de aves, mamíferos y algunos reptiles, pieles montadas en taxidermia, trofeos, invertebrados montados en alfileres, nidos de aves e insectos y la mayoría de los ejemplares botánicos.
 - Categoría 2. Huesos con excepción de fósiles.
 - Categoría 3. Conchas, conchas de cangrejos.
 - Categoría 4. Piedras, fósiles y minerales.
2. *Ejemplares en líquido*: Animales y plantas, organismos enteros o partes, preparaciones histológicas, enteras o partes y piedras, fósiles, minerales.
3. *Documentación*: Archivos en papel, archivos en película y cinta, archivos electrónicos y moldes.

Las colecciones biológicas se dividen en diferentes tipos en base al método de conservación del ejemplar según su naturaleza (ver fig. 1).

Cualquier colección digital que se precie debe considerar las particularidades de cada tipo de colección. Es de interés el definir un soporte lo suficientemente genérico que permita albergar cualquier tipo de colección.

3. Plataforma

Una plataforma consiste en definir una estrategia computacional que sirva de soporte conceptual a nociones teóricas asociando los modelos digitales con información específica relacionada a los objetos de estudio, de manera que puedan

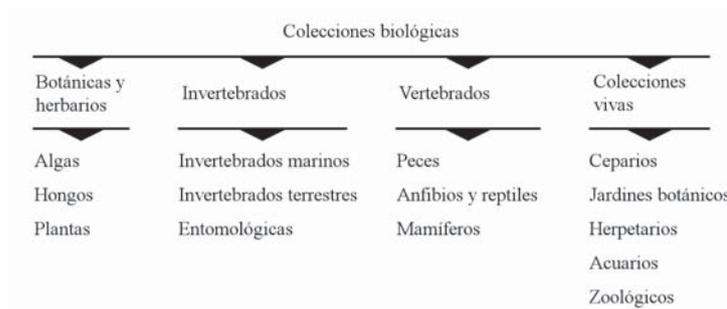


Figura 1: Tipos de colecciones.

ser utilizados para el desarrollo de nuevos trabajos en diferentes ámbitos profesionales. Para tal fin, la plataforma presentada incluye un marco de trabajo, en el cual se basó el software desarrollado. La construcción de una plataforma requiere de una arquitectura de control sobre la cual funcionan las aplicaciones necesarias logrando sistematizar la resolución del problema y su distribución al usuario final. Adicionalmente, como en cualquier plataforma típica, para su funcionamiento se requiere de un sistema operativo, lenguajes de programación, interfaz de usuario, y bibliotecas de uso específico.

3.1. Marco de Trabajo

A partir del dominio problemático planteado en la sección 1, es necesario establecer las directrices para el diseño y desarrollo de la plataforma, permitiendo de esta manera que cumpla con las funciones básicas que hacen al objetivo de la problemática descrita y que pueda reutilizarse para otros dominios.

La plataforma presentada comprende diferentes tareas a tener en cuenta para el desarrollo de una solución las cuales se pueden agrupar en tres grandes etapas. A continuación se presenta una breve descripción de las mismas:

Digitalización 3D del patrimonio. El resultado de esta etapa se verá reflejado en un conjunto de modelos tridimensionales de ejemplares tipo de la colección científica. La misma implica el análisis y definición de parámetros de digitalización de los ejemplares de estudio a partir de sus caracteres particulares.

Las actividades propuestas, de acuerdo a lo definido en trabajos previos [12], implican:

- Adquisición de datos a partir del espécimen, considerando cada región del mismo.
- Procesamiento de datos identificando la información relevante que permita detectar las singularidades de la muestra a digitalizar.
- Generación de modelos 3D construyendo una malla poligonal 3D que representa la superficie del objeto.

- Texturizado del modelo a partir de imágenes de textura de alta resolución.
- Verificación y Corrección manual de los modelos digitales utilizando herramientas de modelado 3D para editar los modelos.

Especificación del esquema conceptual de la colección digital. El resultado de esta etapa será una colección estructurada y organizada de los ejemplares digitales, desarrollada según un esquema conceptual definido para la colección.

Las actividades propuestas implican:

- Una representación simbólica de los ejemplares dentro de un sistema computacional.
- Establecer los ejemplares tipo que reúnen las colecciones analizadas.
- Una solución automática para la organización e interrelación de los datos, así como también, un conjunto de operaciones que se puedan realizar sobre los datos.

Desarrollo de una estrategia para el acceso al contenido de la colección. El resultado de esta etapa es un medio para el acceso y uso de los recursos electrónicos generados en las etapas anteriores, ofreciendo a los usuarios un canal de localización y utilización de los recursos de acuerdo a condiciones preestablecidas.

Las actividades propuestas implican:

- Definición de una licencia que establecerá las condiciones de adquisición mediante un acuerdo de licencia. La licencia debe permitir el uso justo (trato justo, etc.) de toda la información con fines educativos, de enseñanza, no comerciales y de investigación.
- Establecer un punto de acceso a los recursos, de acuerdo a las necesidades de los usuarios, como página web, distribución en DVD, memoria usb, aplicaciones de uso y visualización, entre otros.
- En caso de corresponder, determinar los usuarios autorizados y el método de acceso de los mismos.
- Divulgación del punto de acceso a los usuarios objetivos del recurso electrónico generado.

3.2. Arquitectura

La arquitectura definida para la plataforma presentada fue organizada de acuerdo a lo establecido en [13], donde se consideran tres grandes capas: *Capa de Fuente de Datos*, *Capa de Gestión del Conocimiento*, y finalmente una *Capa de Presentación* que se relaciona con el usuario final.

Considerando las etapas establecidas en el marco de trabajo propuesto y sus correspondientes actividades, en esta subsección se definen tres módulos principales que dan funcionamiento a la plataforma: *Módulo de Digitalización*, *Módulo de Estructuración* y *Módulo de Acceso*.

Debido a que cada uno de los módulos poseen funcionalidades independientes, al momento de procesar una colección científica, es posible paralelizar su ejecución. La Figura 2 ilustra la arquitectura general de la plataforma. A continuación se describirán las principales funcionalidades e interacciones de cada módulo.

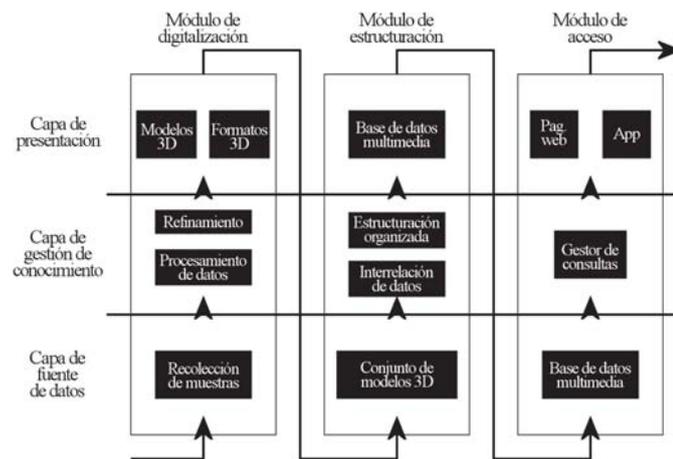


Figura 2: Arquitectura general.

Módulo de Digitalización. Dada una muestra de colección a digitalizar, este módulo inicialmente debe recolectar la información relevante y necesaria para su procesamiento. Una vez procesada toda la información recolectada, en caso de ser necesario, se deben refinar los resultados para lograr una solución lo más aproximada posible a la muestra real. Como tarea final, este módulo proveerá un modelo tridimensional en los formatos indispensables para su correcto uso y portabilidad. La primer columna de la Figura 2 muestra la estructura mencionada para este módulo.

Módulo de Estructuración. A partir del conjunto de modelos 3D obtenidos por el *Módulo de Digitalización*, es necesario procesarlos para definir su interrelación y así determinar una estructuración organizada de los mismos. Para lograrlo se deben tener en cuenta las características relevantes de la colección, el tipo de colección, los ejemplares tipo que la definen, las palabras claves del dominio, entre otros datos. Como se muestra en la segunda columna de la Figura 2 el resultado de este proceso es una *Base de Datos Multimedia*.

Módulo de Acceso. El acceso a la información es un módulo basado en una arquitectura cliente-servidor. Donde la interfaz de usuario será una aplicación cliente (Página Web, App, etc.) que tendrá la tarea de recepcionar las solicitudes del usuario, y gestionar las peticiones necesarias al servidor web (*Gestor de*

Consultas) para dar solución a las necesidades del usuario con el soporte de la *Base de Datos Multimedia*. Para lograrlo el proceso cliente, una vez obtenido el recurso, debe incorporar algún mecanismo que le permita mostrar al usuario el recurso digital solicitado. Dicha visualización deberá tener en cuenta el detalle que se pretende representar para permitir al usuario ocultar o destacar información de los modelos 3D (texturas, polígonos, aristas frontales, aristas traseras, iluminación). La tercera columna de la Figura 2 ilustra el esquema general que debe tener el *Módulo de Acceso* descripto.

4. Caso de estudio

La plataforma propuesta fue validada en un caso de estudio surgido como una labor interdisciplinaria en la cual participaron equipos de trabajo de la *Universidad Nacional de San Luis* pertenecientes a las áreas de la *Unidad de Herpetología* (UH) y el *Laboratorio de Computación Gráfica* (LCG), quienes de manera cooperativa estrecharon lazos de colaboración, para el desarrollo e implementación de nuevas TICs principalmente en Realidad Virtual (RV) y Realidad Aumentada (RA).

En este sentido, el LCG realiza acciones que tienen como objetivo brindar, entre otros, la posibilidad de representar información relevante en forma visual, adaptándola a diferentes problemáticas educativas y de investigación. Por la tanto, la plataforma creada en el LCG, permitirá el **resguardo**, la **visualización** y el **aprovechamiento del potencial pedagógico-didáctico** del patrimonio cultural material e intangible de las distintas áreas, el fomento de la participación de profesionales y finalmente, el desarrollo y utilización de mecanismos computacionales gráficos e innovadores que permitan el acceso a la información en forma funcional y eficiente.

4.1. Unidad de Herpetología

La UH tiene como objetivos: resguardar los ejemplares de anfibios y reptiles, brindar asesoramiento sobre herpetofauna, impulsar la realización de trabajos de investigación en sistemática y biodiversidad, almacenar información digital de los ejemplares y del material bibliográfico, y difundir el conocimiento sobre reptiles y anfibios a través de docencia, capacitación y servicios.

La unidad concentra ejemplares de los especímenes tipo, sitios típicos, material bibliográfico de difícil realización, así como un vasto registro fotográfico, que representa y protege gran parte del patrimonio herpetológico de la región central y patagónica argentina.

Las muestras recolectadas pertenecen a un subconjunto de la colección de ejemplares llamada especímenes tipo, debido a sus características morfológicas (Figura 3).

Una vez determinada la muestra a procesar, se seleccionó la técnica de digitalización a través de fotogrametría, debido a la necesidad de apreciar texturas



Figura 3: Muestras: Especímenes tipo.

en alta calidad, en comparación a los detalles de la malla que es logrado a través del escaneo 3D [12].

Si bien el procesamiento de datos devuelve el modelo tridimensional y texturizado correspondiente al ejemplar de la muestra, fue necesario editar la malla para realizar correcciones particulares como por ejemplo, polígonos aislados, secciones de la malla que no corresponden al espécimen. Por último, se realizó la exportación de los modelos digitales en formatos estándares y portables (.obj y .stl) para ser almacenados en el repositorio online. Las tareas forman parte del **Módulo de Digitalización** descrito anteriormente.

Debido a que el sistema se encuentra en proceso de implementación, cuenta con un **Módulo de Estructuración** base, organizando los modelos digitales respecto a la información básica para su identificación.

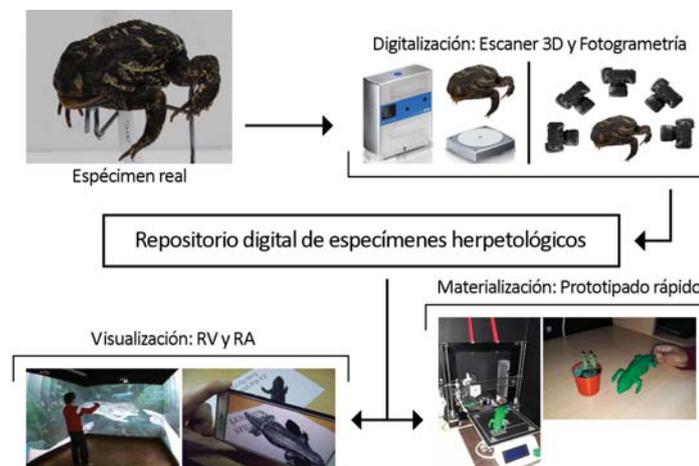


Figura 4: Esquema de creación y divulgación de los modelos tridimensionales de especímenes digitales realizados en este trabajo para la enseñanza y divulgación de la Herpetología.

Finalmente, los especímenes presentes en el repositorio tienen la posibilidad de ser visualizados a través de aplicaciones de RV y RA, como se describió en el **Módulo de Acceso**. Adicionalmente, los modelos pueden ser materializados utilizando impresoras 3D, como se muestra en la figura 4.

5. Conclusiones y Trabajo Futuros

Considerando que la generación de información digital a partir de colecciones tangibles permite preservar los recursos y potenciar su acceso, este trabajo describe una plataforma para crear repositorios digitales 3D de colecciones científicas. El desarrollo hace especial énfasis en las colecciones biológicas.

Con el fin de generalizar los procesos que deben ser llevados a cabo para la creación del repositorio, primeramente se describe un marco de trabajo que define las tareas de la plataforma realizada, para luego detallar los módulos que conforman la arquitectura de la misma. Estos últimos fueron definidos teniendo en cuenta como tareas la *Digitalización* de los recursos tangibles, la *Estructuración* de los datos obtenidos y finalmente la gestión que permite brindar mecanismos de *Acceso* a los recursos digitales. Adicionalmente, se presenta un caso de estudio en proceso, correspondiente a la colección herpetológica de la *Unidad de Herpetología* de la *Universidad Nacional de San Luis*.

La plataforma presentada pretende definir estrategias que permitan replicar lo realizado en otras colecciones y repositorios de valor patrimonial, cultural y social. Para el caso de estudio, el resultado de este trabajo se ve reflejado en un repositorio compuesto por un conjunto de modelos tridimensionales de ejemplares tipo y la disponibilidad de un método sistematizado que será utilizado por el personal de la *Unidad de Herpetología* para la *Digitalización* de futuros recursos. Los modelos fotorrealistas en 3D obtenidos ofrecen nuevas posibilidades para la práctica diaria de la herpetología, protección y conservación de especímenes. Además de ser documentaciones objetivas en 3D, consiguen una conservación fiable del patrimonio herpetológico.

Este trabajo podría ser utilizado como recurso en diversas actividades tales como:

- Actividades de *e-learning*, ya sea en asignaturas curriculares, como cursos de postgrado, brindadas a estudiantes propios o externos.
- Apoyo a asignaturas presenciales, mediante *blended learning*, permitiendo combinar la formación presencial con la formación a través de las TIC, tratando de aprovechar todas las ventajas de éstas en los procesos de aprendizaje presenciales.
- La definición de nuevos instrumentos didácticos a partir de los nuevos recursos disponibles.

Adicionalmente, las configuraciones de acceso permiten romper, por un lado, con las barreras del tiempo, permitiendo acceder mediante visitas online en cualquier momento, y por otro, con la barrera del espacio, brindando el acceso desde cualquier punto del planeta.

Entre los trabajos futuros se encuentran: la definición de herramientas tecnológicas de avanzada que permitan establecer nuevos lineamientos de investigación y estrategias didácticas y la implementación de la metodología con otras colecciones biológicas de la *Universidad Nacional de San Luis*, entre ellas la colección botánica.

Referencias

1. John E Simmons and Yaneth Muñoz-Saba. *Cuidado, manejo y conservación de las colecciones biológicas*. Univesidad Nacional de Colombia, 2005.
2. Marcela Georgina Gómez Zermeño. Bibliotecas digitales: recursos bibliográficos electrónicos en educación básica. *Comunicar: Revista científica iberoamericana de comunicación y educación*, (39):119–128, 2012.
3. David K Yeates, Andreas Zwick, and Alexander S Mikheyev. Museums are biobanks: unlocking the genetic potential of the three billion specimens in the world's biological collections. *Current Opinion in Insect Science*, 18:83 – 88, 2016. Neuroscience * Special Section on Insect phylogenetics.
4. José Saorín, Cecile Meier, Carolina Ruiz Castillo, Jorge Cantero, Dámari Melián Díaz, and Alejandro Bonnet De León. Creación, visualización e impresión 3d de colecciones online de modelos educativos tridimensionales con tecnologías de bajo coste; caso práctico del patrimonio fósil marino de canarias. *Education in the Knowledge Society (EKS)*, 17:89, 12 2016.
5. Guan Wang, Hamid Laga, Ning Xie, Jinyuan Jia, and Hedi Tabia. The shape space of 3d botanical tree models. *ACM Transactions on Graphics*, 37:1–18, 01 2018.
6. Jonathan Lartigue, Tyler Scoville, and Minh Pham. Promoting k-8 learning using oculus rift: Employing virtual reality to increase learning outcomes in elementary biology. In Theo Bastiaens, editor, *Proceedings of E-Learn: World Conference on E-Learning in Corporate, Government, Healthcare, and Higher Education 2014*, pages 1100–1105, New Orleans, LA, USA, October 2014. Association for the Advancement of Computing in Education (AACE).
7. S. Keaveney, C. Keogh, L. Gutierrez-Heredia, and E. G. Reynaud. Applications for advanced 3d imaging, modelling, and printing techniques for the biological sciences. In *2016 22nd International Conference on Virtual System Multimedia (VSMM)*, pages 1–8, Oct 2016.
8. Dallen J Timothy. *Cultural heritage and tourism: An introduction*, volume 4. Channel View Publications, 2011.
9. Claude Lavoie. Biological collections in an ever changing world: Herbaria as tools for biogeographical and environmental studies. *Perspectives in Plant Ecology, Evolution and Systematics*, 15(1):68 – 76, 2013.
10. D. Bakk and R. Urban. Scientific and social value of biological collections. pages 6–9, 08 2014.
11. Edwin Trujillo Trujillo, A Vargas-Triviño, and Salazar-Fajardo . Clasificación, manejo y conservación de colecciones biológicas: una mirada a su importancia para la biodiversidad. *Momentos de Ciencia*, 11:97–106, 02 2014.
12. Nicolás Jofré, Graciela Rodríguez, Yoselie Alvarado, Jacqueline Fernandez, and Roberto Guerrero. *Evaluating an End-to-End Process for Herpetological Heritage Digital Preservation*. Springer International Publishing, Cham, 2019.
13. Vidalina De Freitas and Guillermo Yáber. Una propuesta de arquitectura para los sistemas informáticos de gestión del conocimiento en instituciones de educación superior. *Espacios*, 36:E2, 03 2015.

Streamlining the study of the Tierra del Fuego forest through the use of deep learning

Leonel Viera¹, Federico González¹, Rosina Soler², Lucas Romano¹, Guillermo Feierherd¹ 

¹ Instituto de Desarrollo Económico e Innovación, Universidad Nacional de Tierra del Fuego, Antártida e Islas del Atlántico Sur
Hipólito Irigoyen 880, 9410, Ushuaia, Tierra del Fuego, Argentina
{lviera,fgonzalez,lromano,gfeierherd}@untdf.edu.ar
² Centro Austral de Investigaciones Científicas, CONICET
Houssay 200, 9410, Ushuaia, Tierra del Fuego, Argentina
rosinas@cadic-conicet.gob.ar

Abstract. Understanding plant-herbivorous relationships allows to optimize the way to manage and protect natural spaces. In this paper the study of this relationship in the ñire forests (*Nothofagus antarctica*) of the province of Tierra del Fuego (Argentina) is approached. Using trap cameras to monitor such interaction offers the opportunity to quickly collect large amounts of data. However, to take advantage of its potential, a large investment in trained personnel to analyze and filter the images of interest is required. The present work seeks to establish a path to significantly reduce this obstacle using the advances of machine and deep learning in the recognition of objects from images.

Keywords: machine learning, deep learning, computer vision, trap cameras, forests, image recognition, ñire, antarctic nothofagus.

1 Introduction

1.1 Understanding ecosystems with images

To understand the complexity of forest ecosystems, improve their management and at the same time ensure their conservation, it is necessary to have detailed knowledge about their natural dynamics and that of the organisms that live there (number, location and behavior of animals) [1]. Much of the native forests of Southern Patagonia are currently used for livestock (cows and sheep). The direct impact of domestic animals (foraging, trampling), coupled with the natural presence of wild herbivores (guanacos), affect the growth of young trees and the composition of vegetation [2]. Good management of this activity together with monitoring over time could lead to maintaining productivity without degrading the forest environment.

The use of cameras with motion sensors, called trap cameras, in natural habitats significantly speeds up studies on ecology and wildlife conservation over the past two decades [3]. These have become an essential tool for environmentalists, allowing them to study the size and distribution of populations [4] and evaluate habitat used by animals. While they allow thousands of images to be taken [5], effective data extraction is traditionally performed by people (i.e. experts, previously trained volunteers) and it is so slow and expensive that much of the valuable knowledge in these large data repositories remains untapped.

1.2 Machine Learning and Deep Learning

Machine Learning allows computers to solve tasks without being explicitly programmed to solve them [6]. There are several machine learning algorithms, with those classified as supervised being the most interesting for this work. These work by training the algorithm by entering entries associated with their expected results, in order to obtain a function that may classify future entries [6]. For example, when classifying images, the machine trains with many pairs of images and their corresponding labels, where the image is the input and its correct label (for example, "nocturnal") is the output.

Deep learning [7] allows computers to automatically extract multiple levels of abstraction from raw data. Deep convolutional neural networks are a type of deep learning architecture [7] in which each layer of neurons uses convolutional operations to extract information from small regions that overlap between the previous layers [7]. For the classification, the last layer is generally a function with an output between 0 and 1 per class; adding 1 all class exits. These are interpreted as the estimated probability that an image belongs to a certain class, the higher the more certain is the algorithm that the image belongs to that class [8].

Deep learning has changed the way to solve many difficult problems [7], including voice recognition [9], automatic translation of texts [10], image recognition [11], among others. As a counterpart, deep learning requires a large amount of input data (correctly labeled) and high computational resources.

1.3 Combining forest management and deep learning

In the case treated in this article, we will try to automate the identification and counting of animals from the images captured by trap cameras using deep learning techniques, obtaining a more efficient solution to provide data in projects with these conditions [12].

2 Case study

In 2014, a project to assess the impact of native (guanacos) and domestic (cattle) herbivores on ñire forests (*Nothofagus antarctica*) under livestock production in Tierra del Fuego began. For this, in 4 different forests, fences were installed that

prevented the entry of animals and the vegetation was studied inside and outside these fences. At the same time, trap cameras were installed focusing on both the closed zone and the free sectors. Through these cameras it was possible to evaluate the effectiveness of the closures and the use of animals (feed, rest and others), differentiating seasons of the year and times of day.

The images taken by the cameras were collected from 2015 to 2017, accumulating more than 150,000 photographs. This amount of information produces a high cost of manual image analysis, slows down work and makes it difficult to obtain results.

So far there are several results that show the impact of animals on forest vegetation. For example, both the growth and survival of young trees showed a favorable response to closures. In the closed areas, survival was between 75% and 87%, while outside them it was only 60%. In addition, they achieved greater growth in height. However, this survival varies according to the age of the trees, since the younger (2-3 years) the greater the probability of mortality. Even the smallest plants outside the closures had higher mortality than those of the same age within the closures, as expected. Then, after 5 years the trees were able to survive in the same proportion outside the closures (Fig. 1A).

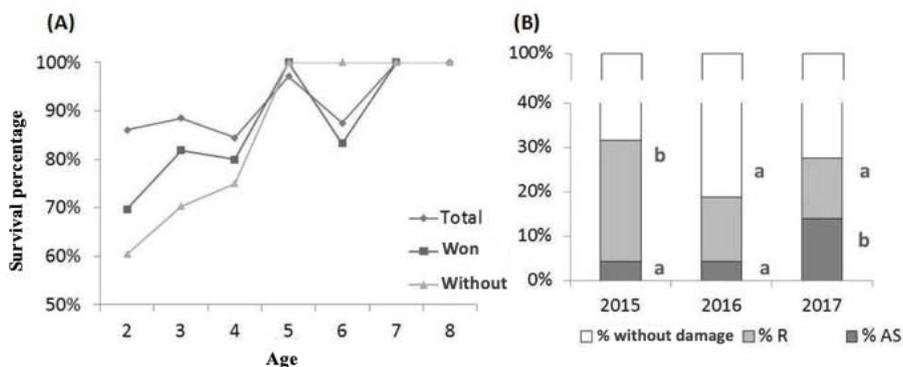


Fig. 1. (A) Survival percentage of the ñire trees inside and outside the closures and according to age. (B) Proportion of damaged trees: eaten by animals (R), desiccation (AS), or without damage. The letters indicate significant statistical differences ($p < 0.05$) between years for each type of damage.

Another example is the identification of two main types of damage: on the one hand branches eaten by animals (called "browsing") and on the other hand drying by climatic factors (frost or drought in spring). The damage by browsing (Fig. 1B) was decreasing over time from the installation of fences, as expected.

These results allow us to understand the effect of animals on the structure and conformation of the forest ecosystem of the province. But it is essential to understand the behavior of the animals (frequency of use of the environment), the time of year with greater use and the possible overlap of presence of domestic and native animals in the forest. Therefore, the use of the entire dataset will increase the accuracy of the conclusions and allow discovering new facets of plant-herbivorous interactions.

2.1 What useful information can be obtained from the images?

For the understanding of plant-herbivore interaction in fire forests and the generation of outputs similar to those shown in the previous section, it is useful to be able to identify the following image properties:

- Presence or absence of animals (photos with animals and without animals). Only photos with animals are of interest.
- Day or night (photos taken at night and photos taken during the day).
- Classify species in images with animals (for example guanacos, cows, foxes, among others).
- Number of animals per species.
- Use that the animal is making of the forest (for example feeding, resting).

Of the characteristics listed, being able to automatically distinguish images with animals is the task that would most contribute to expedite research, since they represent the minority within the data set; only 0.01% estimated. With the separation made, an expert user could quickly identify the rest of the conditions.

2.2 Dataset status

Within the more than 150,000 thousand images different situations can be observed: day and night photos, with high contrast of light, landscapes with and without snow, animals far away or very close to the camera are some examples. This diversity of conditions makes it necessary to have a large number of classified images, with different conditions, to achieve better accuracy when applying deep learning techniques.

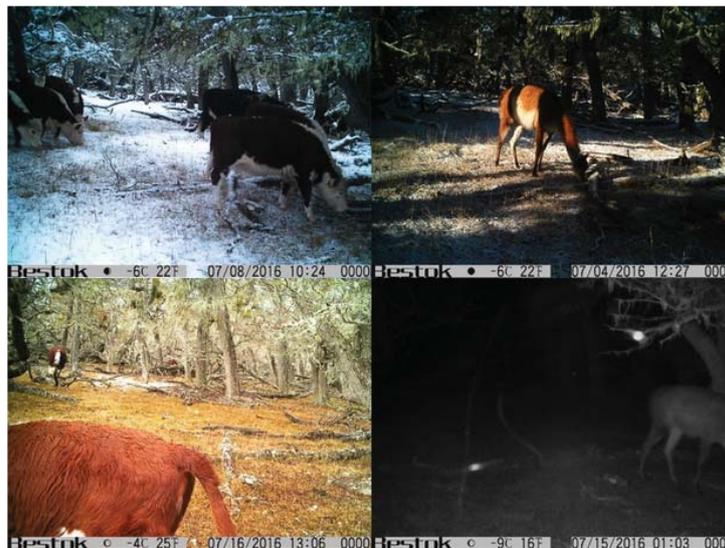


Fig. 2. Examples of the diversity of conditions within the data set. Cows in winter (left above). Guanaco by day with high light contrast (right above). Cows in daylight with a partially photographed animal (left below). Guanaco partially photographed at night (right below).

On the other hand, the trap cameras had different configurations of sensitivity to changes in the environment, which influenced the number of images taken by each of them. Thus, for the same period, there are cameras that have 300 photos while others have 3000, generating a tendency to replicate what happened in a particular forest.

The trap cameras used did not have the possibility of generating metadata that could contribute to improving a rapid classification (for example, identifying day and night with the creation date-time).

Finally, there is only a subset of 1,500 images analyzed by experts, of which 108 are classified as "with animals" and the rest as "without animals". A quick visual evaluation would allow to differentiate the day and night photos.

3 Workplan

After evaluating the state of the data set, it was decided to work on two main aspects: the classification of images (in order to improve the results of the model to be generated) and establish a set of steps to follow to frame the next advances to be made.

3.1 Sorting images, a social experiment

In order to reduce the unclassified data set and obtain a more complete training set [6] for the deep learning model, a web application was designed and implemented that seeks to get volunteers willing to contribute to the project. The idea is simple: the website shows the user random images that must be classified as "with animals" or "without animals" and, at the same time, "by day" or "by night".

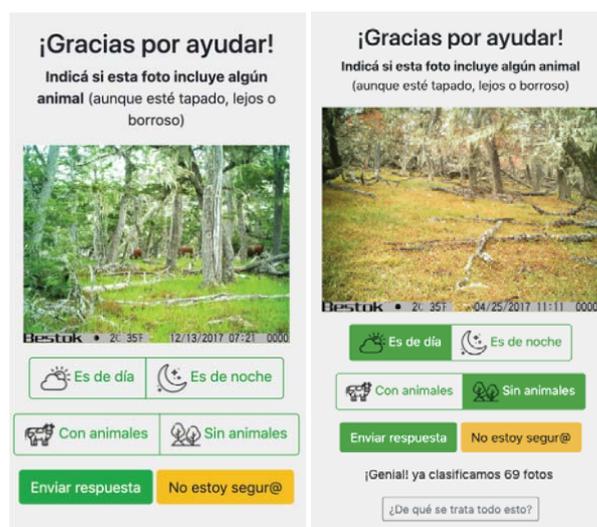


Fig. 3. Mobile version of the image classification screen.

To avoid malicious users or involuntary mistakes, each image must have a minimum of 3 responses and at the same time 90% of them must match the chosen criteria. For example, if three people classify an image as "without animals" and "at night", it is automatically classified with that criterion and will not be shown again to future users. On the contrary, if anyone had differed in the criteria, the image will continue to be shown to future users until reaching a 90% match in their rankings.

Due to this condition, it was necessary to adjust the algorithm that selects random images from the total set since, if they were really random, it would be difficult to get the same image to obtain (at least) three responses. Considering this, the following criteria were designed:

- All images are referenced within a table that has a random order.
- Of all the images, the algorithm limits its random selection to the first 150 unclassified images. These will be shown to the first users.
- As users match their responses and rate a photo, another will take its place. This will always keep a subset of 150 photos pending classification.
- Additionally, expert agents with registered users within the web application can access to revalidate the images or change the classification (if they consider it relevant).

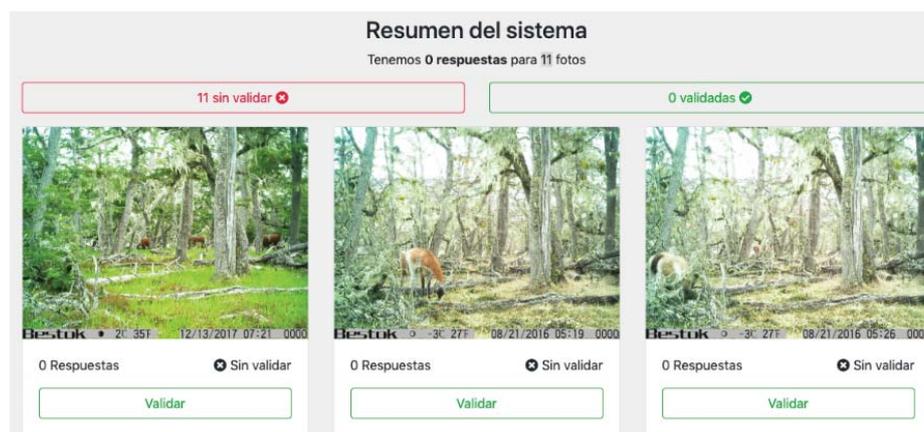


Fig. 4. Summary screen for expert users (with a test database).

During August 2019, it will be promoted in local media and social networks to get people interested in participating. The main public will be inhabitants of the province of Tierra del Fuego. The web application is currently accessible through the link <https://bosques.panalsoft.com>.

3.2 Methodology for classification

With the purpose of marking a coherent course for the resolution of the problem raised, a series of steps to be solved were devised, which will be useful to address the problem in simpler and more manageable tasks.

Preprocessing. The original images of the data set have a dimension of 1280×960 pixels. This size is large in terms of computational calculation, both for training and for evaluation. In that sense, the de facto standard will be followed by reducing the images to 256×256 pixels [7]. Then, we will work on the normalization of colors to unify the tones and facilitate training [13, 14].

Use of known architectures. There are different architectures of deep neural networks, with different number of layers, order, size and purpose [7]. It is intended to use some of the most proven architectures: AlexNet [15], VGG [16], GoogLeNet [17] or ResNet [11].

Pre-training, knowledge transfer. Transfer learning [18] takes advantage of the knowledge obtained to perform a task and applies it to a different and related task. For example, the architectures mentioned above could be pre-trained with the ImageNet data set [19]. Then, it would be re-trained with the set of images of the trap cameras, limiting the specific classifications.

Sort out day and night. This step would involve the first automatic classification, which should be simple due to the high contrast between both situations; saving light-shadow high contrast images. The task has a double purpose, to test the effectiveness of the previous tasks and to introduce the work team in the training of models, adjusting the hyperparameters to achieve more refined results.

Classify presence and absence of animals. This will be the main task and the one that will allow to achieve greater speed in the analysis of the information collected by the trap cameras. In simple words, if achieved correctly, it will be a great filter for experts to focus on the analysis of images with animals. It should be noted that this classification will be carried out in two separate sets provided by the previous task: night and day images. Thus, both the previous classification and this one will be two separate models dedicated exclusively to a particular task.

4 Summary and future work

This work raises the steps to follow to obtain a deep learning model that allows distinguishing (mainly) images with the presence or absence of animals. If a model with acceptable precision is achieved, it would help to speed up the analysis of the Fuegian forest and future projects involving trap cameras. In parallel, a web application was launched that will help classify the original data set, expanding the possibility of achieving a better deep learning model [7].

It is for future work to analyze the classifications achieved with web application and using it to achieve the first models.

Similarly, it would be useful to add new tasks to automate the recognition of the species that appear in the images, the number of animals per species and the use they are making of their environment.

Each of the above tasks should be performed on the sets of nocturnal images with animals and diurnal images with animals. This causes the models to train to be diversified too much. Therefore, it would be interesting to evaluate the possibility of obtaining a model that identifies the absence or presence of animals without requiring a previous filter between night and day.

References

1. Harris G, Thompson R, Childs JL, Sanderson JG: Automatic storage and analysis of camera trap data. *Bull Ecol Soc Am* (2010)
2. Soler RM, Lencinas MV, Bustamante G, Martínez Pastur G: Atributos de la regeneración natural de ñire (*Nothofagus antarctica*) en Tierra del Fuego: beneficios y perjuicios que genera el uso silvopastoril. *Ecosistemas* (2018)
3. O'Connell AF, Nichols JD, Karanth KU: *Camera Traps in Animal Ecology: Methods and Analyses*. Springer, Tokyo (2010)
4. Silveira L, Jacomo AT, Diniz-Filho JAF: Camera trap, line transect census and track surveys: A comparative evaluation. *Biol Conserv* (2003)
5. Fegraus EH, et al. Data acquisition and management software for camera trap data: A case study from the team network. *Ecol Inform* (2011)
6. Mohri M, Rostamizadeh A, Talwalkar A *Foundations of Machine Learning*. MIT Press, Cambridge, MA (2012)
7. Goodfellow I, Bengio Y, Courville A: *Deep Learning*. MIT Press, Cambridge, MA (2016)
8. Bridle JS: Probabilistic interpretation of feedforward classification network out-puts, with relationships to statistical pattern recognition. *Neurocomputing*. Springer, New York (1990)
9. Deng L, Hinton G, Kingsbury B: New types of deep neural network learning for speech recognition and related applications: An overview. 2013 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). IEEE, New York (2013)
10. Sutskever I, Vinyals O, Le QV: Sequence to sequence learning with neural networks. 2014 Advances in Neural Information Processing Systems (NIPS). Neural Information Processing Systems Foundation, La Jolla, CA (2014)
11. He K, Zhang X, Ren S, Sun J: Deep residual learning for image recognition. 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). IEEE, New York (2016)
12. Palmer MS, Packer C: Giraffe bed and breakfast: Camera traps reveal Tanzanian yellow-billed oxpeckers roosting on their large mammalian hosts. *Afr J Ecol* (2018)
13. LeCun YA, Bottou L, Orr GB, Müller KR: Efficient backprop in Neural networks: Tricks of the trade. (2012)
14. Wiesler S, Ney H: A convergence analysis of log-linear training. 2011 Advances in Neural Information Processing Systems (NIPS) (2011)
15. Krizhevsky A, Sutskever I, Hinton GE. Imagenet classification with deep convolutional neural networks. 2012 Advances in Neural Information Processing Systems (NIPS) (2012)
16. Simonyan K, Zisserman A: Very deep convolutional networks for large-scale image recognition. arXiv preprint (2014)
17. Szegedy C, et al: Going deeper with convolutions. 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2015)
18. Yosinski J, Clune J, Bengio Y, Lipson H: How transferable are features in deep neural networks? 2014 Advances in Neural Information Processing Systems (NIPS) (2014)
19. Deng J, et al.: Imagenet: A large-scale hierarchical image database. 2009 IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2009)

Visualization to compare karate motion captures

Dana K. Urribarri¹, Martín L. Larrea^{1,2}, Silvia M. Castro^{1,2}, and Enrico Puppo³

¹ VyGLab (CIC-UNS)

² ICIC (UNS-CONICET)

Depto de Cs. e Ing. de la Computación (UNS)

San Andrés 800, 8000 Bahía Blanca, Buenos Aires, Argentina

{dku, mll, smc}@cs.uns.edu.ar

³ DIBRIS

University of Genoa, Genoa, Italy

enrico.puppo@unige.it

Abstract. Multi-dimensional time series from motion capture (MoCap) provide a rich source of data for human motion analysis, yet they are difficult to process and compare. We address MoCap data related to Karate *katas*, containing predefined sequences of movements, executed independently by several subjects with different timing and speed. We propose a combination of signal processing and data visualization techniques to analyze the misalignment between data from different subjects. We present a web app that implements this proposal, providing a visual comparison of time series, based on Dynamic Time Warping.

Keywords: Dynamic Time Warping, Comparative Visualization, Data Visualization, Multidimensional Data, Motion Capture

1 Introduction

Motion capture (MoCap) data are becoming increasingly important to understand human movements. MoCap data analysis can be used, e.g., to find movements in a database [13], or to compare how different people move [15]. While some tasks are suited for automatic analysis, others may be better performed visually: Bernard et al. [3] emphasize the importance of visual comparison techniques and the challenge to design them. Even though a video rendering provides realistic visualization, comparing movements from different videos is a rather difficult and time-consuming task [14, 15]. Conversely, proper data visualization techniques can provide information about a whole MoCap sequence at a glance.

In this paper, we present a visualization tool based on Dynamic Time Warping, which supports visual comparison between two similar sequences of movements to analyze their time alignment. In order to address codified sequences characterized with neat movements, we take karate *katas* performed by various participants as a case study. Our visualization shows how *source* sequence of movements must be transformed to match a *reference* sequence. This allows, e.g., comparing the performance of karate students with respect to a sequence performed by the teacher, considered to be correct.

2 Previous work

2.1 MoCap visualization

Visualizing a whole MoCap is not an easy task. Li et al. [14] present a review of MoCaps visualization techniques, providing a classification into images summaries, interactive platforms, and animations. Image summaries rely mostly on keyframes [1, 9], i.e., frames that best represent motion in a time window. Malmstrom et al. [15] visualize and compare just a joint from two MoCaps and also include a skeletal view. Hu et al. [9] propose a subject-centered representation called *motion track*, a matrix that only represents angles between skeleton parts, disregarding translation and orientation information; this matrix supports both analyzing one motion sequence and comparing multiple motion sequences. Dimensionality reduction is another technique proposed to visualize time series and MoCaps [20]; however, it is proposed only as a complement for visualization tools, since it helps reveal meaningful information about data clusters but not on interpreting individual information or inspecting changes over time.

As Bernard et al. [3] state, the design of visual strategies to compare MoCaps is also full of challenges regarding data preprocessing, visualization design and interactions. Multiple visualizations focus on comparing data reduced to one 2D or 3D trajectory [4, 8, 9, 13, 22], comparison of patterns [2, 10] or overlapping of keyframes [10]. More recently, Sedmidubsky et al. [19] compare the movements using convolutional neural networks, so the 3D position of the joints are represented as an image. However, they do not intend this image to be part of a visualization system. Just a few visualizations permit to interactively synchronize sequences [22].

2.2 Comparison of 3D trajectories

In the signal processing field, Dynamic Time Warping (DTW) is a very well-known technique to compare two signals; for example, it is commonly used for speech recognition [18]. Bruderlin and Williams [6] stated that signal processing methods, including DTW [5] are suitable for motion data represented with segment lengths and joint angles.

Nowadays, several works use DTW to synchronize and compare movement data [16]. Krüger et al. [13] use DTW as part of a bigger system to find similarities between movements in a database comparing the 3D position of the joints. Vaughan and Gabrys [21] use DTW to analyze trajectory data (x , y , z , force, and pressure) from virtual reality haptic training simulator for epidural needle insertion and evaluate the skill level of trainees; DTW allows them to compare the similarity between two trajectories and to combine together two or more trajectories to produce an average trajectory. Jiang et al. [11] use DTW to evaluate surgical skills for robot-assisted minimally invasive surgery; they propose an additional indicator to include large deviations and velocity changes in different phases of the training tasks into analysis.

3 Data preparation

This work is based on the motion captures corresponding to the Bassai Dai kata of the karate dataset¹ presented in [12] and [17]. The Bassai Dai dataset consists of 20 MoCap sequences by 7 participants with different skills and age. Each MoCap sequence contains information about 25 markers placed on the participant's body (see figure 1). The participants are 2 adults with very high expertise (a Karate teacher and a participant to international Championships); 2 black-belt adults with high expertise; and 4 teenagers with medium expertise. Except for one of the black-belt adults, all are men.

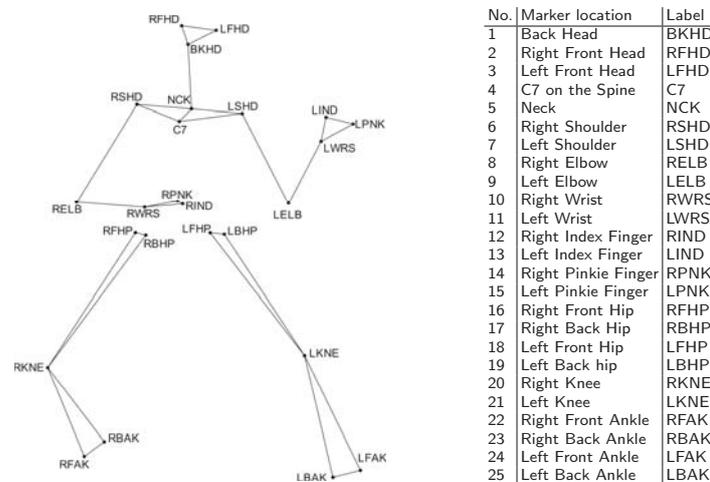


Fig. 1: Labels and positions of the 25 markers that form the MoCaps.

3.1 Data completion

In the skeleton from figure 1, markers are mutually connected with edges that resemble human bones. MoCap sequences are affected by noise and missing data. To complete the missing values of markers, we used a simple strategy considering that the distance between two markers connected by a bone remains constant. For incomplete bones, the position of a missing marker is estimated interpolating the orientation of the bone.

3.2 Data normalization

Data normalization is an inherent problem of sequence comparison. Several approaches exist in the literature. In this work we used an approach similar to the

¹ http://www.infomus.org/karate/eyesweb_dataset_karate_eng.php

one presented by Sedmidubsky et al. [19]; they normalize the data orientation and the size of the skeleton transforming with rotation and translation the 3D position of the joints.

Skeleton alignment. To normalize all the sequences, the skeleton of every frame was transformed in a way that the joint NCK is fixed at point $(0,0,0)$, the segment formed by joints NCK–BKHD is aligned with axis z and the triangle formed by joints NCK–C7–BKHD is aligned with plane yz . Information about the roto-translation of the body with respect to the world reference system is lost; however, to compare the timing of each pose, this information resulted not important.

Space normalization. Finally, in the last step of data preparation, the whole skeleton-aligned MoCap is translated and scaled to fit into the box $(0,0,0)$ – $(1,1,1)$ while the joint NCK is fixed at point $(0.5,0.5,0.5)$. This step reduces distortions due to the different heights of participants.

Data reduction. The basic implementation of DTW needs that both sequences have the same number of frames. Since the captures were taken at a rate of 250Hz and produced sequences of about 25000 frames, every sequence was downsampled, with the Lanczos resampling [7], to have all the same number of frames and a rate of at least 25Hz. In this context, each frame represents a variable amount of time. However, this allows us to compare movements of the sequence and its proportional speeds, regardless of the actual speed at which they were performed.

4 Solution design

We propose a visualization tool that combines different techniques to show how misaligned (hastened or delayed) a *source* MoCap sequence is with respect to a *reference*. In this section A represents the reference MoCap and B represents the source MoCap.

4.1 Time alignment

We use DTW to align in time the two sequences. This technique provides not only a measure of dissimilarity, but also a warping function and, consequently, a misalignment function. DTW computes a mapping between the two sequences that is the optimal match between them. Time warping induced by this mapping provides a measure of dissimilarity between the two sequences. Furthermore, one of the sequences can be transformed by this mapping to obtain two time-aligned sequences.

A dissimilarity measure. The obtain the measure of dissimilarity between to sequences A and B , DTW computes the accumulated distance matrix M between them (see Algorithm 1); that is, it compares every frame of A with every frame of B . The accumulated distance $M_{L,L}$ represents the dissimilarity between the two series and is a measure of the cost of warping one of the sequences to match the other. Euclidean distance is usually used to calculate the distance between frames, and it is the one used in this design; however, any suitable distance function can be used.

Algorithm 1 Accumulated distance matrix M between sequences A and B

Input Two sequences A and B with L frames each one.

Output The accumulated distance matrix M between the two sequences.

```

    ▷ Let  $X_n$  be the frame  $n$  of sequence  $X$ .
1:  $M_{1,1} \leftarrow \text{dist}(A_1, B_1)$            ▷ The distance between the initial frames
2: for all  $i \in [2, L]$  do
    ▷ Accumulated distance between sequence  $B$  and the first frame of sequence  $A$ .
3:    $M_{1,i} \leftarrow \text{dist}(A_1, B_i) + M_{1,i-1}$ 
    ▷ Accumulated distance between sequence  $A$  and the first frame of sequence  $B$ .
4:    $M_{i,1} \leftarrow \text{dist}(A_i, B_1) + M_{i-1,1}$ 
5: for all  $i, j \in [2, L]$  do
    ▷ Minimum accumulated distance between every other pair of frames
6:    $M_{i,j} \leftarrow \text{dist}(A_i, B_j) + \min\{M_{i-1,j-1}, M_{i-1,j}, M_{i,j-1}\}$ 

```

Warping function. The computation of the measure $M_{L,L}$ implies finding the minimal accumulated-distance path from $(1, 1)$ to (L, L) . This *warping path* P is a list of pairs (i, j) that defines a warping from sequence B into sequence A : each frame i in B is warped into a frame j in A . Figure 2 shows the accumulated distance matrix between two sequences and the warping path between them. Ideally, this warping is a function that maps every frame of B into some frame of A . However, in the warping path, there may exist several pairs warping the same frame i of B into different frames of A . Then, a function $F : N \rightarrow N$ that warps sequence B into A is defined as:

$$F(x) = \lfloor \text{mean}_{(x,j) \in P} \{j\} \rfloor \quad (1)$$

4.2 The misalignment function

Given a frame n of sequence B , the function $F(n)$ gives the time frame of A that corresponds to it. The misalignment function $G(n)$ from Equation 2 indicates the distance in frames from n to $F(n)$:

$$G(n) = F(n) - n \quad (2)$$

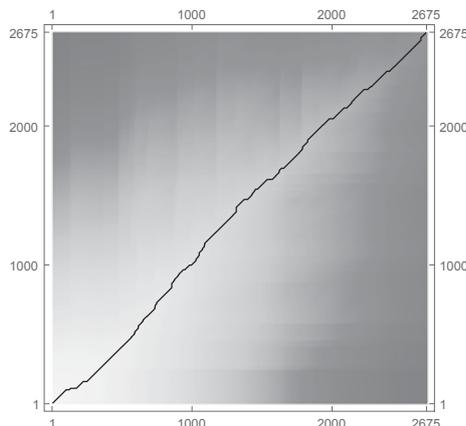


Fig. 2: Accumulated distance matrix and warping path between two sequences of 2675 frames. Graylevel represents distance, the darker the larger. The value of $M_{2675,2675}$ provides a measure of dissimilarity between sequences. Columns represent the frames from sequence B and rows, the frames from sequence A .

If for two consecutive time frames n and $n + 1$, $G(n + 1)$ is greater than $G(n)$ (positive slope), then B is hastened. Conversely, if $G(n + 1)$ is smaller than $G(n)$, then B is delayed. The ideal scenario is when $G(n + 1)$ and $G(n)$ are equal at all frames, which means that B is perfectly synchronized with A .

4.3 Heat map visualization

In a heat map, the color of each pixel encodes information. In this case, the color encodes the position (x, y, z) of each marker: the red, green and blue channels encode the x , y and z coordinates, respectively. Since after space normalization all the positions of the markers lie in the box $(0,0,0)-(1,1,1)$, then the representation as a color is straightforward. One dimension of the image represents the timeline; this means that adjacent pixels in this dimension represent consecutive positions of the same marker. The other dimension represents the markers. In this dimension, adjacency of pixels is not relevant. Figure 3a shows the representation with colors of the pose of Figure 1, and Figure 3b shows a whole MoCap visualized as only one image.

5 The proposed visualization

In order to show the amount of synchronization between two MoCaps, we designed a web visualization that implements the previous design. Figure 4 provides a view of our visualization tool. The left panel shows a table with all available MoCaps to compare; once that the reference and source MoCaps are selected, the *Visualize* button shows the visualization in the right panel.

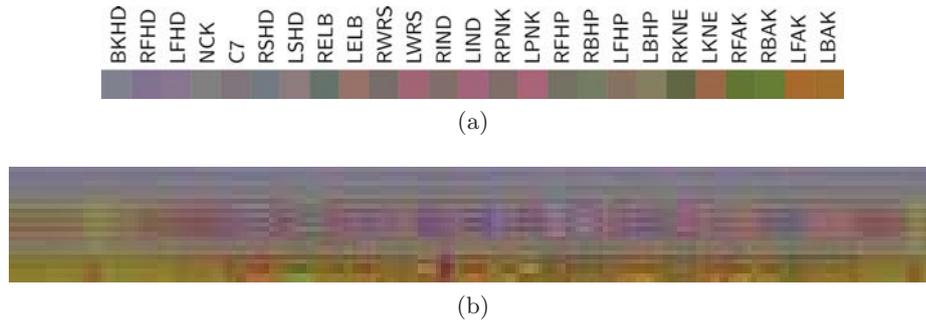


Fig. 3: Heat map visualization of a MoCap. **(a)** Color representation of the pose of Figure 1. The color of each pixel is given by the position of the marker inside the $(0,0,0)$ – $(1,1,1)$ box. **(b)** Color representation of a whole MoCap. The row at the top corresponds to marker BKHD and the one at the bottom, to marker LBAK. Times increases from left to right: the first column on the left corresponds to the first frame of the MoCap, and the last column on the right, to the last frame of the MoCap.

The presented visualizations compare the MoCaps corresponding to the performances of two participants, or two different performances of the same participant; the graphs show differences between the source and the reference:

1. How delayed or hastened each frame of the source is with respect to the reference. A color scale encodes this information: red means delayed, blue means hastened and yellow means on time. The y coordinate of this graph represents the absolute difference between the poses of time-aligned frames. This difference measures the Euclidean distance between markers, considering the collection of N markers as a unique point in a $3N$ -dimensional space. This information gives an idea of how similar the poses are.
2. The misalignment function $G(n)$ between source and reference (Equation 2). The x -axis corresponds to the timeline of the source MoCap. Positive slopes indicate that the source performance is hastened and leaped forward in time with respect to the reference and negative slopes indicate that the performance is delayed. Zero slope means that the source performance is on time.
3. Visualization of both MoCaps with heat maps. The timing of each frame is represented by lines following the same color scales that the first visualization. Two consecutive lines with the same slope indicate that the source performance is on time; however, if the slope of consecutive lines differs, the performance is delayed or hastened.

These three representations of the time alignment of two MoCaps permit to interpret how synchronized the two participants are and how consistent one participant is in performing the sequence of movements.

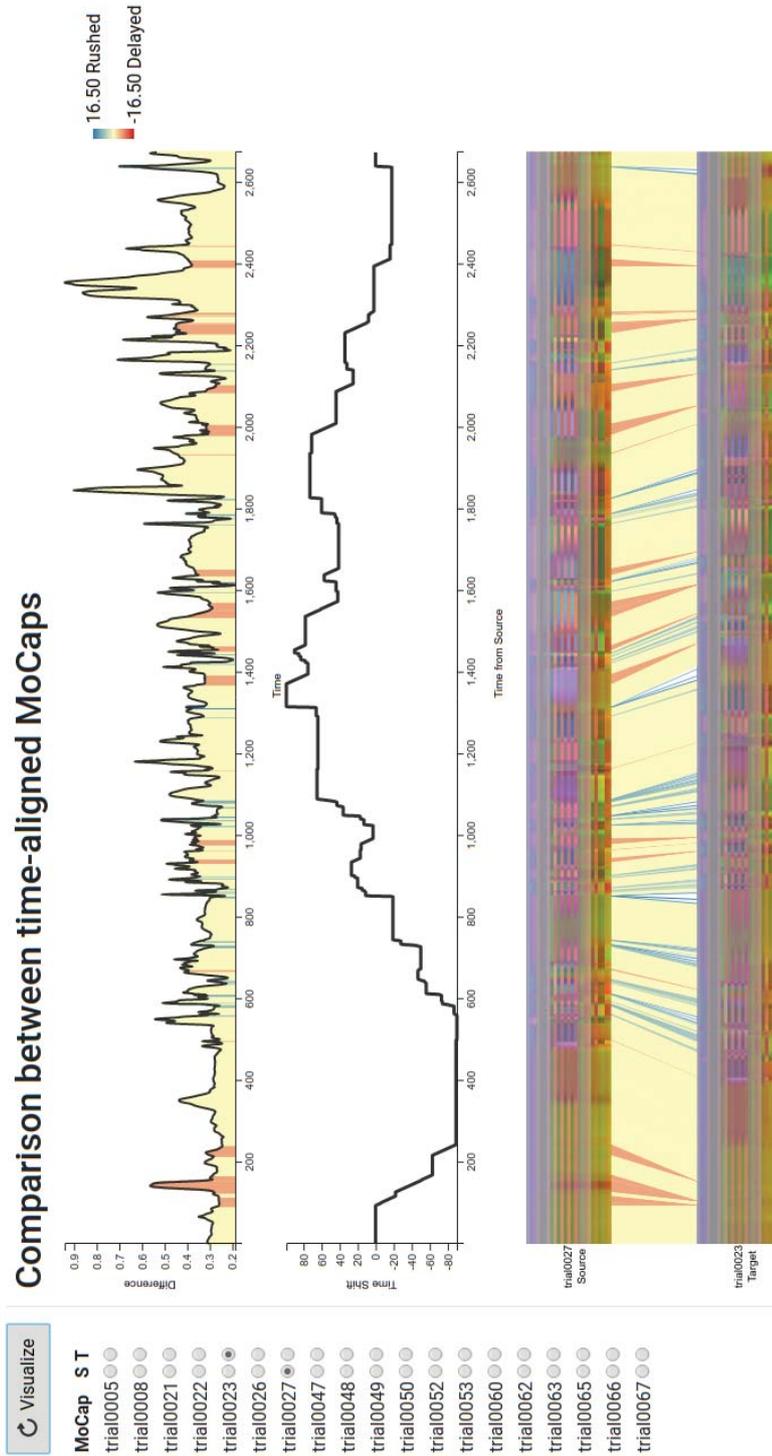


Fig. 4: The left panel shows the list of MoCaps to compare. Results are visualized to the right: At the top, a graph that shows with colors the misalignment between the MoCaps and the absolute difference between time-aligned frames. The middle graph is the misalignment function. The graph at the bottom shows a heat map visualization of the two MoCaps and how they are aligned in time.

6 Implementation notes

The MoCap comparison tool presented in this work was written using Javascript and the D3.js library. A prototype of this applications is available at <http://cs.uns.edu.ar/~dku/mocap-synchrocomparator>. The input data for the application are the preprocessed results of Section 3 *Data preparation* and the DTW for each possible conversion resulting for Equation 1.

7 Conclusions and future work

We have presented a comparative visualization of MoCap sequences that focuses on the time dimension. The visualization gives an overview of the misalignment between two sequences and visual information about when one of them is hastened or delayed with respect to the other. The misalignment color-coding allows the user to spot easily the time frames where the sequences differ. Our tool can be easily extended by showing the corresponding video frame or its related skeleton pose upon user request.

As future work, we plan to improve our tool with suitable interactions for exploration (for example, semantic zoom) and adding the possibility to compare more than two sequences at a time. Regarding the heat map, finding a meaningful order for the markers is somehow desirable. Additionally, we will explore further possibilities with the usage of DTW and comparing the MoCaps considering their actual speed.

Acknowledgments This work was funded by PGI 24/ZN33, Secretaría General de Ciencia y Tecnología, Universidad Nacional del Sur, Bahía Blanca, Argentina; and by the European Union’s Horizon 2020 research and innovation programme under grant agreement n. 824160 (EnTimeMent).

References

1. Assa, J., Caspi, Y., Cohen-Or, D.: Action Synopsis: Pose Selection and Illustration. *ACM Trans. Graph.* 24(3), 667–676 (2005)
2. Bernard, J., Wilhelm, N., Krüger, B., May, T., Schreck, T., Kohlhammer, J.: Motionexplorer: Exploratory search in human motion capture data based on hierarchical aggregation. *IEEE Transactions on Visualization and Computer Graphics* 19(12), 2257–2266 (2013)
3. Bernard, J., Vögele, A., Klein, R., Fellner, D.: Approaches and challenges in the visual-interactive comparison of human motion data. In: *Proceedings of the 12th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications*. vol. 3, pp. 217–224. SciTePress (2017)
4. Bernard, J., Wilhelm, N., Scherer, M., May, T., Schreck, T.: TimeSeriesPaths: Projection-Based Explorative Analysis of Multivariate Time Series Data. In: *Journal of WSCG*, Vol. 20, No. 2. pp. 97–106 (2012)
5. Berndt, D.J., Clifford, J.: Using dynamic time warping to find patterns in time series. In: *Proceedings of the 3rd International Conference on Knowledge Discovery and Data Mining*. pp. 359–370. AAAIWS’94, AAAI Press (1994)

6. Bruderlin, A., Williams, L.: Motion signal processing. In: Proceedings of the 22Nd Annual Conference on Computer Graphics and Interactive Techniques. pp. 97–104. SIGGRAPH '95, ACM (1995)
7. Burger, W., Burge, M.J.: Principles of Digital Image Processing. Core Algorithms. Springer-Verlag London (2009)
8. Cho, K., Chen, X.: Classifying and visualizing motion capture sequences using deep neural networks. CoRR abs/1306.3874 (2013)
9. Hu, Y., Wu, S., Xia, S., Fu, J., Chen, W.: Motion track: Visualizing variations of human motion data. In: 2010 IEEE Pacific Visualization Symposium (PacificVis). pp. 153–160 (2010)
10. Jang, S., Elmqvist, N., Ramani, K.: MotionFlow: Visual Abstraction and Aggregation of Sequential Patterns in Human Motion Tracking Data. IEEE Transactions on Visualization and Computer Graphics 22(1), 21–30 (2016)
11. Jiang, J., Xing, Y., Wang, S., Liang, K.: Evaluation of robotic surgery skills using dynamic time warping. Computer Methods and Programs in Biomedicine 152(Supplement C), 71 – 83 (2017)
12. Kolykhalova, K., Camurri, A., Volpe, G., Sanguineti, M., Puppo, E., Niewiadomski, R.: A multimodal dataset for the analysis of movement qualities in karate martial art. In: Proceedings of the 2015 7th International Conference on Intelligent Technologies for Interactive Entertainment (INTETAIN). pp. 74–78. INTETAIN '15, IEEE Computer Society (2015)
13. Krüger, B., Tautges, J., Weber, A., Zinke, A.: Fast local and global similarity searches in large motion capture databases. In: Proceedings of the 2010 ACM SIGGRAPH/Eurographics Symposium on Computer Animation. pp. 1–10. SCA '10, Eurographics Association (2010)
14. Li, W., Bartram, L., Pasquier, P.: Techniques and approaches in static visualization of motion capture data. In: Proceedings of the 3rd International Symposium on Movement and Computing. pp. 14:1–14:8. MOCO '16, ACM (2016)
15. Malmstrom, C., Zhang, Y., Pasquier, P., Schiphorst, T., Bartram, L.: MoComp: A Tool for Comparative Visualization Between Takes of Motion Capture Data. In: Proceedings of the 3rd International Symposium on Movement and Computing. pp. 11:1–11:8. MOCO '16, ACM (2016)
16. Müller, M.: Information Retrieval for Music and Motion. Springer-Verlag (2007)
17. Niewiadomski, R., Kolykhalova, K., Piana, S., Albornó, P., Volpe, G., Camurri, A.: Analysis of Movement Quality in Full-Body Physical Activities. ACM Trans. Interact. Intell. Syst. 9(1), 1:1–1:20 (2019)
18. Rabiner, L., Juang, B.H.: Fundamentals of Speech Recognition. Prentice-Hall, Inc. (1993)
19. Sedmidubsky, J., Elias, P., Zezula, P.: Effective and efficient similarity searching in motion capture data. Multimedia Tools and Applications 77(10), 12073–12094 (2018)
20. Tanisaro, P., Heidemann, G.: Dimensionality reduction for visualization of time series and trajectories. In: Felsberg, M., Forssén, P.E., Sintorn, I.M., Unger, J. (eds.) Image Analysis. pp. 246–257. Springer International Publishing (2019)
21. Vaughan, N., Gabrys, B.: Comparing and combining time series trajectories using dynamic time warping. Procedia Comput. Sci. 96(C), 465–474 (2016)
22. Wilhelm, N., Vögele, A., Zsoldos, R., Licka, T., Krüger, B., Bernard, J.: FuryExplorer: Visual-Interactive Exploration of Horse Motion Capture Data. In: Visualization and Data Analysis (VDA 2015) (2015)



CACIC 2019

Congreso Argentino de Ciencias de la Computación

XVI Workshop Bases de Datos y Minería de Datos

Coordinadores

Bertone, Rodolfo (UNLP)
Alfonso, Hugo (UNLPam)
Reyes, Nora (UNSL)

Comité Científico

Bender, Cristina (Argentina)	Hasperué, Waldo (Argentina)	Pedreira, Óscar (España)
Brisaboa, Nieves R. (España)	Herrera, Norma (Argentina)	Pollo Cattaneo, María Florencia (Argentina)
Chávez, Edgar (Mexico)	Lanzarini, Laura (Argentina)	Romero, Francisco Pascual (España)
Corbalán, Leonardo (Argentina)	Leguizamón, Guillermo (Argentina)	Seco, Diego (Chile)
Cuevas, Alfredo Simón (Cuba)	Luaces, Miguel R. (España)	Serrano, Jesús (España)
Deco, Claudia Deco (Argentina)	Motz, Regina (Uruguay)	Spositto, Osvaldo (Argentina)
Errecalde, Marcelo (Argentina)	Olivas Varela, José Ángel (España)	
Figuroa, Karina (México)	Paramá Gabía, José Ramón (España)	
Gagliardi, Edilma Olinda (Argentina)	Paredes, Rodrigo (Chile)	

Temas

Modelo y Diseño de Datos
Verificación y Validación de Software
Bases de Datos Objeto
Bases de Datos Difusas
Bases de Datos Temporales, Espaciales y Espacio-Temporales
Bases de Datos Geográficas
Bases de Datos de Conocimiento
Bases de Datos Textuales y Multimedia
Bases de Datos en Web
Base de Datos y XML
Otro tipo de Bases de Datos
Data Warehouse
Data Mining
Datos: Procesamiento y Exploración
Clasificación y Análisis de Asociación
Clustering: Agrupamiento de datos
Detección de Anomalías
Aprendizaje Automático

ANÁLISIS DE TÉCNICAS DE RASPADO DE DATOS EN LA WEB – APLICADO AL PORTAL DEL ESTADO NACIONAL ARGENTINO

Roxana Martínez¹, Rocío Rodríguez¹, Pablo Vera¹, Christian Parkinson¹

¹ Centro de Altos Estudios en Tecnología Informática (CAETI).
Universidad Abierta Interamericana (UAI), Ciudad Autónoma de Buenos Aires, Argentina
{Roxana.Martinez; Rocioandrea.Rodriguez; Pablomartin.Vera;
Christian.Parkinson}@uai.edu.ar

Resumen. Entender la importancia de los datos como herramienta es fundamental para el avance en la tecnología. Garantizar la calidad de los datos no es sencillo, por lo que es vital estar atentos a los procesos de recolección de los mismos. Este trabajo permitirá obtener datos precisos, actualizados y completos. Para ello, se analizan distintas herramientas de raspado de datos (web scraping) que existen en el mercado y se estudian las modalidades de extracción de datos de cada una de ellas en base al caso de estudio del sitio web del Estado Nacional Argentino (Ministerio de Modernización de Argentina), con el fin de extraer datos de los trámites y servicios disponibles para los ciudadanos.

Palabras Claves: Extracción de Datos, Scraping, Big Data, Datos Abiertos.

1. Introducción

La innovación en la tecnología informática se encuentra en pleno auge y cada vez es más estrecha su alcance y relación en los distintos ambientes que utiliza la población para los entornos tecnológicos.

Uno de los puntos a tener en cuenta es que los datos son el centro básico de transformación digital y que sólo pueden brindar su máximo potencial, si se exploran correctamente las innovaciones tecnológicas que los utilizan. “*Los datos ahora se han convertido en el activo más valioso del mundo, más que el petróleo*” [1].

Una de las características más sobresalientes en la actualidad es el concepto de “*algoritmos y la acumulación de grandes volúmenes de información, los así llamados Big Data. Se trata de un proceso de automatización que, lejos de remitir a la vieja idea de lo automático como repetición, genera incesantemente diferencia, establece rangos de acción, permite niveles cada vez mayores de interacción y, por ello mismo, suscita nuevas formas, sutiles y sofisticadas, de control social*” [2].

Con respecto a la innovación de los datos, existen varias observaciones a tener en cuenta, pero las más destacadas son “*las cuatro V de la innovación de los datos: volumen, la cantidad de datos; velocidad, la rapidez con que se crean; variedad, los tipos de datos involucrados; y veracidad, su precisión*” [3]. La comprensión de cada

uno de estos ítems es fundamental para un correcto aprovechamiento de la información.

Los datos abiertos posibilitan el conocimiento abierto, al que las personas pueden acceder sin restricciones, utilizándolo en forma libre y gratuita. Este paradigma busca generar soluciones que sean beneficiosas para los ciudadanos y generar un bien público que se lleve a cabo de manera colaborativa junto con datos abiertos. Los datos abiertos *“son una infraestructura básica para la creación de negocios y de productos y servicios. Para hacer un análisis de su utilidad hay que tener en cuenta que no es igual el acceso a los mismos y su difusión que su reutilización”* [4].

1.1 Web Scraping

El método de web scraping es una técnica utilizada mediante programas de software para extraer información de sitios web. *“Usualmente, estos programas simulan la navegación de un humano en la World Wide Web ya sea utilizando el protocolo HTTP manualmente, o incrustando un navegador en una aplicación”* [5].

Básicamente si se realiza la copia de datos de una página web y se almacenan en una base de datos, se considera que es un proceso de extracción de datos. Si, en lugar de hacerlo de forma manual, se utilizan robots o bots que automatizan todo el procedimiento anteriormente comentado, se hablará de “web scraping”.

El web scraping está muy *“relacionado con la indexación de la web, la cual indexa la información de la web utilizando un robot y es una técnica universal adoptada por la mayoría de los motores de búsqueda. Sin embargo, el web scraping se enfoca más en la transformación de datos sin estructura en la web (como el formato HTML) en datos estructurados que pueden ser almacenados y analizados en una base de datos central”* [6].

1.2 Web Scraping versus Web Crawling

Existen dos técnicas que por lo general se suelen confundir. Si bien, poseen una relación entre ambas y parte de la técnica es similar, hay una diferencia bien marcada en cuanto a una metodología comparada con otra. En el caso de Web Scraping, se conoce como el *“raspado web”* o bien *“raspado de datos”* y en el caso de Web Crawling se conoce como el *“rastreo web”*.

Particularmente el *“rastreo web”* tiene que ver con el proceso de lograr ubicar diversos datos en Internet con el fin de almacenar todas las palabras relevantes o bien palabras claves de búsqueda, y que las mismas se puedan alojar en una base de datos. Mediante esta técnica se tiene como resultado un pequeño repositorio de palabras en un almacenamiento de acceso, para luego lograr manipularlas. Por otra parte, cada una de las palabras extraídas, posee una identificación, que permite saber de qué link (hipervínculo) fue obtenido, por lo que esta técnica, además, permite que se guarden dichos links como parte del proceso de rastreo. Los mismos pueden complementarse con una exploración como parte de la indexación de la base de datos, enlazados a un tema particular. El proceso de rastreo web se lleva a cabo con un software que permite acceder a diversos sitios web, y luego, realizar una lectura de la página en

forma completa para crear un índice de motores de búsqueda. Los principales motores de búsqueda del mercado que utilizan esta técnica son: Google o Bing, los cuales poseen un programa de este estilo, lo que también se conoce más comúnmente como: "araña web" o "bot". Estos programas permiten generar un índice para luego poder realizar consultas contra el índice generado, para así localizar páginas web que coincidan con la consulta de filtros realizadas. Otro de los fines que se le puede dar a este tipo de método es un uso de minería de datos para el análisis de diversas propiedades estadísticas sobre los datos extraídos. Finalmente, se puede lograr un servicio más sofisticado de monitoreo de datos para generar un aviso o alerta a los usuarios que requieran determinada información del rastreo.

Como desafío de esta técnica se puede optar por un modelo de extracción de datos donde se busca información nueva o bien actualizada de manera que surja un comportamiento proactivo de esta actividad.

En resumen, el rastreo web es el método que realizan los motores de búsqueda, es decir, se busca cualquier tipo de información, en cambio, el raspado web está apuntado a determinados sitios web específicos para localizar datos determinados.

A continuación, se muestra la Figura 1 en la que se puede ver una comparativa de pasos básicos entre ambos métodos. En la parte izquierda, se muestra Web Crawler, se observa que se rastrea y se visitan los distintos sitios web para luego armar la lista y así lograr indexarla. Como paso final, se almacenan los datos en una base de datos, la cual es utilizada posteriormente. En la parte derecha, se muestra el Web Scraping, el cual permite analizar un sitio web y así recolectar los datos específicos que son interesantes a identificar del sitio web, como paso final, éstos se almacenan en diversos formatos como ser: XML, SQL o bien formato de archivo Microsoft Excel.

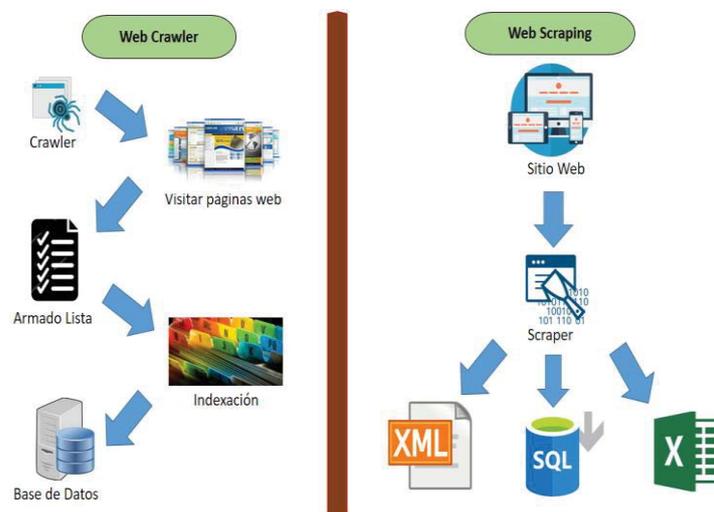


Fig. 1. Comparativa de los pasos, entre el método de Web Scraping y Web Crawling.

2 Herramientas de Raspado de Datos

Las herramientas de raspado web cumplen la función de extracción de información de sitios web, también se los conoce como herramientas de recolección/extracción de datos web. Esta técnica no requiere el método de copiar y pegar, sino que apunta a una forma de extraer los datos en un formato determinado para que luego sea accesible.

La idea central es localizar datos puntuales de los sitios web y almacenarlos para su posterior utilización. Es decir, de un sitio web, puede ser interesante obtener sólo algunos datos, y el resto de lo que se analiza puede ser descartado. Existen algunos servicios de monitoreo frente a los cambios de datos en sitios web, lo que permite mantener actualizado lo previamente recuperado con la técnica de web scraping.

Existen en el mercado diversas herramientas disponibles de raspado de datos. Mediante Google Trends se consideraron las más populares para generar una lista acotada que es la que se presenta en la Tabla 1. Cabe destacar que todas las herramientas listadas tienen licenciamiento gratuito o bien que permitan tener una versión trial para probarlas y compararlas entre sí.

Tabla 1. Herramientas de Raspado de Datos para extraer datos de sitios web.

Herramienta	Descripción
Import.io [8]	Elabora sus propios datasets mediante la importación de los datos de una página web específica. Raspado de miles de páginas web. No requiere de programación, ya que es una plataforma automatizada de extracción. Además, en su versión paga, posee varias funcionalidades más, como ser: Pericias, escalabilidad Integración e Informes.
Web Scraper [9]	Es un Plugin, extensión de Google Chrome llamado Web Scraper. Diseñado para quienes no tienen conocimiento de programación, posee funciones básicas de extracción de dato y no es tan amigable.
Dexi.io [10]	Esta herramienta permite la recopilación de datos de cualquier sitio web. Posee un editor basado en navegador para configurar rastreadores y extraer datos en tiempo real. Es bastante complejo y poco amigable su utilización, es utilizada por usuarios avanzados.
ParseHub [11]	Gestiona tareas de extracción de datos y administra páginas web que usan JavaScript y AJAX. Para el reconocimiento de documentos complejos utiliza aprendizaje automático. Posee una versión web y de escritorio.
Outwit Hub [12]	Se destaca por su interfaz, ya que es muy amigable, lo que la hace fácil de usar, y a su vez, posee grandes características de reconocimiento de datos. Es una herramienta genérica, con un amplio espectro de uso, que va desde la extracción de datos ad hoc sobre temas de investigación específicos hasta la toma extensiva diaria de datos en línea para poblar sitios web.
Scrapestorm [13]	Modo inteligente: basado en algoritmos de inteligencia artificial, esta herramienta identifica de manera inteligente los datos de lista, datos tabulares y botones de paginación sin tener que establecer reglas manualmente.

Herramienta	Descripción
Octoparse [14]	Es utilizada por programadores y analistas de datos. Tecnología de aprendizaje automático. Posee una extensión Cloud Extraction que permite raspado programado en tiempo real. Rotación de IP automática: cuando se configura una tarea de extracción para ejecutarse en la nube, las solicitudes se realizan en el sitio web de destino a través de varias IP, lo que minimiza las posibilidades de ser rastreado y bloqueado.

En la Tabla 2, se muestran los criterios de comparación entre las herramientas de extracción de datos, para ello se tiene en cuenta: Si son herramientas instalables o bien se accede sólo por web; Si la interfaz es amigable y posee varias funcionalidades desde la aplicación; Se identifican los formatos en los que se pueden exportar los datos; y si brinda documentación para manipular la herramienta.

Tabla 2. Comparativa de Herramientas de Raspado de Datos.

Herramienta	Instalable/ Web/Plugin	Interfaz amigable	Funciones	Monitoreo de Cambios	Formatos	Documen- tación	Gratis/ Pago
Import.io [8]	Sitio Web	SI	Muy completas	SI	XLSX; CSV y JSON	SI	Ambos
Web Scraper [9]	Plugin Chrome	SI	Básicas	NO	CSV; JSON	Poca	Gratis
Dexi.io [10]	Sitio Web	NO	Completas	SI	CSV; JSON	SI	Pago (tiene Trial)
ParseHub [11]	Instalable (MAC, Windows, Linux)	SI	Muy Completas	NO	JSON; EXCEL	SI	Ambos
Outwit Hub [12]	Instalable (MAC, Windows, Linux)	SI	Básicas	SI	EXCEL; JSON; XML; CSV; SQLite; TXT; HTML; SQL	SI	Gratis
Scrapestorm [13]	Instalable (MAC, Windows, Linux)	MEDIA	Muy Completas	SI	EXCEL; CSV; TXT; HTML; MySQL; MongoDB, SQL Server, PostgreSQL y WordPress	SI	Ambos
Octoparse [14]	Instalable Windows Se requiere Framework (.NET3.5 SP1)	SI	Muy Completas	SI	EXCEL; JSON; CSV; HTML; DB	SI	Ambos

2.1 Consideraciones ante la elección de una técnica de Web Scraping

Ante la elección de una mejor técnica de Web Scraping, es necesario tener presente los siguientes puntos:

- a) ¿Sólo es importante extraer información de una página web?
- b) Los datos extraídos: ¿serán utilizados directamente en otra página web, o bien serán analizados en una PC?
- c) ¿Es importante hacer una sincronización de actualización con el sitio web que se está realizando la técnica de extracción de datos?

Si se responde a cada una de estas preguntas se logrará entender la opción que más conviene para llevar a cabo para una minería de datos. Cada una de estas preguntas, permite identificar que se necesita para realizar un raspado de datos.

Si se necesita recuperar datos una única vez y luego procesar dicha información, entonces, es posible utilizar las herramientas de los navegadores, ya que permiten obtener el resultado en un formato CSV y XML, lo cual es totalmente manejable.

Si lo que se pretende es utilizar la extracción de datos como herramienta para tener actualizada la información y como siguiente paso, utilizar ésta en una página web, entonces se puede pensar en dos opciones: a) Utilizar un servicio web como ser: import.io [8], Dexi.io [10], ParseHub [11] o bien otra opción que permita el raspado constante de la información en un sitio web. Cada uno de ellos requiere conocimiento de lenguajes de programación, ya que, en este caso, no se utilizan las herramientas de los navegadores (plugin). b) Otra opción es el scraping web a medida, por medio de programación. Es decir, se realiza el programa de software de raspado de página web a medida, con el fin de extraer sólo los datos que sean necesarios, luego es fundamental realizar la integración y esta información pueda ser actualizada en un determinado período. Tanto para este caso, como para el anterior, también es necesario tener conocimientos de lenguaje de programación y de integración de API. Cabe destacar que en este trabajo se realizó un análisis de las herramientas completas, no así de las opciones de extracción de datos que brinden servicio de API únicamente.

3 Caso de Estudio

A continuación, se propone un caso de estudio en el que se desea extraer datos del sitio web del Estado Nacional Argentino (Ministerio de Modernización de Argentina) [15]. Los datos a extraer son puntualmente los 14 trámites y servicios que pueden realizar los ciudadanos argentinos desde el sitio web, catalogados por categorías junto con su descripción y link. Se extraerán los datos mediante la técnica de raspado de datos. Para esto se aplicarán las distintas herramientas presentadas en la sección anterior. En la Figura 2 se muestran los 14 trámites y servicios a extraer con las herramientas de raspado de datos.



Fig. 2. Trámites y servicios del sitio web del Estado Nacional Argentino (Ministerio de Modernización).

4 Resultados de la comparativa

En base a los resultados arrojados, se muestra a modo de resumen, el análisis de las herramientas de raspado de datos teniendo en cuenta los siguientes criterios en la Tabla 3: a) ¿Logra extraer los 14 trámites y servicios disponibles?; b) Tiempo de extracción tomando como “Rápido” a un rango de 1 segundo a 5 segundos; “Medio” en un rango de 6 segundo a 10 segundos; y “Alto” en un rango de más de 10 segundos; c) Nivel de trabajo de depuración para obtener los datos limpios extraídos de la herramienta, tomando como “Alto”; “Medio” y “Bajo”; d) Dificultad a nivel de extracción.

Para comparar los resultados de la extracción de datos se ha exportado todo a CVS que es el formato común de todas las herramientas comparadas. Algunas herramientas permitían exportar a JSON (tal como se indicó en la tabla 2) pero para comparar lo obtenido fue necesario exportar a CVS que todas lo tienen habilitado.

Tabla 3. Comparativa de Resultados de las Herramientas de Raspado de Datos.

Herramienta	Efectividad de Extracción	Tiempo [Rápido/Medio/Lento]	Separación de los datos	Depurar resultado [Alto/Medio/ Bajo]	Dificultad de Extracción [Alta/Media/Baja]
<i>Import.io</i> [8]	14 registros	MEDIO	“,”	MEDIO: Se muestra el texto plano exportado correctamente. Se exporta una columna adicional en la que repite el URL. Se exportan 2 columnas extras.	BAJA
<i>Web Scraper</i> [9]	43 registros	RÁPIDO	“,”	ALTO: Se extrae más datos de los solicitados, como ser números de IDs e información redundante.	MEDIA
<i>Dexi.io</i> [10]	14 registros	MEDIO	“,”	MEDIO: Se muestran caracteres especiales en lugar de acentos. Se exporta una columna adicional de control de errores. No se extrajeron links.	MEDIA
<i>ParseHub</i> [11]	14 registros	RÁPIDO	“,”	MEDIO: Se muestran caracteres especiales en lugar de acentos. Se exporta una columna adicional que repite el URL. Tiene problemas si los datos originales tienen coma ya que lo toma como separador de campo.	BAJA
<i>Outwit Hub</i> [12]	16 registros	RÁPIDO	“,” y “;”	MEDIO: Tiene dos separadores “,” (coma) y “;” (punto y coma); pero se extrae información de más, provenientes de la sección anterior.	BAJA
<i>Scrapestorm</i> [13]	17 registros	MEDIO	“,”	ALTO: Se muestran caracteres especiales en lugar de acentos. Se exportan registros de más que surgen del raspado.	BAJA
<i>Octoparse</i> [14]	1 registro (con 15 tramites, uno duplicado)	MEDIO	“,”	ALTO: Todos los datos son exportados juntos en un mismo registro. No se comprende bien la división si los datos extraídos tienen “,” (coma). No se extrajeron links.	MEDIA

Para seleccionar una de estas herramientas, se descartan aquellas que no logran recuperar los 14 registros. Luego se analiza también los inconvenientes que se presentan al depurar el resultado obtenido, por ejemplo, aquellas herramientas que tienen problemas con los acentos ó las comas del texto original. El caso de estudio

considerado, tiene el nombre de un servicio con dos comas, y así también tiene acentos en nombres o descripciones; lo cual genera problemas en la exportación con algunas herramientas. Algunas herramientas traen registros vacíos, duplicados, etc. Lo que no permite una extracción directa de lo obtenido. Como puede observarse en la tabla 3, sólo 3 herramientas permitieron traer correctamente los 14 registros esperados. Por los resultados obtenidos se selecciona como herramienta para la extracción de datos a Import.io [8], ya que, si bien es una herramienta que requiere de un trabajo de limpieza de datos de un nivel medio, posee una buena identificación de los registros extraídos para su reutilización e identificación de datos abiertos. A comparación de las otras herramientas, no presenta caracteres extraños en los acentos, además, permite una extracción de datos en forma amigable y con múltiples funciones, a través de una licencia gratuita. Por otra parte, se destaca el tiempo de rapidez en la extracción. En la Figura 3 se muestran los datos raspados de la herramienta Import.io [8], identificando los 14 trámites y servicios del sitio web.

#	Tramites y Servicios	New column
1	Pareja y familia Trámites relacionados con estar en pareja, tener un hijo o enfrentar el fallecimiento de un ser querido.	Pareja y familia
2	Emergencias Te ofrecemos una lista de números y consejos a seguir.	Emergencias
3	Adultos mayores Consejos, servicios y orientación para realizar trámites de jubilaciones y pensiones.	Adultos mayores
4	Trabajar Todos los servicios por si sos empleado, monotributista, querés capacitarte o estás buscando trabajo.	Trabajar
5	Beneficios para artistas Beneficios para artistas y agenda de actividades para disfrutar.	Beneficios para artistas
6	Argentinos en el mundo Si te encontrás en otro país o vas a hacer un viaje, chequeá información útil.	Argentinos en el mundo
7	Estudiar Niveles y modalidades educativas, programas, becas y reválidas de títulos.	Estudiar
8	Documentación ciudadana Te guiamos en trámites tales como sacar el DNI o pasaporte y cambio de género, entre otros.	Documentación ciudadana
9	Consumidores Información sobre tus derechos, herramientas para comprar o contratar servicios, y pautas para hacer un reclamo.	Consumidores
10	Emprender, innovar, crecer Herramientas de financiamiento y capacitación para emprendedores y PYMEs.	Emprender, innovar, crecer
11	Personas con discapacidad Servicios para atención de la salud, asignaciones familiares, viajar gratis y más trámites.	Personas con discapacidad
12	Violencia y abuso Ante una situación de violencia o abuso no dudes en llamar y denunciar.	Violencia y abuso
13	Acceder a una vivienda Información sobre créditos, programas de vivienda para compra o mejoras y derechos de inquilinos y propietarios.	Acceder a una vivienda
14	Tránsito y automotor Información para circular en vía pública y para comprar, registrar, renovar, asegurar o vender tu auto o moto.	Tránsito y automotor

Fig. 3. Se muestran los trámites y servicios raspados del sitio web.

5 Conclusiones y Trabajos Futuros

Si bien existen diversas herramientas en el mercado no todas han dado por resultado una extracción correcta de los registros a recuperar en el caso de estudio seleccionado. Por lo que es importante analizar las herramientas existentes en cuanto al resultado generado, lo cual se realizó analizando los CSV exportados. Luego se contemplan otros parámetros como la dificultad y el tiempo de extracción, siendo el parámetro más decisivo la efectividad en el resultado obtenido. Como se trató en este

trabajo, el scraping es una técnica que se utiliza para extraer datos de cualquier sitio web, siempre y cuando se arme el mapa de los datos a extraer. Mediante la extracción de datos de los sitios web gubernamentales, se puede exportar a un formato más sencillo y así lograr analizarlos y cruzarlos con mayor facilidad, disponibilizándolos como datos abiertos. Para recopilar automáticamente y mostrar esta información, los scrapers, permiten la manipulación de los datos con el fin de obtener un almacenamiento de datos abiertos y que puedan ser compartidos para su posterior análisis estadístico. Estos datos compartidos podrían ofrecerse en un entorno donde a través de crowdsourcing los usuarios puedan escribir comentarios sobre estos trámites o servicios, realizar aportes en caso de detectar errores en la información ofrecida. La manipulación de estos datos extraídos es una línea de trabajo futuro.

Referencias

- [1] Sinead Garvan, BBC, “Cambridge Analytica: cómo Netflix retrata el mayor escándalo de privacidad en las redes sociales en *Nada es privado*”, Disponible en: <https://www.bbc.com/mundo/noticias-49122905>
- [2] Costa, F., & Rodríguez, P. (2018). ALGORITMOS, BIG DATA Y AUTOMATIZACIÓN SOCIAL. AVATARES de la Comunicación y la Cultura, (15).
- [3] Estrada, J. C. H., Silva, I. A. M., & Páez, J. O. B. (2018). Big Data: Ventajas y desventajas-aplicaciones y tecnologías para implementar el servicio. COMITÉ CIENTÍFICO CICOM 2018, 44.
- [4] Abella, A., Ortiz-de-Urbina-Criado, M., & De-Pablos-Heredero, C. (2018). Indicadores de calidad de datos abiertos: el caso del portal de datos abiertos de Barcelona. El profesional de la información (EPI), 27(2), 375-382.
- [5] Khabsa, M., & Giles, C. L. (2014). The number of scholarly documents on the public web. PLoS one, 9(5), e93949.
- [6] Andrés, O. R., Pulido, J. R. G., Guillermo, A., & Morales, J. R. H (2019). Recuperación de metadatos e indicadores de impacto para publicaciones científicas mediante servicios de Google académico.
- [7] Google Trends, “Descubre qué está buscando el mundo”, Disponible en: <https://trends.google.com/trends/?geo=US>
- [8] Import.io, “Proporcionar los datos web que informan las decisiones comerciales cotidianas”, Disponible en: <https://www.import.io>
- [9] Web Scraper, “More than 250,000 users proud of using our solutions”, Disponible en: <https://webscraper.io/>
- [10] Dexi.io, “Web Automation Software [Scraping ETL API AI]”, Disponible en: <https://dexi.io/>
- [11] ParseHub, “A web scraping tool that is easy to use”, Disponible en: <https://www.parsehub.com/>
- [12] Outwit Hub, “OutWit Hub explores the depths of the Web for you, automatically collecting and organizing data and media from online sources”, Disponible en: <https://www.outwit.com/products/hub/>
- [13] Scrapestorm, “AI-Powered Visual Web Scraping Tool”, Disponible en: www.scrapestorm.com
- [14] Octoparse, “Extraiga fácilmente cualquier dato web”, Disponible en: <https://www.octoparse.com/product>
- [15] Estado Nacional Argentino (Ministerio de Modernización de Argentina), “Argentina.gob.ar”, Disponible en: <https://www.argentina.gob.ar/>

Consultas Métrico Espaciales

Adrián Planas¹, Andrés Pascal¹, Norma Herrera²

¹ Univ. Tec. Nacional, Fac. Concepción del Uruguay, Dpto. Sistemas de Información
pladnic@gmail.com, andrespascal2003@yahoo.com.ar

² Univ. Nac. de San Luis, Departamento de Informática, San Luis, Argentina
nherrera@unsl.edu.ar

Abstract. Los espacios métricos permiten modelar bases de datos que soportan búsquedas por similitud, es decir, búsquedas de objetos parecidos a uno dado. Las bases de datos espaciales permiten almacenar y recuperar eficientemente datos que poseen alguna componente espacial. Existen aplicaciones donde resulta de interés realizar búsquedas por similitud pero teniendo en cuenta también el componente espacial. Este tipo de consultas no puede resolverse eficientemente ni con índices espaciales, ni con índices métricos. En este artículo abordamos el estudio de estas consultas con el fin de formalizarlas y proponer un método eficiente para su resolución. Para ello, presentamos el MeTree, una combinación del índice métrico FQT con el índice espacial R-Tree que permite consultas combinando ambos aspectos.

Keywords: Consultas Métrico-Espaciales, Búsquedas por Similitud, Índices Métricos, Índices Espaciales.

1 Introducción

En las últimas tres décadas, las bases de datos han incorporado paulatinamente la capacidad de almacenar tipos de datos no estructurados tales como imágenes, sonido, texto, video, elementos geométricos, etc. La problemática de almacenamiento y búsqueda en éstas difiere notablemente de la correspondiente a las bases de datos clásicas. Estos tipos de datos no poseen una longitud fija, por lo cual son más difíciles de organizar dentro de una estructura, normalmente no pueden ser ordenados, y la búsqueda exacta carece de interés en este ámbito. Es en este contexto donde han surgido nuevos modelos de bases de datos capaces de cubrir eficaz y eficientemente las necesidades de almacenamiento y búsqueda de estas aplicaciones. Entre estos modelos se encuentran las bases de datos espaciales, que permiten mantener un registro de ubicación de un objeto en un espacio n-dimensional, y los espacios métricos, que constituyen un modelo genérico que permite las búsquedas por similitud.

El objetivo de las búsquedas por similitud [1] es encontrar objetos que poseen características similares a uno dado bajo cierto criterio. Esta clase de funcionalidad ha sido utilizada en muchas áreas de la informática. Por ejemplo, en reconocimiento de patrones, las consultas por similitud se pueden usar para clasificar nuevos objetos de acuerdo a objetos cercanos ya clasificados; en sistemas de recomendación se pueden utilizar para generar recomendaciones personalizadas basadas en las preferencias del usuario; en bases de datos de imágenes se pueden usar para búsquedas por contenido.

Dado que los tipos de datos de estos sistemas pueden variar considerablemente (imágenes, cadenas, secuencias de proteínas, sonido, texto libre, etc), es deseable contar con un modelo genérico que los abarque, y que permita diferentes funciones de comparación. En este aspecto, el modelo más utilizado es el definido por los Espacios Métricos, que permiten cualquier función de similitud, siempre que cumpla con ciertas propiedades.

Un espacio métrico es un par (U, d) donde U es un universo de objetos y $d: U \times U \rightarrow R^+$ es una función de distancia definida entre los elementos de U que mide la similitud (disimilitud, en realidad) entre ellos; esto significa que a menor distancia, más cercanos o similares son los objetos. Esta función d cumple con las propiedades características de una función métrica:

- (a) $\forall x, y \in U, d(x, y) \geq 0$ (positividad)
- (b) $\forall x, y \in U, d(x, y) = d(y, x)$ (simetría)
- (c) $\forall x \in U, d(x, x) = 0$ (reflexividad)
- (d) $\forall x, y, z \in U, d(x, y) \leq d(x, z) + d(z, y)$ (desigualdad triangular)

Esta última propiedad, la desigualdad triangular, es de suma importancia para descartar elementos durante la búsqueda. La base de datos será entonces un subconjunto finito $X \subseteq U$ de cardinalidad n . Una de las consultas típicas en este modelo es la búsqueda por rango, que se denota $(q, r)_d$. Dado un elemento $q \in U$ al que llamaremos *query*, y un radio de tolerancia r , una búsqueda por rango consiste en recuperar los objetos de X que estén a distancia a lo sumo r de q , es decir:

$$(q, r)_d = \{x \in X / d(q, x) \leq r\}$$

Por otro lado, el modelo espacial ha aumentado su uso significativamente desde la década de los 90', principalmente como soporte para los Sistemas de Información Geográfica (SIG). Con el crecimiento de Internet, la cantidad de objetos que poseen una ubicación espacial se ha incrementado exponencialmente. Por ejemplo, los dispositivos móviles han permitido la generación de cantidades enormes de datos georeferenciados que se comparten a través de la Web; principalmente fotos con ubicación.

Las bases de datos espaciales están formadas por información estructurada a la que se le añaden objetos geométricos: puntos, polilíneas o polígonos, bajo un sistema de referencia predefinido. Dichas bases pueden contener cientos de miles o millones de estos objetos sobre los cuales se realizan consultas con operaciones tales como intersección, adyacencia, inclusión y muchas otras. Tales operaciones suelen ser costosas desde el punto de vista computacional, por lo cual se requieren métodos de acceso que disminuyan considerablemente la cantidad de comparaciones necesarias para resolver una consulta.

Existen aplicaciones donde resulta de interés realizar búsquedas por similitud pero teniendo en cuenta también la componente espacial. Un caso típico son los SIG, donde los elementos con ubicación espacial en muchos casos requieren ser consultados también por similitud. Por ejemplo, en un mapa con fotos de edificios y construcciones ampliamente conocidas, sería de interés, dada una imagen de una construcción, encontrar las imágenes similares dentro de un área geográfica particular. O en lugar de una imagen, simplemente buscar por similitud el nombre de un punto de interés más cercano a la ubicación donde uno se encuentra, ya que en ocasiones no se conoce cómo se escribe. Estas consultas implican una búsqueda que tiene en cuenta tanto el aspecto espacial como la similitud del elemento que se consulta y, en el caso de segundo ejemplo, son muy comunes cuando se utilizan Sistemas de Posicionamiento Global (GPS). Un caso particular de este problema, la búsqueda de documentos geográficos, ha sido estudiado previamente dando lugar a índices tales como el IR-Tree [2], pero que están diseñadas sólo para documentos.

Si bien estas consultas se pueden resolver utilizando índices espaciales y métricos por separado para luego encontrar la respuesta final, es mucho más eficiente contar con métodos de acceso que estén diseñados específicamente para resolverlas. En este artículo se presenta

una solución general a este problema, que sirve para cualquier objeto que sea consultado por similitud y ubicación a la vez.

Este artículo está organizado de la siguiente manera. En la Sección 2 se presenta un breve resumen del trabajo relacionado. En las Secciones 3 y 4 planteamos el problema de consultas métrico-espaciales y proponemos un primer índice para resolverlas con eficiencia, el MeTree. En la Sección 5 se muestra su evaluación experimental, y por último en la Sección 6 se presentan las conclusiones y el trabajo futuro.

2 Trabajo Relacionado

En esta sección se resumen los desarrollos más importantes en cuanto a los métodos de acceso métricos y espaciales.

2.1 Índices Métricos

Existen varios métodos de acceso diseñados para acelerar la búsqueda por similitud en espacios métricos genéricos. En general, pueden ser clasificados en dos categorías: métodos basados en particiones compactas [3], [4], [5], [6] y métodos basados en pivotes [7], [8], [9], [10]. Los métodos basados en particiones compactas dividen el espacio en regiones representadas por centros e intentan descartar las regiones alejadas del objeto que se consulta. Mientras que los basados en pivotes almacenan las distancias precalculadas de cada objeto hacia los pivotes y las utilizan con el mismo de fin. En ambos casos se utiliza la desigualdad triangular para reducir la cantidad de elementos a comparar con la consulta. En general, los métodos basados en pivotes obtienen mejores resultados en cuanto a la performance de las consultas.

Dentro de los basados en particiones compactas, el BST [11], [12] es un árbol binario construido recursivamente. Utiliza un centro con un radio de cobertura para representar cada partición. El GHT [13] utiliza dos centros por cada nodo del árbol y agrupa los elementos de acuerdo al centro más cercano a cada uno de ellos. GANT [14] es una generalización del GHT. Utiliza particiones de Voronoi del espacio. Existe una versión dinámica de este último, llamada EGANT [15]. SAT [16] utiliza un modelo complementario a los diagramas de Voronoi, los grafos de Delaunay. También existen extensiones de SAT, dinámicas, y para memoria secundaria [17], [18]. El M-tree [4] es un árbol balanceado optimizado para memoria secundaria, que surge como una adaptación natural de la familia de los B-Trees a más de una dimensión. Existen muchas variantes del M-Tree, tales como Slim-Tree [6], DBM-Tree [19] y CM-Tree [20]. El D-index [5] es una estructura que utiliza una función hash para mapear objetos en buckets. LC [3] utiliza una lista de clusters que mejora la eficiencia en la búsqueda a costa de hacer menos eficiente su construcción. BP [21] es un árbol no-balanceado pensado para espacios métricos de alta dimensionalidad.

Respecto a los métodos basados en pivotes, AESA [22] emplea una tabla que registra todas las distancias entre los objetos de la base de datos. Para reducir el tamaño de dicha tabla, se han propuesto distintas variantes. Por ejemplo, LAESA [9] solo guarda las distancias hacia un conjunto de pivotes seleccionados. EP [23] selecciona un conjunto de pivotes sin redundancia, que cubren la base de datos completa. Clustered Pivot-Table [24] agrupa las distancias precalculadas para mejorar aún más la eficiencia de las búsquedas. El BKT [25] fue uno de los primeros índices métricos basados en pivotes, y está diseñado para distancias discretas. Es un árbol donde cada nodo contiene un pivote distinto y todos los elementos que

se encuentran a la misma distancia de pivote se ubican en el mismo nodo hijo. El FQT [26] es similar al BKT, pero utiliza el mismo pivote para todos los nodos del mismo nivel, reduciendo así la cantidad de comparaciones entre la consulta y los pivotes. VPT [10] es un árbol binario diseñado para distancias continuas y su versión r-aria es el MVPT [27].

Existen también métodos híbridos, que combinan particiones compactas con pivotes, por ejemplo el PM-Tree [28], que utiliza pivotes sobre regiones definidas por un M-Tree, o el M-Index [29], que agrupa objetos utilizando distancias precalculadas hacia sus pivotes más cercanos.

2.1 Índices Espaciales

El procesamiento de consultas espaciales implica la ejecución de operaciones geométricas complejas y costosas. Considerando que las bases de datos espaciales suelen contener grandes cantidades de objetos geométricos, realizar un recorrido secuencial para resolver una consulta espacial no es una solución práctica en la mayoría de los casos, por lo cual en las aplicaciones reales es necesario el uso de índices espaciales.

Los métodos de acceso espacial también pueden clasificarse en dos categorías [30]: estructuras dirigidas por el espacio, y estructuras dirigidas por los datos. Las primeras están basadas en la partición de un espacio 2D en regiones rectangulares. Los objetos se mapean en las regiones de acuerdo a algún criterio geométrico. En el segundo caso las particiones están basadas en la distribución del conjunto de objetos que se está indexando.

El Grid File [31] divide el espacio en celdas fijas para indexar puntos. Cada celda está asociada a una página de disco, donde se almacenan secuencialmente los objetos contenidos en dicha celda. Si la celda se llena, la celda se divide en dos. Existen variantes del mismo para indexar rectángulos. El QuadTree [32] subdivide el plano en cuatro cuadrantes del mismo tamaño, y realiza esta misma operación sobre cada cuadrante, dando como resultado un árbol 4-ario. Cada nodo representa a todos los objetos contenidos en su cuadrante. El QuadTree ha sido utilizado ampliamente tanto para la búsqueda espacial como para distintas tareas de procesamiento de imágenes. El Bitplane Quadtree (BQ-Tree) [33] es una variante diseñada para datos geoespaciales de gran escala. El K-D-Tree [34] almacena puntos k-dimensionales subdividiendo el espacio en forma alternativa en paralelepípedos rectángulos ortogonales a cada eje de coordenadas. Existe una familia de métodos llamada “Space Filling Curve”, que se basan en funciones que transforman un espacio n-dimensional en una sola dimensión, conservando ciertas propiedades de proximidad. Un ejemplo es el Hilbert Space Filling Curve [35].

Respecto a los métodos dirigidos por los datos, los más importantes son la familia de índices R-Tree [36], que constituyen una generalización del B-Tree a más de una dimensión. Son árboles M-arios de rectángulos donde cada nodo contiene espacialmente a todos los elementos del subárbol del que es raíz. El árbol es balanceado y cada nodo interior salvo la raíz, contiene entre m y M hijos, y se almacena en una página de disco. Son los índices espaciales más utilizados actualmente por los motores de bases de datos comerciales.

3 Modelo Métrico-Espacial

Las aplicaciones donde tienen sentido las consultas métrico-espaciales tienen las siguientes características:

- No se pueden realizar búsquedas exactas sobre los objetos: los elementos de la base de datos no tienen un identificador (o un grupo de atributos) que se pueda utilizar como clave de búsqueda, o si existe, no se conoce en el momento de la consulta.
- Los objetos tienen una ubicación (y/o forma) espacial.
- Los resultados de una consulta tienen que satisfacer requisitos tanto de similitud como de posición espacial.
- La base de datos contiene una cantidad suficientemente grande de objetos o el tiempo de respuesta ante una consulta debe ser suficientemente reducido como para que no tenga sentido realizar una búsqueda secuencial.

Sea U el universo de objetos válidos, el modelo Métrico-Espacial se define mediante el par (U, d) , donde para todo $o \in U$, la función $s(o) \in S$ devuelve el componente métrico del objeto y $e(o) \in E$ su aspecto espacial. La función métrica d , es la medida de disimilitud y está definida como $d: S \times S \rightarrow R^+$. Por ejemplo, si $o \in U$ es un punto de interés en un mapa, $s(o)$ podría ser una cadena que representa su nombre o una foto de su frente, mientras que $e(o)$ será el punto que indica su ubicación.

Una consulta por rango métrico e intersección espacial, se denota a través de la 3-upla $(q, r, g)_d$ y se define formalmente de la siguiente manera:

$$(q, r, g)_d = \{o \in X / d(s(o), q) \leq r \wedge \text{intersects}(e(o), g)\}$$

siendo $X \subseteq U$, la base de datos, q el aspecto métrico de la consulta, r el radio de búsqueda que representa el valor máximo de disimilitud aceptada, y g su aspecto geométrico (punto, polilínea o polígono).

Una forma trivial de resolver una consulta métrico-espacial es el uso de dos índices, uno métrico y el otro espacial. Luego, ante una consulta $(q, r, g)_d$, se procede de la siguiente manera:

1. Realizar la búsqueda por similitud $(q, r)_d$ sobre el índice métrico, devolviendo el conjunto L como resultado,
2. Realizar una búsqueda de los elementos que se intersectan con g , sobre el índice espacial, devolviendo como resultado el conjunto M ,
3. Por último, realizar la intersección $L \cap M$ para obtener el resultado final.

La desventaja de esta solución es que no aprovecha la información métrica y espacial al mismo tiempo para el descarte de elementos. Una mejor estrategia es el diseño de un índice que integre ambos aspectos y permita consultas métricas, espaciales y métrico-espaciales.

4 Método de Acceso Métrico-Espacial: el MeTree

En este trabajo se presenta un primer índice métrico-espacial, el MeTree, diseñado en base a una variante del índice métrico FQT (que permite distancias continuas) y a la familia de índices espaciales R-Tree. El MeTree es un árbol r-ario en el cual cada nodo contiene un rectángulo que contiene espacialmente a todos sus hijos, tal como el R-Tree. Pero además, cada nodo tiene un intervalo métrico que representa el valor mínimo y máximo de las distancias de todos los elementos del subárbol hacia un pivote correspondiente al nivel del nodo (como un FQT). La altura del árbol es fija en principio, y está determinada por la cantidad de pivotes con la cual se decide indexar. Sin embargo, esta cantidad puede ser extendida en cualquier momento, agregando nuevos pivotes y por lo tanto, nuevos niveles. Una hoja del árbol contendrá apuntadores a objetos que sean similares y que además, estén cercanos espacialmente.

Un problema importante es que puede haber elementos similares muy alejados espacialmente. En los niveles inferiores del árbol esta situación produciría rectángulos demasiado grandes, afectando considerablemente a la eficiencia del índice. Para resolver este problema se establecieron restricciones a cumplir para que dos elementos puedan pertenecer a la misma hoja, y se decidió permitir que los intervalos métricos de nodos hermanos no sean excluyentes, como así tampoco los rectángulos.

Un nodo interior del MeTree es una 3-upla $(i_m, f_m, rect)$, donde i_m, f_m es el intervalo métrico y $rect$ el rectángulo que contiene espacialmente a todos los hijos de dicho nodo. A su vez, cada nivel del árbol (salvo la raíz) tiene asociado un pivote.

En la Fig. 1 se presenta un ejemplo de la estructura de un MeTree. En la parte superior se muestran los nodos del árbol y los intervalos métricos de cada nodo, de acuerdo las distancias de los elementos al pivote de cada nivel. Por razones de legibilidad, los rectángulos de cada nodo se muestran por separado, en la parte inferior de la figura. Como se puede ver, puede haber superposición tanto entre intervalos métricos de los nodos hermanos, como en los rectángulos que representan el espacio que ocupan.

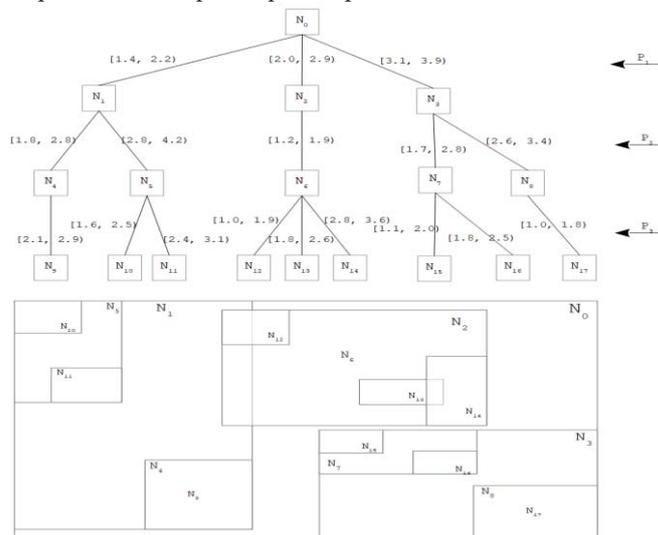


Fig. 1. Estructura de un MeTree.

Ante una inserción se procede de la siguiente manera. Sea o el objeto a insertar y p_n el pivote correspondiente al nivel n , se calcula el costo de agregar el objeto en cada uno de los nodos hijos del nodo actual y se obtiene el mínimo. Si dicho costo es menor que *infinito*, el objeto se añade al hijo de costo mínimo, y en caso contrario se agrega un nuevo hijo. El costo se calcula como la suma ponderada del costo métrico y el costo espacial. El costo métrico es cero si $i_m \leq d(s(o), p_n) \leq f_m$, es decir, si la distancia del objeto al pivote del nivel se encuentra dentro del intervalo métrico del nodo. En caso contrario es igual al menor incremento del intervalo, necesario para incluir a $d(s(o), p_n)$. Si el intervalo aumentado es mayor que el tamaño máximo permitido, el costo es *infinito*. Un procedimiento similar se utiliza para calcular el costo espacial. Si el aspecto espacial del objeto a insertar se encuentra ubicado dentro del rectángulo del nodo, el costo es cero. Para calcular el costo de aumentar el rectángulo para incluir al objeto se calcula el incremento necesario de su diagonal, y si es mayor a un valor máximo, nuevamente se considera *infinito*. Cuando se añade un elemento a un nodo existente, tanto su intervalo métrico como su rectángulo pueden aumentar de

tamaño. El tamaño máximo de los intervalos métricos es un parámetro fijado de antemano. En el caso de los rectángulos, la diagonal máxima depende del nivel del nodo. La raíz contiene el espacio completo donde pueden estar los elementos. En cada nivel la diagonal máxima es la mitad de la diagonal del nivel padre. Es decir que los tamaños disminuyen logarítmicamente.

Cuando se realiza una consulta métrico-espacial $(q, r, g)_d$, se visitan los nodos que cumplen las condiciones $im-r \leq d(q, p_n) \leq fm+r$ y $intersects(g, rect)$ hasta alcanzar las hojas, obteniendo un conjunto de candidatos. Luego se recorre secuencialmente este conjunto y se comparan los elementos con la consulta para obtener el resultado final. En la Fig. 2 se muestra el pseudocódigo de la consulta.

```

ConsultaMeTree(q, r, g)
  nodosAVisitar:=hijosDe(raíz) // conjunto de hijos
  nivelActual:=1
  WHILE |nodosAVisitar|>0 AND nivelActual <=MaxNiv:
    nodosNuevoNivel:=[] // conjunto vacío
    dist:=distanciaMetrica(q, pivote(nivelActual))
    FOR padre IN nodosAVisitar:
      FOR hijo IN hijosDe(padre):
        IF (dist BETWEEN hijo.im-r AND hijo.fm+r):
          IF intersects(g, hijo.rect):
            nodosNuevoNivel += hijo // agregar hijo
      nodosAVisitar:= nodosNuevoNivel
    nivelActual += 1
  // en nodosAVisitar quedan las hojas de candidatos
  Resultado:=[]
  FOR hoja IN nodosAVisitar:
    elementos:=elementosDe(hoja)
    FOR elem IN elementos:
      dist:= distanciaMetrica(q, s(elem))
      IF (dist<=r) AND (intersects(g, e(elem))):
        Resultado += e
  Return Resultado

```

Fig. 2. Pseudocódigo de Consulta a un MeTree.

Este mecanismo aprovecha tanto el aspecto métrico como el espacial para descartar ramas del árbol, lo que aumenta la eficiencia significativamente en comparación a la solución trivial.

5 Resultados Experimentales

Como parte de este trabajo se implementó el MeTree y se realizaron pruebas preliminares sobre una base de datos de 1514 puntos de interés correspondientes al microcentro de la Ciudad Autónoma de Buenos Aires (Fig. 3). Las consultas se realizaron sobre los nombres de los puntos de interés, utilizando la distancia de Levenshtein como función métrica, y su ubicación geográfica.

Se definieron 100 consultas compuestas por nombres similares a los elementos de la base de datos, radios aleatorios y polígonos de distintos tamaños y formas. Se eligieron aleatoriamente elementos de la base de datos para utilizarlos como pivotes y se fijó el tamaño máximo de los intervalos métricos.



Fig. 3. Sección del mapa utilizado en los experimentos.

Las consultas se ejecutaron utilizando la solución trivial planteada anteriormente y el MeTree y se compararon sus resultados. Ya que tanto la función de distancia métrica como la operación de intersección de dos elementos geométricos suelen ser costosas, se tomó como medida de comparación la cantidad de elementos candidatos resultantes sobre los cuales se hace la comprobación secuencial de ambos aspectos. Los resultados obtenidos para cada una de las consultas se muestran en la Fig. 4.

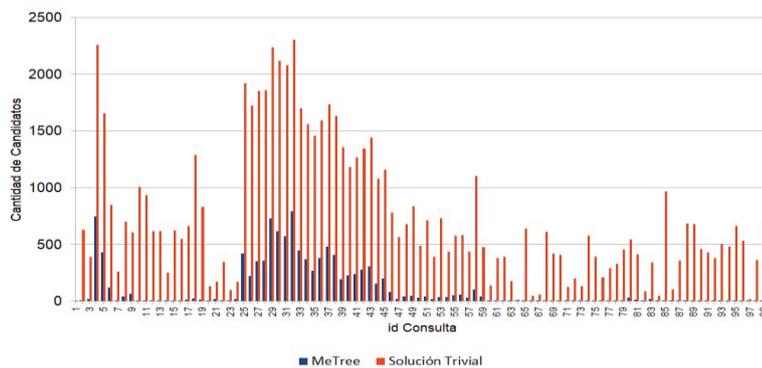


Fig. 4. Costos del MeTree y de la Solución Trivial.

En promedio, la cantidad de candidatos por consulta, que luego deben ser comparados con la consulta para determinar si forman parte del resultado final o no, fue 104,24 para el MeTree y 740,09 para la solución trivial. La diferencia en eficiencia es significativa y se explica por el hecho de que el descarte mediante ambos aspectos a la vez reduce considerablemente la cantidad de elementos a evaluar durante el recorrido del árbol.

Si bien los resultados son promisorios, estos experimentos son los primeros de una serie de pruebas planificadas que brindarán mayor detalle y confianza en el funcionamiento del índice bajo distintas situaciones y con mayores cantidades de datos.

6 Conclusiones y Trabajo Futuro

En este trabajo formalizamos el modelo métrico-espacial y presentamos el método de acceso MeTree, que resuelve consultas con restricciones por similitud y espaciales al mismo tiempo. Trabajos de investigación anteriores limitaban las búsquedas por similitud y espacial sólo a

búsquedas de documentos. El índice presentado en este artículo permite trabajar con cualquier tipo de objeto, siempre que exista una distancia métrica de comparación.

Un aspecto interesante es que el mismo índice se puede utilizar para realizar búsquedas por similitud o búsquedas espaciales por separado también, es decir que en principio no sería necesario contar con índices espaciales y métricos adicionales. Pero para comprobar esta afirmación aún se requiere verificación de los resultados bajo distintas situaciones y mayores cantidades de datos, ya que los experimentos realizados aún no son suficientes.

Trabajo futuro

1. Calcular la complejidad temporal del método en forma analítica, tema en el cual estamos trabajando actualmente.
2. Realizar experimentos con otros tipos de datos (imágenes, por ejemplo), con funciones de distancia de mayor complejidad y mayor cantidad de datos (estamos preparando una base de más de un millón de elementos)
3. Analizar y realizar modificaciones al procedimiento de inserción, para disminuir el tamaño de los rectángulos y de los intervalos métricos y reducir la superposición.

Referencias

1. Chávez, E., Navarro, G., Baeza-Yates, R. and Marroquín, J.L. Searching in metric spaces. *ACM Computing Surveys*, 33(3):273-321, September (2001)
2. LI, Zhisheng; LEE, Ken C. K.; ZHENG, Baihua; LEE, Wang-Chien; LEE, Dik Lun; and WANG, Xufa. IR-Tree: An Efficient Index for Geographic Document Search. *IEEE Transactions on Knowledge and Data Engineering*. (2011)
3. E. Chavez and G. Navarro, "A compact space decomposition for effective metric indexing," *Pattern Recognition Letters*, 26(9), pp. 1363–1376, (2005)
4. P. Ciaccia, M. Patella, and P. Zezula, "M-tree: An efficient access method for similarity search in metric spaces," in *VLDB*, pp. 426–435. (1997)
5. V. Dohnal, C. Gennaro, P. Savino, and P. Zezula, "D-index: Distance searching index for metric data sets," *Multimedia Tools Appl.*, 21(1), pp. 9–33, (2003)
6. C. T. Jr., A. Traina, B. Seeger, and C. Faloutsos, "Slim-trees: High performance metric trees minimizing overlap between nodes," in *ICDE*, pp. 51–65. (2000)
7. W. Burkhard and R. Keller, "Some approaches to best-match file searching," *Commun. ACM*, 16(4), pp. 230–236, (1973)
8. C. T. Jr., R. F. S. Filho, A. J. M. Traina, M. R. Vieira, and C. Faloutsos, "The Omni-family of all-purpose access methods: A simple and effective way to make similarity search more efficient," *VLDB J.*, 16(4), pp. 483–505, (2007)
9. L. Mico, J. Oncina, and R. C. Carrasco, "A fast branch & bound nearest neighbour classifier in metric spaces," *Pattern Recognition Letters*, 17(7), pp. 731–739, (1996)
10. P. N. Yianilos, "Data structures and algorithms for nearest neighbor search in general metric spaces," in *SODA*, pp. 311-321.(1993)
11. I. Kalantari and G. McDonald, "A data structure and an algorithm for the nearest point problem," *IEEE Trans. Software Eng.*, 9(5), pp. 631–634, (1983)
12. H. Noltemeier, K. Verbarg, and C. Zirkelbach, "Monotonous bisector Trees — A tool for efficient partitioning of complex scenes of geometric objects," in *Data Structures and Efficient Algorithms*, pp. 186–203, (1992)
13. J. K. Uhlmann, "Satisfying general proximity/similarity queries with metric trees," *Inf. Process. Lett.*, 40(4), pp. 175–179, (1991)
14. S. Brin, "Near neighbor search in large metric spaces," in *VLDB*, pp. 574–584, (1995)
15. G. Navarro and R. U. Paredes, "Fully dynamic metric access methods based on hyperplane partitioning," *Inf. Syst.*, 36(4), pp. 734–747, (2011)

16. G. Navarro, "Searching in metric spaces by spatial approximation," *VLDB J.*, 11(1), pp. 28–46, (2002)
17. L. Britos, A. M. Printista, and Nora Reye, "DSACL+-tree: A dynamic data structure for similarity search in secondary memory," in *SISAP*, pp. 116–131, (2012)
18. G. Navarro and N. Reyes, "Dynamic spatial approximation trees for massive data," in *SISAP*, pp. 81–88, (2009)
19. M. R. Vieira, C. T. Jr., F. J. T. Chino, and A. J. M. Traina, "DBM-Tree: A dynamic metric access method sensitive to local density data," *J. Inf. Data Management*, 1(1), pp. 111–128, (2010)
20. L. Aronovich and I. Spiegler, "CM-tree: A dynamic clustered index for similarity search in metric databases," *Data Knowl. Eng.*, 63(3), pp. 919–946, (2007)
21. J. Almeida, R. D. S. Torres, and N. J. Leite, "BP-tree: An efficient index for similarity search in high-dimensional metric spaces," in *CIKM*, pp. 1365–1368, (2010)
22. E. Vidal, "An algorithm for finding nearest neighbors in (approximately) constant average time," *Pattern Recognition Letters*, 4(3), pp. 145–157, (1986)
23. G. Ruiz, F. Santoyo, E. Chavez, K. Figueroa, and E. S. Tellez, "Extreme pivots for faster metric indexes," in *SISAP*, pp. 115–126, (2013)
24. J. Mosko, J. Lokoc, and T. Skopal, "Clustered pivot tables for I/O optimized similarity search," in *SISAP*, pp. 17–24 (2011)
25. W. Burkhard and R. Keller, "Some approaches to best-match file searching," *Commun. ACM*, 16(4), pp. 230–236, (1973)
26. R. A. Baeza-Yates, W. Cunto, U. Manber, and S. Wu. "Proximity matching using fixed-queries trees," in *CPM*, pp. 198–212, *SODA*, (1994)
27. T. Bozkaya and M. Ozsoyoglu, "Distance-based indexing for high dimensional metric spaces," in *SIGMOD*, pp. 357–368, (1997)
28. T. Skopal, J. Pokorny, and V. Snasel, "PM-tree: Pivoting metric tree for similarity search in multimedia databases," in *ADBIS*, pp. 803–815, (2004)
29. D. Novak, M. Batko, and P. Zezula, "Metric Index: An efficient and scalable solution for precise and approximate similarity search," *Inf. Syst.*, 36(4), pp. 721–733, (2011)
30. Rigaux P, Scholl M, Voisard A. "6 - Spatial Access Methods". En *Spatial Databases*, Morgan Kaufmann, San Francisco, pp 201–266, (2002)
31. Nievergelt J, Hinterberger H, Sevcik K.C. The Grid File: An Adaptable, Symmetric Multikey File Structure. *ACM Trans. Database Syst* 9(1):38–71, (1984)
32. Finkel, R. A.; Bentley, J. L. "Quad Trees A Data Structure for Retrieval on Composite Keys". *Acta Informatica*. Springer-Verlag. 4: 1–9, (1974)
33. Zhang J, You S, Gruenwald L. Parallel QuadTree coding of large-scale raster geospatial data on gpgpus. En *Proceedings of the 19th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, ACM, New York, pp 457–460, (2011)
34. Bentley J. L. Multidimensional binary search trees used for associative searching. *Commun. ACM* 18(9):509–517, (1975)
35. Castro J, Burns S. Online Data Visualization of Multidimensional Databases Using the Hilbert Space Filling Curve, *Lecture Notes in Computer Science*, vol 4370, Springer Berlin Heidelberg, chap 9, pp 92–109, (2007)
36. Guttman A. R-trees: a dynamic index structure for spatial searching. *SIGMOD Rec* 14(2):47–57, (1984)

Data science methodologies selection with hierarchical analytical process and personal construction theory

Karina B. Eckert¹⁻² and Paola V. Britos³

¹ National University of Misiones, Posadas, Misiones, Argentina

² Gastón Dachary University, Posadas, Misiones, Argentina

³ Applied Computer Lab, National University of Río Negro, El Bolsón, Río Negro, Argentina
karinaeck@gmail.com, pbritos@unrn.edu.ar

Abstract. The amount of data currently available for Strategic Decision Making is substantial; which is why Data Science find itself in apogee in various areas where it can be applied. Expertise respecting the areas' methodologies is fundamental; which is why, the objective of this paper is to compare and ponder them, for which, Analytic Hierarchy Process, was utilized along with linguistic tags and Personal Construction Theory, with the purpose of establishing and prioritizing characteristics according to their degree of compliance in real validation cases. The sub-criteria were grouped in different levels, conforming a hierarchy for the present problem. The validation case consisted in determining causes for breakdowns in new automobiles as they are being transported from the factory to the concessionaires; in which the proposed model proved useful and MoProPEI could be identified as the most adequate methodology.

Keywords: Data Science Methodologies, Analytic Hierarchy Process, Personal Construction Theory, Linguistic tags, Criteria.

1 Introduction

Multiple Criteria Decision Making (MCDM) can be seen as a useful tool for Decision Making (DM) and of great potentiality for Systems Engineering (SE) processes. There exists a superposition on the multicriteria and systemic approaches on a conceptual and operational level. At a conceptual level, when the established objectives begin to conflict with each other and it is required to find an equilibrium or compromise. At an operational level SE can be understood as a sequence of steps in which it is necessary to evaluate and choose among different alternatives or criteria at all times. These methods allow to approach a problem subjacent of subjectivity in an organized and systemic manner, which helps rationalize a complex process. [1], [2].

A popular MCDM method is Analytic Hierarchy Process (AHP), created by Saaty [3], with the purpose of searching for a systemic practice to define priorities and support complex DM [4]. The advantages of using AHP lie, among other aspects, in that among MCDM techniques, it is one of the few that provides a theoretical axiomatic; from a practical point of view, it is characterized by its good performance; providing a flexible, adaptable, robust and easy to understand model [1], [5], [6].

In some scenarios decision makers have a very limited amount of information to specify their preferences on multiple pair comparisons; which is why a deeper analysis is required rather than a direct comparison [7]. Assessing and selecting Data Science (DS) methodologies is one of these scenarios.

Despite the fact that the majority of DS methodologies have been evaluated and validated by the community, these are not without flaw, for example the ones related to project management [8], [9]. Selecting a methodology can be complicated, especially for novices; while this is an essential task for experts in the area [10], [11].

The objective of this article is to determine which of the assessed DS methodologies (P³TQ [12], CRISP-DM [13] y MoProPEI [14]) is the most robust for real applications. In order to achieve this, the first three stages of Personal Construction Theory (PCT) are used, which initially include a dialogue to determine how the expert thinks, and to identify which are their priorities and most important factors [15], [16], from that, linguistic tags are established, the sub-criteria involved in DS methodologies are defined y and the hierarchic structure of the problem is conformed. Sub-criteria are established based on a degree of compliance function, depending on the validation case, which is then integrated to AHP, to finally obtain the resulting ponderations for each methodology.

The present article is structured in the following way: Preliminary concepts referred to DS, AHP and PCT can be found in Section 2. Posteriorly the proposed model is described in Section 3; which is validated using a real-life case in Section 4. Finally, conclusions are presented in Section 5

2 Preliminary concepts

2.1 Data Science

Currently, informatics systems can generate and store a vast amount of data at a low cost ; which results in these growing exponentially and making them impossible to process using common methods [10], [11]. A substantial amount of attention must be given to the importance and implications of data for DM; given that they are a great advantage for it [10]. In consequence, there is an increasing number of companies that take decisions based on data, improving their performance in an operational and financial way [17].

Data Science was previously referred to as Data Mining or Information Mining; which over time changed its designation as it grew. The concept in this case is the extraction of knowledge from data and technology that incorporate these principles [11], [18].

DS consists in a group of fundamental principles, guided by a specific methodology, which help and guide the extraction of knowledge from data; it includes several techniques, algorithms and tools which ease the exhaustive and automatic processing of data; allowing to identify useful knowledge which is not possible to be detected in plain sight [10], [11], [18]. In order to forecast results, areas such as statistics, math, behavioral science, computing and predictive analysis are included [19]. The objec-

tive is to obtained knowledge specialized for DM, from results represented as models or patterns [11], [20], [21].

There exists a diverse variety of techniques and algorithms for data processing and knowledge extraction applicable in this area; however, DS involves much more. DS provides professionals a structure and a group of principles that bring a framework in order to systematically treat knowledge extraction problems; where methods to treat data and methodologies utilized in these projects are transcendental [10], [18].

An amount of studies comparing existing methodologies exists shown in [20], [21], [22], [23], [24] y; from which methodologies such as Catalys (known as P3TQ) [12] and CRISP-DM [13] can be highlighted. Based on different studies, recommendations and our own study of the methodologies; the aforementioned where selected along with MoProPEI [14].

2.2 Analytic Hierarchy Process

AHP involves the following activities or steps [1], [5], [25], [26], [27], [28]:

Firstly, the decision problem must be modelled as a hierarchic structure. Situating the main objective in the upper level. In the level below, criteria such as attributes, secondary objective or parameters from which preferences are transformed or justified can be found. In some cases, criteria can be divided in sub-criteria forming another descending hierarchy. Finally, in the last level, alternatives are presented.

When establishing priorities using paired comparison, the objective is to define the relative weights for the criteria, said numeric values indicate the importance or relative priority between C_i and C_j as criteria, respecting the element in the immediately superior level. In order to achieve this, the fundamental scale proposed by Saaty was utilized; which ranges from an equivalent importance (value 1), with two criteria that contribute equally to achieve the objective, to an extreme degree of importance (value 9) where the evidence that benefits one criteria over another is the highest possible in the affirmation order. Numbers in the scale represent the importance proportion of an element respect another in relation to criteria or an objective which they share.

Making use of the fundamental scale, the decision maker must determine the assigned weight for each criterion, completing for this the matrix. For a matrix of these characteristics it is true that the maximum eigenvector λ_{max} is a positive real number and that there exists an eigenvector Z , which elements associated to this vector are positive. Posteriori the eigenvector must be normalized so that the summation becomes a unit.

An incoherence error in the pairing comparison process generates a matrix and an eigenvector which are unrepresentative; which in turn results in a contradiction since it violates the transitivity of the values, in order to correct this Saaty proposed the Consistency Ratio(CR) in order to evaluate coherence in the decisions made by the decision maker, which is shown in equation 1:

$$CR = CI/RI \quad (1)$$

CI is defined in equation 2, where λ_{max} is the maximum value in the matrix and n and its order. RI is a measurement utilized to improve the consistency of the decisions

accounting for the dimension of the matrix. Simulating 100.000 randomly generated inverse matrixes [29], average RI was defined. For adequate consistency, Saaty indicated that CR must not be greater than 10% ($CI \leq 0,10$); the closer to 0, the greater the consistency; in the opposite case, decisions must be further revised.

$$CI = (\lambda_{max} - n)/(n - 1) \quad (2)$$

If there exist sub-criteria, their global weight must be calculated a priori associated to them; following the same procedure, but in this case the paired comparisons must be performed, in order to determine the relative importance to the criteria immediately above in the hierarchy (local priority). To calculate relative global importance, the product of the different weights of each one of the sub criteria and criteria is calculated, following the hierarchy from the most inferior part to the top of this one, this procedure is known as "Hierarchic composition"

Saaty proposed the use of a method known as pondered summation; which consists in finding the global priority vector p , which adds priorities obtained from criteria and alternatives. The p_i components in the vector belong to total priorities associated to each alternative A_i , reflecting the total value which the decision maker has for each alternative, for the aforementioned, the following expression can be used: (3):

$$p_i = \sum_{j=1}^n (w_j \cdot r_{ij}) \quad i = 1, 2, \dots, m \quad (3)$$

Where w_j corresponds to the associated weights from each of the considered criteria and r_{ij} are the components of the normalized matrix. In order to solve the decision problem and to determine the best alternative which will be the greater pondered summation, sorting the alternatives based on these values will be enough.

2.3 Personal Construction Theory

Personal Construction Theory (PCT) was proposed by Kelly, it is a technique for the extraction of knowledge; which consists in becoming aware of the inconsistencies in the value scales, given that each person has their own view of the world. For a particular domain all of the aspects that the expert finds important must be included and must be represented as elements and their decomposition in bipolar characteristics; which are then evaluated. It is considered as a classification test divided in five stages [15], [16].

3 Proposed Model

The proposed model is divided in a series of steps:

1. Characteristics recognition

Based on methodology studies and experts' opinion, the criteria, sub-criteria and their hierarchic structure were selected. The first two stages of PCT were utilized in order to work with the experts: First, the elements are identified (DS methodologies) and secondly the characteristics (criteria y sub-criteria).

2. Linguistic tag definition from completeness level

The purpose of this step is to avoid ambiguities in the definition and the completeness of the criteria, a narrow scale ranging from 1 to 9 was utilized, where 1 indicated that the sub-criteria shouldn't be analyzed; 2 to 9 represent interval values in the form of percentages respecting the fulfillment of this aspect inside the case study. Depending on if the sub-criteria is a positive or negative aspect, the values in the scale defined between 2 and 9 can be inverted. For example, two sub-criteria are exposed in Table 1, the first one being Portability which is a positive aspect, the values(percentages) are positive and in an ascendant way, whereas for Transformation Costs, which is a negative aspect, the percentages are shown in descendent way.

Table 1. Linguists tags for sub-criteria

Portability		Transformation Cost	
1	Not analyzed	1	Not analyzed
2	1% to 13% portability	2	98% to 100% transformation cost
3	14% to 27% portability	3	84% to 97% transformation cost
4	28% to 41% portability	4	70% to 83% transformation cost
5	42% to 55% portability	5	56% to 69% transformation cost
6	56% to 69% portability	6	42% to 55% transformation cost
7	70% to 83% portability	7	28% to 41% transformation cost
8	84% to 97% portability	8	14% to 27% transformation cost
9	98% to 100% portability	9	1% to 13% transformation cost

3. Establishing a hierarchic structure

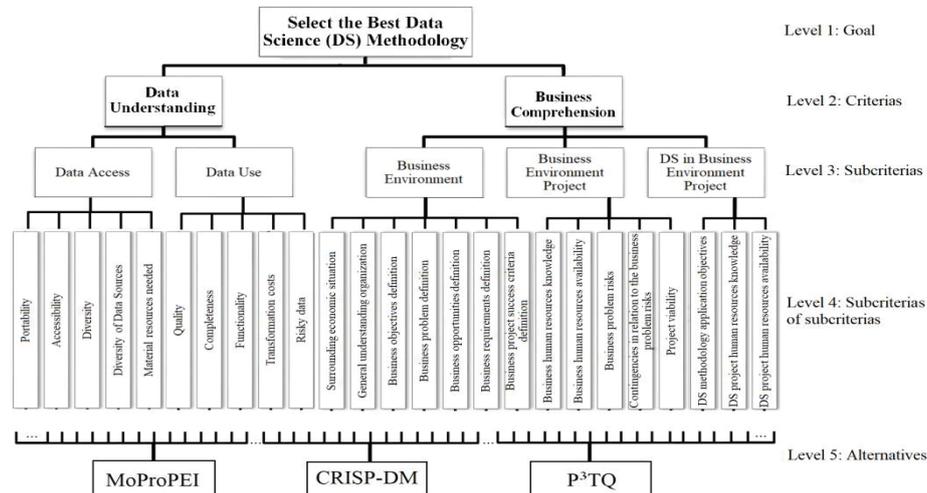


Fig. 1. Hierarchical structure to select the best DS methodology

The aforementioned steps allowed to identify the criteria and sub-criteria that must be taken into account when selecting a DS methodology; which were grouped in different levels, therefore establishing a hierarchy for the problem. In Fig.1. the obtained structure can be visualized; the main objective can be observed in the first level (Selecting the best DS methodology); the second level is where the two main branches from the methodologies and DS projects, the understanding of the data and the business comprehension are defined; in the third level data access and data usage can be found, with 5 sub-criteria for each in level 4 on one side; and on the other, business environment, project environment and DS in the business project on the third level, with seven, five and three sub-criteria respectively (level 4); the sub-criteria in the last level are compared based on each of the methodologies visible in level 5.

4. Making arrays with the sub-criteria in the fourth level

For this step a grill-type matrix was made (PCT Stage 3), for which were set the extreme or bipolar values, the worst and best case, which represent values 1 and 9(See Table 1). Each expert filled the matrix with the corresponding values taking into account the previously established scale. For example, sub-criteria with their bipolar values in each extreme and their respectively assigned ponderation for each methodology can be found in Table 2.

Table 2. Compliance last level sub-criteria

	P³TQ	CRISP-DM	MoPro PEI	
Data portability is not analyzed	8	5	9	98% to 100% data portability
Data diversity is not analyzed	3	4	2	0% to 13% data diversity

5. Paired comparison for criteria and sub-criteria

Starting from the values indicated by the expert in the previous step, matrixes were completed based on the sub-criteria in the fourth level; taking the difference between the assessments as absolute values, plus one (Ex: $5-5=0+1$, both sub criteria have the same level of preference; $7-5=2+1$, the first sub-criteria has a preference of 3 over the second one). The purpose of this was to complete the paired matrixes based on assessed matrixes with linguistic tags (see example in Table 2) fitting the values of the Saaty scale and placing them in their corresponding places inside the matrix

For the criteria in levels 2 and 3, other paired-comparison matrixes where made in order to ease the expert's choices; where the preference value respect another is marked with an X (based on the Saaty scale). The aforementioned is seen in Table 3, where the comparison between the criteria for data access and data usage is exemplified; for which the expert assigned a ponderation of 5, that is to say, that data usage possesses a great importance over data access. From these matrixes the corresponding ones where completed according to what Saaty proposed, defining their importance grouping them by criteria and sub-criteria based on the defined hierarchy.

Table 3. Criteria by pairs comparison

Data Access					Data Use			
Extreme importance: 9	Very strong importance: 7	Strong importance: 5	Moderate importance: 3	Equal importance: 1	Moderate importance: 3	Strong importance: 5	Very strong importance: 7	Extreme importance: 9
						X		

Posteriorly, their reciprocal values were incorporated, all the matrixes were normalized and the ponderations for each of them were defined. Table 4 shows the continuity of what was proposed in Table 3.

Table 4. Pairwise comparison matrix

	Data Access	Data Use	Normalized Matrix		Weightings
Data Access	1	1/3	0,25	0,25	0,25
Data Use	3	1	0,75	0,75	0,75

6. Coherence control

The previous step was followed by an assessment of the consistency of the decisions made by the experts based on the paired matrixes from levels 2 to 4; for this, the quotient for the matrixes was estimated along with the approximations of consistency for each of them, as indicated in equations 1 and 2 (Section 2).

7. Establishing final ponderations

The local and global priorities were established using the relative weight of the criteria for each level; this was followed by obtaining the total priorities associated to each alternative; using pondered summation (See equation 3).

4 Validation Cases

In order to verify the proposed model, two real validation cases were utilized, the first one has the objective of determining random breakdowns in new automobiles as they are being transported from factories to concessionaires and the second one, causes for college desertion. The results obtained for the first scenario are summarized below; for which experts chose the values when analyzing each methodology.

Fig. 2 shows the ponderations obtained for each methodology respect from the sub-criteria in the third level. In this way, the preponderations for MoProPEI can be appreciated over the other two for these sub-criteria; it can also be noted that for the understanding of data, P3TQ obtained a better performance compared to CRISP-DM, however this was not the case in business comprehension, where in a general way CRISP-DM obtained better weights.

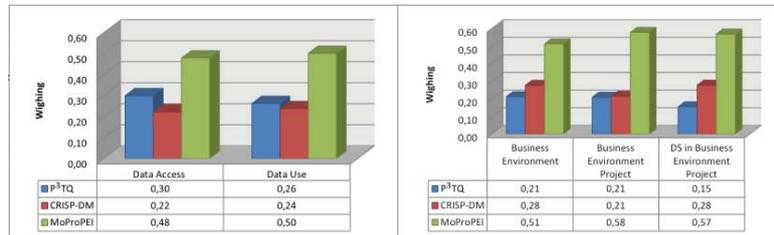


Fig. 2. Third level hierarchy weightings

Inside the understanding of data, the sub-criteria for data access have a ponderation of 0,25 whereas data usage has 0,75 respect from this criterion. Concerning business comprehension, the weights assigned to sub-criteria for business environment and project environment are of 0,20 each and for DS in business project it is 0,75 respect from this criterion. Taking into account the aforementioned ponderations and the ones obtained in level 3, the criteria located in the second level are shown in Table 5.

Table 5. Second level hierarchy weightings

	Data Understanding	Business Comprehension
P ³ TQ	0,07	0,11
CRISP-DM	0,17	0,17
MoProPEI	0,26	0,23

Going up in the hierarchy taking into account that ponderations in inferior levels and that Data Understanding and Business Comprehension are equally important; Fig. 3. Shows the final ponderations obtained, which show that MoProPEI obtained a ponderation of 53%, followed by CRISP-DM with 25% and lastly P³TQ with 22%. This clearly indicates that given the assessed criteria and sub-criteria, MoProPEI is the most adequate and complete for the present case a, obtaining a ponderation larger than the other two combined.

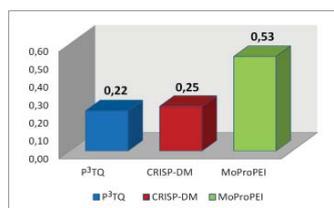


Fig. 3. Final weightings by methodologies

5 Conclusions

Based on the proposed model, the analysis and the results, it can be highlighted that the model that integrates AHP, linguistic tags and PCT is beneficial to identify the

primordial characteristics inside a DS project; as well as to establish de compliance level for each of the sub-criteria of each methodology using defined linguistic tags and grill-type matrixes used to complete these aspect therefore avoiding ambiguity. It was proven that these techniques can be integrated to obtain positive results.

Concerning determining causes for breakdowns in automobiles as they are being transported from factories to concessionaries, the proposed model obtained the ponderations for each methodology, where MoProPEI can be highlighted over the remaining two, being this one the selected one for the aforementioned validation case. For the other validation case, the results were similiary.

As future research it is expected to develop software that implements the proposed model, validation for new DS project and even other areas.

References

1. García Cascales, M. del S.: Métodos para la comparación de alternativas mediante un Sistema de Ayuda a la Decisión (S.A.D.) y “Soft Computing,” (2009).
2. Romero, C.: Análisis de las Decisiones Multicriterio. Isdefe, Madrid, España (1996).
3. Saaty, T.L.: The analytic hierarchy process. McGraw-Hill, New York (1980).
4. Forman, E.H., Gass, S.I.: The Analytic Hierarchy Process—An Exposition. *Operations Research*. 49, 469–486 (2001). <https://doi.org/10.1287/opre.49.4.469.11231>.
5. Russo, R. de F.S.M., Camanho, R.: Criteria in AHP: A Systematic Review of Literature. *Procedia Computer Science*. 55, 1123–1132 (2015). <https://doi.org/10.1016/j.procs.2015.07.081>.
6. Kou, G., Lin, C.: A cosine maximization method for the priority vector derivation in AHP. *European Journal of Operational Research*. 235, 225–232 (2014). <https://doi.org/10.1016/j.ejor.2013.10.019>.
7. Jalao, E.R., Wu, T., Shunk, D.: A stochastic AHP decision making methodology for imprecise preferences. *Information Sciences*. 270, 192–203 (2014). <https://doi.org/10.1016/j.ins.2014.02.077>.
8. Pytel, P., Britos, P., García Martínez, R.: Proposal and Validation of a feasibility Model for Information Mining Projects. Presented at the 25th International Conference on Software Engineering and Knowledge Engineering , Boston, USA.
9. Vanrell, J.Á., Bertone, R.A., García Martínez, R.: Modelo de proceso de operación para proyectos de explotación de información. Presented at the XVI Congreso Argentino de Ciencias de la Computación (2010).
10. Waller, M.A., Fawcett, S.E.: Data Science, Predictive Analytics, and Big Data: A Revolution That Will Transform Supply Chain Design and Management. *Journal of Business Logistics*. 34, 77–84 (2013). <https://doi.org/10.1111/jbl.12010>.
11. Eckert, K., Britos, P.V.: Modelo basado en la toma decisiones con criterios múltiples para la elección de metodologías de data science. Presented at the XX Workshop de Investigadores en Ciencias de la Computación (2018).
12. Pyle, D.: *Business Modeling and Data Mining*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA (2003).

13. Chapman, P., Clinton, J., Kerber, R., Khabaza, T., Reinartz, T., Shearer, C., Wirth, R.: CRISP-DM 1.0: Step-by-Step Data Mining Guide, <http://tinyurl.com/crispdm>, (2000).
14. Martins, S., Pesado, P., García Martínez, R.: Propuesta de Modelo de Procesos para una Ingeniería de Explotación de Información: MoProPEI. *Revista Latinoamericana de Ingeniería de Software*. 2, 313–332 (2014).
15. Britos, P., Rossi, B., García Martínez, R.: Notas sobre didáctica de las etapas de formalización y análisis de resultados de la técnica de emparrillado. Un Ejemplo. In: *Proceedings del V Congreso Internacional de Ingeniería Informática*. pp. 200–209 (1999).
16. García Martínez, R., Britos, P.V.: *Ingeniería de Sistemas Expertos*. Nueva Librería (2004).
17. McAfee, A., Brynjolfsson, E.: Big data: the management revolution. *Harv Bus Rev*. 90, 60–68 (2012).
18. Provost, F., Fawcett, T.: Data Science and its Relationship to Big Data and Data-Driven Decision Making. *Big Data*. 1, 51–59 (2013). <https://doi.org/10.1089/big.2013.1508>.
19. Hazen, B.T., Boone, C.A., Ezell, J.D., Jones-Farmer, L.A.: Data quality for data science, predictive analytics, and big data in supply chain management: An introduction to the problem and suggestions for research and applications. *International Journal of Production Economics*. 154, 72–80 (2014). <https://doi.org/10.1016/j.ijpe.2014.04.018>.
20. Rodríguez Montequín, M.T., Álvarez Cabal, J.V., Mesa Fernández, J.M., González Valdés, A.: Metodologías para la realización de proyectos de Data Mining. Presented at the VII Congreso Internacional de Ingeniería de Proyectos, Pamplona España (2003).
21. Moine, J.M.: Metodologías para el descubrimiento de conocimiento en bases de datos: un estudio comparativo, <http://hdl.handle.net/10915/29582>, (2013).
22. Moine, J.M., Gordillo, S.E., Haedo, A.S.: Análisis comparativo de metodologías para la gestión de proyectos de minería de datos. Presented at the XVII Congreso Argentino de Ciencias de la Computación (2011).
23. Giraldo Mejia, J.C., Jiménez Builes, J.A.: Caracterización del proceso de obtención de conocimiento y algunas metodologías para crear proyectos de minería de datos. *Revista Latinoamericana de Ingeniería de Software*. (2013).
24. Palacios, H.J.G., Pantoja, G.A.H., Navarro, A.A.M., Puetaman, I.M.A., Toledo, R.A.J.: Comparative between CRISP-DM and SEMMA for data cleaning of MODIS products in a study of land use and land cover change. In: *2016 IEEE 11th Colombian Computing Conference (CCC)*. pp. 1–9 (2016). <https://doi.org/10.1109/ColumbianCC.2016.7750789>.
25. Saaty, T.L.: *Fundamentals of Decision Making and Priority Theory With the Analytic Hierarchy Process*. RWS Publications (2000).
26. Saaty, T.L.: How to make a decision: The analytic hierarchy process. *European Journal of Operational Research*. 48, 9–26 (1990). [https://doi.org/10.1016/0377-2217\(90\)90057-I](https://doi.org/10.1016/0377-2217(90)90057-I).
27. Liu, B., Kong, F.: Research and application of sidewall stability prediction method based on analytic hierarchy process and fuzzy integrative evaluation method. *Natural Science*. 4, 142 (2012).
28. Saaty, T.L.: Decision making with the analytic hierarchy process. *International Journal of Services Sciences*. 1, 83–98 (2008). <https://doi.org/10.1504/IJSSci.2008.01759>.
29. Aguarón, J., Moreno Jiménez, J.M.: The geometric consistency index: Approximated thresholds. *European Journal of Operational Research*. 147, 137–145 (2003). [https://doi.org/10.1016/S0377-2217\(02\)00255-2](https://doi.org/10.1016/S0377-2217(02)00255-2).

Ensamblado *ad hoc* de clasificadores para la detección de cáncer de mama usando Scikit-learn

Andrés Maciel Cardozo, Gustavo Sosa-Cabrera, María E. García-Díaz

Facultad Politécnica
Universidad Nacional de Asunción
Asunción, Paraguay
amaciel@est.pol.una.py, {gdsosa, mgarcia}@pol.una.py
<http://www.pol.una.py>

Resumen En el presente, el cáncer de mama es uno de los cánceres más frecuentes y es la segunda causa de muerte en mujeres en todo el mundo. Asimismo, cada vez es más difícil ignorar el constante e intenso aumento de la importancia de los enfoques de minería de datos en los diagnósticos médicos. En este sentido, central a toda la disciplina de la minería de datos, encontramos la clasificación como la tarea preponderante en el proceso de toma de decisiones para los médicos. A día de hoy, una gran cantidad de clasificadores se han propuesto en la literatura. Sin embargo, teniendo en cuenta la cantidad de personas afectadas por el cáncer, merece la pena seguir desarrollando técnicas que puedan contribuir en mejores formas de diagnóstico. En lo que a este estudio concierne, se ha considerado introducir un *ensamblado de clasificadores*, propuesto en razón al balance entre sus factores individuales de *sesgo* y *varianza*. Demostrando la correctitud de la metodología adoptada para la conjunción *ad hoc* de los clasificadores, los resultados empíricos de este estudio proporcionan evidencia de una mejor clasificación de los tumores como maligno o benigno, en cuanto a precisión se refiere.

Palabras Claves: Cáncer de mama · Clasificación de tumores · Ensamblado de clasificadores · Sesgo · Varianza · Bagging · Boosting · Scikit-learn

1. Introducción

Durante el último siglo, el cáncer ha sido una de las principales causas de muerte donde el cáncer de mama es uno de los más frecuentes; y de hecho, es la segunda causante de muerte en mujeres en todo el mundo [13].

Por otra parte, entre las numerosas aplicaciones de la minería de datos y el aprendizaje automático, la tarea de clasificación es un área que cobra cada vez más importancia en diagnósticos médicos asistidos. En la tarea de clasificación cada instancia pertenece a una clase la cual se indica mediante el valor discreto de un atributo que llamamos la clase de la instancia.

Hasta la fecha, una considerable cantidad de propuestas de nuevos clasificadores, variantes de los ya existentes y *ensamblado* de los mismos ha sido publicada en la literatura científica.

En este sentido, se desarrolla un creciente interés por el ensamblado de clasificadores por ser considerado como el siguiente nivel de tendencia en la búsqueda de mejoras en la tarea de clasificación [9]. Básicamente, este enfoque consiste en la construcción de un clasificador más robusto y preciso, esto es, mediante la combinación de las predicciones de varios clasificadores atendiendo las fortalezas y debilidades que cada modelo algorítmico individualmente posee.

Entre tanto, el cáncer de mama es un tumor maligno que amenaza gravemente la salud de la mujer a nivel mundial y constituye una prioridad debido a su elevada incidencia¹. Adicionalmente y como es bien conocido, es posible obtener mejores resultados en el contexto de los diagnósticos médicos apoyados por el análisis de los datos, ya que se equilibra el conocimiento de los expertos humanos al describir problemas y objetivos con las capacidades de clasificación de los computadores [12].

Surge, por tanto, la constante necesidad de nuevas investigaciones a los estudios previos [1,16] y que contribuyan para la toma de decisiones de los médicos en el diagnóstico de cáncer de mama mediante la aplicación de técnicas de minería de datos; siendo éste, en efecto, el propósito del presente estudio.

Los resultados de esta investigación demuestran que el proceso de *ensamblado de clasificadores* propuesto en razón al balance entre el *sesgo* y la *varianza* de cada clasificador, permite la construcción de un mecanismo mejorado de clasificación de tumores maligno o benigno, en cuanto a precisión se refiere.

De manera estructural, este trabajo primeramente sienta los fundamentos teóricos respecto a los clasificadores y los factores principales, sesgo y varianza, así como las técnicas de ensamblado. Posteriormente, en la sección de Materiales y Métodos, se presentan los conceptos de *Bagging* y *Boosting* seguido de los algoritmos de clasificación que se contemplan en este estudio. En la sección de experimentos se evalúan los planteamientos iniciales para finalmente converger las ideas en la conclusión.

2. Fundamentos Teóricos

Clasificador. Desde el punto de vista matemático, en [11], [4] se precisa que un problema de clasificación puede verse como una colección L de N ejemplos etiquetados que pueden definirse de la siguiente manera:

$$L = \{(x_n, y_n), n = 1, 2, \dots, N, y \in [y^1, y^2, \dots, y^y]\} \quad (1)$$

donde cada ejemplo (x_n, y_n) viene determinado por un vector de atributos x_n y una etiqueta de clase y_n pertenecientes a algunas de las Y clases del problema $[y^1, y^2, \dots, y^y]$. El algoritmo de clasificación resultante deberá de construir un clasificador que prediga, dado un nuevo vector de características x (no incluido

¹ <https://seer.cancer.gov/statfacts/html/breast.html>

en L) la clase Y a la que pertenece usando el conocimiento contenido en el conjunto inicial de datos L . Formalmente, un clasificador es un modelo o función M que predice la etiqueta de clase y para un ejemplo de entrada dado x , es decir, $\mathcal{Y} = M(x)$, donde $x = (x_1, x_2, \dots, x_d)^T \in \mathbb{R}^d$ es un punto en el espacio d -dimensional e $y \in \{c_1, c_2, \dots, c_k\}$ es su clase predicha.

Sesgo y Varianza. En [20] definen al sesgo y a la varianza como la pérdida esperada para la función de pérdida al cuadrado. Ofrece información importante sobre el problema de clasificación, ya que puede descomponerse en términos de sesgo y varianza. Ambas son herramientas poderosas de las estadísticas de la teoría de muestreo para analizar escenarios de aprendizaje supervisado que tienen funciones de pérdida cuadrática [2]. Dado un objetivo fijo y el tamaño del conjunto de entrenamiento, $sesgo^2$ mide hasta qué punto la estimación promedio del algoritmo de aprendizaje (sobre todos los conjuntos de entrenamiento posibles del tamaño del conjunto de entrenamiento dado) coincide con el objetivo. En tanto que la *varianza*, mide cuánto fluctúa la suposición del algoritmo de aprendizaje para los diferentes conjuntos de entrenamiento del tamaño dado. Matemáticamente Sesgo (*Ses*) y varianza (*Var*) se definen:

$$Ses_x^2 = \frac{1}{2} \sum_{y \in Y} [P(Y_f = y|x) - P(Y_H = y|x)]^2 \quad Var_x = \frac{1}{2} (1 - \sum_{y \in Y} P(Y_H = y|x)^2).$$

En general, la pérdida esperada puede atribuirse a un *alto sesgo* o a una *alta varianza*, que suele ser una compensación entre estos dos términos. Idealmente, se busca un equilibrio entre estas tendencias opuestas, es decir, se prefiere un clasificador con un *sesgo aceptable* (reflejando suposiciones específicas de dominio o conjunto de datos) y una variación tan baja como sea posible.

2.1. Técnicas de Ensamblado

En [20], se menciona que un clasificador se llama inestable si las pequeñas perturbaciones en el conjunto de entrenamiento resultan en grandes cambios en la predicción o el límite de decisión. Los clasificadores de alta varianza son inherentemente inestables, ya que tienden a sobre ajustar los datos. Por otro lado, los métodos de alto sesgo suelen adecuarse a los datos y, por lo general, tienen una varianza baja. Los métodos de conjunto crean un clasificador combinado utilizando la salida de múltiples clasificadores básicos que se entrenan en diferentes conjuntos de datos. Dependiendo de cómo se seleccionen los conjuntos de entrenamiento y de la estabilidad de los clasificadores básicos, los clasificadores de conjunto pueden ayudar a estabilizar la varianza y el sesgo, lo que lleva a un mejor rendimiento general del modelo.

3. Materiales y Métodos

Dadas las pruebas realizadas en el presente artículo, se describe a continuación tanto los algoritmos de clasificación seleccionados, como así también el

Bagging	Boosting
<p>Genera múltiples conjuntos de entrenamiento y llama al algoritmo de aprendizaje modelo base con cada uno de ellos para producir una conjunción de modelos base [10]. $M^k(x) = \text{sign}(\sum_{i=1}^K M_i(x))$</p>	<p>En [2] se describe como una técnica que genera un conjunto de clasificadores y los vota. El algoritmo de aprendizaje genera los clasificadores en paralelo. $G(X) = \text{sign}(\sum_{m=1}^M a_m G_m(X))$</p>
<p>Intenta disminuir la varianza ya que cada modelo está construido de forma independiente.</p>	<p>Intenta disminuir el sesgo ya que los nuevos modelos están influenciados por el rendimiento de los modelos construidos previamente</p>

Cuadro 1. Cuadro comparativo entre Bagging y Boosting.

entorno experimental utilizado, en el Cuadro 1 se puede observar la comparación entre *Bagging* y *Boosting*.

3.1. Algoritmos de Clasificación

Random Forest(RF). Cada árbol de decisión que se crea se hace a partir de una secuencia aleatoria de un subconjunto de datos del conjunto de entrenamiento [9].

Extra Trees Forest(ETF). Construye un conjunto de árboles de regresión o decisión sin afinar. Sus dos diferencias principales con otros métodos de conjunto basados en árboles son que divide nodos al elegir puntos de corte completamente al azar y que utiliza toda la muestra de aprendizaje [6] [17].

K-Neighbors(KNN). Encuentra un grupo de k objetos en el conjunto de entrenamiento que están más cerca del objeto de prueba, y basa la asignación de una etiqueta en el predominio de una clase en particular en este conjunto [19].

Linear Support Vector Machine(SVM). Técnica utilizada para el aprendizaje supervisado para resolver problemas de clasificación y regresión basados en hiperplanos [8].

Decision Tree(DT). Estructura en la cual los datos son divididos de acuerdo a un criterio. Cada nodo del árbol denota un test sobre un atributo. Cada rama representa un resultado del test, y las hojas del nodo representan clases [15] [3].

Redes Neuronales(RN). Las entradas \mathbf{x}_i se recopilan de las neuronas anteriores (o el conjunto de datos) y se combinan mediante una función de combinación como \sum , que luego se ingresa en una función de activación (generalmente no lineal) para producir una respuesta de salida \mathbf{y} , que luego se canaliza otras neuronas. [4]:

Gaussian Naive Bayes(NB). Las Redes Bayesianas representan las dependencias que existen entre los atributos a través de una distribución de probabilidad condicional en un grafo dirigido acíclico [15].

AdaBoost(AB). Se encarga de analizar un número de algoritmos débiles, que serán árboles de decisión pequeños de profundidad 1, que tienen poca precisión

a través del conjunto de entrenamiento modificado en cada iteración. Secuencialmente formará uno nuevo más preciso con la combinación de todos ellos asignando un peso en la votación final para la elección de las mejores reglas de cada uno [2].

Logistic Regression(LR). Procedimiento cuantitativo donde la variable dependiente toma valores en un conjunto finito [9].

Gradient Boosting(GB). El modelo de conjunto construido es también una suma ponderada de clasificadores débiles. La conexión se realiza entre expansiones aditivas por etapas y minimización de descenso más pronunciado [5].

4. Experimentos y Análisis de Resultados

Para realizar los experimentos, se ha optado por el uso del *Scikit-learn*, ya que es un módulo de Python² que integra una amplia gama de algoritmos de aprendizaje automático de última generación para problemas supervisados y no supervisados de mediana escala [14], a su vez se optó por el uso del dataset de cáncer de mama de Wisconsin, ya integrado en el *Scikit-learn* para la realización de las pruebas [18]. Siguiendo con el proceso experimental, se seleccionan los siguientes algoritmos para entender las ventajas con respecto al *sesgo* y la *varianza* que se puede extraer. Al observar el Cuadro 2, las características con respecto al tipo de algoritmo, los modelos según su tipo, se hace el análisis sobre los mismos de manera de buscar el resultado adecuado. Se tiene en cuenta la precisión del modelo, debido a que el dataset utilizado se encuentra con una distribución de clases balanceada, lo que permite obtener un resultado aceptable para la interpretación final de los datos. La implementación de estas pruebas están disponibles en un repositorio público³.

Alg.	Var.	Sesgo	Definición
<i>RF</i>	Alta	Bajo	$C_{rf}^B(x) = v\{C_b(x)_1^B\}$
<i>ETF</i>	Alta	Bajo	$f_t(x) = \sum_{x^i, y^i \in I_s} a_i k_t(x^i, x)$
<i>KNN</i>	Alta	Bajo	$c_z = \operatorname{arg}_{v \in L} \max \sum_{y \in N} w_i x I(v = \operatorname{class}(c_y))$
<i>SVM</i>	Alta	Bajo	$K(x, x_i) = \operatorname{sum}(x * x_i)$
<i>DT</i>	Alta	Bajo	$I(P) = -(p_1 * \log_2(p_1) + p_2 * \log_2(p_2) + \dots + p_n * \log_2(p_n))$
<i>RN</i>	Alta	Bajo	$y_i = g_1(\sum_{j=1}^L w_{ij} s_j) = g_1(\sum_{j=1}^L w_{ij} (g_2(\sum_{r=1}^N t_{jr} x_r)))$
<i>NB</i>	Baja	Alto	$V_{nb} = \operatorname{argmax}_{v_j} V_j \prod_i P(a_i v_j)$
<i>AB</i>	Baja	Alto	$C * (x) = \operatorname{arg}_{y \in Y} \max \sum_{i: C_i(x)=y} \log \frac{1}{\beta_i}$
<i>LR</i>	Baja	Alto	$\operatorname{logit}(p(y = 1 x)) = w_0 x_0 + w_1 x_1 + \dots + w_m x_m = \sum_{i=0}^m w_i x_i = w^t x$
<i>GB</i>	Baja	Alto	$F_m = (x) = F_{m-1}(x_i) + p_m h(x_i; \mathbf{a}_m)$

Cuadro 2. Algoritmos ordenados por *varianza* y *sesgo*.

² <https://www.python.org>

³ <https://github.com/andresmacielc/Stacking>

Lo primero es definir las variables a ser utilizadas y los objetivos a ser encontrados al realizar el análisis de aprendizaje supervisado, para que el paciente sepa si posee o no cáncer de mama. Una vez finalizado, se avanza al siguiente paso, el cual implicará normalizar los datos [7]; con esto se busca solución a diferentes ruidos, que se presentan en el *dataset*; luego se procede a dividir el *dataset* en conjunto de entrenamiento y testeo. Se utiliza el paquete *model selection()*, instalado por defecto en la biblioteca *scikit-learn*. Con la validación cruzada se hacen las pruebas a la precisión de la clasificación de algoritmos en las versiones *bagging* de cada uno, de modo de retratar las diferencias que presentan los resultados finales de dicha estimación.

```
clf_array = [rf, et, knn, svc, dt, nb, mlp]
for clf in clf_array:
    scores = cross_val_score(clf, X_train_scaled, y_train, cv=3,
                             n_jobs=-1)
    bagging_clf = BaggingClassifier(clf,
                                    max_samples=0.8, max_features=1.0, random_state=seed)
    bagging_scores = cross_val_score(bagging_clf, X, y, cv=3,
                                     n_jobs=-1)
    print("Media de: {1:.3f}, std: (+/-) {2:.3f}
          [Bagging {0}]\n".format
          (clf.__class__.__name__,
           bagging_scores.mean(), bagging_scores.std()))
```

El código muestra los resultados en las ejecuciones estudiadas, con la aclaración que los algoritmos *RF*, *ETF* y *KNN* poseen un balance de 100 de estimaciones en los 2 primeros y 100 *neighbors* en el tercero. Los resultados en la figura 1.

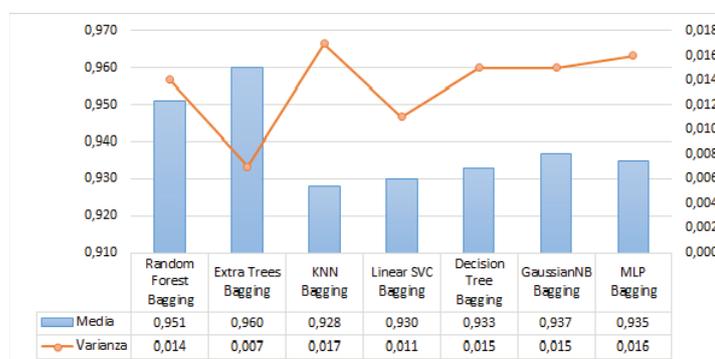


Figura 1. Comparación de precisión en la técnica del *bagging*.

Las versiones del *bagging* presentan mayor varianza, debido a que algunos de los algoritmos fueron ejecutados con baja cantidad de estimadores. En determinados casos el *bagging* es efectivo con datos limitados. El enfoque del *bagging*

consiste en usar un par de sub-muestras y agruparlas, si la precisión del conjunto es mayor que la de los modelos básicos, funciona. Nótese que el uso de submuestras más grandes no garantiza un mejor resultado. El *bagging* puede mejorar el conjunto en gran medida cuando tiene un modelo inestable; sin embargo, cuando sus modelos son más estables, entrenados en submuestras mayores, las mejoras de *bagging* se reducen.

Para el caso del *boosting*, se analiza el funcionamiento de los algoritmos *AB* y *GB*, en los cuales no se tuvieron en cuenta los algoritmos utilizados en el *bagging*. Luego se realizan las comparaciones mediante la implementación del siguiente *script*:

```
boost_array = [ada_boost, grad_boost]
eclf = VotingClassifier(estimators=[('Ada Boost', ada_boost),
('GradientBoost', grad_boost)], voting='hard')
labels = ['Ada Boost', 'Grad Boost', 'Ensemble']
for clf, label in zip([ada_boost, grad_boost, eclf],
labels):
    scores = cross_val_score(clf, X_train_scaled, y_train, cv=10,
scoring='accuracy')
    print("Media: {0:.3f}, std: (+/-) {1:.3f}
    [{2}]" .format(scores.mean(), scores.std(), label))
```

En la Figura 2, se muestra los resultados comparativos obtenidos.

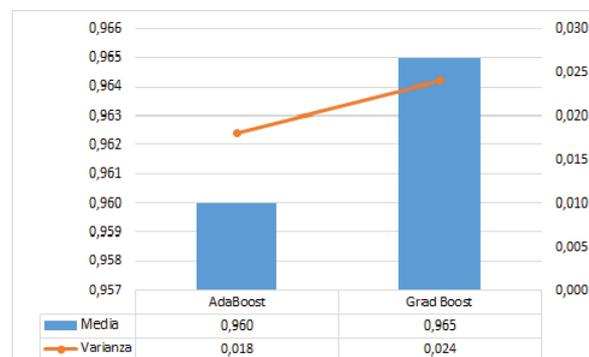


Figura 2. Comparación de precisión en la técnica del boosting.

En la Figura 2, las diferencias de resultados presentan una baja variación entre ellos, se observa a su vez que el *sesgo* es más preciso en el algoritmo *GB*, ya que al basarse en el descenso de gradiente, realiza búsquedas más precisas en el conjunto de datos entrenados erróneamente.

Realizadas las clasificaciones correspondientes, se procede a la combinación de los algoritmos que son implementados en el *ensamblado*, los cuales, basados

en las características que presentan los modelos descritos anteriormente, tanto en el *bagging* como en el *boosting*, permiten encontrar la mejor combinación de los métodos, aplicados a los clasificadores base y a los específicos, de manera que el resultante obtenga estabilidad en cuanto al *sesgo* y a la *varianza*. Para el ensamblado se utilizaron todos los algoritmos básicos implementados previamente, sumados a estos los algoritmos *RF* y *ETF*, los cuales ya implementan el método de *bagging* por defecto, y los algoritmos *AB* y *GB* que realizan lo propio desde el lado del *boosting*, de modo a que el ensamblado final realice el aprendizaje tanto de los clasificadores simples, del *bagging* y del *boosting*.

Para dicho propósito se utiliza la librería *StackingClassifier*, en donde para la implementación del *ensamblado*, es necesaria la selección de un clasificador final que pueda funcionar como base de los resultados propuestos por los anteriores, dado el hecho que la mayoría de los algoritmos utilizados previamente se caracterizan por poseer una *varianza* más alta, seleccionar un algoritmo de *sesgo* alto funciona como un nivelador de las variaciones que presentan los clasificadores, entonces se opta por el algoritmo de *LR*, resultando de la siguiente forma:

```
sclf = StackingClassifier(classifiers=
[rf, et, knn, svc, mlp, nb, dt, adab, gd], meta_classifier=lr)

classifier_array = [rf, et, knn, svc, mlp, nb, dt, adab, gd,
sclf]
labels = [clf.__class__.__name__ for clf in classifier_array]

for clf, label in zip(classifier_array, labels):
    cv_scores = cross_val_score(clf, X, y, cv=10,
scoring='accuracy')
    print("Media: %0.4f (+/- %0.4f) [%s]" % (cv_scores.mean(),
cv_scores.std(), label))
```

En la Figura 3 se muestran los resultados comparativos obtenidos.

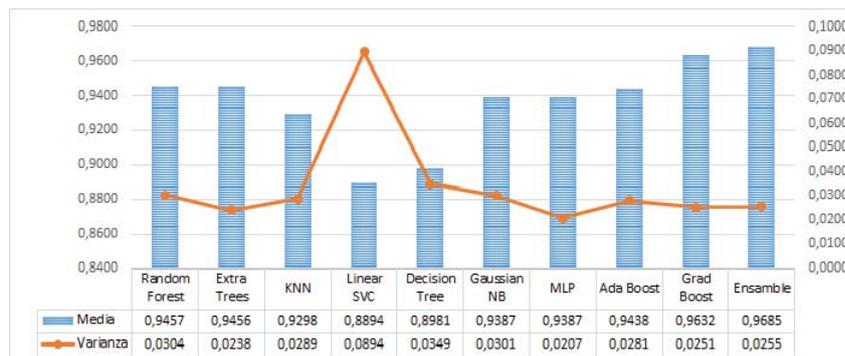


Figura 3. Comparación de precisión de Algoritmos Individuales vs Ensamblado.

Puede observarse en la Figura 3 que la combinación de los algoritmos posee el mejor *sesgo* del conjunto, y una *varianza* estable respecto a los demás.

A diferencia de los ejemplos anteriores, esta vez se buscó una mayor estabilidad en cuanto a la precisión de los algoritmos se refiere, como se detalló en [20], el aumento de la *varianza* de un algoritmo implica indefectiblemente en su reducción en el *sesgo*, y viceversa, esta situación de por sí es algo inevitable, sin embargo, para casos particulares de algunos algoritmos, como serían el *KNN* y *SVM* o el *RF*, en los cuales aumentando la cantidad de *neighbors*, la cantidad de estimadores o de parámetros, puede obtener un aumento en el *sesgo*, y al mismo tiempo una reducción de la *varianza*.

Dicho de otra manera, en la figura 3 se puede observar la prueba de dicha afirmación al comparar las estimaciones de los algoritmos *KNN* y *DT*. En el caso del primero se definieron los *neighbors* lo que permitió el aumento del *sesgo*, por otra parte, en el segundo no se definieron mayores cantidad de parámetros, por lo que la cantidad de *varianza* se mantiene alta, a su vez, el *sesgo* permanece bajo. A través del balance obtenido con la combinación de los algoritmos se llega a un *multiclasificador* con estabilidad relativa en términos de *sesgo* y *varianza* y con una exactitud en la precisión mejorada tomando las características de su combinación.

5. Conclusión

En este estudio, se ha contribuido a evidenciar los resultados de la aplicación de un *ensamblado de clasificadores* propuesto en razón al balance entre el *sesgo* y la *varianza* de cada clasificador para la detección de cáncer de mama mediante el uso de herramientas de software libre. Tomados en conjunto, los resultados obtenidos proporcionan apoyo a la premisa que el ensamblado de los clasificadores en bruto, no garantiza una mejora automática en cuanto a la precisión de la clasificación, ya que al momento de realizar un apilado de algoritmos, deberían considerarse los factores identificados y referidos en esta investigación.

Finalmente, basándose en la presencia de un aumento en la precisión obtenida para la clasificación de tumores maligno o benigno, es posible subrayar la necesidad de seguir trabajando para incrementar el entendimiento sobre la conjunción de clasificadores y su generalización en otros dominios de aplicación.

Referencias

1. AGARAP, A. F. M. On breast cancer detection: an application of machine learning algorithms on the wisconsin diagnostic dataset. In *Proceedings of the 2nd International Conference on Machine Learning and Soft Computing* (2018), ACM, pp. 5–9.
2. BAUER E., K. R. An empirical comparison of voting classification algorithms: Bagging, boosting, and variants. *Machine Learning* (2018).
3. BLANCO F., ORTIZ A., R. G. M. R. C. Y. Clasificadores y multiclasificadores com cambio de concepto basados en arboles de decision. *Inteligencia Artificial. Revista Iberoamericana de Inteligencia Artificial* (2010).

4. D., L. *DISCOVERING KNOWLEDGE IN DATA An Introduction to Data Mining*. John Wiley and Sons, Inc., 2005.
5. G., F. Greedy function approximation: a gradient boosting machine. *The Annals of Statistics* (2001).
6. GEURTS P., ERNST D., W. L. Extremely randomized trees. *Machine Learning* (2006).
7. HAN J., K. M. *Data Mining: Concepts and Techniques, Second Edition*. Morgan Kaufmann Publishers, 2006.
8. HUINCALEF R., URRUTIA G., I. G. M. D. Recognition of surface irregularities on roads: a machine learning approach on 3d models. *Congreso Argentino de Ciencias de la computacion* (2018).
9. J., Z. *Comparativa y Analisis de Algoritmos de Aprendizaje Automatico para la Prediccion del Tipo Predominante de Cubierta Arborea*. Benemérita Universidad Autónoma de Puebla, 2017.
10. KOTSIAANTIS S. B., P. P. E. Combining bagging and boosting. *World Academy of Science, Engineering and Technology International Journal of Mathematical and Computational Sciences* (2007).
11. MARTÍNEZ-MUÑOZ G., S. A. Comites de arboles igp.
12. OLARTE, E., PANIZZI, M. D., AND BERTONE, R. A. Segmentación de mercado usando técnicas de minería de datos en redes sociales. In *XXIV Congreso Argentino de Ciencias de la Computación (La Plata, 2018)*. (2018).
13. O'SULLIVAN, C. C., LOPRINZI, C. L., AND HADDAD, T. C. Updates in the evaluation and management of breast cancer. In *Mayo Clinic Proceedings* (2018), vol. 93, Elsevier, pp. 794–807.
14. PEDREGOSA, F., VAROQUAUX, G., GRAMFORT, A., MICHEL, V., THIRION, B., GRISEL, O., BLONDEL, M., PRETTENHOFER, P., WEISS, R., DUBOURG, V., ET AL. Scikit-learn: Machine learning in python. *Journal of machine learning research* 12, Oct (2011), 2825–2830.
15. PORTUGAL R., C. M. Ensamble de algoritmos bayesianos con arboles de decision, una alternativa de clasificacion.
16. SALAMA, G. I., ABDELHALIM, M., AND ZEID, M. A.-E. Breast cancer diagnosis on three different datasets using multi-classifiers. *Breast Cancer (WDBC)* 32, 569 (2012), 2.
17. WEHENKEL L., ERNST D., G. P. Ensembles of extremely randomized trees and some generic applications. *RTE-VT workshop* (2006).
18. WOLBERG, W. H., STREET, W. N., AND MANGASARIAN, O. L. Breast cancer wisconsin (diagnostic) data set. *UCI Machine Learning Repository [http://archive.ics.uci.edu/ml/]* (1992).
19. WU X., K. V. *The top ten algorithms in Data Mining*. Chapman and Hall, 2009.
20. ZAKI M., M. W. *Data Mining and Analysis: Fundamental Concepts and Algorithms*. Cambridge University Press New York, NY, USA ©2014, 2013.

Evaluación de la precisión de acierto de un conjunto desbalanceado mediante la combinación de clasificadores

Nancy B. Ganz¹, Alicia E. Ares¹ y Horacio D. Kuna²

¹ Instituto de Materiales de Misiones (IMAM-CONICET), Facultad de Ciencias Exactas Químicas y Naturales, Universidad Nacional de Misiones, Posadas, Misiones, Argentina.

² Instituto de Investigación, Desarrollo e Innovación en Informática (IIDII), Facultad de Ciencias Exactas Químicas y Naturales, Universidad Nacional de Misiones, Posadas, Misiones, Argentina.

{nancy.bea.ganz, a.e.ares, hdkuna}@gmail.com

Resumen. El campo de la ciencia de datos ha tenido muchos avances respecto a la aplicación y desarrollo de técnicas en el sector de la salud. Sin embargo, no se han encontrado trabajos que apliquen procesos o técnicas de aprendizaje automático específicamente al campo de la odontología, como es el caso de los injertos o implantes dentales. En este trabajo se estudia la aplicación de varios clasificadores para evaluar la precisión de clasificación de forma integrada sobre un conjunto desbalanceado, referido a historias clínicas de implantes dentales recolectados en puntos característicos de la Provincia de Misiones, Argentina. El modelo abarcó la combinación de los clasificadores: Random Forest, SVC, KNN, Naive Bayes y un Perceptrón Multicapa. La integración de las predicciones se realiza a través del promedio de las mismas, empleando un umbral y pesos en los clasificadores. Como resultado, cada clasificador logró hasta un 68% de acierto de la clase minoritaria, mientras que la integración de todas las predicciones permitió alcanzar el 72% de casos correctamente identificados como fracaso.

Palabras claves: ensamble de clasificadores, predicción, clase minoritaria, postoperatorio, implantes dentales.

1 Introducción

La combinación de modelos puede ser elemental a la hora de tomar una decisión, debido a que trata de obtener la solución más eficiente para un problema en cuestión. Es posible combinar decisiones obtenidas con el mismo o distintos clasificadores de base [1].

Los métodos de combinación de clasificadores, son aquellos donde dado un conjunto de clasificadores ya entrenados, combinan de distintas formas los resultados para retornar un valor más preciso que el de los clasificadores individuales [2]. La integración suele ser más precisa, porque los datos de entrenamiento pueden no proporcionar información suficiente para elegir el mejor clasificador y en esta situación la combinación es la mejor opción [3].

En este trabajo se estudia la aplicación de varios métodos de clasificación para la predicción del resultado postoperatorio (éxito o fracaso) de un conjunto de datos de implantes dentales. Con el objetivo de aumentar el acierto de la clase minoritaria o de

los fracasos. El conjunto de datos se basa en historias clínicas de pacientes que se han sometido a procesos quirúrgicos de colocación de implantes dentales en la Provincia de Misiones, Argentina. El procedimiento que se propone en este trabajo utiliza los clasificadores: Random Forest (RF) [4], SVC [5], KNN [6], Naive Bayes (NB) [7] y Perceptrón Multicapa (MLP) [8]. El ensamble consistió en aplicar pesos a los clasificadores y promediar sus predicciones. Finalmente, para asignar la etiqueta de clase final se aplicó un umbral.

La contribución de este trabajo es un modelo de aprendizaje automático para la predicción del resultado postoperatorio (éxito o fracaso) en implantes dentales, el cual es un dominio de poco conocimiento. Asimismo, demostramos que los sistemas de múltiples clasificadores también pueden ser aplicados al estudio de caso, ya que permite lograr mejores rendimientos de clasificación que los alcanzados de manera individual por los clasificadores.

El resto del documento está estructurado de la siguiente manera. En la sección 2 se exponen antecedentes en la combinación de clasificadores, en la sección 3 se describe el enfoque propuesto, la sección 4 presenta los resultados experimentales llevados a cabo, y en la sección 5 se exhibe las principales conclusiones y líneas de investigación futuras.

2 Trabajos Relacionados

Existen varios trabajos de investigación sobre la combinación o integración de clasificadores, para mejorar el acierto de predicción o inclusive para no sesgar la decisión sobre los resultados de un solo clasificador [9]. A continuación, se presentan y examinan algunos trabajos previos sobre este tema.

Miao et al. [10] proponen un procedimiento para mejorar la precisión en la identificación de genes mediante la integración de los clasificadores SVM, RF y máquina de aprendizaje externo (ELM). Luego del entrenamiento y predicción con los tres clasificadores, los resultados fueron combinados a través del método de votación mayoritaria [9]. Lograron obtener una precisión mayor a través de la integración de las predicciones que de forma individual. De igual manera, Catal y Nengir [11] han utilizado los clasificadores NB y SVM mediante la técnica de voto mayoritario para la integración de las predicciones. Demostraron a través de sus experimentaciones sobre varios conjuntos de datos que los sistemas de clasificadores múltiples mejoran la precisión. Otro trabajo de similares características es el de Pandey y Taruna [12], donde proponen un clasificador integrado utilizando un árbol de decisión J48, KNN y agregación de estimadores de una dependencia (AODE), sobre un conjunto de datos de rendimiento académico de estudiantes de ingeniería. En este modelo, cada clasificador individual genera su valor de predicción y se integran a través del producto de las probabilidades, donde la etiqueta de clase final está representada por el máximo de una probabilidad posterior. Además, Mukhopadhyay et al. [13], utilizan diferentes métodos de combinación para el reconocimiento de imágenes en documentos, con el mismo clasificador de base MLP. Consiguen una precisión máxima del 98,45% con una mejora del 7%, sobre el clasificador individual de mejor desempeño.

Inspirados en las ideas anteriores, proponemos un procedimiento de aprendizaje automático mediante la utilización de múltiples clasificadores para un dominio de poco conocimiento, como es el caso de los implantes dentales. El enfoque propuesto fue capaz de superar el porcentaje de acierto logrado por cada clasificador.

3 Materiales y Métodos

La metodología de trabajo que se plantea, consistió en llevar a cabo un conjunto de pasos para obtener el grado de pertenencia del atributo clase. El enfoque propuesto se resume en la Figura 1.

Para el propósito de este trabajo, se utilizó un conjunto de datos de historias clínicas de pacientes que se han sometido a procesos quirúrgicos de colocación de implantes dentales en la Provincia de Misiones, Argentina. Este conjunto de datos estaba representado a través de 4 dimensiones: datos del paciente (antecedentes y condiciones médicas de los pacientes a la hora de la intervención), datos del implante (características del implante utilizado por el especialista implantólogo), datos de la fase quirúrgica (procedimiento de intervención quirúrgica y mejoramiento del lecho óseo del paciente) y datos del seguimiento postoperatorio (resultado del proceso de colocación del implante, es decir si el proceso de oseointegración implante/tejido-óseo tuvo éxito o fracasó). El mismo contaba con un total de 1165 tuplas, 34 características y tenía la particularidad de ser un conjunto desbalanceado, es decir atributo de clase binario con 1009 casos etiquetados como éxito y 156 como fracaso.

Primeramente, se procedió a leer el conjunto de datos y seleccionar la clase objetivo a predecir (característica que representa el resultado del proceso postoperatorio), para pasar a seleccionar las características más relevantes a través de la combinación de los métodos: Information Gain[14], Gain Ratio[15], Random Forest importance[4], Relief[16] y Chi Squared[17]. El procedimiento de combinación de estos métodos se describe en el trabajo [18].

Luego de obtener el subconjunto de características más importantes, se procedió a dividir el conjunto de datos. Una estrategia común consiste en tomar todos los datos etiquetados y dividirlos en subconjunto de entrenamiento y evaluación, normalmente con una proporción del 70 al 80 % para entrenamiento y un 20 al 30 % para evaluación o prueba [19]–[21]. Esta división va a depender en gran medida del número total de muestras y del modelo a entrenar [22], [23]. En nuestro caso se dividió los datos de forma aleatoria para preservar la distribución de ambas clases en: 70 % para entrenamiento y 30 % para evaluación [24]–[27]. Garantizando que todos los casos se encuentren representados en ambos conjuntos.

Conjuntamente, se definió cada uno de los clasificadores seleccionados para el estudio de caso. Todos fueron empleados de forma probabilística. Estos algoritmos de aprendizaje automático poseen hiper parámetros que deben configurarse para lograr su mejor rendimiento. Por esta razón, se precisó:

1. Un espacio de búsqueda: se definió valores para los hiper parámetros y se fue ajustando en función de la medida de rendimiento.

2. Un algoritmo de optimización o ajuste: se empleó el método GridSearchCV [28], es costoso en cuanto a rendimiento, pero permite cubrir todo el espacio de búsqueda.
3. Un método de evaluación: se utilizó validación cruzada de 10 iteraciones.
4. Una medida de rendimiento: se fijó la métrica precisión de equilibrio [29].

Toda la implementación del enfoque propuesto se realizó sobre Python con la librería “Scikit-learn” [30]. En la Tabla 1, se exponen los hiper parámetros que se buscó ajustar para lograr el mejor desempeño de cada uno de los clasificadores, sobre el conjunto de datos utilizado.

Tabla 1. Hiper parámetros y espacio de búsqueda definido para los clasificadores: RF, SVC, KNN, NB y MLP.

Clasificadores	Parámetros	Espacio de búsqueda
RF	<i>n_estimators</i>	<i>range (1, 200)</i>
	<i>criterion</i>	<i>gini, entropy</i>
	<i>bootstrap</i>	<i>True, False</i>
SVC	<i>kernel</i>	<i>linear, rbf, poly</i>
	<i>C</i>	<i>range (1, 10)</i>
	<i>gamma</i>	<i>range (1, 10)</i>
	<i>degree</i>	<i>range (1, 10)</i>
KNN	<i>n_neighbors</i>	<i>range (1, 100)</i>
	<i>weights</i>	<i>uniform, distance</i>
	<i>p</i>	<i>manhattan, euclidean</i>
NB	<i>alpha</i>	<i>[0, 0.1, 0.2, 0.3, ..., 0.9, 1]</i>
	<i>fit_prior</i>	<i>True, False</i>
	<i>class_prior</i>	<i>[0.5,0.5], [0.4,0.6], [0.6,0.4]</i>
MLP	<i>hidden_layer_sizes</i>	<i>range (1,5)</i>
	<i>activation</i>	<i>logistic, tanh, relu</i>
	<i>alpha</i>	<i>[0.0001, 0.05]</i>
	<i>solver</i>	<i>lbfgs, sgd, adam</i>
	<i>learning_rate</i>	<i>constant, invscaling</i>

Tanto para el entrenamiento como para la predicción se obtuvieron métricas para evaluar el rendimiento final. Estas métricas fueron: matriz de confusión [31] y exactitud [29].

La integración de las predicciones consistió en multiplicar para cada tupla el valor de probabilidad de cada clase, obtenida por cada clasificador por el peso asignado al mismo. El peso fue determinado mediante una búsqueda en cuadrícula utilizando un parámetro de prueba con valores comprendidos entre 0 y 1. Esta búsqueda fue sometida a una validación cruzada de 10 iteraciones, donde se medió la exactitud [31], [32] de cada clasificador para la clase en cuestión, seleccionando el valor que logró la mejor precisión [33], [34].

Finalmente, se aplicó un umbral para determinar la etiqueta de clase final. Este umbral también estuvo determinado por una búsqueda en cuadrícula, utilizando valores

comprendidos entre 0.1 y 0.5 con incrementos de 0.1 en cada prueba. Se seleccionó el valor del umbral que permitió obtener el mejor resultado de clasificación (matriz de confusión).

La probabilidad de la etiqueta de clase estuvo dado por:

$$prob[c][i] = ((prob.RF.c * w_1) + (prob.SVC.c * w_2) + (prob.KNN.c * w_3) + (prob.MNB.c * w_4) + (prob.MLP.c * w_5))/5$$

$$prob[predic][i] = np.where(prob[P] \geq \mu, 'FRACASO', 'EXITO')$$

Donde:

- ❖ i : número de tupla.
- ❖ c : etiqueta de clase, E (*EXITO*) o F (*FRACASO*).
- ❖ $prob.[RF | SVC | KNN | NB | MLP].c$: probabilidad obtenida por el clasificador RF, SVC, KNN, NB o MLP para la clase E o F.
- ❖ $w_1, w_2 \dots w_5$: peso asignado a cada clasificador en función de su desempeño para c . Donde $w \in \mathbb{Q}$, $\{w / 0 \leq w \leq 1\}$, siendo 0 el peso menor y 1 el mayor.
- ❖ $prob[c][i]$: valor de probabilidad final de c , promediada y multiplicada por los pesos definidos para cada clasificador.
- ❖ μ : valor de probabilidad necesario para considerar a la etiqueta c como F.
- ❖ $probl[predic][i]$: etiqueta de clase final asignada en base al umbral μ .

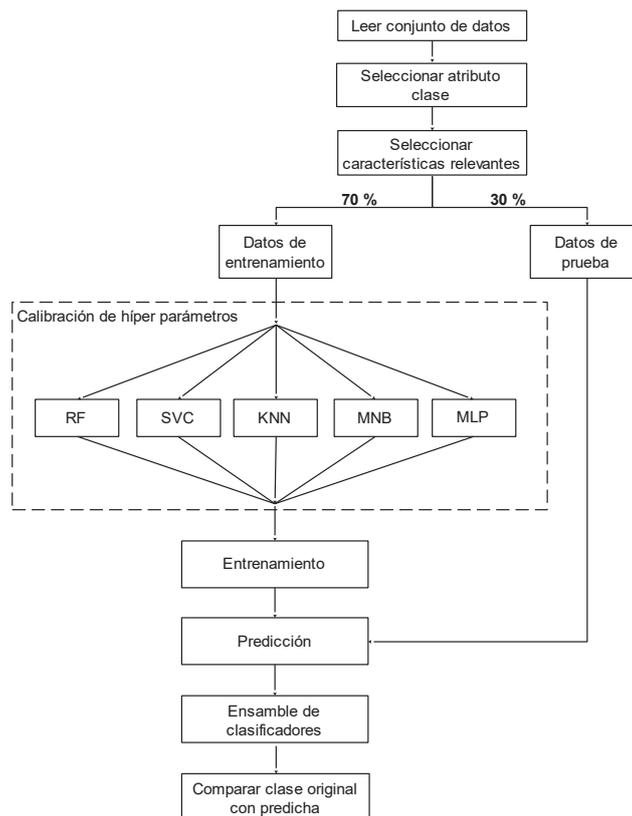


Fig. 1. Enfoque propuesto para la integración de las predicciones de los clasificadores utilizados sobre el conjunto de datos desbalanceado de implantes dentales.

4 Resultados

A continuación, se presentan los resultados logrados en este trabajo de investigación sobre los datos de prueba del conjunto de datos utilizados.

Luego de la selección de características sobre el conjunto de implantes dentales, se logró un subconjunto conformado por 14 características de un total de 34. En la tabla 2 se presentan las matrices de confusión y el porcentaje de acierto (exactitud) obtenido por cada clasificador de forma individual.

Tabla 2. Matiz de confusión y exactitud alcanzada por los clasificadores RF, SVC, KNN, NB y MLP sobre el conjunto de datos de implantes dentales.

Clasificador	EXITO	FRACASO	Exactitud
<i>RF</i>	292 16	8 34	93%
<i>SVC</i>	294 17	6 33	93%
<i>KNN</i>	294 17	6 33	93%
<i>NB</i>	250 24	50 26	79%
<i>MLP</i>	281 16	19 34	90%

Una vez encontrados los pesos para cada clasificador con la validación cruzada, se multiplicó las predicciones por el peso definido a cada clasificador, se promedió y se aplicó el umbral.

El valor del umbral que permitió obtener el mejor resultado de clasificación fue $\mu \geq 0.3$. Es decir, todas aquellas probabilidades mayores o iguales a 0.3 para la clase fracaso fueron consideradas verdaderos fracasos, los casos contrarios fueron considerados como éxitos.

El ensamble permitió confeccionar una matriz de confusión del enfoque propuesto. En la tabla 3 se aprecia que se logró incrementar el acierto de la clase minoritaria (fracaso).

Tabla 3. Matiz de confusión y exactitud alcanzada por el enfoque propuesto sobre el conjunto de datos de implantes dentales.

Clasificador	EXITO	FRACASO	Exactitud
<i>Enfoque propuesto</i>	284 14	16 36	91%

En la figura 2, se observa que si bien el rendimiento de la integración de las predicciones no fue la mejor opción para la clase éxito, no quiere decir que éste haya sido el peor en comparación con las predicciones individuales. Mientras que en la figura

3, se aprecia que la integración de las probabilidades fue la mejor opción, ya que permitió obtener el mayor porcentaje de acierto de la clase minoritaria.

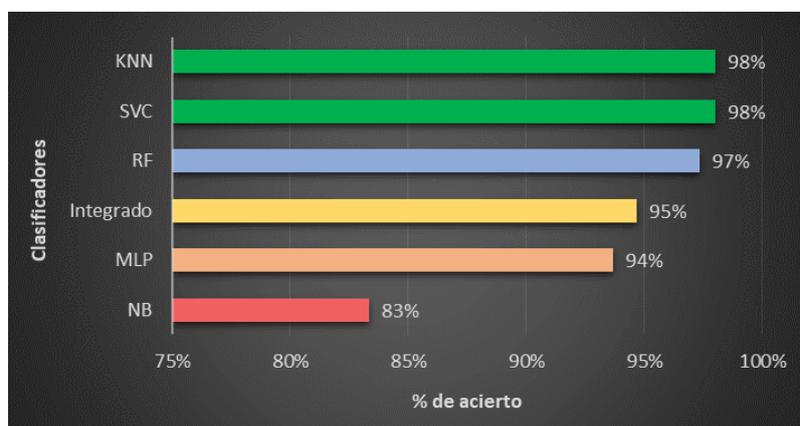


Fig. 2. Eficiencia de cada clasificador para la clase “EXITO” en comparación con el enfoque propuesto.

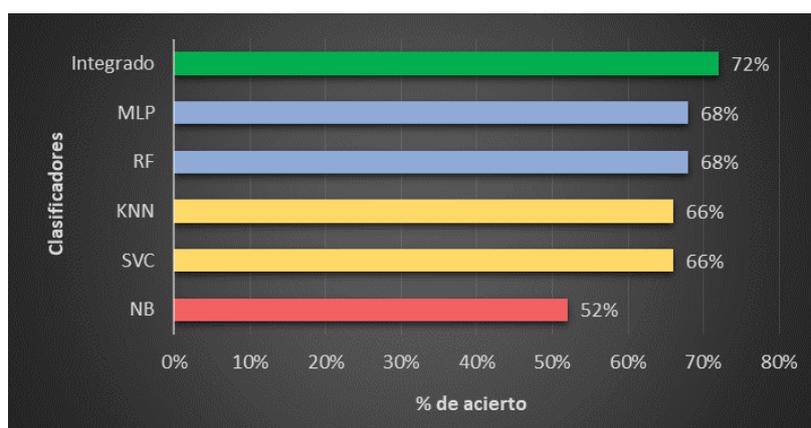


Fig. 3. Eficiencia de cada clasificador para la clase “FRACASO” en comparación con el enfoque propuesto.

Mediante la experimentación realizada sobre el conjunto de datos de implantes dentales, la integración de las probabilidades de los clasificadores logró el mejor porcentaje de acierto de la clase minoritaria, en comparación con el rendimiento de los clasificadores de forma individual. Donde se logró un acierto del 72% de forma integrada, mientras que de forma individual los clasificadores lograron alcanzar un máximo del 68% de acierto de los fracasos. Si bien la exactitud del enfoque propuesto no fue el mejor, fue el que más fracasos acertó.

Los expertos consultados en patologías bucales y rehabilitación compleja en implantología oral, de distintos puntos de la provincia de Misiones, Argentina,

coincidieron y remarcaron que en este campo de estudio es menos delicado etiquetar un caso como fracaso, que etiquetarlo como éxito cuando era un eventual fracaso.

5 Conclusiones y futuras líneas de investigación

Las técnicas de clasificación se basan en conceptos de estimación diferentes, tratando siempre de asociar las mejores características de cada uno de ellos, combinándolos de alguna forma. La finalidad de este trabajo fue la aplicación de varios clasificadores, para predecir el resultado del proceso postoperatorio de un conjunto de implantes dentales y tratar de aumentar el acierto de la clase minoritaria. Se buscó demostrar que en ocasiones es mejor integrar las predicciones de los clasificadores, para no sesgar la decisión sobre un solo resultado. Además, utilizar predicciones integradas permite conocer diversos resultados para un mismo caso, ya que se utilizan más de un clasificador y esto posibilita una asignación de etiqueta o clasificación más precisa.

Como resultado, cada clasificador logró hasta un 68% de acierto para la clase minoritaria, mientras que la integración de todas las predicciones permitió alcanzar el 72% de casos correctamente identificados como fracasos.

Se alcanzó apreciar que si bien KNN y SVC fueron los clasificadores que mejor acierto de la clase éxito lograron, no fue así para la clase fracaso. De la misma manera MLP y RF fueron los clasificadores que mejor acierto de fracasos alcanzaron, no siendo así para la clase éxito. Por este motivo es mejor tener la opinión o predicción de varios modelos, para comparar resultados a la hora de la toma de decisiones.

En cuanto al clasificador NB se valoró que su rendimiento o porcentaje de acierto fue el menor para ambas clases. Se buscó calibrar y optimizar todo lo posible, y aun así no se lograron resultados comparables a los demás clasificadores.

Finalmente, se plantea como trabajo futuro validar el enfoque propuesto con otros conjuntos de datos del área de la salud o la medicina. Conjuntamente, se plantea validar con conjuntos de datos artificiales. Además, se propone la inclusión o ampliación de los clasificadores utilizados, para evaluar la posibilidad de ajustar el porcentaje de acierto de ambas clases.

Referencias

- [1] Y. Lu, "Knowledge integration in a multiple classifier system," *Appl. Intell.*, vol. 6, no. 2, pp. 75–86, 1996.
- [2] L. I. Kuncheva, *Combining Pattern Classifiers: Methods and Algorithms*, 2nd ed. John Wiley & Sons, 2014.
- [3] M. Mohandes, M. Deriche, and S. O. Aliyu, "Classifiers Combination Techniques: A Comprehensive Review," *IEEE Access*, vol. 6, pp. 19626–19639, 2018.
- [4] L. Breiman, "Random Forest," *Mach. Learn.*, vol. 45, no. 1, pp. 5–32, 2001.
- [5] C. Chang and C. Lin, "LIBSVM: A Library for Support Vector Machines," *ACM Trans. Intell. Syst. Technol.*, vol. 2, no. 3, pp. 1–39, 2011.
- [6] N. S. Altman, "An Introduction to Kernel and Nearest-Neighbor

- Nonparametric Regression,” *Am. Stat.*, vol. 46, no. 3, pp. 175–185, 1992.
- [7] C. D. Manning, P. Raghavan, and H. Schütze, “Text classification and Naive Bayes,” in *Introduction to Information Retrieval*, Cambridge University Press, 2009, pp. 253–287.
- [8] B. Irie and Sei Miyake, “Capabilities of Three-layered Perceptrons,” *IEEE International Conf. Neural Networks*, pp. 641–648, 1988.
- [9] J. Fierrez, A. Morales, R. Vera-Rodriguez, and D. Camacho, “Multiple classifiers in biometrics. part 1: Fundamentals and review,” *Inf. Fusion*, vol. 44, no. December 2017, pp. 57–64, 2018.
- [10] Y. Miao, H. Jiang, H. Liu, and Y. dong Yao, “An Alzheimers disease related genes identification method based on multiple classifier integration,” *Comput. Methods Programs Biomed.*, vol. 150, pp. 107–115, 2017.
- [11] C. Catal and M. Nangir, “A sentiment classification model based on multiple classifiers,” *Appl. Soft Comput. J.*, vol. 50, pp. 135–141, 2017.
- [12] M. Pandey and S. Taruna, “Towards the integration of multiple classifier pertaining to the Student’s performance prediction,” *Perspect. Sci.*, vol. 8, pp. 364–366, 2016.
- [13] A. Mukhopadhyay, P. Singh, R. Sarkar, and M. Nasipuri, “A Study of Different Classifier Combination Approaches for Handwritten Indic Script Recognition,” *J. Imaging*, vol. 4, no. 2, pp. 1–21, 2018.
- [14] C. E. Shannon, “A Mathematical Theory of Communication,” *Bell Syst. Tech. J.*, vol. 27, no. 3, pp. 379–423, 1948.
- [15] J. R. Quinlan, “Induction of Decision Trees,” *Mach. Learn.*, vol. 1, no. 1, pp. 81–106, 1986.
- [16] K. Kira and L. A. Rendell, “A Practical Approach to Feature Selection,” *Mach. Learn. Proc.*, pp. 249–256, 1992.
- [17] K. Pearson, “On the criterion that a given system of deviations from the probable in the case of a correlated system of variables is such that it can be reasonably supposed to have arisen from random sampling,” *London, Edinburgh, Dublin Philos. Mag. J. Sci.*, vol. 50, no. 302, pp. 157–175, 1900.
- [18] N. B. Ganz, F. A. Domínguez, A. E. Ares, and H. D. Kuna, “Selección de características mediante la combinación de métodos para evaluar la precisión de clasificación en un conjunto de datos de implantes dentales,” *XXI Workshop de Investigadores en Ciencias de la Computación*. San Juan, Argentina, pp. 263–267, 2019.
- [19] G. Singh, “Surface Treatment of dental implants: A review,” *IOSR J. Dent. Med. Sci.*, vol. 17, no. 2, pp. 49–53, 2018.
- [20] M. Moran and G. Gordon, “Curious Feature Selection,” *Inf. Sci. (Ny)*, vol. 485, pp. 42–54, 2019.
- [21] I. Sumaiya Thaseen and C. Aswani Kumar, “Intrusion detection model using fusion of chi-square feature selection and multi class SVM,” *J. King Saud Univ. - Comput. Inf. Sci.*, vol. 29, no. 4, pp. 462–472, 2017.
- [22] X. Fan and H. Shin, “Road vanishing point detection using weber adaptive local filter and salient-block-wise weighted soft voting,” *IET Comput. Vis.*, vol. 10, no. 6, pp. 503–512, 2016.
- [23] Y. Quan, Y. Xu, Y. Sun, and Y. Huang, “Supervised dictionary learning with multiple classifier integration,” *Pattern Recognit.*, vol. 55, pp. 247–260, 2016.

- [24] B. T. Pham, M. D. Nguyen, K. T. T. Bui, I. Prakash, K. Chapi, and D. T. Bui, “A novel artificial intelligence approach based on Multi-layer Perceptron Neural Network and Biogeography-based Optimization for predicting coefficient of consolidation of soil,” *Catena*, vol. 173, no. September 2018, pp. 302–311, 2019.
- [25] G. Isabelle, W. Maharani, and I. Asror, “Analysis on Opinion Mining Using Combining Lexicon-Based Method and Multinomial Naïve Bayes,” *2018 Int. Conf. Ind. Enterp. Syst. Eng. (ICoIESE 2018)*, vol. 2, no. IcoIESE 2018, pp. 214–219, 2019.
- [26] K. Bhattacharjee and M. Pant, “Hybrid Particle Swarm Optimization-Genetic Algorithm trained Multi-Layer Perceptron for Classification of Human Glioma from Molecular Brain Neoplasia Data,” *Cogn. Syst. Res.*, vol. 58, pp. 173–194, 2019.
- [27] D. Chong, N. Zhu, W. Luo, and X. Pan, “Human thermal risk prediction in indoor hyperthermal environments based on random forest,” *Sustain. Cities Soc.*, vol. 49, no. April, p. 101595, 2019.
- [28] scikit-learn, “Tuning the hyper-parameters of an estimator,” 2019. [Online]. Available: https://scikit-learn.org/stable/modules/grid_search.html#grid-search. [Accessed: 04-Jul-2019].
- [29] M. Sokolova and G. Lapalme, “A systematic analysis of performance measures for classification tasks,” *Inf. Process. Manag.*, vol. 45, no. 4, pp. 427–437, 2009.
- [30] scikit-learn, “scikit-learn: Machine Learning in Python,” 2019. [Online]. Available: <https://scikit-learn.org/stable/>. [Accessed: 04-Jul-2019].
- [31] R. Susmaga, “Confusion Matrix Visualization,” in *Intelligent Information Processing and Web Mining*, Berlin, Heidelberg: Springer Berlin Heidelberg, 2004, pp. 107–116.
- [32] H. He and E. A. Garcia, “Learning from imbalanced data,” *IEEE Trans. Knowl. Data Eng.*, vol. 21, no. 9, pp. 1263–1284, 2009.
- [33] D. Ruano-Ordás, I. Yevseyeva, V. B. Fernandes, J. R. Méndez, and M. T. M. Emmerich, “Improving the drug discovery process by using multiple classifier systems,” *Expert Syst. Appl.*, vol. 121, pp. 292–303, 2019.
- [34] H. F. Nweke, Y. W. Teh, G. Mujtaba, and M. A. Al-garadi, “Data fusion and multiple classifier systems for human activity detection and health monitoring: Review and open research directions,” *Inf. Fusion*, vol. 46, no. June 2018, pp. 147–170, 2019.

Exploratory Analysis of a New Corpus for Political Alignment Identification of Argentinian Journalists

Viviana Mercado¹, Andrea Villagra¹, and Marcelo Luis Errecalde^{1,2}

¹ Laboratorio de Tecnologías Emergentes - Unidad Académica Caleta Olivia
Universidad Nacional de la Patagonia Austral
{vmercado, avillagra}@uaco.unpa.edu.ar

² Laboratorio de Investigación y Desarrollo en Inteligencia Computacional
Universidad Nacional de San Luis
merrecalde@unsl.edu.ar

Abstract. Political alignment identification is an author profiling task that aims at identifying political bias/orientation in people's writings. As usual in this kind of field, a key aspect is to have available adequate data sets so that the data mining and machine learning approaches can obtain reliable and informative results. This article takes a step in this direction by introducing a new corpus for the study of political alignment in documents of Argentinian journalists. The study also includes several kinds of analysis of documents of pro-government and opposition journalists such as sentiment analysis, topic modelling and the analysis of psycholinguistic indicators obtained from the *Linguistic Inquiry and Word Count* (LIWC) system. From the experimental results, interesting patterns could be observed such as the topics both types of journalists write about, how the sentiment polarities are distributed and how the writings of pro-government and opposition journalists differ in the distinct LIWC categories.

Keywords: Text Mining, Exploratory Data Analysis (EDA), Author Profiling, Journalist Political Alignment, Sentiment Analysis, Topic Modelling, LIWC

1 Introduction

Political alignment identification (PAI) in a text or document is a form of *author profiling* (AP), one of the main tasks of *authorship analysis* (AA) together with authorship attribution/determination, plagiarism detection and style inconsistency detection. PAI, the same as other AP tasks like the detection of depressed people or with different personality traits, paedophiles and suicides is a challenging task within the automatic analysis of texts since it involves, in general, the use of representations of texts that capture stylistic and content aspects of their authors. In this context, a particular area within the PAI is that which is oriented to the study of political orientation in texts written by journalists, and which we will refer to now as *journalistic texts*. We will consider as journalistic texts that information that a journalist publishes in various media such as a personal blog, an article written in a mass media such as a newspaper or the content expressed in a book of his authorship.

The PAI has been applied to texts generated by regular users of social media such as Twitter [1, 2] although more recently it has been done with the documents produced by journalists [3]. In [4], political speech in Twitter has been analyzed with LIWC during the 2008' German electoral campaign. The same tool, LIWC, was used to determine

the psychological state and personality of the candidates for the presidency and vice presidency of the US in the 2004 campaign [5] and the language used by the New York's mayor, R. Giuliani [6] throughout his term. Regarding texts in Spanish language, in [7] the linguistic style of the candidates of the main political parties in the Spanish general elections of 2008 and 2011 is analyzed. On the other hand, the Spanish dictionary of LIWC was applied to analyze the political speech and tweets of the candidates in the elections of Galicia in 2012 [8].

The previous approaches are related to our work but, as far as we know, there are no PAI studies of journalistic texts in Spanish. In this work, we will make a first approach to the PAI in journalistic texts in Spanish, in particular, of texts generated by Argentinian journalists. The task, in this case, will be to group all the documents of "pro-government" journalists on one side and "opponents" on the other one. In that way, it will allow in the future to visualize it as a binary classification ("pro-government" versus "opponent") problem. Our objectives, in the long run, will be to answer the following research questions:

1. What are the appropriate forms of document representation for this task?
2. What are the most effective learning algorithms to use with those representations?
3. What is the impact of the dimensionality reduction approaches in the representations of the documents?
4. How related are the results obtained with similar studies with journalistic texts written in other languages?

In the present article, a first step is taken to achieve these objectives by introducing a new corpus for the study of political alignment in documents of Argentinian journalists. The study also includes several kinds of analysis of documents of pro-government and opposition journalists such as sentiment analysis, topic modelling and the analysis of psycholinguistic indicators obtained from the LIWC system.

The rest of the article is organized as follows: Section 2 describes the PAI corpus introduced in the present article with statistics and metrics of the whole data set and of each of both involved classes; Section 3 gives some results obtained from a topic modelling and sentiment analysis; Section 4 goes further in analyzing the PAI corpus taking into account psycholinguistic indicators obtained from the LIWC system. Section 5 finishes this article by giving the main conclusions obtained from our study and some future work.

2 Corpus Description

Our work was focussed on generating a collection of Argentinian journalistic documents obtained from news blogs, online newspapers, books, etc. It consists of 196 documents belonging to 10 journalists: 5 of them that clearly support the actions of the Argentine government in the period 2012 to 2015 and 5 of them that explicitly express themselves against the government in that period. The data set was split into two groups of documents according to the political orientation of the journalists. Thus, 98 documents belonging to the 5 pro-government journalists were selected for the *gov* (pro-government) class and the 98 remaining documents of the opposition journalists were used to build the *oppo* (opposition) class. In that way, a balanced corpus with 2 classes was obtained.

To select the documents some guidelines were taken into account:

- Texts correspond to Spanish documents written by Argentinian journalists.

- Texts refer to different political aspects related to the Argentinian government in the period 2012-2015, such as government actions, politicians’ declarations, corruption cases, treatment of laws, etc.
- All the documents contain “ formal text ”, that is to say, they do not present common “ informal ” aspects of content from social media such as abbreviations, slang expressions, typos, hyperlinks, labels, figures, and emoticons.
- From each journalist, between 18 and 20 documents were taken from his/her personal blog, articles in online newspapers or digital books of his/her authorship.
- Each journalist was clearly identified as pro-government or opponent.
- The same proportion of male and female journalists were kept between both categories.

After collecting the documents, they were manually labeled as belonging to the two above-mentioned classes *gov*, and *oppo*. Table 1 shows information about how the documents were finally distributed in both classes and what were the sources (online newspaper, blog or digital books) from they were obtained.

Class	Newspapers	Blogs	Books	Total
<i>gov</i>	50	46	2	98
<i>oppo</i>	60	36	2	98

Table 1: Distribution of documents in classes and source.

2.1 Corpus statistics

Before proceeding with a more elaborated analysis of the corpus, some basic statistics were obtained to get some insights about the general characteristics of the documents. First of all, the *number of words* per document was analyzed for each document/article of both classes. Table 2 shows the minimum, maximum, mean and standard deviation values for the number of words in the documents of the *gov*, *oppo* classes and the whole corpus (*gov + oppo*).

Class	Minimum	Maximum	Mean	St. Dev.
<i>gov</i>	139	36619	1865.18	5031.56
<i>oppo</i>	236	3423	1243.01	733.71
<i>gov + oppo</i>	139	36619	1554.09	3608.91

Table 2: Number of words in documents: minimum, maximum, mean and standard deviation values per class.

As we can see, although the sizes of the shortest documents (Min) are similar for both classes (139 vs 236), they differ considerably in the longest ones (36619 vs 3423). That can be observed more clearly in Figures 1 and 2 that show the number of words (at the left) and the histogram (at the right) for the documents in the *gov* and the *oppo* classes. There, for instance, Figure 1a shows that there are some documents in the class

gov whose sizes exceeds 34000 words. However, as the Figure 1b confirms, most of the documents in the *gov* class do not exceed 5000 words with only a couple of documents (that correspond to books) whose sizes are between 34000 and 36000 words. Figure 2 shows that the *oppo* class has a “smoother” distribution in the number of words of its documents, with sizes that oscillate between 200 and 3500 words approximately.

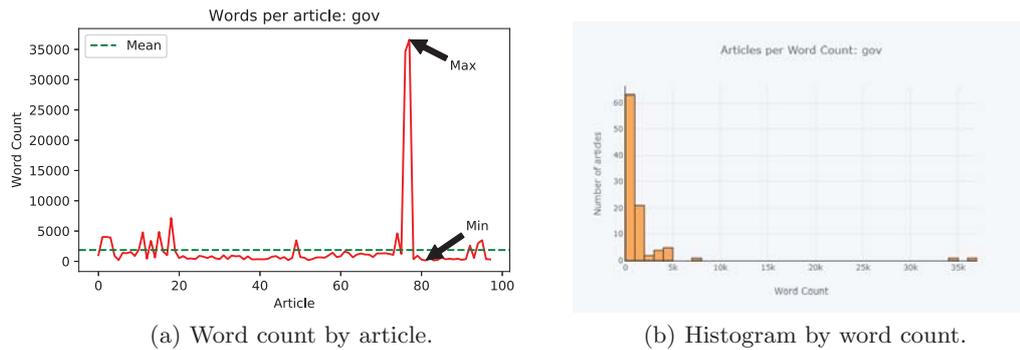


Fig. 1: Class *gov*: number of words per article

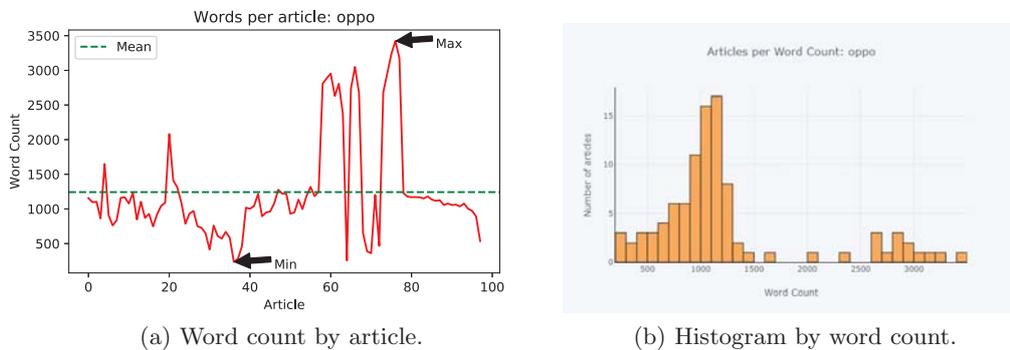


Fig. 2: Class *oppo*: number of words per article

Regarding the total number of words and the size of the vocabulary of the whole corpus and of each class, we will first introduce the following notation: let $D_{\mathcal{C}}$ be the set of documents belonging to our corpus \mathcal{C} ; let $D_{\mathcal{G}}$ and $D_{\mathcal{O}}$ be the sets of documents belonging to the pro-government and opposition journalists respectively, $D_{\mathcal{G}} \subset D_{\mathcal{C}}$, $D_{\mathcal{O}} \subset D_{\mathcal{C}}$, $D_{\mathcal{C}} = D_{\mathcal{G}} \cup D_{\mathcal{O}}$, $D_{\mathcal{G}} \cap D_{\mathcal{O}} = \emptyset$. Let $|D_{\mathcal{C}}|$ be the total number of words in our corpus \mathcal{C} , with similar meanings for $|D_{\mathcal{G}}|$ and $|D_{\mathcal{O}}|$. Besides, let $\mathcal{V}_{\mathcal{C}}$, $\mathcal{V}_{\mathcal{G}}$, and $\mathcal{V}_{\mathcal{O}}$ be the *vocabularies*³ of $D_{\mathcal{C}}$, $D_{\mathcal{G}}$, and $D_{\mathcal{O}}$, respectively. Table 3 gives some statistics related to these collection of documents.

³ The vocabulary of a collection of documents is the set of *distinct* words that appear in that collection.

$ D_C $	$ \mathcal{V}_C $	$ D_G $	$ \mathcal{V}_G $	$ \mathcal{V}_G / D_G $	$ D_O $	$ \mathcal{V}_O $	$ \mathcal{V}_O / D_O $	$ \mathcal{V}_G \cap \mathcal{V}_O $	$ \mathcal{V}_G \setminus \mathcal{V}_O $	$ \mathcal{V}_O \setminus \mathcal{V}_G $
280343	24323	167844	18497	0.11	112499	13146	0.11	7320	11177	5826

Table 3: Statistics on the documents of the whole corpus (D_C), documents of pro-government (D_G) and opposition (D_O) journalists.

One of the first things we can observe from Table 3 is that although when the number of words in the whole corpus is high ($|D_C| = 280343$), the number of *distinct* words (the size of the *vocabulary*) is relatively small ($|\mathcal{V}_C| = 24323$). That differs from the size of vocabularies in texts from social media which usually are bigger. A possible cause of this is that writings in social media are usually informal and prone to have abbreviations and typos, increasing in that way the number of distinct words. Another interesting datum from Table 3 is that the vocabulary of pro-government journalists is considerably bigger than the one of opposition journalists ($|\mathcal{V}_G| = 18497$ $|\mathcal{V}_O| = 13146$). One cause of this might be that due to the greater number of words in the documents of pro-government journalists ($|D_G| > |D_O|$) we will probably have a greater number of distinct words. For this reason, as vocabulary richness estimation it is frequently used the ratio between the size of vocabulary and the number of words in the collection. In that case, we can see that those metrics for pro-government ($|\mathcal{V}_G|/|D_G|$) and opposition ($|\mathcal{V}_O|/|D_O|$) journalists are the same.

Finally, it is worth to note that the intersection between pro-government and opposition vocabularies ($|\mathcal{V}_G \cap \mathcal{V}_O|$) is very small compared to the vocabulary of the whole corpus ($|\mathcal{V}_C|$). That means that many words are used by a group and not by the other one, and the other way around. This point can be easily observed in Table 3 where the number of words used by pro-government and not by opposition journalists ($|\mathcal{V}_G \setminus \mathcal{V}_O|$) is 11177 and $|\mathcal{V}_O \setminus \mathcal{V}_G| = 5826$. For instance, some words of $\mathcal{V}_G \setminus \mathcal{V}_O$ are “milicos”, “globitos”, “latinoamericana”, “egoísmo”, and “ultraderecha” and some words of $\mathcal{V}_O \setminus \mathcal{V}_G$ are “monárquica”, “tribunera”, “dilapidado”, “negociaron”, “hitlerismo”, and “avaricia”.

Another analysis that is usually informative is to measure the “relevance” of the terms in the corpus according to some specific metric. For instance, an approach is estimating the importance of a term according to the weight that it would receive in a particular document representation scheme, such as *tf-idf*. This scheme (*tf-idf*) is a weighted model commonly used for information retrieval problems. It is an unsupervised model in the sense that when weighting a term in a document, it does not take into account any information about the class that document belong to; for instance, in our corpus, if we take as terms the word uni-grams, the terms with the highest *tf-idf* value are: “colegio”, “años”, “comisión”, “madre”, “dijo”, “perón”, “día”, “plata”, “dos”, “decía”, “chica”, “mujeres”, “dice”, “mamá”, “después”, “flaco”, “casa”, “alicia”, “néstor”, and “cristina”. Taking as terms words 2-grams, the terms with the highest *tf-idf* value are: “próximo gobierno”, “cinco años”, “santa fe”, “derechos humanos”, “clase media”, “muerte néstor”, “años después”, “día siguiente”, “néstor kirchner”, “muchas veces”, “provincia buenos”, “primera vez”, “cristina dijo”, “procurador general”, “gils carbó”, “cristina fernández”, “buenos aires”, “santa cruz”, “néstor cristina” and, “río gallegos”. Finally, “magdalena ruiz guiñazú”, “economía axel kicillof”, “ministro economía axel”, “asignación universal hijo”, “josé pablo feinmann”, “joaquín morales solá”, “gobernador provincia buenos”, “da mucha bronca”, “triple crimen general”, “derechos humanos cidh”, “cristina fernández kirchner”, “mil millones dólares”, “manuel abal medina”, “juan manuel abal”, “madres plaza mayo”, “comisión interamericana derechos”, “aumento mínimo imponible”, “ciudad buenos aires”, “interamericana derechos humanos”,

and “provincia buenos aires” are the terms with the highest *tf-idf* values when word 3-grams are used as terms.

A second approach to measure the importance of terms, is considering *supervised* metrics that capture the importance of each term concerning its class/category, such as χ^2 and *information gain*, among others. They are usually used in *feature selection* processes to determine what are the most informative features to be preserved in the document representation. Here, we will calculate χ^2 scores for all the terms consisting in word 2-grams and the top 20 are shown in Figure 3.

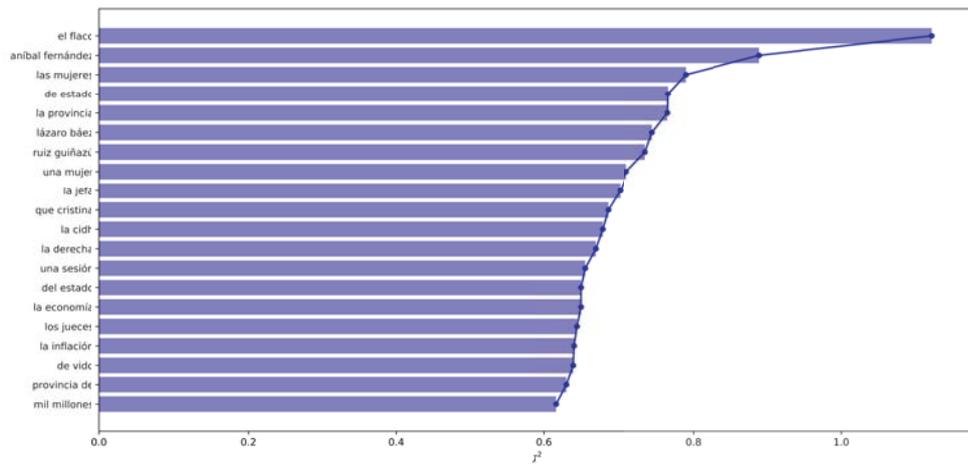


Fig. 3: Top 20 word 2-grams according to χ^2 scores.

A third alternative that usually shows interesting information about the importance of features, is to obtain the *coefficients* of a model learned in a training stage of a prediction task with the whole corpus. In that case, those coefficients reflect how the learned model *weights each feature* for that task. In that way, we can look at the largest coefficients, and see which words these correspond to. For instance, Figure 4 shows a bar chart with the 25 largest and 25 smallest coefficients of a logistic regression model, with the bars showing the size of each coefficient. Word 2-grams are used as terms and the negative coefficients on the left belong to terms that according to the model are indicative of pro-government alignment, while the positive coefficients on the right belong to terms that according to the model indicate an article written by an opposition journalist. Most of the terms are quite intuitive, in the sense that reflect aspects very recognizable in texts from both political alignment, such as “grandes medios”, “medios dominantes”, and “nacional popular” indicating pro-government journalists, while “ruta dinero”, “lázaro baez” and, “lavado dinero” indicate opposition documents.

3 Topic Modelling and Sentiment Analysis

Topic modeling is an umbrella term describing a class of text analysis methods whose task is assigning each document to one or multiple *topics*, usually without supervision. A good example of this is news data, which might be categorized into topics like “politics”, “sports”, “finance”, and so on. Intuitively, a topic is a group of words that appear together frequently. In that context, “topics” obtained by a topic modeling process might

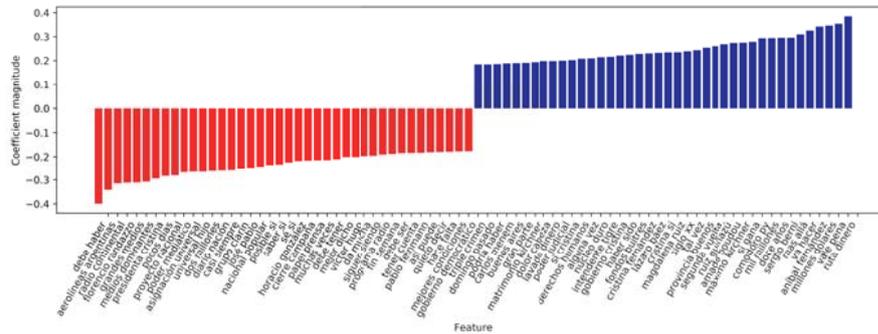


Fig. 4: Largest and smallest coefficients of logistic regression trained on tf-idf features.

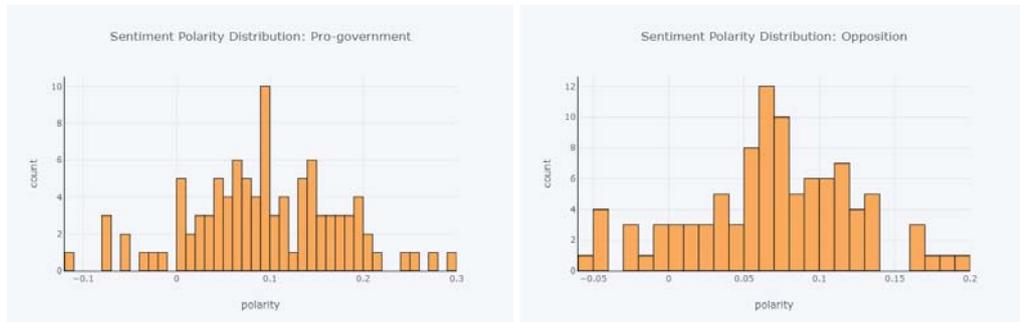
not be what we would normally call a topic in everyday speech. In other words, the obtained groups (topics) might or might not have a semantic meaning clearly identifiable by a person. Often, when people talk about topic modeling, they refer to one particular decomposition method called *Latent Dirichlet Allocation* (LDA). LDA not only tries to find groups of words (the topics) that appear together frequently. It also requires that each document can be understood as a “mixture” of a subset of the topics.

As an example, applying LDA to the pro-government and opposition documents and setting the number of topics to 100, several topics with intuitive meaning are obtained. Table 4 shows the first 20 words of some of those topics, three of the pro-government documents on the left column and three of the opposition documents on the right one. There, it can be observed that pro-government topics have to do with women’s rights (*Topic #9*), Argentine debt with vulture funds (*Topic #27*) and social security plans (*Topic #49*), while opposition topics are related to communication media and journalists (*Topic #46*), some events related to what was popularly known as “the cause of the ephedrine” (*Topic #89*), and the relationship between the official Argentine cult and the Pope and some politicians (*Topic #91*).

Pro-government topics	Opposition topics
<i>Topic #9:</i> voto mujeres décadas femenino incluso ciclo siglo luchas derecho quiera derrota banderas fitzgerald capital feministas siglos diez políticamente evita consigue	<i>Topic #46:</i> radio mitre canal censura lanata tn intento sabemos intervenir oyentes diario adecuación marcelo pánico convertirse puesta usureros colegas clarín clarín
<i>Topic #27:</i> fondos deuda buitres final tema ministro documento kicillof soberana necesidad procesos comunicado litigiosidad previsibilidad reestructuración anexo conformes soberanas deudas líderes	<i>Topic #89:</i> aníbal desmentida triple kilos crimen efedrina quilmes melnyk clara importación agosto negocio granero morsa perez nombres indispensable junto lanatta quizá
<i>Topic #49:</i> auh asignación pobreza fondos plan implicó reparación ése cfk octubre previsionales diputados pasaba narváez proyectos dirigencia impulso región decreto corporaciones	<i>Topic #91:</i> papa francisco iglesia ayuda mirada hombres página uca bergoglio cuervo vaticano guillermo michetti sienten larroque carrío explican quedaron opositores alegría

Table 4: Some topics of pro-government and opposition journalists.

Another usual analysis of the texts in a corpus is determining the *polarity* of each document by averaging the polarity of its component words. A popular tool for this task is **TextBlob** that calculates sentiment polarity in the range of $[-1, 1]$ where 1 means *positive* sentiment and -1 means *negative* sentiment. In that way, it is usual to show some articles with the highest/lowest or even close to neutral (zero) sentiment polarity score or give some distribution of the articles according to their polarity scores. Due to space restrictions, we will only analyze the last alternative presenting in Figure 5 the sentiment polarity distributions of both pro-government (Figure 5a) and opposition (Figure 5b) journalists. There, it can be observed a greater frequency of pro-government articles on higher scores (around 0.1) than the one shown by the opposition journalists (around 0.05). Besides, the highest positive score achieved by opposition journalists (0.2) is surpassed by several pro-government articles. The other way around, the global lowest (negative) score is also obtained by pro-government journalists (less than -0.1) indicating that pro-government articles show the greatest variation range in polarity scores.



(a) Histogram of pro-government articles.

(b) Histogram of opposition articles.

Fig. 5: Sentiment Polarity Distribution in pro-government and opposition journalists.

4 LIWC-based Analysis

LIWC is a tool developed by the American psychologist J. Pennebaker and colleagues [9] and have been used in several studies related to psychological aspects of individuals. LIWC calculates the proportions of certain grammatical, lexical, and semantic markers, as well as markers belonging to other categories (up to 90 text features depending on the version). In our study, we used the most recent version of LIWC, LIWC2015 [10]. For each text file, LIWC2015 generates approximately 90 output variables as one line of data to an output file. This data record includes the file name and word count, 4 summary language variables (analytical thinking, clout, authenticity, and emotional tone), 3 general descriptor categories (words per sentence, percent of target words captured by the dictionary, and percent of words in the text that are longer than six letters), 21 standard linguistic dimensions (e.g., percentage of words in the text that are pronouns, articles, auxiliary verbs, etc.), 41 word categories tapping psychological constructs (e.g., affect, cognition, biological processes, drives), 6 personal concern categories (e.g., work, home, leisure activities), 5 informal language markers (assents, fillers, swear words, netspeak), and 12 punctuation categories (periods, commas, etc.).

Properties of documents generated by LIWC2015 have been used as document representations in several studies and also to analyze how these measures differ between articles of different classes. This last approach will be the one used in the present article. We will first identify what are the features/characteristics in which there are statistical differences between both classes and then we will show more detailed information on some of them. Since the distribution of the feature values is not known and we cannot make any assumption about it, we used, the same as similar works with LIWC features [11], the (non-parametric) Wilcoxon signed-rank test for comparing paired data samples with a p-value < 0.05 for statistical significance. The null hypothesis (H_0) that we are trying to refute is that there is no statistically significant relationship between the mean value of a feature belonging to the pro-government class and the mean value of the same feature belonging to the opposition class. In that context, we determined significant statistical differences in 34 LIWC categories. For instance, pro-government journalists show a greater use of *verbs*, *adverbs*, *first person singular* (“yo”, “mi”, “mio”), *social processes* (“compañero”, “hablar”, “ellos”) and, *perceptual processes* (“mirar”, “escuchar”, “sentir”). Opposition journalists, on the other hand, make a higher use of words with a *length* > 6 letters and make more references to expressions related to *money* (“dinero”, “efectivo”, “adeudar”). As an example, Figure 6 shows comparative boxplots of pro-government and opposition journalists for 2 LIWC categories with statistically significant differences: Perceptual processes and Money.

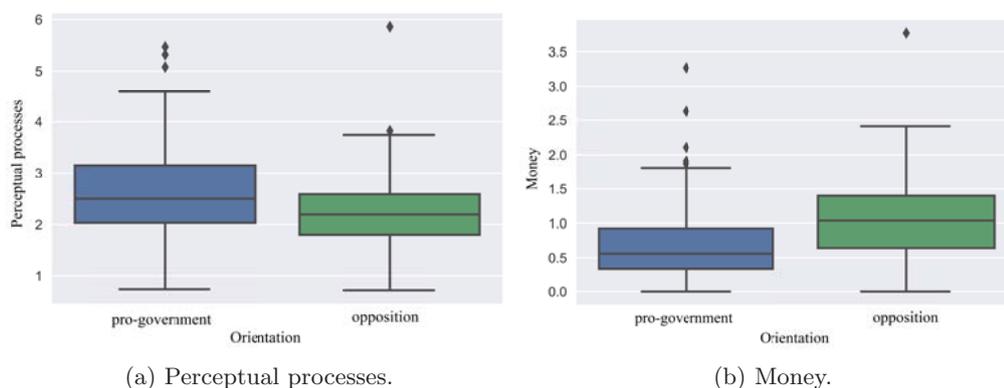


Fig. 6: Comparative box-plots for Perceptual processes and Money categories.

5 Conclusions and Future Work

This article introduces a new corpus for political alignment identification of Argentinian Journalists. In that context, a comprehensive analysis of that corpus has been carried out which included the study of the corpus statistics, topic modelling and sentiment analysis, and a comparison of texts based on LIWC categories. As a result of this analysis, some interesting patterns were identified that reveal evident differences between the writings of pro-government and opposition journalists. For instance, they differ in how their sentiment polarities are distributed, what are the topics they talk about and, in their values for many categories of the LIWC system.

As future work, we plan to use the different types of information obtained in the present work in the representation of documents for supervised (classification) and, non-supervised (clustering) tasks. Thus, the idea is using LIWC-based and LDA/topics-based features in text classification tasks, and comparing them against classical (bag of words) and more recent approaches like deep neural-networks with word embeddings.

Finally, we will reorganize the documents of the corpus analyzed in the present work according to the *authors* of these documents. In that way, we will have ten different classes (one for each journalist) and the task will be addressed as an *authorship attribution* task. An interesting point, in this case, will be determine how the hardness of this task is incremented when the authorship attribution is constrained to journalists of the same political orientation.

Acknowledgments. The authors acknowledge the support of Universidad Nacional de la Patagonia Austral. The last author also acknowledges the constant support afforded by the Universidad Nacional de San Luis.

References

1. Raviv Cohen and Derek Ruths. Classifying political orientation on twitter: It's not easy! In *Proceedings of the Seventh International AAAI Conference on Weblogs and Social Media*, 2013.
2. Michael D. Conover, Bruno Gonçalves, Jacob Ratkiewicz, Alessandro Flammini, and Filippo Menczer. Predicting the political alignment of twitter users. In *Proceedings of 3rd IEEE Conference on Social Computing (SocialCom)*, pages 192–199, 10 2011.
3. Konstantina Lazaridou and Ralf Krestel. Identifying political bias in news articles. *Bulletin of the IEEE TCCL*, 12, 2016.
4. Andranik Tumasjan, Timm O. Spenger, Philipp G. Sandner, and Isabelle M. Welp. Predicting elections with twitter: What 140 characters reveal about political sentiment. *Proceedings of the Fourth International AAAI conference on Weblogs and Social Media (ICWSM)*, Washington, S. 178-185., (2010).
5. Richard. B. Slatcher, Cindy K. Chung, James W. Pennebaker, and Lori D. Stone. Winning words: individual differences in linguistic style among u. s. presidential and vice presidential candidates. *Journal of Research in Personality*, vol. 41, pp.63-75., (2007).
6. James W. Pennebaker and Thomas C. Lay. Language use and personality during crises: analyses of mayor rudolph giuliani's press conferences. *Journal of Research in Personality*, vol. 36, pp.271-282., (2002).
7. María Jesús Carrera-Fernández, Joan; Guàrdia-Olmos, and Maribel Peró-Cebollero. Linguistic style in the mexican electoral process: Language style matching analysis. *Revista Mexicana de Psicología*, 31(2), 138-152., (2014).
8. Mercedes Fernández-Cabana, José Rúas-Araújo, and Maria Teresa Alves-Pérez. Psicología, lenguaje y comunicación: análisis con la herramienta liwc de los discursos y tweets de los candidatos a las elecciones gallegas de 2012. *Anuario de Psicología*, 44(2), pp.169-184., (2014).
9. James W.a Pennebaker, Roger J. Booth, and Marta E. Francis. Linguistic inquiry and word count (liwc). *[Software]*., (2001).
10. James W. Pennebaker, Ryan L. Boyd, Kayla Jordan, and Kate Blackburn. *The development and psychometric properties of LIWC2015*. University of Texas at Austin, Austin, TX, 2015.
11. José Rúas, Mercedes Fernández, and Iván Puentes. Aplicación de la herramienta liwc al análisis del discurso político. los mítines de los candidatos en las elecciones al parlamento de galicia de 2012. In *Actas del 2do Congreso Nacional sobre Metodología de la Investigación en Comunicación*, pages 47–64, 2013.

Extensión de Proceso de Explotación de Información para la Detección de Asociaciones Espaciales Difusas

Giovanni Daián Rottoli^{1,2,3}, Hernán Merlino³

¹ Programa de Doctorado en Ciencias Informáticas, Facultad de Informática, Universidad Nacional de La Plata (UNLP). La Plata, Buenos Aires, Argentina.

² Programa de Beca Doctoral para el Fortalecimiento de Áreas I+D+i. Universidad Tecnológica Nacional (UTN). Argentina
rottolig@frcu.utn.edu.ar

³ Grupo de Investigación en Sistemas de Información (GISI), Universidad Nacional de Lanús (UNLa). Lanús, Buenos Aires, Argentina.
hmerlino@gmail.com

Resumen Las asociaciones espaciales brindan conocimiento que resulta de sumo interés para la inteligencia de negocios en dominios varios. Estas asociaciones están constituidas tanto de relaciones espaciales como no espaciales que se dan entre los objetos de una base de datos, las cuales invitan a ser modeladas haciendo uso de la teoría de conjuntos difusos debido a la falta general de límites claros para determinar cuándo un par de objetos pertenece o no a la relación en cuestión, producto de la misma naturaleza de los datos. Un ejemplo es la relación de cercanía entre dos objetos: ¿a qué distancia los mismos dejan de estar cerca entre sí? En consecuencia, en el presente trabajo se extiende un proceso de explotación de información para la detección de asociaciones espaciales basado en la minería de subgrafos frecuentes considerando además relaciones difusas. Se presenta una prueba de concepto utilizando datos generados para tales fines.

Keywords: Asociaciones Espaciales, Spatial Data Mining, Explotación de Información Espacial, Lógica Difusa, Subgrafos Frecuentes

1. Introducción

La explotación de información espacial se encarga de la búsqueda de conocimiento interesante y novedoso en bases de datos espaciales para dar soporte a la toma de decisiones estratégicas [1]. En este contexto, los patrones de asociación resultan de interés para los tomadores de decisiones [2].

Un dato espacial $D = \{\{s_j\}, \{v_k\}\}$ se compone por un conjunto de n atributos espaciales $\{s_j\}, j = 1..n$, que relacionan cada objeto espacial inequívocamente con una ubicación determinada del espacio, y un conjunto de m atributos no espaciales $\{v_j\}, k = 1..m$, que recogen datos relevantes para el dominio del problema en la ubicación detallada [3].

La dependencia espacial de la información admite extraer relaciones espaciales que se encuentran explícitas, las cuales a su vez pueden ser: 1. relaciones geométricas, si se basan en la distancia euclidiana (vecindad, por ejemplo); 2. relaciones topológicas, independientes de transformaciones como el escalado, la rotación y traslación (e.g., intersección, inclusión); 3. relaciones direccionales, relacionadas con la posición de un objeto con respecto a otro (norte, sur, arriba, abajo); 4. relaciones mixtas, si intervienen combinaciones de relaciones anteriormente mencionadas (perpendicularidad, paralelismo) [4–6].

Estas relaciones se asocian con conceptos cuya naturaleza no es posible de ser definida mediante umbrales estrictos, trasladándose esta incertidumbre a los patrones de asociación espaciales en los que las mismas intervienen. Frente a este escenario, la lógica difusa resulta una herramienta adecuada para modelarlas [8, 9].

En trabajos anteriores [7], se ha propuesto un proceso de explotación de información para el minado de asociaciones espaciales que pueden incluir relaciones y datos de distintos tipos y considera fenómenos como la heterogeneidad espacial.

El proceso de explotación de información mencionado deja abierta la posibilidad de utilizar relaciones difusas entre los datos, cuestión que se aborda en el presente trabajo. Para esto, en la sección 2 se ahonda en el concepto de relaciones espaciales (sección 2.1) y de asociaciones espaciales (sección 2.2). Luego, en la sección 3 se propone la adaptación del proceso en cuestión para contemplar tanto relaciones *CRISP* como difusas. Posteriormente, en la sección 4 se realizan pruebas de concepto del proceso propuesto utilizando datos generados para dicho propósito. Finalmente, en la sección 5 se presentan conclusiones y trabajos futuros.

2. Estado del Arte

2.1. Relaciones Espaciales

Sea D una base de datos espaciales, una relación $R \subseteq D^2$ es llamada *espacial* si y solo si está definida mediante un predicado binario $P(x, y) | x, y \in D$ que involucra a los atributos espaciales de los objetos evaluados [6].

La dependencia espacial de cada objeto de la base de datos es la que define implícitamente las distintas relaciones que se pueden extraer entre los mismos. A modo de ejemplo, la relación espacial $V \subseteq D^2$, con $x, y \in D$ y $\lambda \in R^+$, definida como:

$$xVy \leftrightarrow \sqrt{(lat_x - lat_y)^2 + (long_x - long_y)^2} < \lambda \quad (1)$$

es la relación de vecindad entre dos puntos en el espacio utilizando la distancia euclidiana entre sus atributos latitud y longitud.

El cómputo de estas relaciones implica una operación *join* entre dos subconjuntos de datos haciendo que su cálculo sea, por lo general, costoso, dependiendo de la cardinalidad de los mismos.

Las relaciones espaciales pueden clasificarse según su naturaleza en *Geométricas*, *Direccionales*, *Topológicas* e *Híbridas* [10, 11].

Tabla 1: Ejemplo de funciones de distancia.

Nombre	Fórmula
Distancia Euclidiana	$d_E(\vec{P}, \vec{Q}) = (\sum_{i=1}^n (p_i - q_i)^2)^{\frac{1}{2}}$
Distancia de Chebyshev	$d_C(\vec{P}, \vec{Q}) = \max_{i=1}^n (p_i - q_i)$
Distancia Manhattan	$d_M(\vec{P}, \vec{Q}) = \sum_{i=1}^n p_i - q_i $
Distancia Geográfica	$d_G(P, Q) = R((\Delta\phi)^2 + (\cos(\phi_m)\Delta\lambda)^2)^{\frac{1}{2}}$

Las relaciones geométricas están basadas en la geometría euclidiana y suelen ser el tipo de relaciones más utilizadas entre datos espaciales. Se requiere, por lo usual, el cálculo de distancias entre los objetos de la bases de datos (Tabla 1) [12].

Las relaciones direccionales involucran información sobre la orientación relativa de un objeto en un espacio 2D o 3D. Generalmente son definidas en función del concepto de “vecindario”, adoptando para tal propósito las ya mencionadas distancia Manhattan y la de Chebyshev, obteniéndose respectivamente cuatro vecinos (norte, sur, este y oeste) u ocho vecinos (norte, noreste, este, sudeste, sur, sudoeste, oeste, noroeste).

Las relaciones topológicas son independientes de los conceptos de distancia y dirección y no se ven afectadas por transformaciones lineales como traslación, escalado y rotación. Es utilizado frecuentemente el modelo de 9 intersecciones propuesto por [13] para modelar las distintas interacciones topológicas entre dos instancias de cualquier tipo de dato espacial vectorial: líneas, puntos o polígonos. Las mismas poseen diferentes configuraciones que varían entre los distintos tipos de objetos espaciales. Por ejemplo, entre dos polígonos existen ocho posibles relaciones: *disjuntos*, *contiene_a*, *dentro_de*, *iguales*, *conoce_a*, *ubre*, *cubierto_por*, *superpuestos*.

Las relaciones híbridas surgen a partir de la combinación de relaciones clasificadas en dos o más categorías de relaciones diferentes. Por ejemplo, las relaciones de paralelismo y perpendicularidad entre dos líneas son tanto geométricas, debido a que se basan en el ángulo de incidencia de ambos elementos, como topológicas, ya que la propiedad se mantiene invariante frente a transformaciones de traslación, rotación y escalado.

Esta variedad de funciones de tipos de relaciones espaciales y el costo computacional que requiere su cálculo, hacen de la preparación de los datos una tarea considerablemente más compleja que en el explotación de información tradicional.

2.2. Asociaciones Espaciales

Un patrón de asociación se define como un predicado compuesto con alta probabilidad de ocurrencia. Este patrón se dice espacial si al menos uno de los predicados simples que lo componen define una relación espacial [2]. Un ejemplo de una asociación espacial es:

$$Es(x, 'Escuela') \wedge Es(y, 'ZonaResidencial') \wedge Contiene(y, x)$$

La relevancia de un patrón es medida utilizando distintas métricas. El soporte (*support*) es una de las más utilizadas, calculándose sobre una base de datos transaccional como la probabilidad de ocurrencia conjunta de los predicados (o valores de sus atributos). En una base de datos espaciales esta medida es equivalente.

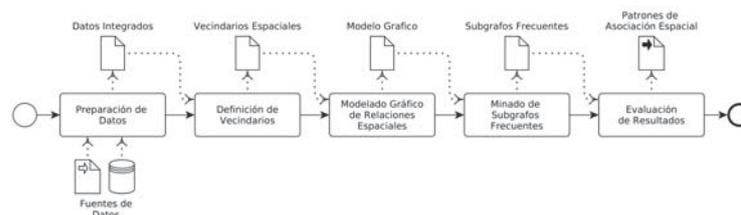


Figura 1: Proceso de Explotación de Información para el descubrimiento de Asociaciones Espaciales [7].

Asimismo, al igual que en la búsqueda de asociaciones tradicional, es posible generar reglas de asociación a partir del cálculo de la confianza de las posibles configuraciones de antecedente y consecuente sobre los patrones de asociación descubiertos [15], la cual se calcula utilizando la probabilidad condicional: $Es(x, \text{'Escuela'}) \wedge Contiene(y, x) \rightarrow Es(y, \text{'ZonaResidencial'})$

3. Extensión del Proceso de Explotación de Información

Previamente, en [7], se ha propuesto un proceso de explotación de información para el descubrimiento de asociaciones espaciales (Fig. 1) que tiene en consideración el fenómeno de la heterogeneidad espacial [16] y la utilización de datos de entrada de naturaleza y tipos varios.

El proceso en cuestión modela las relaciones espaciales mediante una estructura de multigrafo $G = (V, E, L, K, \phi, P_l, P_k)$. Los conjuntos que conforman dicha estructura son V , conjunto de vértices que modelan los objetos espaciales independientemente de su forma; E , conjunto de aristas que modelan las relaciones entre los objetos espaciales; L y K , conjuntos de etiquetas para los vértices y las aristas respectivamente que modelan las características de los mismos. Adicionalmente, la función $\phi : E \rightarrow \{\{v_1, v_2\} | v_1 \in V \wedge v_2 \in V \wedge v_1 \neq v_2\}$ es la función de incidencia de las relaciones espaciales sobre los objetos espaciales, $P_l \subseteq V \times L$ es la relación de etiquetado de características de los objetos espaciales, y $P_k \subseteq E \times K$ es la relación de etiquetado de características de las relaciones espaciales.

A partir de esta herramienta de modelado, se propone la búsqueda de subgrafos frecuentes en el grafo de relaciones original. Esta actividad se lleva a cabo mediante el uso de algoritmos especializados en esta tarea [17, 18]. Cada uno de estos subgrafos representan aquellas relaciones espaciales que se dan frecuentemente de forma conjunta.

El artículo previamente mencionado propone la posibilidad de utilización de relaciones difusas, mas no especifica el procedimiento para hacerlo. En consecuencia, a continuación se propone la representación de relaciones difusas mediante una estructura de multigrafo equivalente, el minado de los subgrafos frecuentes, y la obtención de un valor representativo de los grados de pertenencia de las instancias involucradas en los patrones obtenidos.

3.1. Modelado Gráfico de Relaciones Espaciales Difusas

Los umbrales rígidos para la definición de relaciones entre datos espacialmente referenciados no permiten modelar la realidad de manera apropiada. Póngase como

ejemplo la relación de vecindad: si el criterio tomado establece una distancia máxima de 50 metros entre dos objetos para que estos sean vecinos, ¿es justo excluir un par de datos por encontrarse separados por 51 metros?. Asimismo, ¿Cuál es el límite estricto de una zona de bosque? .

Generalmente hacemos uso de etiquetas lingüísticas como “cerca” o “lejos” para referimos a las relaciones espaciales. Estos conceptos no poseen propiamente límites bien establecidos, por lo que son factibles de ser modelada utilizando herramientas como la teoría de conjuntos difusos (*fuzzy sets*) [14].

Un conjunto difuso se define mediante una función de pertenencia $\mu_{\tilde{A}}(x) : U \rightarrow [0, 1]$, que representa el grado en el cual $x \in \tilde{A}$, siendo x un objeto del universal considerado. Consecuentemente, una relación difusa \tilde{R} es un subconjunto difuso del producto cartesiano clásico $A_1 \times A_2 \times \dots \times A_n$ [19].

A modo de ejemplo, dada la relación de vecindad previamente definida de manera rígida (Ec. 1), es posible extenderla de forma difusa mediante una función de pertenencia como la presentada en la ecuación 2. Consecuentemente, dos objetos espaciales que se encuentren a una distancia menor a λ_1 pertenecerán a la relación con $\mu = 1$; si se encuentran a una distancia comprendida entre λ_1 y λ_2 serán vecinos con $0 < \mu < 1$; y si se alejan a mayor no estarán relacionados.

La definición de las funciones de pertenencia dependerán del dominio del problema abordado en cada caso.

$$\forall x, y \in S, \mu_{\text{Vecinos}}((x, y)) = \begin{cases} 1 & \text{si } \delta \leq \lambda_1 \\ (-\frac{1}{25}\delta + 3)^{\frac{1}{2}} & \text{si } \lambda_1 < \delta < \lambda_2 \\ 0 & \text{si } \delta \geq \lambda_2 \end{cases} \quad (2)$$

siendo $\delta = \text{Distancia}(x, y)$

Para la construcción de las aristas del grafo, entonces, primero deben ser extraídos los grados de pertenencia de las tuplas de objetos espaciales a las relaciones difusas evaluadas. Luego, se modelará una arista por cada tupla perteneciente al soporte de cada relación difusa, siendo éste el conjunto rígido $S(\tilde{A})$ que contiene todos los elementos del universal U cuyo grado de pertenencia al conjunto difuso \tilde{A} sea no nulo. En símbolos: $S(\tilde{A}) = \{x/x \in U \wedge \mu_{\tilde{A}} > 0\}$ [20].

De esta forma, el conjunto de aristas se conforma a partir de las relaciones cuyas tuplas pertenezcan en algún grado a la relación, sin importar el valor de este. Posteriormente se evaluarán los grados de pertenencia de las instancias de los patrones de asociación detectados.

3.2. Minado de Subgrafos Frecuentes

A partir del grafo generado, tal como se especifica en el proceso presentado en [7], se buscan aquellos subgrafos que se repiten frecuentemente en el grafo original haciendo uso de algoritmos de búsqueda de subgrafos frecuentes en un único gran grafo [17, 18]. Los autores proponen el uso del algoritmo SUBDUE ya que hace uso de una medida de prevalencia de los subgrafos basada en la compresión de la información [21].

Posteriormente, por cada uno de los subgrafos frecuentes, se calculan los grados de pertenencia de las instancias de las relaciones que intervienen en el mismo, obteniendo la mínima pertenencia conjunta a las mismas. Para esto haremos uso de una T-norma sobre los grados de pertenencia de cada objeto espacial interviniente en cada una de las instancias del patrón, a los predicados en los que participa [15]. Por último calcularemos

mediante una regla heurística el grado de pertenencia conjunto de las instancias, por ejemplo, mediante el promedio de los mismos.

Si bien la T-norma Mín es la más habitual, esto no reflejaría correctamente la relación existente entre los objetos que conforman la instancia de la regla. Por ejemplo, sea un predicado compuesto $Vecinos(x, y) \wedge Norte(x, y)$ con grados de pertenencia $\mu_1 = (0,5; 1)$ y $\mu_2 = (0,5; 0,7)$, si se utiliza la T-norma Mín, el grado de cumplimiento de la instancia, en ambos casos es 0.5, si bien la segunda instancia cumple en menor medida el segundo predicado.

Por este motivo se propone utilizar la familia de T-normas de Hamacher [22], que permiten tener mayor flexibilidad en los posibles modelos al controlar la tasa de caída de la función mediante su parámetro p (Ec. 3, Fig 2). Además, esta T-norma es equivalente a la T-norma producto cuando $p=1$.

$$T_p^H = \begin{cases} 0 & \text{si } p=x=y=0 \\ \frac{xy}{p+(1-p)(x+y-xy)} & \text{en caso contrario} \end{cases} \quad (3)$$

Pongamos como ejemplo un predicado compuesto $P_1 \wedge P_2 \wedge P_3$ que constituye un subgrafo frecuente, siendo $a_k, k = 1,4$, instancias del mismo con grados de pertenencia a cada una de las relaciones mostrados en la tabla 2. Aplicando la T-norma de Hamacher en cada fila obtenemos el grado de pertenencia descrito en la última columna. Luego, el promedio de dichos grados de pertenencia es 0,36 resultando una medida representativa del cumplimiento del patrón de todos los objetos espaciales intervinientes.

4. Prueba de Concepto

4.1. Materiales y Métodos

El proceso de explotación de información para el descubrimiento de asociaciones espaciales (Fig. 1) posee 5 actividades principales entre las que se encuentran la preparación de los datos y la determinación de regiones de interés mediante la definición de vecindarios. Debido a que la extensión realizada en el presente documento no las involucra, se realizarán las pruebas a partir de la actividad de modelado de relaciones mediante grafos. Supondremos entonces que los datos se encuentran preparados y que cada conjunto de datos corresponde a un vecindario espacial.

Se han realizado pruebas en 3 conjuntos de datos (Fig. 4). Estos consisten en 300 puntos espaciales, clasificados en 3 tipos, 3 regiones y 2 líneas [23]. Las regiones parten

μ	P_1	P_2	P_3	$\otimes_{p=0}^H$
a_1	0.9	0.3	0.5	0.23
a_1	0.8	0.7	0.6	0.43
a_1	0.1	0.4	0.2	0.06
a_1	0.9	1.0	0.8	0.73
			μ_{\otimes}	0.36

Tabla 2: Ejemplo de uso de T-Norma de Hamacher con $p=0$

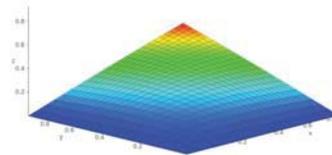


Figura 2: T-norma de Hamacher con $p = 0$.

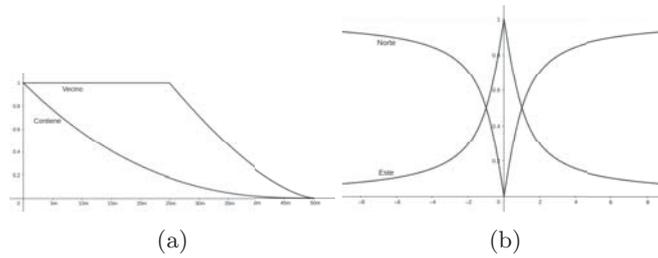


Figura 3: (a) Funciones de pertenencia para las relaciones Cerca y Contiene, en función de la distancia en metros. (b) Funciones de pertenencia para las relaciones Norte y Este, en función de la pendiente de la recta entre dos puntos.

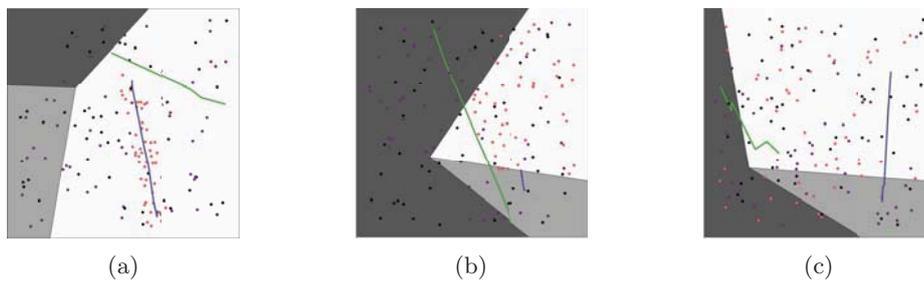


Figura 4: Conjuntos de datos de prueba.

el espacio en 3 partes de manera aleatoria. Las líneas y los puntos se disponen deliberadamente sobre el espacio para conformar ciertos patrones que se desean encontrar, incorporando también un error aleatorio. La prueba de concepto busca determinar si el proceso descubre adecuadamente los patrones de asociación que han sido forzados, y si el valor de pertenencia difuso describe adecuadamente la situación.

Las relaciones difusas binarias que se han utilizado son: “Cerca”, “Contiene”, “Norte”, “Este” y sus negaciones (Fig. 3), algunas de las cuales se detallan en las siguientes ecuaciones. Las relaciones euclidianas se evalúan entre los puntos más cercanos a la geometría involucrada, las direccionales se obtienen a partir del ángulo relativo a las posiciones de los objetos espaciales, y las topológicas se basan en el modelo de 9 relaciones mencionado previamente.

$$\mu_{Cerca}(x, y) = \begin{cases} 1 & \text{si } \delta \leq 25 \\ (-\frac{1}{25} Dist(x, y) + 2)^{\frac{3}{2}} & \text{si } 25 < Dist(x, y) < 50 \\ 0 & \text{si } Dist(x, y) \geq 50 \end{cases} \quad (4)$$

$$\mu_{Contiene}(x, y) = \begin{cases} 1 & \text{si } x \text{ contiene a } y \\ (-\frac{1}{50} Dist(x, y) + 1)^{\frac{5}{2}} & \text{si } Dist(x, y) < 50 \\ 0 & \text{si } Dist(x, y) \geq 50 \end{cases} \quad (5)$$

$$\mu_{Norte}(A, B) = \begin{cases} 1 & \text{si } A_x = B_x \wedge A_y > B_y \\ |\frac{2}{\pi} \tan^{-1} | \frac{A_y - B_y}{A_x - B_x} || & \text{si } A_x \neq B_x \wedge A_y > B_y \\ 0 & \text{en caso contrario} \end{cases} \quad (6)$$

Los cálculos han sido realizados alrededor de un tipo de punto de referencia (puntos rojos, *target*) y codificadas dentro de un archivo JSON adecuado para ser entrada del algoritmo SUBDUE. Una vez extraídos los subgrafos frecuentes, se ha calculado los grados de pertenencia de los predicados intervinientes.

Tanto la generación de los datos como la extracción de relaciones difusas fue realizada utilizando Python3.7 y las librerías Geopandas, Pandas, Shapely y Numpy, disponibles en el entorno Anaconda (<https://www.anaconda.com/>). QGIS3.8 (<https://www.qgis.org/es/site/>) fue utilizado para la validación de los datasets de pruebas. La extracción de los subgrafos frecuentes fue realizada con la herramienta SUBDUE (<http://ailab.wsu.edu/subdue/>).

4.2. Resultados

Por cada subgrafo se analizó cada una de las combinaciones de relaciones posibles para determinar cómo influyen los grados de pertenencia individuales en el grado de pertenencia conjunto del patrón. En el primer caso (Tabla 3) se puede observar que $\bar{\mu}_{\otimes} = 1$ para el primer patrón, al estar todos los puntos rojos contenidos en la región blanca (Fig. 4,a). Al contemplar también la relación de vecindad con la línea azul, se reduce el grado de pertenencia conjunto. Esto significa que existen ciertos puntos que no cumplen totalmente la relación de vecindad.

En el segundo caso se agrega un error aleatorio, por lo que existen ciertos puntos rojos que se encuentran incluidos en la región blanca pero no totalmente, lo cual reduce el valor de $\bar{\mu}_{\otimes}$ (Fig. 4,b). Al incluir la relación direccional Norte, se nota que los puntos se ubican en cierto grado al norte de la línea azul, cuestión que impacta en el grado de pertenencia. De igual forma sucede al considerar la misma relación pero con la línea verde, por lo que se concluye que los objetos espaciales cumplen de forma escasa con dicha propiedad. Si bien el subgrafo frecuente los incluye por haberse considerado el soporte de la relación difusa, es necesario entonces filtrar las aristas del subgrafo cuyo grado de pertenencia sea bajo.

Adicionalmente, con el tercer conjunto de datos se han encontrado tres subgrafos distintos que contemplan las relaciones consideradas. En el primer caso, se nota la posibilidad de que los puntos rojos no se encuentren al este de ninguna de las dos líneas, estando claramente al este de la línea verde. Esto sucede debido a la definición de la relación difusa (Tabla 3, fila 7,9).

Por último, en el tercer patrón detectado, se evidencia cómo la relación direccional no afecta significativamente al valor de la relación topológica, por lo que es un patrón de conocimiento que describe adecuadamente el conjunto de datos. Al agregar la relación direccional con la línea verde se reduce de nuevo, mas sigue estando dentro de valores tolerables.

Los patrones descubiertos describen adecuadamente los conjuntos de datos, y representan de forma apropiada los grados de pertenencia de las instancias de datos a cada uno de las regularidades descubiertas. Sin embargo, es considerable el tiempo y los recursos que demanda la ejecución del algoritmo de minado de subgrafos frecuentes debido a que esta es un problema cuya complejidad escala considerablemente al agregarse aristas y vértices. En futuros trabajos deberá evaluarse este aspecto del proceso.

5. Conclusión

Se ha presentado una extensión del proceso de explotación de información para el descubrimiento de asociaciones espaciales para incorporar relaciones difusas al mismo,

Patrón	Compresión	μ_{\otimes}
$rojo(p) \wedge blanca(r) \wedge contiene(r, p)$	0.219	1.0
$rojo(p) \wedge blanca(r) \wedge azul(l) \wedge contiene(r, p) \wedge cerca(p, l)$	0.219	0.838
$rojo(p) \wedge blanca(r) \wedge contiene(r, p)$	0.119	0.948
$rojo(p) \wedge blanca(r) \wedge contiene(r, p) \wedge azul(l) \wedge norte(p, l)$	0.119	0.815
$rojo(p) \wedge blanca(r) \wedge verde(l) \wedge contiene(r, p) \wedge norte(p, l)$	0.119	0.598
$rojo(p) \wedge blanca(r) \wedge azul(l_1) \wedge verde(l_2) \wedge contiene(r, p) \wedge norte(p, l_1) \wedge norte(p, l_2)$	0.119	0.618
$rojo(p) \wedge verde(l) \wedge \neg este(p, l)$	0.077	0.489
$rojo(p) \wedge azul(l) \wedge \neg este(p, l)$	0.077	0.951
$rojo(p) \wedge verde(l) \wedge este(p, l)$	0.068	0.578
$rojo(p) \wedge blanca(r) \wedge contiene(r, p)$	0.062	0.947
$rojo(p) \wedge blanca(r) \wedge contiene(r, p) \wedge azul(l) \wedge \neg este(p, l)$	0.062	0.954
$rojo(p) \wedge blanca(r) \wedge contiene(r, p) \wedge azul(l_1) \wedge \neg este(p, l_1) \wedge verde(l_2) \wedge este(p, l_2)$	0.062	0.832

Tabla 3: Resultados obtenidos.

lo que permite modelar la realidad asociada a los datos espacialmente referenciados de manera más adecuada.

Esta extensión se lleva a cabo mediante la aplicación de una T-norma sobre el grado de pertenencia de cada instancia del patrón, medida que se promedia para obtener un valor representativo del grado en el cual todos los datos involucrados cumplen la regularidad descubierta.

Se ha realizado una prueba de concepto con datos generados para tal fin, obteniendo resultados que representan adecuadamente las situaciones planteadas. Queda pendiente la prueba sobre datos reales y la evaluación de eficiencia de la aplicación de las herramientas asociadas al proceso.

Agradecimientos

La investigación presentada en este artículo fue parcialmente financiada por el Programa de Doctorado para el Fortalecimiento de Áreas I+D+i (2016-2020) de la Universidad Tecnológica Nacional y por el Proyecto de Investigación 80020160400001LA de la Universidad Nacional de Lanús. Los autores agradecen a Karen Nerak por sus comentarios sobre la redacción y al Prof. Dr. Ramón García-Martínez quien ha motivado estos estudios y ya no se encuentra entre nosotros.

Referencias

1. Mennis, J., & Guo, D. “Spatial data mining and geographic knowledge discovery—An introduction.” *Computers, Environment and Urban Systems* 33.6 (2009): 403-408.
2. Koperski, Krzysztof, and Jiawei Han. “Discovery of spatial association rules in geographic information databases.” *International Symposium on Spatial Databases*. Springer, Berlin, Heidelberg, (1995).
3. Leung, Yee. “Knowledge Discovery in Spatial Data.” Ed. Manfred M. Fischer et al. 1st ed. Vol. 45. Hong Kong: Springer, (2011).
4. Agrawal, Rakesh, and Ramakrishnan Srikant. “Fast algorithms for mining association rules.” *Proc. 20th int. conf. very large data bases(VLDB)*. Vol. 1215. (1994).
5. Appice, Annalisa et al. “Discovery of Spatial Association Rules in Geo-Referenced Census Data : A Relational Mining Approach.” *Intelligent Data Analysis 7* (2003): 541–566.

6. Rottoli, Giovanni Daián. "Formulación de Procesos para una Ingeniería de Explotación de Información Espacial." *XXI Ibero-American Conference on Software Engineering*. Doctoral Symposium. Bogotá, Colombia, (2018).
7. Rottoli, Giovanni Daián, Hernán Merlino, and Ramón García-Martínez. "Proceso para el descubrimiento de asociaciones espaciales mediante subgrafos frecuentes." *XXIV Congreso Argentino de Ciencias de la Computación* (La Plata, 2018).. 2018.
8. Shi, Wenzhong, Anshu Zhang, and Geoffrey I. Webb. "Mining significant crisp-fuzzy spatial association rules." *International Journal of Geographical Information Science* 32.6 (2018): 1247-1270.
9. Ladner, Roy, Frederick E Petry, and Maria A Cobb. "Fuzzy Set Approaches to Spatial Data Mining of Association Rules." *Transactions in GIS* 7.1 (2003): 123-138.
10. Appice, Annalisa, et al. "Discovery of spatial association rules in geo-referenced census data: A relational mining approach." *Intelligent Data Analysis* 7.6 (2003): 541-566.
11. Malerba, Donato. "A relational perspective on spatial data mining." *International Journal of Data Mining, Modelling and Management* 1.1 (2008): 103-118.
12. Maiti, Sandipan, and R. B. V. Subramanyam. "Mining behavioural patterns from spatial data." *Engineering Science and Technology, an International Journal* 22.2 (2019): 618-628.
13. Egenhofer, Max J., and Robert D. Franzosa. "Point-set topological spatial relations." *International Journal of Geographical Information System* 5.2 (1991): 161-174.
14. Gader, Paul D. "Fuzzy spatial relations based on fuzzy morphology." *Proceedings of 6th International Fuzzy Systems Conference*. Vol. 2. IEEE, 1997.
15. Laube, Patrick, Mark De Berg, and Marc Van Kreveld. "Spatial support and spatial confidence for spatial association rules." *Headway in Spatial Data Handling*. Springer, Berlin, Heidelberg, 2008. 575-593.
16. Miller, Harvey J. "Tobler's first law and spatial analysis." *Annals of the Association of American Geographers* 94.2 (2004): 284-289.
17. Bianco, Santiago. "Análisis Comparativo de Algoritmos de Minería de Subgrafos Frecuentes." *Revista Latinoamericana de Ingeniería de Software* 4.2 (2016): 111-142.
18. Jiang, Chuntao, Frans Coenen, and Michele Zito. "A survey of frequent subgraph mining algorithms." *The Knowledge Engineering Review* 28.1 (2013): 75-105.
19. Zadeh, Lotfi A. "Fuzzy sets." *Information and control* 8.3 (1965): 338-353.
20. Lazzari, L., et al. "Los conjuntos borrosos y su aplicación a la programación lineal." Facultad de Ciencias Económicas, Universidad de Buenos Aires, Buenos Aires (2001).
21. Ketkar, Nikhil S., Lawrence B. Holder, and Diane J. Cook. "Subdue: Compression-based frequent pattern discovery in graph data." *Proceedings of the 1st international workshop on open source data mining: frequent pattern mining implementations*. ACM, 2005.
22. Fodor, János C., and Tibor Keresztfalvi. "A characterization of the Hamacher family of t-norms." *Fuzzy Sets and Systems* 65.1 (1994): 51-58.
23. Rottoli, Giovanni Daián. "Datos para la prueba de la extensión difusa del proceso para SARM", *Mendeley Data*. (2019). v1. <http://dx.doi.org/10.17632/b254wfhfkg.1>

Genericidad de funciones: El *quid* para la incorporación de dominios en un Sistema Funcional Inductivo basado en Haskell

Félix Gómez, Pedro González, Derlis Salinas,
Gustavo Sosa-Cabrera, María Elena García-Díaz

Universidad Nacional de Asunción, Paraguay
{fgomez,pgonzalez,dsalinas,gdsosa,mgarcia}@pol.una.py
<http://www.pol.una.py>

Resumen Hoy en día, una inconmensurable cantidad de programas informáticos se encuentran en ejecución generando información propia de su comportamiento, estos tipos de historial son amplia y generalmente conocidos como *Logs*. Sin embargo, a pesar de los avances en la *inferencia funcional inductiva* para trabajar con los datos, hasta el momento se ha prestado escasa atención a la automatización del procesamiento analítico de estos tipos de registros de eventos. En este sentido, puesto que la alta expresividad de los *Lenguajes de Programación Declarativos* es una noción ampliamente aceptada, en este trabajo se aborda las implicancias prácticas de la *Programación Funcional Inductiva* aplicado en el dominio específico de los *Logs* para su inferencia asistida *ex professo*.

Palabras Claves: Funciones Genéricas · Logs · Dominio Específico · Conocimiento de Fondo · Inferencia Funcional Inductiva · Haskell

1. Introducción

A día de hoy, cada vez es más difícil ignorar la cantidad no conmensurable de programas informáticos de diverso índole que se encuentran en ejecución generando registros sobre su propio funcionamiento, ya sea informativamente o indicando anomalías. Estos tipos de historial son ampliamente conocidos como *Logs* y se constituyen como un recurso vital para la administración de los sistemas ya que permiten tanto el monitoreo periódico como así también el diagnóstico de problemas. En adición, puesto que el gran volumen de registros de eventos serializados a diario hace que sea poco práctico inspeccionar manualmente los *logs*, analizarlos de manera efectiva sigue siendo un gran desafío [10].

Aunque una importante labor se ha llevado a cabo, tanto por parte de la academia, como también por la industria; las 2 alternativas propuestas hasta el momento para el análisis de *logs* consisten básicamente en (1) la extracción de patrones mediante expresiones regulares (e.g. *grok*¹) ó (2) más recientemente, por medio de aplicación de técnicas tradicionales de minería de datos (TTMD).

¹ <https://logz.io/blog/logstash-grok>

Sin embargo, ambos enfoques respectivamente presentan una serie de inconvenientes importantes como ser (1) la escritura manual de reglas consume mucho tiempo y es propenso a errores; (2) la fase de aprendizaje de las *TTMD* poseen una complejidad temporal en función al tamaño del conjunto de datos y a la vez requieren una considerable cantidad de registros para evitar el *sobre-ajuste*.

En este sentido, prolifera la preocupación en entornos *TICs*, puesto que ínterin los *logs* crecen en cantidad, volumen y complejidad; los administradores de sistemas deben solucionar los problemas lo más rápido posible para satisfacer a los clientes, proteger la imagen corporativa y mantener las ventas [9].

Por otra parte, estudios previos recientes han señalado que a diferencia de los enfoques estándar de aprendizaje automático utilizado en las *TTMD*, la Programación Funcional Inductiva (*IFP*, por sus siglas en inglés) aborda el problema de aprender programas pequeños pero complejos, de unos pocos ejemplos representativos de *entrada/salida*, generados a medida que el usuario transforma una o muy pocas instancias o campos particulares de los datos, como lo hacemos los humanos análogamente [2]. Asimismo, es una noción ampliamente aceptada la gran expresividad de la Programación Declarativa del cual descende la *IFP*.

Surge por tanto, la oportunidad de dar luz a la investigación de la *IFP* en el contexto de la automatización del proceso de inferencia asistida de *logs* y de hecho, este es el primer estudio en el que se realiza la incorporación de este tipo de *dominio* en un *Sistema Inductivo Funcional* (*SIF*).

Los resultados de esta investigación demuestran que mediante un tratamiento adecuado de la *genericidad de funciones* como factor principal, es posible la explotación de las ventajas del *IFP* en cuanto a inferencia de *logs* se refiere.

2. Trabajos Relacionados

Una de las razones del éxito de los sistemas *IFP* es el uso de lenguajes específicos de dominio (*DSL*, por sus siglas en inglés). Así *FlashFill*[3], como ejemplo de ello, puede realizar transformaciones sintácticas de cadenas utilizando formas restringidas de expresiones regulares, condicionales y bucles en tablas de hojas de cálculo. Sistemas *IFP* que también tuvieron protagonismo son *Igor-II* [6] y *ADATE* [11]. Sin embargo, éstos no han sido actualizados recientemente ni puestos en práctica. *Igor-II* impone una restricción estricta en el conjunto de ejemplos dado como especificación y *ADATE* requiere una gran habilidad para la síntesis de programas simples. Además, debido a la ausencia de memorización, tienen una desventaja en la velocidad práctica en comparación con *MagicHaskeller*, que puede iniciar la síntesis con su tabla de memorización llena de expresiones[5], como se menciona en el apartado 3.3 de la siguiente sección.

3. Materiales y Métodos

3.1. Logs

Registro del “qué sucedió, cuándo y por quién” del sistema. Los *logs* de aplicaciones graban cronológicamente las operaciones durante el funcionamiento de

la aplicación y vienen categorizados en tres niveles principales, donde representan la información de depuración, información narrativa o eventos de la lógica del negocio. A continuación, los tipos de *logs* de interés para este estudio.

Logs de Spring. Se refiere al proyecto *Spring Framework*². Un *framework web* es un conjunto de componentes que ofrece mecanismos para la implementación de las capas del negocio y presentación web [1]. *Spring Framework* proporciona soporte para diferentes arquitecturas de aplicaciones, incluidos mensajería, datos transaccionales y persistencia en ambiente web.

Logs de Apache³. Para administrar un servidor web de manera efectiva, es necesario obtener comentarios sobre la actividad y el rendimiento del servidor, así como sobre cualquier problema que pueda estar ocurriendo. El servidor HTTP *Apache* proporciona capacidades de registro muy completas y flexibles, y con ellos describimos dos divisiones principales, *Error Log* y *Access Log*.

3.2. Programación Declarativa

Se caracteriza por una *alta expresividad* basado en la utilización de bloques de construcción como funciones, recursión y el uso de patrones, de manera a especificar más la solución antes que su cálculo de bajo nivel [8].

Programación Funcional. Deriva de la *Programación Declarativa*, prescinde del uso de bloques condicionales o ciclos. Su característica principal reside en el uso de expresiones y funciones, donde un programa es una función pura. Siendo función en términos matemáticos obedeciendo sus fundamentos.

Función total. Toda función relaciona dos conjuntos a través de una operación algebraica. Una función total es aquella que está definida para todos los posibles valores del conjunto dominio.

Programación Inductiva. (*IP, por sus siglas en inglés*) desarrolla métodos en base al razonamiento inductivo, esto es, de lo específico a lo general.

IFP. Una vertiente en auge de la *IP* es la Síntesis de Programas Inductivos (*IPS, por sus siglas en inglés*). Generalmente en *IP*, infinitos programas semánticamente diferentes cumplen con una misma especificación, por lo tanto, se necesita definir criterios para elegir entre ellos. Estos criterios se denominan sesgo inductivo, que se dividen en: (1) Sesgo de restricción: El sistema *IPS* solo es capaz de generar un subconjunto específico de las funciones de un dominio, debido a que tiene un lenguaje restringido, incapacidad computacional o sus operadores no logran alcanzar cada programa; (2) Sesgo de preferencia: El sistema depende del orden en que se explora el espacio del problema, causando que la organización de las soluciones dependa del orden de exploración.

La búsqueda en los métodos *IPS* funcionales se basa en generar-y-probar. A diferencia de técnicas de aprendizaje automático convencionales donde los ejemplos de E/S son utilizados para la construcción de modelos, en *IPS*, los ejemplos de E/S se utilizan única y exclusivamente para probar los programas generados. La *IFP* es el marco para la programación automatizada para lograr la síntesis de programas funcionales recursivos a partir de ejemplos de E/S.

² <https://github.com/spring-projects/spring-framework>

³ <https://httpd.apache.org>

3.3. Domain Specific Induction (DSI)

Karmiloff-Smith (1994) define a un *dominio* como el “conjunto de representaciones que sostiene un área específica de conocimiento” (p. 23)[7]. Ejemplos de dominio podrían ser las fechas, los correos electrónicos, nombres de personas y *logs*. DSI es un sistema que pretende la automatización del procesamiento de datos o *Data Wrangling* mediante la implementación del software *MagicHaskell*[4], sumado a un conjunto de funciones definidas para un cierto dominio y una interfaz de acceso web para la interacción humano-computador.

Por otra parte, la interfaz web permite, a partir de datos de E/S, dónde la Entrada es el valor a ser procesado y la Salida es el resultado esperado, transformarlo a un predicado para utilizarlo como valor de entrada en el *MagicHaskell*. Éste retorna como resultado la función f que satisface el predicado. Luego la función f es aplicada al resto de entradas del *DSI* a fin de obtener los valores para cada columna de salida. Desde la interfaz web, se envían los predicados al *MagicHaskell* a través de un puerto abierto en el servidor, donde se ejecuta una instancia del sistema para cada dominio abarcado. *MagicHaskell* al encontrar una o varias funciones que resuelven el problema enviado en el predicado, lo devuelve en una cadena con el siguiente formato:

```
(\a -&gt; function1 a), (\a -&gt; function2 a), (\a -&gt; function3 a)
```

El proceso *data wrangling* culmina cuando el sistema *DSI* selecciona la primera función de la lista: `function1 a`, dónde `a` se reemplazará por el valor de cada entrada para evaluar la función hasta completar todos los campos de salida.

MagicHaskell. Un sistema IFP que adopta el enfoque de generar-y-probar, llamado *MagicHaskell*, el cual al recibir un conjunto de ejemplos de E/S, primero infiere el tipo de expresiones deseadas. Luego, genera expresiones del mismo tipo que se pueden expresar con las funciones de la biblioteca de componentes, combinadas con aplicaciones de funciones y abstracciones *lambda*. La generación se realiza de forma exhaustiva a partir del más corto. Luego se comparan con la especificación en orden y las que pasan la prueba son las funciones sintetizadas.[4]

MagicHaskell puede sintetizar programas funcionales cortos sin restricción del espacio de búsqueda en base a cualquier conocimiento previo, gracias a su gran tabla de memorización. El sistema recibe un predicado, en este caso, una expresión *lambda* con la siguiente estructura: $\lambda f \rightarrow f \text{ “entrada”} == \text{ “salida”}$. Para reducir el espacio de búsqueda, se necesita limitar la cantidad de funciones, conocido como *Conocimiento de Fondo Específico de un Dominio* (DSBK, por sus siglas en inglés), para ello, se generan los prototipos de las funciones en el archivo “primitives”, con prioridades por función, donde 0 es el más prioritario; a este número de funciones se lo denomina Amplitud o *Breadth* (b). Otro valor importante es la Profundidad o *Depth* (d), que limita la profundidad con la cual se combinarán las funciones disponibles para solucionar un predicado. La relación entre estos dos valores se da en la cota superior asintótica, siendo una de orden potencial exponencial dado $O(b^d)$. Considerando esto, el valor de d es clave en la fase de entrenamiento, ya que con una cantidad alta de b y d , la fase

de entrenamiento podría no culminar, debido a que el computador no posee los suficientes recursos para realizar todas las combinaciones posibles y almacenarlas en su Tabla de Memorización, obedeciendo al criterio de *sesgo de restricción*.

4. Experimentos

4.1. Réplica de experimentos y verificación de resultados de *DSI*

Con el objetivo de montar el escenario experimental del presente estudio validando a la vez su correctitud, se han verificado y reproducidos los resultados obtenidos en *DSI*[2], basado en su implementación⁴ se comprueba la ejecución e interacción con *MagicHaskell* en la búsqueda de funciones, con dominios como fechas, correos, nombres y palabras. Para realizar los experimentos se utiliza una máquina virtual con una vCPU, dos gigabytes de memoria RAM y un disco de almacenamiento de 25 gigabytes. El sistema operativo utilizado es Ubuntu 18.04.2 para arquitecturas x64. Acorde a estas especificaciones, se ha empleado un aproximado de cuatro horas reloj en modificaciones para hacerlo funcionar.

4.2. Definición de nuevo dominio y la generación de funciones

En este estudio, se amplía el alcance de *DSI*[2] con la inclusión del dominio de *Logs*. Inicialmente incorporando *logs* del framework de *Spring*, ya que éste tiene una estructura básica sumamente completa, definida en la Sección 3.1.

Se determina generar las funciones para la extracción de porciones de datos como las fechas, el nivel del *log*, el nombre de la clase Java en donde se registró el *log*, y el mensaje del *log*. Basado en ello, se generan las correspondientes funciones en Haskell: (1) *extractDate*; (2) *getLogLevel*; (3) *getClass*; (4) *getMessage*.

Conforme a la estructura de un *log* de *Spring*, se generan estas funciones sin antes haber analizado a profundidad las características de las funciones ya presentes en *MagicHaskell* ni las funciones generadas para *DSI*; el punto principal es la construcción de funciones que realicen de la manera más sencilla posible la extracción del dato solicitado a partir de una cadena de texto. Para añadir las funciones a *MagicHaskell* se sigue el siguiente procedimiento:

- Crear un módulo en el directorio */MagicHaskell*. Realizar las importaciones necesarias y agregar las nuevas funciones correspondientes al dominio.
- En el archivo *Individuals.hs*, se debe agregar el prototipo de cada función nueva, dentro del parámetro “totals”, con el siguiente formato:

```
++ $(p [| function :: [Char] ->[Char] |] )
```
- Agregar el nombre del módulo en el archivo *MagicHaskell.cabal*, dentro del parámetro “Build-depends”.
- Compilar el programa desde el directorio */MagicHaskell*, mediante:

```
cabal install --global
```
- Generar las primitivas con el comando:

```
MagicHaskell --dump-primitives > primitives
```

⁴ <https://github.com/liconoc/DataWrangling-DSI>

En pruebas de ejecución se somete a *MagicHaskell* con predicados como:

```
f "2017-03-03 13:02:50.608 INFO ... info message" == "IndexController"
```

Se obtiene la función `getClass` que convierte el *input* en el *output* esperado. *MagicHaskell* puede retornar como solución una función compuesta equivalente a la búsqueda; la función `getClass` es equivalente a `extractDate(getClass)` ya que ante un mismo *input* retornan exactamente el mismo *output*.

Considerando lo mencionado en la Sección 3.3, una desventaja de *MagicHaskell* es el potencial exceso de tiempo en la fase de entrenamiento. La cantidad de tiempo está directamente ligada a los parámetros *depth* y *breadth*. Se maneja como hipótesis la posibilidad de que el agregado de predicados adicionales sobre un dominio pudiera mejorar el entrenamiento y los resultados se encuentren en instancias de ejecución con menores valores de *depth*. Para tal efecto, se efectúa un experimento buscando obtener variaciones, alteración de resultados o algún comportamiento anómalo que pudiera registrarse. Se añadió diez predicados diferentes por función, totalizando cincuenta nuevos predicados. Añadidos los predicados, los resultados de las pruebas son exactamente iguales a cuando *MagicHaskell* se ejecuta sin ningún predicado añadido. Esto confirma que los predicados adicionales y pretendidos como *DSBK* para el *MagicHaskell* no generan alteraciones en su comportamiento en la búsqueda de funciones, obedeciendo la teoría de los sistemas *IPS* (Sección 3.2).

4.3. Ampliación del dominio de *Logs*

Se contempla la necesidad de ampliar el dominio agregando otra estructura de *logs*, mediante *logs* del servidor web *Apache* (Sección 3.1). Para tal efecto, se generan funciones para la extracción de fechas, inicialmente para *logs* de acceso de *Apache*, mediante la función `extractApacheDate`. Consultar definición en repositorio publicado⁵.

En pruebas de ejecución, se someten predicados similares a:

```
f "109.24.239.55 ... "-" "-" == "13/Nov/2018:17:30:26 -0300"
```

Efectivamente, ante un predicado así, *MagicHaskell* responde con la función `extractApacheDate`, sin embargo, esta función representa un caso muy específico, ya que se trata de la extracción de la fecha solo para este tipo de *log*. Entonces, nace el cuestionamiento sobre la posibilidad de realizar funciones capaz de abarcar un mayor espectro de casos posibles, teniendo en cuenta la inmensa variedad de estructuras y tipos de *logs* existentes por *frameworks*.

4.4. Propuesta de caracterización de funciones: Genéricas.

Dado que los *DSBK* contienen funciones para la manipulación de un determinado tipo de dato, muchas de ellas pueden resolver problemas relacionados con otros tipos de datos. Además, si los datos pertenecen a un dominio y el problema en cuestión termina siendo una tarea muy exclusiva que pertenece a ese

⁵ <http://www.github.com/derlissalinas/dsi>

dominio, se necesitan funciones más precisas para obtener resultados correctos. En pos de caracterizar las funciones a añadir para un dominio, se contempla la caracterización estableciendo dos límites. Primero, con funciones **genéricas**, basada en dos principios que se adoptan para representar la idea propuesta:

Especificidad. Las funciones generadas deben retornar únicamente los datos esperados, implicando validaciones en los datos de retorno. Por ejemplo, la función `extractApacheDate` de la Sección anterior, extrae la fecha solamente para los *logs* de *Apache*, del tipo `Access`, sin embargo, carece de una validación para afirmar que lo retornado es un dato que pertenezca al dominio de fechas.

Generalidad. Las funciones deben ser capaces de abarcar un mayor espectro de casos posibles dentro de un mismo dominio, realizando las correspondientes validaciones por medio de condicionales.

Teniendo los casos de *logs* de *Spring* y *Apache*, se generan las funciones de ejemplo basadas en la **especificidad** para la extracción de fechas, denominadas `getDateSpring` y `getDateSpringArray`. De igual forma, se generan las funciones `getDateApache` para `Access` y `Error`. Por otra parte, se genera un ejemplo de una función genérica, que se aferra al principio de **generalidad**. La implementación está presente en el *script* `C2_Logs.hs` del repositorio⁶.

Mediante condicionales se evalúan todas las funciones específicas hasta determinar la que retorne un valor distinto de vacío.

4.5. Propuesta de caracterización de funciones: Concretas

Es posible generar funciones de concretas, simples y puntuales para un caso en particular. Respecto a la Sección anterior, estas funciones guardan similitud con la simplicidad del principio de *especificidad*, pero sin la validación correspondiente. Tales son los casos de las funciones generadas en la Sección 4.2 para *logs* del *framework Spring*. A continuación se muestra una función como ejemplo para la extracción de fechas.

```
Prelude>extractDate :: [Char] -> [Char]
      extractDate x = unwords (take 2 (splitStringByPunctuation x " "))
```

Estas funciones son generadas netamente para un caso en particular, es decir, obedecen sólo a predicados que posean *logs* de *Spring* como *input*, y por otra parte, en el *output*, lo que explícitamente cada función es capaz de evaluar.

5. Resultados y Discusión

La caracterización de funciones para el agregado de un nuevo dominio no es, en absoluto, determinante, ya que está sujeto a factores intrínsecos de cada dominio que lo diferencia de otros, por lo cual, no es posible decir, por ejemplo, que un conjunto de funciones concretas se corresponde como la mejor solución para todos los dominios existentes que pudiesen añadirse a *DSI*. Al definir un dominio y pretender añadirlo a *DSI*, debe considerarse lo mencionado en la

⁶ <http://www.github.com/derlissalinas/dsi>

Sección 3.3, los parámetros *depth* y *breadth* son determinantes al momento de contemplar la caracterización ya que éstos valores determinan el tamaño de la tabla de memorización.

Depth. El parámetro más relevante dentro de la caracterización determina la cantidad de composiciones posibles entre las funciones del *MagicHaskell*, lo cual implica un aumento exponencial en la cantidad de combinaciones posibles.

Breadth. Cantidad de funciones que generemos para nuestro dominio.

Tomando el caso del dominio de *Logs*, es sabido que existe un enorme número de estructuras diferentes de registros si se considera la cantidad de *frameworks* existentes con la consecuente posibilidad de modificación de dicha estructura básica en una aplicación, pero una vez definida la estructura del *log*, la misma está sujeta a escasas variaciones, por lo cual, es plausible considerar la creación de funciones concretas que operen directamente sobre dicha estructura de registro. Esto implica que la cantidad de funciones a definir es directamente proporcional a la necesidad de datos a obtener de un *log*, como se mencionó en la Sección 4.2, cinco funciones en el *DSBK* bastaron para satisfacer la necesidad del caso.

Por otra parte, considerando un dominio como el de fechas, existe una amplia gama de funciones a definir para el tratamiento de datos, dadas las diferentes variaciones y estándares que existen, aunque así como con el dominio de *Logs*, pueden crearse funciones concretas para manejar cada caso o cada necesidad de obtener un dato a partir de cada *input*. Es sabido que una fecha en formato “DD/MM/AA” puede representarse también como “DD/month/AA”, donde el mes se define por nombre y no por valor numérico. Ante la necesidad de funciones que abarquen más de un idioma, nace la cuestión en definir funciones de tipo concreta o genérica, ya que un listado de meses de un idioma es definido como una función, no variable, lo cual implica que se deberá definir una función para cada idioma añadido. Esto eleva proporcionalmente la cantidad de funciones disponibles aumentando el valor de *breadth*. Como ejemplos se toman a las funciones `convertMonth`, que convierte el nombre de un mes en su valor numérico y a `getMonthName`, que extrae el nombre de un mes embebido en una cadena de caracteres. Una solución al caso puede verse como la definición de funciones genéricas, que realicen las validaciones contemplando los idiomas, evitando añadir todas las definiciones de funciones dentro del archivo *Individuals.hs* y en las *primitivas* de *MagicHaskell* reduciendo drásticamente el valor de *breadth*.

Ante una situación que requiera añadir un dominio, es deber analizar la cantidad de funciones a crear, a fin de contemplar mantenerlas como concretas, o bien, si los datos a extraer requieren de varias funciones específicas cuyas funcionalidades sean similares, implementar una función genérica que englobe todas esas funciones; este último es aplicable a las funciones que precisan un nivel de abstracción superior a las funciones concretas, como el caso de la extracción de fechas independientemente al formato de entrada, o como se menciona anteriormente, el caso de extracción del mes sin importar el idioma.

En la Figura 1, se presenta una comparación de tiempo de entrenamiento en segundos para las funciones `convertMonth` y `getMonthName` en sus versiones

concretas y genéricas, considerándose el agregado de veinte idiomas con sus respectivas lista de meses, variando el valor de *depth* en cada ejecución.

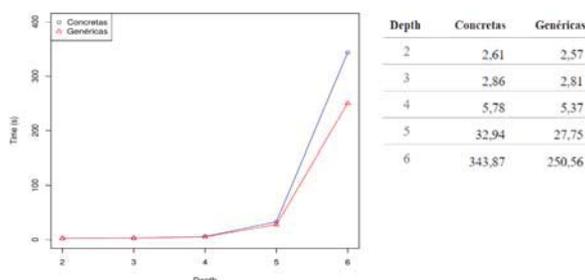


Figura 1. Tiempos de entrenamiento con funciones genéricas y concretas, sobre *depth*.

Un requisito determinante para la creación de funciones y el posterior agregado de un dominio a *DSI* y por ende al *MagicHaskell*, es que las funciones deben de estar definidas para todos los valores posibles de entrada, como una función total⁷, evitando excepciones. Éste requisito es propio del *MagicHaskell* para que las funciones sean contempladas en la fase de entrenamiento y agregadas en la *tabla de memorización*. Al definir funciones deben de contemplarse todos los casos posibles de *inputs* y que la función sea capaz de evaluarlo.

La *Programación Funcional Inductiva* en la síntesis de funciones implica criterios para realizar la selección. Dichos criterios son los sesgos inductivos. En el ambiente del *MagicHaskell*, el *sesgo de preferencia* se manifiesta al momento de ampliar el *DSBK*, agregando los prototipos de las funciones de un determinado dominio al archivo de configuración *Individuals.hs*, ya que el orden en el cual dichos prototipos son colocados, influye para que sean encontradas con menor o mayor facilidad una función. Mismo sesgo se manifiesta mediante el archivo *Primitives*, en donde se indica explícitamente el orden de prioridades para las funciones antes de la ejecución de una nueva instancia del *MagicHaskell*. Ambos casos se deben tener en cuenta para el agregado de un dominio al ambiente de *DSI*. Cabe resaltar que ha quedado como incógnita el porqué de situaciones donde habiéndose cumplido y verificado las condiciones mencionadas, *MagicHaskell* no retorna las funciones definidas.

6. Conclusión y Líneas de Trabajo Futuro

La gran cantidad de *Logs* generándose constantemente en los programas informáticos evidencian la necesidad existente de automatizar su procesamiento y análisis a fin de generar una información útil a partir de éstos. Incorporar el dominio de *Logs* al entorno de *DSI* resulta un proceso poco trivial, en vista a que

⁷ <http://mathworld.wolfram.com/>

se ha detectado como factor determinante a la *genericidad* de las funciones de un dominio. En paralelo, se ha demostrado que *MagicHaskeller* es una herramienta viable para la inferencia de *Logs*. Este aporte podría enmarcar la aceptación de la alta expresividad de los Lenguajes de Programación Declarativos para la solución de necesidades en ambientes poco explorados hasta el momento.

En términos generales, considerando futuros dominios complejos como cadenas de texto o mensajes, implicará caracterizar funciones en cierto grado de *genericidad*. En ello, el *quid* de la incorporación de dominios reside en el balanceo entre casos donde se determine la *genericidad* por función. Basado en lo expuesto, los trabajos futuros podrían centrarse en esclarecer la incógnita respecto al *MagicHaskeller* en los casos donde no retorna funciones previamente definidas. Además, la ampliación de factores para la caracterización, como el análisis de la factibilidad de variación del orden y los valores de prioridades en prototipos, siendo éstos sesgos propios de *MagicHaskeller*.

Referencias

- [1] M. A. Constanzo y S. I. Casas. «Usabilidad de framework web: identificación de problemas y propuesta de evaluación». En: *XXIV Congreso Argentino de Ciencias de la Computación (La Plata, 2018)*. 2018.
- [2] L. Contreras-Ochando. «General-purpose Declarative Inductive Programming with Domain-Specific Background Knowledge for Data Wrangling Automation». En: (2018). DOI: <https://arxiv.org/abs/1809.10054>. arXiv: 1809.10054.
- [3] S. Gulwani. «Automating String Processing in Spreadsheets Using Input-Output Examples». En: *POPL '11 Proceedings of the 38th annual ACM SIGPLAN-SIGACT symposium on Principles of programming languages* (2011), págs. 317-330.
- [4] S. Katayama. «Systematic search for lambda expressions». En: *In Proceedings Sixth Symposium on Trends in Functional Programming*. 2005.
- [5] S. Katayama. «Towards Human-Level Inductive Functional Programming». En: *Artificial General Intelligence 10* (2015), págs. 111-120. DOI: <https://doi.org/10.1007/978-3-319-21365-1>.
- [6] E. Kitzelmann. «A Combined Analytical and Search-Based Approach to the Inductive Synthesis of Functional Programs.» En: (2010).
- [7] Robert Kowalski. «Algorithm= logic+ control». En: *Communications of the ACM 22.7* (1979), págs. 424-436.
- [8] John W Lloyd. «Practical Advantages of Declarative Programming.» En: *GULP-PRODE (1)*. 1994, págs. 18-30.
- [9] M. Mizutani. «Incremental mining of system log format». En: *2013 IEEE International Conference on Services Computing*. IEEE. 2013, págs. 595-602.
- [10] Adam Oliner, Archana Ganapathi y Wei Xu. «Advances and challenges in log analysis». En: *Communications of the ACM 55.2* (2012), págs. 55-61.
- [11] R. Olsson. «Inductive functional programming using incremental program transformation.» En: *Artificial Intelligence 74.1* (1998), págs. 55-81.

Grafos y Temporalidad en Bases de Datos

Una aplicación sobre Datos Abiertos Enlazados

Carlos Buckle¹, Marcos Zárate^{1,2}, Leo Ordinez¹, Damián Barry¹

¹ Laboratorio de Investigación en Informática (LINVI), Facultad de Ingeniería, Universidad Nacional de la Patagonia San Juan Bosco (UNPSJB), Puerto Madryn, Argentina.

² Center for the Study of Marine Systems, Patagonian National Research Center, (CENPAT-CONICET), Puerto Madryn, Argentina.

cbuckle@unpata.edu.ar, zarate@cenpat-conicet.gob.ar,
{leo.ordinez,demian.barry}@gmail.com

Resumen

Diferentes tipos de sistemas han encontrado una alternativa de modelado utilizando grafos. En particular, las bases de datos. Una necesidad común a todos ellos ha sido la de incorporar la dimensión temporal. Este trabajo presenta un análisis de las principales teorías y modelos desarrollados para *Grafos Temporales* y *RDF Temporal* y los vincula bajo una misma clasificación simple de acuerdo a las estrategias de modelado utilizadas, como instrumento orientador al momento de resolver diseños sobre grafos con temporalidad. Adicionalmente, enfoca y amplía sobre las posibilidades de RDF para la incorporación de semántica temporal. El aporte aplicado del trabajo consiste en la extensión, hacia un RDF Temporal, de un modelo conceptual denominado *BiGe-Onto* del dominio de la biodiversidad marina. La extensión desarrollada utiliza la ontología OWL-Time (estándar de W3C), la cual además, amplía las capacidades de descubrimiento de información por parte de investigadores que estudian fenómenos de conservación de especies que requieren ser interpretados en el tiempo.

Palabras clave: Grafos Temporales, RDF Temporal, Datos Abiertos Enlazados, OWL-Time, Bases de datos temporales.

1. Introducción

La teoría de grafos es una poderosa herramienta que permite estudiar diferentes problemas, sociales, naturales o de ingeniería. Una gran parte de los problemas a resolver pueden abordarse con una propuesta de solución orientada a grafos. Entre los sistemas que se pueden modelar con grafos están aquellos en los cuales la dimensión temporal es clave para una representación correcta, como es el caso de los sistemas dinámicos o sistemas que deben registrar cambios históricos en los objetos y/o sus relaciones. En este campo

comienzan a cobrar trascendencia los *Grafos Temporales* [1] referenciados como *temporal graphs*, *temporal networks*, *dynamics graphs* o *time-varying graphs*.

Un grafo temporal (G,T) refleja la dinámica de un sistema en el cual la existencia de los arcos entre los nodos del grafo G puede variar dentro del tiempo T . Incluso, los nodos pueden tener un tiempo de vida limitado en el sistema. La estructura temporal de activaciones de arcos y nodos en un grafo temporal permite realizar análisis de adyacencia, proximidad, centralidad, transitividad, camino más corto y otros, los cuales deben ser adaptados respecto de los análisis clásicos sobre un grafo estático. Detrás del grafo temporal existe un *grafo subyacente*, que es estático y se obtiene contemplando todos los nodos y todos los arcos que se activan en algún momento del tiempo de vida del sistema.

Respecto del modelado de Grafos Temporales, se han desarrollado muchas propuestas, que varían de acuerdo a las características del problema a resolver. A rasgos generales, se identifican dos mecanismos clásicos de modelado, uno es el de *versionado* (*snapshots*) y otro es el de *etiquetado* (*timestamp*). El *versionado* se basa en generar una nueva versión cada vez que cambia el estado del grafo. El *etiquetado* sin embargo, propone un único grafo agregando etiquetas de tiempo a los elementos sujetos a cambio.

De la variedad de sistemas que pueden aplicar modelos de Grafos Temporales, el presente trabajo se enfoca al modelado de *bases de datos temporales* [2], las cuales han sido inicialmente desarrolladas sobre bases de datos relacionales. Estas bases de datos manejan típicamente dos dimensiones temporales: el *tiempo de validez* y el *tiempo de transacción*. El primero es el período de tiempo en que un dato es válido en el mundo modelado. El segundo es el instante en que los datos son actualizados en la base de datos.

Las bases de datos orientadas a grafos, se caracterizan por incorporar tanto a los esquemas como a las instancias de los datos como grafos y manipularlos con operaciones también orientadas a grafos. Estos modelos comenzaron a desarrollarse a principio de los 80s; puede verse un compendio de su historia y principales trabajos en [3]. Si bien en su momento no llegaron a ser demasiado influyentes, han tomado especial trascendencia con la necesidad de manejar información de naturaleza interconectada con el surgimiento de la Web Semántica [4], las redes sociales [5] y los datos enlazados [6].

En este contexto, ha cobrado relevancia el uso de Resource Description Framework¹ (*RDF*) como una alternativa de datos estructurados en un grafo y sobre el cual hay un volumen creciente de información representada actualmente en la Web. Un grafo RDF $G=(S,P,O)$ es un multigrafo dirigido y rotulado, formado por tripletas (s,p,o) las cuales representan sentencias que conectan un sujeto (s) y un objeto (o) por medio de un predicado (p). Cada tripleta es un arco $s \rightarrow^p o$ en el grafo. Debe considerarse el conjunto $S=(\hat{I} \cup \beta)$, donde \hat{I} son todos los recursos del sistema identificados por una *IRI* (*Internationalized Resource Identifier*) y β son *Blank-Nodes* que representan recursos no identificados por IRIs; el conjunto $P=\hat{I}$ y el conjunto $O = \hat{I} \cup \beta \cup \mathcal{L}$, donde \mathcal{L} es el

¹ <https://www.w3.org/RDF/>

conjunto de literales permitidos. Además de permitir el modelado de datos, RDF provee el lenguaje de consultas *SPARQL*².

Al igual que los grafos clásicos, los grafos RDF han tenido la necesidad de escalar incorporando la dimensión temporal, dando origen a los *RDF Temporales* [1].

El tránsito hacia los RDF Temporales ha abordado problemas similares al de los grafos clásicos hacia Grafos Temporales y muchos trabajos de ambas áreas, han analizado problemáticas equivalentes. Sin embargo, no se encuentran referencias cruzadas entre trabajos de sendos corpus de conocimiento.

Este trabajo realiza un análisis de modelos de referencia propuestos por diferentes autores para Grafos Temporales y para RDF Temporales y los pone en común, clasificándolos según las estrategias de modelado, con el objetivo de aportar una herramienta que oriente la tarea de diseño sobre grafos con temporalidad. Adicionalmente, considerando la tendencia que han seguido las bases de datos orientadas a grafos, se enfoca específicamente en RDF, con el objetivo de generar una aplicación concreta de implementación de RDF Temporal con incorporación de semántica. La misma, se aplica sobre un conjunto de datos abiertos enlazados denominado *BiGe-Onto*³, desarrollado por el Centro Científico Tecnológico CONICET-CENPAT⁴. Este desarrollo extiende el modelo existente con entidades y propiedades temporales basadas en la ontología OWL-Time, estándar de W3C, la cual incorpora nuevas capacidades de descubrimiento de información relacionada con el tiempo.

El resto del documento está estructurado de la siguiente manera: la Sección 2 clasifica trabajos de referencia de Grafos Temporales y RDF Temporal de acuerdo a sus estrategias de modelado. En la Sección 3, se orienta especialmente a las posibilidades de RDF para proveer semántica temporal sobre los datos y presenta una ontología de uso en la comunidad. La Sección 4 implementa dicha ontología sobre un caso real de un repositorio de Datos Abiertos Enlazados registrados por CONICET-CENPAT y muestra un ejemplo de consulta con semántica temporal. Finalmente, la Sección 5 presenta las conclusiones y trabajos futuros.

2. Estrategias de modelado de la dimensión temporal

En teoría y modelado de Grafos Temporales y RDF Temporales, se han desarrollado muchas propuestas, que varían de acuerdo a las características del problema a resolver, pero que se encuadran en dos estrategias principales: *versionado (snapshots)* o *etiquetado (timestamp)*. A continuación se describe cada una de ellas y posteriormente se identifican trabajos de referencia de diferentes autores y se los clasifica de acuerdo a la

²Protocol and RDF Query Language (SPARQL) <https://www.w3.org/TR/rdf-sparql-query/>

³ *BiGe-Onto*: <http://www.w3id.org/cenpat-gilia/bigeonto>

⁴Centro Científico Tecnológico CENPAT Consejo Nacional de Investigaciones Científicas y Técnicas (CONICET) - Puerto Madryn - Argentina <http://www.cenpat-conicet.gob.ar/>

estrategia de modelado utilizada, confiando que esta organización será orientadora para la tarea de resolución de problemas particulares sobre grafos con temporalidad.

a) *Versionado*: Es un modelo basado en secuencia de *snapshots* que reflejan el estado del grafo en instantes precisos del tiempo. El modelo define $G_t=(V_t, E_t)$ como el conjunto de arcos y nodos vigentes en el instante t , como se muestra en Fig. 1. De esta manera, el grafo subyacente se obtiene como: $G = G_1 \cup G_2 \cup \dots \cup G_n$.

Se denomina $\Delta t_{i,i+1}$ al período de tiempo entre el G_i y G_{i+1} . El modelo supone que en esos períodos no hay cambios en el grafo o bien, son despreciables.

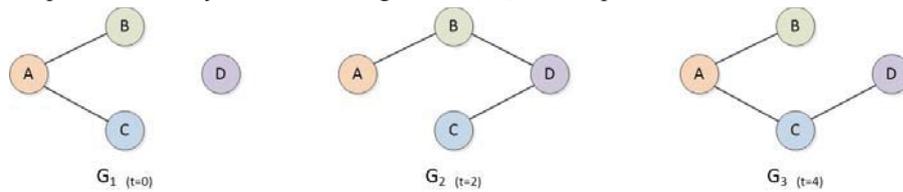


Figura 1: Grafo con temporalidad aplicando el mecanismo de *versionado*.

Este tipo de modelo es apropiado en sistemas con naturaleza de cambio periódico, en los cuales lo más habitual es trabajar sobre el snapshot más reciente, o bien sobre una ventana de tiempo de unos pocos snapshots, siendo eventual la necesidad de acudir a snapshots lejanos en el pasado.

b) *Etiquetado*: Los modelos de versionados no son adecuados cuando son necesarias evaluaciones históricas sobre toda la línea de tiempo, por ejemplo “¿Cuántas veces y durante cuánto tiempo estuvieron conectados el objeto x y el objeto y en el sistema?”. Este tipo de problemas requieren tener presente el grafo subyacente con todos los intervalos de tiempo en los que los arcos y los nodos estuvieron (o estarán) vigentes en el sistema. Para estos casos, los trabajos analizados proponen grafos *etiquetados*. Un grafo etiquetado se define como $G=(V, E, \mathcal{L})$, donde \mathcal{L} es el conjunto de etiquetas. En particular, en un grafo con tiempo, llamaremos \mathcal{LT} al conjunto de etiquetas que rotulan una referencia o restricción temporal (instante, intervalo o duración). De esta forma, el conjunto $E \subseteq (V \times V \times \mathcal{LT})$ y $V \subseteq (V \times \mathcal{LT})$. Dentro de estos modelos se detectan dos mecanismos principales: Etiquetado para definir *secuencias de contactos* entre nodos y Etiquetado para definir *intervalos de validez* de arcos y nodos.

b.1) *Etiquetado para secuencia de contactos*: Los arcos del grafo temporal se definen como tripletas $e_{ij}(v_i, v_j, t)$ la cual denota que el contacto entre el nodo v_i y el nodo v_j se realizó en el instante t . Como lo muestra la Fig. 2.a).

Este tipo de modelo es apropiado en casos en los que solo importa reflejar el instante de contacto entre nodos, aplicable por ej. en comunicaciones o cuando se necesario representar series de tiempo.

b.2) *Etiquetado para Intervalos de validez*: Los arcos del grafo temporal se definen como tripletas $e_{ij}(v_i, v_j, [t_s, t_e))$ la cual denota que el arco entre el nodo v_i y el nodo v_j tiene vigencia sólo en el intervalo $[t_s, t_e)$ donde t_s es el instante de inicio y t_e es el fin abierto, dado que en ese instante comienza un nuevo intervalo de validez. Ver Fig. 2.b).



Figura 2: a) Etiquetado con secuencia de contactos. b) Etiquetado con Intervalos de validez.

2.1. Modelos de referencia para Grafos Temporales

Entre los modelos de referencia que han propuesto mecanismo de *versionado* podemos citar a [7] donde además se definen cálculos de métricas temporales sobre grafos, como el *camino temporal* entre dos nodos. $Wt_{[k,k+L]}(i,j)$ es el camino entre i y j en el período definido por los snapshots $(k,k+L)$, considerando que $Wt_k(v_1,v_2)$ es el camino entre los nodos v_1 y v_2 en el snapshot G_{t_k} , el camino temporal entre i y j se define como $Wt_k(i,x_1), Wt_{k+1}(x_1,x_2), \dots, Wt_{k+L}(x_p,j)$.

En [8] se propone un mecanismo de *versionado* para modelar la actividad de la red neuronal del cerebro y propone una implementación basada en matrices de contacto y matrices de distancia para cada snapshot G_r . Por otra parte en [9] se propone el modelado con versiones en donde suma arcos de *espera* entre los nodos de diferentes versiones y define métricas.

Entre los modelos de referencia que han propuesto mecanismos de *etiquetado* está [10] donde se presenta un modelo de *secuencia de contactos* en el cual los arcos del grafo temporal se definen como tripletas $e_{ij}(v_i, v_j, t)$ que indican que el contacto entre el nodo v_i y el nodo v_j se realizó en el instante t . En [11] se presenta un modelo basado en *intervalos de validez*, donde la función *presencia* (ρ) puede ser aplicada tanto a arcos como a nodos y se trata de una función binaria, que retorna 1 si un determinado arco está presente en un determinado momento del tiempo y una función de latencia (ζ) que indica el tiempo que permanecen conectados. Considerando T como el tiempo de vida del sistema, define el grafo temporal G como: $G = (V, E, T, \rho, \zeta)$, donde la función *presencia* sobre un determinado arco e será verdadera en $[t, t+\zeta(e)]$.

2.2. Modelos de referencia para RDF Temporal

Dentro de los trabajos mayormente citados en RDF Temporal podemos mencionar [1] y [12], quienes han desarrollado modelos de *etiquetado*, estos definen la problemática del tiempo en RDF y proponen un modelo de tripletas $(s,p,o)[t]$ donde t es un *timestamp*. En [13] extienden su propuesta a tripletas $(s,p,o)[i]$ donde i es un *intervalo de validez* y definen relaciones básicas entre intervalos de tiempo como *meets*, *before*, *overlaps*, etc, que permiten realizar inferencia temporal. En [14] se propone un mecanismo para optimizar consultas con SPARQL sobre un modelo de *versionado*

temporal. Un abordaje similar se desarrolla en [15] con el objetivo de optimizar consultas y almacenamiento. En [16] los autores proponen un mecanismo de *etiquetado de intervalos de validez* de los arcos, mediante tripletas $(s,p:a,o)$ donde $a \in A$ es el conjunto de todos los intervalos de tiempo $[t_s, t_e]$. Los autores definen la *transitividad temporal* y algoritmos con evaluaciones experimentales.

Hay trabajos planteados para aplicaciones concretas, por ejemplo en [17] con un *modelo de secuencia de contactos* aplicado al streaming de lecturas de sensores, como series de tiempo. En este trabajo asignan un timestamp a cada frame del streaming $r[t]$, donde $t \in T$ es un espacio de tiempo discreto. Otro caso aplicado es el de [18], un *modelo de intervalos de validez* sobre una base de datos histórica de empleados, con entidades temporales sobre las que se definen consultas SPARQL y se aplican filtros por solapamiento de intervalos de tiempo de validez.

La clasificación de trabajos que se ha realizado en esta sección permite orientar la tarea de identificar la estrategia de modelado temporal más conveniente al momento de resolver un problema concreto, tanto sobre Grafos Temporales como RDF Temporal. Para el caso de uso que desarrolla este trabajo en la Sección 4, se ha utilizado un *modelo de etiquetado de intervalos de validez y secuencias de contacto* sobre una base de datos RDF.

3. Grafos RDF y semántica temporal en los datos

En sentido amplio, RDF es una herramienta conceptual que permite modelado de datos y metadatos en la web. RDF no es un grafo clásico, sino un hipergrafo dirigido, dado que sus arcos (s,p,o) pueden ser ternarios, permitiendo que el predicado (p) de una tripleta puede ser sujeto u objeto en otra, como es el caso del *SubPropertyOf*⁵ de la cual se muestra su uso en el caso de uso de la Fig. 3.

Un modelo de datos implementado en RDF, contiene al esquema y a las instancias de datos en el mismo grafo. Esto es posible mediante el uso del RDF Schema (RDFS), que permite definir el esquema de los datos en base a jerarquías de Clases/Subclases y Propiedades/Subpropiedades. Este andamiaje permite extender los modelos incorporando semántica sobre los datos y definir ontologías.

Una ontología [19] es un conjunto de conceptos relacionados que da significado a un hecho o contexto de la vida real. En el caso particular de bases de datos temporales, es propio hablar de una ontología de conceptos temporales que defina propiedades de los recursos que dependen del tiempo. Un ejemplo es *OWL-Time*⁶ definida por la W3C, la cual provee un amplio vocabulario que permite expresar hechos basados en secuencias de instantes, intervalos o duraciones. Esta ontología, además define propiedades que describen relaciones entre intervalos (*overlaps*, *meets*, *before*, *after*, etc.), duraciones, calendarios y diferentes granularidades y descripciones de tiempo. Esta forma de

⁵ Sub-Property: https://www.w3.org/TR/rdf-schema/#ch_subpropertyof

⁶ OWL-Time: <https://www.w3.org/TR/owl-time/>

representar el conocimiento agrega la posibilidad de definir una lógica temporal para luego realizar inferencias [20] y manejar información con cierto grado de incertidumbre, por ejemplo, intervalos o periodos de tiempo de los cuales se desconocen las duraciones exactas o los instantes concretos de inicio y fin.

4. Una Aplicación sobre Datos Abiertos Enlazados

En esta sección se presenta un caso de uso que reúne ideas vertidas en este trabajo y muestra la implementación de una ontología temporal basada en OWL-Time. La experiencia se aplica sobre una base de datos espacio-temporal real con 190 mil tripletas RDF, que registra ocurrencias de observaciones de peces recolectadas en campañas exploratorias sobre el Atlántico Sur Occidental por investigadores de CONICET-CENPAT y almacenadas en un triple-store RDF sobre *GraphDB*⁷.

Este repositorio forma parte del Sistema Nacional de Datos del Mar, dentro del programa de Grandes Instrumentos y Bases de Datos de Argentina y fue diseñado para publicar Datos Abiertos Enlazados (*LOD*) con el objetivo de ofrecer acceso abierto a los datos primarios de investigación. Los LOD [6] son un camino estándar hacia la integración de datos abiertos heterogéneos y de diferentes dominios [21]. Están basados en RDF y permiten interconectar datos en la web, pues son comprensibles por las máquinas, autodescriptos y con significado explícitamente definido.

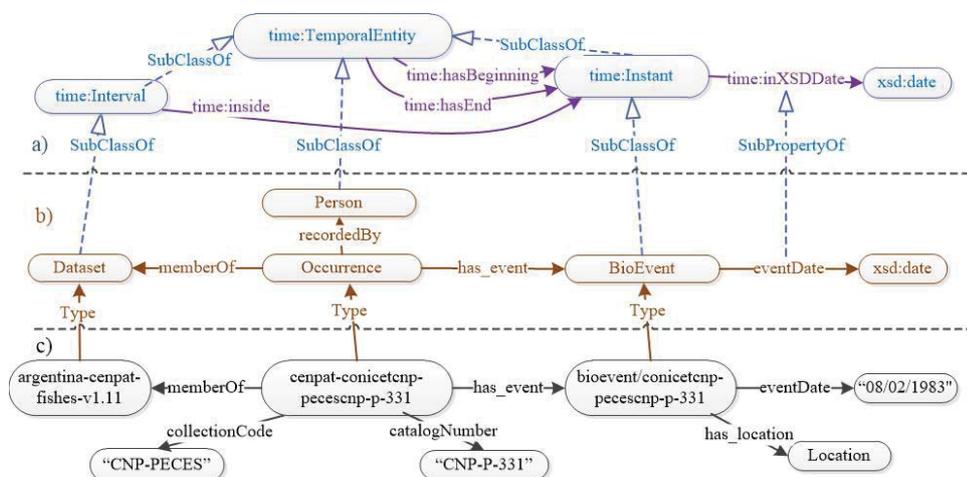


Figura 3: a) Implementación de OWL-Time para extender *BiGe-Onto*. b) Fragmento de la Ontología *BiGe-Onto* actual. c) Instancia actual de observaciones en tripletas RDF.

⁷ GraphDB: <http://graphdb.ontotext.com/>

El modelo sobre el que se realizó la experiencia responde a la ontología *BiGe-Onto* en la que se definen ocurrencias (*Occurrences*) de observaciones de peces recolectadas por un investigador (*Person*) y catalogadas en colecciones (*Dataset*). Cada evento de observación (*BioEvent*) se realiza sobre un determinado punto geográfico (*Location*) en una fecha determinada. En la *Fig. 3.b*) se muestran algunas de las clases *BiGe-Onto* y en la *Fig 3.c*) se muestra un ejemplo de la instancia de base de datos RDF, que en el repositorio utilizado cuenta con 5426 observaciones de peces.

En este trabajo, se incorporó temporalidad a dos entidades del modelo: *Person* y *Dataset* en base a clases y propiedades tomadas de OWL-Time, como se muestra en la *Fig. 3 a)*, generalizando también la entidad *BioEvent*, la cual ya tenía características temporales. Es importante destacar que esta extensión no resultó “traumática” en términos de reingeniería de modelo y ha aportado potencialidades de análisis temporal, como se muestra al final de esta sección.

A continuación se muestran las extensiones realizadas sobre el modelo actual:

- a) Se implementan las clases *TemporalEntity*, *Instant* e *Interval* y las propiedades *hasBeginning*, *hasEnd*, *inside* e *inXSDDate*, de acuerdo a la definición *OWL-Time*⁸.
- b) Se extiende a *Person* como una entidad temporal (*TemporalEntity*), utilizando el mecanismo de *etiquetado de intervalo de validez*. Esto permite establecer el período de permanencia de cada investigador dentro del sistema. Registrando su fecha de ingreso (*hasBeginning*) y de egreso (*hasEnd*). De la siguiente manera:

```
foaf:Person rdf:type owl:Class rdfs:subClassOf time:TemporalEntity
```

- c) Se extiende a *Dataset* como una entidad tipo *Interval*. Esto permite conocer el intervalo de tiempo que abarca cada colección. Si bien se aplica el mecanismo de *etiquetado de intervalos de validez*, tiene una característica especial, el dataset no incluye intervalo propio con fecha de inicio y de fin claros, sino que se define dinámicamente, a medida que se incorporan observaciones al dataset. La ontología permite definir este tipo de intervalos como una secuencia de instantes (*time:inside*).

```
void:Dataset rdf:type owl:Class rdfs:subClassOf time:Interval
```

- d) Se generaliza *BioEvent* y su propiedad *eventDate* como *Instant*. Esto no extiende funcionalidad pero incorpora a los eventos de observaciones dentro de la ontología. El mecanismo que se aplica es *etiquetado de secuencia de contactos*

```
bigeonto:BioEvent rdf:type owl:Class rdfs:subClassOf time:Instant;
                  rdfs:subClassOf dwc:Event.
dwc:eventDate rdf:type owl:DatatypeProperty;
              rdfs:subPropertyOf time:inXSDDate;
              rdfs:domain bigeonto:BioEvent rdfs:range xsd:dateTime
```

Para mostrar un resultado con la ontología implementada, se plantea la siguiente consulta: “*Qué hechos sucedieron entre 1957 y 1966?*”. La consulta en SPARQL es:

⁸ Términos del Namespace *time*: en OWL-Time <https://www.w3.org/2006/time>

```

SELECT (COUNT(?s) as ?sucesos) ?entidad ?propiedad
WHERE {
  ?o a time:Instant.
  ?s ?propiedad ?o .
  ?s a ?entidad .
  ?entidad rdfs:subClassOf time:TemporalEntity .
  ?o time:inXSDDate ?date.
  BIND (substr(xsd:string(?date),1,4) as ?daten) .
  FILTER(?daten >= "1957" && ?daten <= "1966") }
GROUP BY ?entidad ?propiedad

```

El resultado arrojado es “188 observaciones de peces, 53 registros en Datasets y 2 ingresos de nuevos científicos investigadores al staff”. Esta consulta se puede ejecutar accediendo desde la interfaz de *GraphDB* mediante el siguiente link⁹, con las credenciales: `usr:guest pass:cd.lod` y seleccionando el repositorio “*BiGe-Test*” desde el menú “*choose repository*”.

De esta manera se demuestra la utilidad de la aplicación de la ontología y las capacidades de consulta semántica. Esta característica es sumamente importante en áreas de estudio como la biodiversidad dado que ayuda a los investigadores a comprender fenómenos de conservación, como la distribución de especies en periodos de tiempo.

5. Conclusiones

La mayoría de los problemas pueden modelarse con grafos y entre ellos los que requieren contemplar la dimensión temporal, como es el caso de las bases de datos temporales. Hay destacados trabajos de referencia en el área de *Grafos Temporales* y en el área de *RDF Temporal*. Este trabajo muestra que la problemática de incorporar la dimensión temporal ha sido abordada con estrategias comparables en ambas áreas y aporta una clasificación de estrategias de modelado mayormente utilizadas y usos sugeridos, como soporte a la resolución de problemas de grafos con temporalidad.

Teniendo en cuenta que la tendencia de las bases de datos de grafos se inclina a grafos RDF, este trabajo profundiza sobre sus ventajas para la incorporación de semántica y contribuye con un aporte aplicado sobre un caso real, en un repositorio de *Datos Abiertos Enlazados* del Sistema Nacional de Datos del Mar. Este desarrollo ha permitido mostrar que identificando las entidades temporales del modelo es posible extenderlo, sin necesidad de una reingeniería costosa y que los resultados obtenidos aportan al descubrimiento de información por parte de los investigadores que estudian fenómenos de la biodiversidad en una línea de tiempo.

Los trabajos futuros se orientarán, en lo que refiere a teoría de grafos, a analizar un posible homomorfismo entre Grafos Temporales y RDF Temporal y en cuanto a desarrollo con ontologías de tiempo, a incorporar descripciones cualitativas sobre la información temporal para luego aplicar reglas de inferencia, como así también otros elementos, que permitan el razonamiento temporal sobre el modelo.

⁹ Consulta SPARQL: <http://web.cenpat-conicet.gob.ar:7200/sparql?savedQueryName=sucesos-1966&execute>

6. Referencias

- [1] C. Gutierrez, C. Hurtado, and A. Vaisman, “Temporal RDF,” *Lecture Notes in Computer Science*. pp. 93–107, 2005.
- [2] C. J. Date, H. Darwen, and N. Lorentzos, *Time and Relational Theory: Temporal Databases in the Relational Model and SQL*. Morgan Kaufmann, 2014.
- [3] R. Angles and C. Gutierrez, “Survey of graph database models,” *ACM Computing Surveys*, vol. 40, no. 1. pp. 1–39, 2008.
- [4] T. Berners-Lee, J. Hendler, and O. Lassila, “The Semantic Web,” *Scientific American*, vol. 284, no. 5. pp. 34–43, 2001.
- [5] C. Lozares, “La teoría de redes sociales,” *Papers. Revista de Sociología*, vol. 48. p. 103, 1996.
- [6] T. Heath and C. Bizer, *Linked Data: Evolving the Web Into a Global Data Space*. Morgan & Claypool Publishers, 2011.
- [7] V. Nicosia, J. Tang, C. Mascolo, M. Musolesi, G. Russo, and V. Latora, “Graph Metrics for Temporal Networks,” *Understanding Complex Systems*. pp. 15–40, 2013.
- [8] M. J. Williams and M. Musolesi, “Spatio-temporal networks: reachability, centrality and robustness,” *R Soc Open Sci*, vol. 3, no. 6, p. 160196, Jun. 2016.
- [9] V. Kostakos, “Temporal graphs,” *Physica A: Statistical Mechanics and its Applications*, vol. 388, no. 6. pp. 1007–1023, 2009.
- [10] P. Holme and J. Saramäki, “Temporal Networks as a Modeling Framework,” *Understanding Complex Systems*. pp. 1–14, 2013.
- [11] A. Casteigts, P. Flocchini, W. Quattrociocchi, and N. Santoro, “Time-Varying Graphs and Dynamic Networks,” *Ad-hoc, Mobile, and Wireless Networks*. pp. 346–359, 2011.
- [12] C. Gutierrez, C. Hurtado, and A. Vaisman, “Introducing Time into RDF,” *IEEE Transactions on Knowledge and Data Engineering*, vol. 19, no. 2. pp. 207–218, 2007.
- [13] C. Hurtado and A. Vaisman, “Reasoning with Temporal Constraints in RDF,” *Principles and Practice of Semantic Web Reasoning*. pp. 164–178, 2006.
- [14] J. Tappolet and A. Bernstein, “Applied Temporal RDF: Efficient Temporal Querying of RDF Data with SPARQL,” *Lecture Notes in Computer Science*. pp. 308–322, 2009.
- [15] J. D. Fernández, J. Umbrich, A. Polleres, and M. Knuth, “Evaluating Query and Storage Strategies for RDF Archives,” *Proceedings of the 12th International Conference on Semantic Systems - SEMANTiCS 2016*. 2016.
- [16] O. Udrea, D. R. Recupero, and V. S. Subrahmanian, “Annotated RDF,” *ACM Transactions on Computational Logic*, vol. 11, no. 2. pp. 1–41, 2010.
- [17] A. Rodriguez, R. McGrath, Y. Liu and J. Myers, “Semantic management of streaming data,” *Proceedings of the 2nd International Conference on Semantic Sensor Networks*, vol. 522, no. CEUR-WS. org, pp. 80–95, 2009.
- [18] Mcbride and Butler, “Representing and querying historical information in RDF with application to E-discovery,” *HP Laboratories Technical Report*. 2009.
- [19] T. Gruber, “Ontology,” in *Encyclopedia of Database Systems*, L. Liu and M. T. Özsu, Eds. Boston, MA: Springer US, 2009, pp. 1963–1965.
- [20] S. Batsakis, E. Petrakis, I. Tachmazidis, and G. Antoniou, “Temporal representation and reasoning in OWL 2,” *Semantic Web*, vol. 8, no. IOS Press, pp. 981–1000, 2017.
- [21] K. Janowicz, P. Hitzler, B. Adams, D. Kolas, I. I. Vardeman, and Others, “Five stars of linked data vocabulary use,” *Semantic Web*, vol. 5, no. 3, pp. 173–176, 2014.

***k*-TVT: a flexible and effective method for early depression detection**

Leticia C. Cagnina^{1,2}, Marcelo L. Errecalde¹, Ma. José Garciarena Ucelay¹,
Dario G. Funez¹, Ma. Paula Villegas^{1,2}

¹ Laboratorio de Investigación y Desarrollo en Inteligencia Computacional
Universidad Nacional de San Luis (UNSL)
Ejército de los Andes 950, (5700) San Luis, Argentina

² Consejo Nacional de Investigaciones Científicas y Técnicas (CONICET), Argentina
{lcagnina, merreca, mjgarciaarenaucelay, funezdario, villegasmariapaula74}@gmail.com

Abstract. The increasing use of social media allows the extraction of valuable information to early prevent some risks. Such is the case of the use of blogs to early detect people with signs of depression. In order to address this problem, we describe *k*-temporal variation of terms (*k*-TVT), a method which uses the variation of vocabulary along the different time steps as concept space to represent the documents. An interesting particularity of this approach is the possibility of setting a parameter (the *k* value) depending on the urgency (earliness) level required to detect the risky (depressed) cases. Results on the early detection of depression data set from eRisk 2017 seem to confirm the robustness of *k*-TVT for different urgency levels using SVM as classifier. Besides, some recent results on an extension of this collection would confirm the effectiveness of *k*-TVT as one of the state-of-the-art methods for early depression detection.

Keywords: Early Risk Prediction, Early Depression Detection, Text Representation, Semantic Analysis Techniques, Temporal Variation of Terms.

1 Introduction

Early risk detection (ERD) on the Internet is an important research area due to the impact it might have in fields like health when people suffer depression, anorexia or other disorders that can threaten life. In the field of security, there is a latent risk when criminals and sex offenders try to attack using web technologies. In this context, detection of depression is a major public health concern and a leading cause of disability. It is clear that the development of computational tools that help detecting depressed people, whether they are diagnosed as depressive or not, is becoming a very relevant task as demonstrated by the increase in publications in the specialized literature [1-4]. In particular, a scenario that has started to receive more attention is the one referred as early depression detection (EDD), that is, detecting depressive persons *as soon and accurate as possible* [5]. However, EDD, like other ERD tasks, usually presents several challenging aspects to the standard machine learning field: 1)

unbalanced data sets, 2) classification with partial information and, 3) the classification time decision.

An additional difficulty in research on EDD is the scarcity of resources (data sets) publicly available for experimentation and development of automatic detection systems. For this reason, the primary objective in [6] was to provide the first collection to study the relationship between depression and language usage by means of machine learning techniques. Another important contribution of that work is the proposal of a new error measure called Early Risk Detection Error ($ERDE_{\sigma}$) that simultaneously evaluates the accuracy of the classifiers and the delay in making a prediction. The σ parameter serves as the deadline for decision making, i.e. if a correct positive decision is made in time $k > \sigma$ it will be evaluated by $ERDE_{\sigma}$ as if it were incorrect (false positive). In that way, σ parameter allows to specify the urgency (earliness) level required for a task, that is to say, the lower the σ value the sooner a depressed user needs to be detected.

Beyond the effectiveness of TVT [7] in the first EDD pilot task, we found several limitations. First of all, the “heuristic” value of 4 “chunks” (short pieces of text) was completely empiric, that is, it mainly addressed the problem of balancing the minority class. However, no effort was dedicated to analyze what would be the impact of varying the number of chunks or determining what is the relation between the used number of chunks and the urgency level specified by the σ value. Besides, although different algorithms (like SVM, Random Forest, Multinomial Naive Bayes) were tested and different parameters were considered (like the probability threshold τ), no guidance was provided to select an adequate TVT configuration that produces acceptable results for the EDD task with different requirements in the earliness level.

The present work addresses the above mentioned drawbacks by presenting k -TVT, a generalization of the TVT method that allows to vary the number of chunks k considered for the minority class. Depending on the urgency (earliness) required in a particular scenario (specified by the σ parameter), it is possible to select for k -TVT a proper number of chunks k that obtains acceptable $ERDE_{\sigma}$ values. Furthermore, we provide some guidance about which could be robust learning algorithms and appropriate thresholds τ to be combined with the specific selected k value.

In order to fulfill these objectives, our study should be able to answer the following research questions:

- RQ1: Are the number of chunks k and the level of urgency σ related? If this is the case, which would be the correct values of k for the different σ values?
- RQ2: Once the number of chunks k has been selected, what is the influence of the probability threshold τ used to classify a user as depressed? Does it vary depending on the degree of earliness determined by σ ?
- RQ3: Are the results obtained with those configurations of k -TVT comparable with other state-of-the-art methods?

Hence, in Section 2 we describe k -TVT, the method proposed in this article for EDD. Section 3 describes the data set used in our experiments. In Section 4, different studies are carried out to answer the research questions mentioned above. Finally, Section 5 summarizes the main conclusions obtained and possible future works are suggested.

2 The Proposed Method

Our method is based on the *concise semantic analysis* (CSA) technique proposed in [8] and later extended in [9] for author profiling tasks. Therefore, we first present in Section 2.1 the key aspects of CSA and then explain in Section 2.2 how we instantiate CSA with concepts derived from the terms used in the temporal chunks analyzed by an ERD system at different time steps.

2.1 Concise Semantic Analysis

Standard text representation methods such as Bag of Words (BoW) suffer of two well-known drawbacks. First, their high dimensionality and sparsity; second, they do not capture relationships among words. CSA is a semantic analysis technique that aims at dealing with those shortcomings by interpreting words and documents in a space of *concepts*. Differently from other semantic analysis approaches such as *latent semantic analysis* (LSA) [10] and *explicit semantic analysis* (ESA) [11] which usually require huge computing costs, CSA interprets words and text fragments in a space of concepts that are close (or equal) to the category labels. For instance, if documents in the data set are labeled with q different category labels (usually no more than 100 elements), words and documents will be represented in a q -dimensional space. That space size is usually much smaller than standard BoW representations which directly depend on the vocabulary size (more than 10000 or 20000 elements in general).

To explain the main concepts of the CSA technique we first introduce some basic notation that will be used in the rest of this work.

Let $D = \{\langle d_1, y_1 \rangle, \dots, \langle d_m, y_m \rangle\}$ be a training set formed by n pairs of documents (d_i) and variables (y_i) that indicate the concept the document is associated with, $y_i \in C$ where $C = \{c_1, \dots, c_q\}$ is the *concept space*. For the moment, consider that these concepts correspond to standard category labels although, as we will see later, they might represent more elaborate aspects. In this context, we will denote as $V = \{t_1, \dots, t_v\}$ to the vocabulary of terms of the collection being analyzed.

2.1.1. Representing Terms in the Concept Space

In CSA, each term $t_i \in V$ is represented as a vector $t_i \in \mathbb{R}^q$, $t_i = \langle t_{i,1}, \dots, t_{i,q} \rangle$. Here, $t_{i,j}$ represents the degree of association between the term t_i and the concept c_j and its computation requires some basic steps that are explained below. First, the raw term-concept association between the i th term and the j th concept, denoted w_{ij} , will be obtained. If $D_{c_u} \subseteq D$, $D_{c_u} = \{d_r \mid \langle d_r, y_s \rangle \in D \wedge y_s = c_u\}$ is the subset of the training instances whose label is the concept c_u , then w_{ij} might be defined as Equation 1 shows.

$$w_{ij} = \sum_{\forall d_m \in D_{c_j}} \log_2 \left(1 + \frac{tf_{im}}{\text{len}(d_m)} \right) \quad (1)$$

where tf_{im} is the number of occurrences of the term t_i in the document d_m and $len(d_m)$ is the length (number of terms) of d_m .

As noted in [8] and [9], direct use of w_{ij} to represent terms in the vector \mathbf{t}_i could be sensible to highly unbalanced data. Thus, some kind of normalization is usually required and, in our case, we selected the one proposed in [9]:

$$t'_{ij} = \frac{w_{ij}}{\sum_{i=1}^{|V|} w_{ij}} \quad (2) \quad t_{ij} = \frac{t'_{ij}}{\sum_{j=1}^{|C|} w_{ij}} \quad (3)$$

where Equation 2 normalizes weights in proportion to the $|V|$ terms of each class and Equation 3 normalizes term weights in order to make them comparable among the $|C| = q$ categories/concepts. With this last conversion we finally obtain, for each term $t_i \in V$, a q -dimensional vector \mathbf{t}_i , $\mathbf{t}_i = \langle t_{i,1}, \dots, t_{i,q} \rangle$ defined over a space of q concepts. Up to now, those concepts correspond to the original categories used to label the documents. Later, we will use other more elaborated concepts.

2.1.2. Representing Documents in the Concept Space

Once the terms are represented in the q -dimensional concept space, those vectors can be used to represent documents in the same concept space. In CSA, documents are represented as the central vector of all the term vectors they contain [8]. Terms have distinct importance for different documents so it is not a good idea computing that vector for the document as the simple average of all its term vectors. A previous work in BoW ([12]) has considered different statistic techniques to weight the importance of terms in a document such as $tfidf$, $tfig$, $tf\chi^2$ or tf/f , among others. Here, we will use the approach used in [9] for author profiling that represents each document d_m as the weighted aggregation of the representations (vectors) of terms that it contains (see Equation 4).

$$\mathbf{d}_m = \sum_{t_i \in d_m} \left(\frac{tf_{im}}{len(d_m)} \times \mathbf{t}_i \right) \quad (4)$$

Thus, documents are also represented in a q -dimensional concept space (i.e., $\mathbf{d}_m \in \mathbb{R}^q$) which is much smaller in dimensionality than the one required by standard BoW approaches ($q \ll v$).

2.2 k -Temporal Variation of Terms

In Subsection 2.1 we said that the concept space C usually corresponds to standard category names used to label the training instances in supervised text categorization tasks. In this scenario, which in [8] is referred as direct derivation, each category label simply corresponds to a concept. However, in [8] also are proposed other alternatives like *split derivation* and *combined derivation*. The former uses the low-level labels in hierarchical corpora and the latter is based on combining semantically related labels in

a unique concept. In [9] those ideas are extended by first clustering each category of the corpora and then using those subgroups (sub-clusters) as new concept space.¹

As we can see, the common idea to all the above approaches is that once a set of documents is identified as belonging to a group/category, that category can be considered as a concept and CSA can be applied in the usual way. We take a similar view to those works by considering that the positive (minority) class in ERD problems can be augmented with the concepts derived from the sets of partial documents read along the different time steps. In order to understand this idea, it is necessary to first introduce a sequential work scheme as the one proposed in [6] for research in ERD systems for depression cases.

Following [6], we will assume a corpus of documents written by p different individuals ($\{I_1, \dots, I_p\}$). For each individual I_l ($l \in \{1, \dots, p\}$), the n_l documents that he has written are provided in chronological order (from the oldest text to the most recent one): $D_{I_l,1}, D_{I_l,2}, \dots, D_{I_l,n_l}$. In this context, given these p streams of messages, the ERD system has to process every sequence of messages (in the chronological order they are produced) and has to make a binary decision (as early as possible) on whether or not the individual might be a positive case of depression. Evaluation metrics on this task must be time-aware, so an early risk detection error (ERDE) is proposed. This metric not only takes into account the correctness of the (binary) decision but also the delay taken by the system to make the decision.

In a usual supervised text categorization task, we would only have two category labels: *positive* (risk/depressive case) and *negative* (non-risk/non-depressive case). That would only give two concepts for a CSA representation. However, in ERD problems there is additional temporal information that could be used to obtain an improved concept space. For instance, the training set could be split in h “chunks”, $\hat{C}_1, \hat{C}_2, \dots, \hat{C}_h$, in such a way that \hat{C}_1 contains the oldest writings of all users (first $(100/h)\%$ of submitted posts or comments), chunk \hat{C}_2 contains the second oldest writings, and so forth. Each chunk \hat{C}_m can be partitioned in two subsets \hat{C}_m^+ and \hat{C}_m^- , $\hat{C}_m = \hat{C}_m^+ \cup \hat{C}_m^-$ where \hat{C}_m^+ contains the positive cases of chunk \hat{C}_m y \hat{C}_m^- the negatives ones of this chunk.

It is interesting to note that we can also consider the data sets that result of concatenating chunks that are contiguous in time and using the notation \hat{C}_{i-j} to refer to the chunk obtained from concatenating all the (original) chunks from the i th chunk to the j th chunk (inclusive). Thus, \hat{C}_{1-h} will represent the data set with the complete stream of messages of all the p individuals. In this case, \hat{C}_{1-h}^+ and \hat{C}_{1-h}^- will have the obvious semantic specified above for the complete documents of the training set.

The classic way of constructing a classifier would be to take the complete documents of the p individuals (\hat{C}_{1-h}) and use an inductive learning algorithm such as SVM to obtain that classifier. As we mentioned earlier, another important aspect in EDS systems is that the classification problem being addressed is usually highly unbalanced. That is, the number of documents of the majority/negative class (“non-depression”) is significantly larger than of the minority/positive class (depression). More formally, following the previously specified notation $|\hat{C}_{1-h}^-| \gg |\hat{C}_{1-h}^+|$.

¹ In that work, concepts are referred as profiles and subgroups as sub-profiles.

An alternative to balance the classes would be to consider that the minority class is formed not only by the complete documents of the individuals but also by the partial documents obtained in the different chunks. Following the general ideas posed in CSA, we could consider that the partial documents read in the different chunks represent “temporal” concepts that should be taken into account. In this context, one might think that variations of the terms used in these different sequential stages of the documents may have relevant information for the classification task. With this idea in mind, the method proposed in this work named *k-temporal variation of terms* (*k*-TVT) arises, which consists in enriching the documents of the minority class with the partial documents read in the first *k* chunks. These first chunks of the minority class, along with their complete documents, will be considered as a new concept space for a CSA method.

Therefore, in *k*-TVT we first determine the number *k* of initial chunks that will be used to enrich the minority (positive) class. Then, we use the document sets \hat{C}_1^+ , \hat{C}_{1-2}^+ , ..., \hat{C}_{1-k}^+ and \hat{C}_{1-h}^+ as concepts for the positive class and \hat{C}_{1-h}^- for the negative class. Finally, we represent terms as documents in this new $(k + 2)$ -dimensional space using the CSA approach explained in Section 2.1.

3 Data Set and Pilot Task

Our study was carried out on the data sets provided by *eRisk 2017 pilot task*² and *eRisk 2018 Lab*³ on *early risk prediction for depression* [5, 13]. They are collections of writings (posts) of Social Media users taken from *Reddit*. There are two categories of users: “depressed” (or positive) and “non-depressed/control” (or negative). For each user, the data set contains a sequence of writings (in chronological order) divided into 10 chunks. The first chunk contains the oldest 10% of the messages, the second chunk contains the second oldest 10%, and so forth. Table 1 summarizes, for both data sets, the number of users for each class.

Table 1. Data sets for depression task.

	Training		Test	
	Depressed	Non-Depressed	Depressed	Non-Depressed
eRisk 2017	83	403	52	349
eRisk 2018 ⁴	135	752	79	741

4 Experimental Study

We will use the data sets described previously: eRisk2017 and eRisk2018. Research questions RQ1 and RQ2, will be analyzed in Subsection 4.1. We use a cross-validation

² <http://early.irlab.org/2017/index.html>

³ <http://early.irlab.org/2018/index.html>

⁴ The eRisk 2018 training set is the join of eRisk 2017 training and test sets.

study on the whole data set eRisk2017 in order to evaluate if it is possible to select an appropriate number of chunks k to obtain good $ERDE_{\sigma}$. Also, the incidence of the probability threshold τ on the classifier's performance is deeply analyzed. In Subsection 4.2, experimental results are compared with the ones previously published in [5, 13]. Thus, research question RQ3 will be answered.

4.1 Setting the k Parameter

We present an exploratory analysis that allows us a deeper understanding of the relationship between the number of initial chunks k used by k -TVT and the urgency level specified by σ . In that way, we could give some guidance about a reasonable number k for the different thresholds σ .

To carry out this study we perform a 5-fold cross validation with different versions of k -TVT using 3 learning algorithms: Support Vector Machine (SVM), Naïve Bayes (NB) and Random Forest. The implementations of these algorithms correspond to those provided in the Python scikit-learn library with the default parameters.

The performance of the classifiers was assessed using the $ERDE_{\sigma}$ measure and the parameter σ was varied considering the values: 5, 10, 25, 50 and 75. Because σ represents some type of urgency in detecting depression cases, we want to analyze how k -TVT performs under different levels of urgency. Note that $\sigma = 5$ means a high urgency (a quick decision should be made) and $\sigma = 75$ represents the lowest urgency (there is more time to make a decision) in detecting the positive cases.

As we stated before, k -TVT defines concepts that capture the sequential aspects of the ERD problems and the variations of vocabulary observed in the distinct stages of the writings. Thus, different number k of chunks that will enrich the minority (positive) class could have an impact in the $ERDE_{\sigma}$ measure. In this study, the k value was varied in the (integer) range $[0, 5]$.

In each chunk, classifiers usually produce their predictions with some "confidence", in general, the estimated probability of the predicted class. Therefore, we can select different thresholds τ considering that an instance is assigned to the target class when its associated probability p is greater (or equal) than certain threshold τ ($p \geq \tau$). Our study considered 4 different settings for the probabilities assigned for each classifier: $p \geq 0.9$, $p \geq 0.8$, $p \geq 0.7$ and $p \geq 0.6$. Note that once a classifier determines that an instance is positive in a specific chunk, that decision remains inalterable until chunk 10. Due to space constraints, only the best results are shown (Table 2).

The performance of k -TVT was compared against those obtained with a standard bag of words (BoW) representation with different weighting schemes: boolean, term-frequency (tf) and tf -inverse document frequency ($tfidf$). The best results with BoW were achieved with $tfidf$ scheme, SVM as learning algorithm and different thresholds τ . These results were adopted as a baseline in the subsequent experiments.

Table 2 shows the best values obtained by k -TVT for the temporal-aware measure $ERDE_{\sigma}$ with the different urgency levels σ . As we can see in the first row of the table, for the lowest σ ($\sigma = 5$ and $\sigma = 10$), k -TVT obtains the best $ERDE_5$ and $ERDE_{10}$ (highlighted in boldface) with the minimum k value, that is, $k = 0$ and the SVM classifier using $p \geq 0.8$. Those two numbers are around a 5% better than the ones corresponding to the baseline (BoW-SVM, $p \geq 0.7$). However, it is interesting to notice in the same

row, the baseline is better than 0-TVT when $\sigma = 25, 50$ and 75 are used. Those results are a preliminary evidence that k -TVT performance, as measured by $ERDE_{\sigma}$, effectively depends on the selected number of chunks k . That is, 0-TVT does not seem to produce as good results for higher σ values as the ones obtained with $\sigma = 5$ and 10 . The same fact is confirmed by the best $ERDE_{25}$ and $ERDE_{50}$ obtained by k -TVT using the first 4 initial chunks ($k = 4$), SVM as learning algorithm and a lower probability, $p \geq 0.7$. Finally, the best $ERDE_{75}$ was obtained with 4-TVT using Naïve Bayes with the same probability ($p \geq 0.7$). Here, the lowest $ERDE_{75}$ indicates that 4-TVT-NB outperformed the baseline in almost 1 unit which constitutes a good value.

Table 2. Best results of 5-fold cross validation on the whole eRisk2017 data set.

		ERDE					
		τ	5	10	25	50	75
Best $ERDE_{5-10}$	0-TVT-SVM	0.8	13.58	12.48	12.04	11.40	11.08
	BoW-SVM	0.7	14.13	13.18	11.75	10.97	10.49
Best $ERDE_{25-50}$	4-TVT-SVM	0.7	14.10	12.51	11.00	9.57	9.17
	BoW-SVM	0.6	14.42	13.20	11.28	10.38	9.70
Best $ERDE_{75}$	4-TVT-NB	0.7	14.49	12.72	11.05	9.67	8.74
	BoW-SVM	0.6	14.42	13.20	11.28	10.38	9.70

In summary, from this study we can conclude that when there is a high urgency level (low σ values) in detecting depression cases, the best performance is obtained with 0-TVT. As we decrease the level of urgency in the detection ($\sigma \geq 25$), 4-TVT performs well and it can detect the positive cases with enough accuracy. It seems that while more information enriches the k -TVT representation, more confident can be the classifier, therefore better $ERDE_{\sigma}$ can be obtained. It is also worth to note that for $\sigma \in \{5, 10, 25, 50\}$ SVM classifier obtained the best results demonstrating thus enough robustness. Using k -TVT the classifiers obtained the predictions with highest probability: 0.8 and 0.7 while if BoW is used, the threshold is lower (around 0.7 and 0.6).

Since SVM performs well in most of the cases, we can suggest it as an acceptable algorithm to be combined with generic k -TVT and it will be used in next subsection.

4.2 Performance of k -TVT - eRisk's Train and Testing Sets

Here we analyze our approach against some of the state-of-the-art methods in order to answer RQ3. In this way, our k -TVT results are directly compared with those obtained by the different groups participating in the tasks and published in [5,13]. Thus, we reproduce the same conditions faced by the participants: we first work on the data set released on the training stage for obtaining the models and then, these are tested on the test sets.

Table 3 shows the values obtained with k -TVT and SVM considering $p \geq 0.7$ (4-TVT) and $p \geq 0.8$ (0-TVT). The best values reported until now are also shown in this table. The complete description of those methods can be found in [5,13].

The results reveal several interesting aspects. First of all, we can confirm the hypothesis originated from the previous study regarding that lower k values for k -TVT

produce better $ERDE_{\sigma}$ when σ is low (high urgency level). Also, when there are low urgency levels, it is better to set k with higher values. With both k -TVT and probability thresholds ($\tau = 0.7$ and 0.8), the $ERDE_5$ measures are better than the best published for the eRisk 2017 task. For $ERDE_{50}$ the 4-TVT outperforms the best published for eRisk 2017, while 4-TVT obtains a value slightly worse in eRisk 2018, although better than 0-TVT.

Table 3. Comparison between best results in eRisk 2017 pilot task and eRisk 2018 Lab.

eRisk 2017	$ERDE_5$	$ERDE_{50}$	eRisk 2018	$ERDE_5$	$ERDE_{50}$
0-TVT	12.04	10.67	0-TVT	8.78	7.39
4-TVT	12.66	8.99	4-TVT	8.82	6.95
FHDO	12.7	10.39	FHDO-BCSGB	9.50	6.44
UNSLA ⁵	13.66	9.68	UNSLA ⁶	8.78	7.39

5 Conclusions

In this article we present *k-temporal variation of terms* (k -TVT), a flexible and effective method for early depression detection. k -TVT considers the variation of vocabulary along the different time steps as concept space for document representation. The flexibility of k -TVT is given by the possibility of setting a parameter (the k value) depending on the urgency level (the threshold σ) required to detect the risky (depressed) cases.

We obtained interesting evidence about the relationship between the k -parameter and the required level of earliness (σ threshold) in the predictions. For low σ values (high urgency) a low number of chunks ($k = 0$) is an adequate representation while for low urgency (higher σ), the use of higher value ($k = 4$) seems to be better.

Interestingly 0-TVT and 4-TVT show to be competitive (in fact better) than state of the arts methods participating in the EDD tasks. Besides, a very relevant aspect of the k -TVT representation is the complete domain independence because it is only based on the vocabulary present in the training corpus. It does not depend on features specifically derived for the depression problem or other costly hand-crafted features. Even more, the mechanism used to determine the classification time does not need to be adapted to a particular domain. This makes k -TVT suitable for implementation in other early risk tasks such as the early detection of anorexia or pedophiles without virtually any cost of migrating from one domain to another.

In that context, as future work, we plan to apply the k -TVT approach to other problems that can be directly tackled as early risk detection such as sexual predation and suicide discourse identification.

⁵ In eRisk 2017 pilot task UNSLA is an assembly of several methods which includes 4-TVT representation.

⁶ In eRisk 2018 Lab, UNSLA is the same method 0-TVT-SVM.

Acknowledgments. This work was partially funded by CONICET and Universidad Nacional de San Luis (UNSL) - Argentina.

References

1. Guntuku, S. C., Yaden, D. B., Kern, M. L., Ungar, L. H., Eichstaedt, J. C.: Detecting depression and mental illness on social media: an integrative review. In: *Current Opinion in Behavioral Sciences* 18, pp 43-49. 2017.
2. De Choudhury, M., Gamon, M., Counts, S., Horvitz, E.: Predicting depression via social media. In: *Proceedings of the Seventh International AAAI Conference on Weblogs and Social Media (ICWSM'13)*, pp 128-137. 2013.
3. De Choudhury, M., Counts, S., Horvitz, E.: Social media as a measurement tool of depression in populations. In: *Proceedings of the 5th Annual ACM Web Science Conference, ACM*, pp 47-56. 2013.
4. Park, M., McDonald, D. W., Cha, M.: Perception differences between the depressed and non-depressed users in twitter. In: *Proceedings of the Seventh International AAAI Conference on Weblogs and Social Media*, pp 476-485. 2013.
5. Losada D.E., Crestani F., Parapar J. eRISK 2017: CLEF Lab on Early Risk Prediction on the Internet: Experimental Foundations. In: Jones G. et al. (eds) *Experimental IR Meets Multilinguality, Multimodality, and Interaction. CLEF 2017. Lecture Notes in Computer Science*, vol 10456, pp 346-360. Springer, Cham. 2017.
6. Losada, D. E., Crestani, F.: A test collection for research on depression and language use. In: Fuhr N. et al. (eds) *Experimental IR Meets Multilinguality, Multimodality, and Interaction. CLEF 2016. Lecture Notes in Computer Science*, vol 9822, pp 28-39. Springer, Cham. 2016.
7. Errecalde, M. L., Villegas, M. P., Funez, D. G., Garciarena Ucelay, M. J., Cagnina, L. C.: Temporal variation of terms as concept space for early risk prediction. In: *Working Notes of CLEF 2017 - Conference and Labs of the Evaluation Forum, Vol 1866*, 2017.
8. Li, Z., Xiong, Z., Zhang, Y., Liu, C., Li, K.: Fast text categorization using concise semantic analysis, *Pattern Recognition Letters* 32 (3), pp 441-48. 2011.
9. López-Monroy, A. P., Montes y Gómez, M., Escalante, H. J., Villaseñor-Pineda, L., Stamatatos, E.: Discriminative subprofile specific representations for author profiling in social media. In: *Knowledge-Based Systems* 89, pp 134-147. 2015.
10. Deerwester, S., Dumais, S. T., Furnas, G. W., Landauer, T. K., Harshman, R.: Indexing by latent semantic analysis. In: *Journal of the ASIS* 41 (6), pp 391-407. 1990.
11. Gabrilovich, E., Markovitch, S.: Wikipedia-based semantic interpretation for natural language processing. In: *JAIR* 34 (1), pp 443-498. 2009.
12. Lan, M., Tan, C., Su, J., Lu, Y.: Supervised and traditional term weighting methods for automatic text categorization, *IEEE TPAMI* 31 (4), pp 721-735. 2009.
13. Losada, D. E., Crestani F., Parapar J. Overview of eRisk: Early Risk Prediction on the Internet. In: Bellot P. et al. (eds) *Experimental IR Meets Multilinguality, Multimodality, and Interaction. CLEF 2018. Lecture Notes in Computer Science*, vol 11018, pp 343-361. Springer, Cham. 2018.

Modelo NoSQL de Propuestas Académicas de Grado

Vizcaino Corsetti Fanny¹, Smail Ana², Pompei Sabrina³

Instituto de Investigación y Transferencia de Tecnología (ITT)⁴

Comisión de Investigaciones Científicas (CIC)

Escuela de Tecnología (ET)

Universidad Nacional del Noroeste de la Provincia de Buenos Aires (UNNOBA)

Sarmiento Nro 1119 3er Piso, Junín (B), Argentina

fbvizcainocorsetti@comunidad.unnoba.edu.ar ; {ana.smail,
sabrina.pompei}@itt.unnoba.edu.ar

Abstract. La sobre-duración de los estudios universitarios es una de las problemáticas actuales en la educación superior argentina que nos plantea la necesidad de identificar factores determinantes a fin de tomar decisiones en pos de disminuir la duración real de las carreras. Distintos factores provocan esta sobre-duración ya sea por causas extrínsecas a los estudiantes como intrínsecas a los mismos. La UNNOBA no es ajena a esta problemática nacional.

El presente artículo muestra un avance del trabajo realizado en pos de obtener un modelo NoSQL de grafos para las propuestas académicas de grado de la universidad. Esto implica el diseño de grafos tanto para los planes de estudio como para los trayectos académicos de los egresados. Se espera que este modelo permita a futuro mediante el análisis de los grafos resultantes buscar evidencias sobre problemáticas propias del currículo que inciden en la sobre-duración de las propuestas.

Keywords: base de datos noSQL, base de datos orientada a grafos, sobre-duración de propuestas académicas

1 Introducción

Actualmente los modelos relacionales luchan y se debaten en dominios altamente conectados frente a alternativas NoSQL, sistemas de bases de datos que a diferencia del modelo relacional no usan SQL como lenguaje de consultas. En el mundo relacional el incremento de relaciones y conectividad se traduce en incremento de “join” lo que reduce el rendimiento y la adaptabilidad en respuesta a los cambios de negocio. Frente a esto, la creciente demanda de análisis basado en relaciones ha puesto en escena a los

¹ Becario ITT

² Docente Investigador - ITT

³ Docente Investigador - ITT

⁴ ITT - Centro Asociado CIC

grafos como modelo para representar los datos y sus relaciones a fin de optimizar el descubrimiento de conocimiento, implementados en bases de datos NoSQL orientadas a grafos (BDOG).

Por definición un grafo es un conjunto de nodos y aristas. Cuando modelamos con grafos las entidades son representadas por los nodos, y el modo en que estas entidades se relacionan mediante las aristas. Esta estructura de propósito general nos permite modelar todo tipo de escenarios.

En las BDOG los datos conectados se almacenan como datos conectados, es decir, donde hay conexiones en el dominio, hay conexiones en los datos. Por esta razón el modelo de grafos es particularmente útil cuando los datos a almacenar tienen multitud de interrelaciones entre sí, y cuando la importancia recae más en estas interrelaciones que en los propios datos. Es el caso de nuestro objeto de estudio, dado que un plan de una propuesta de grado está determinado por las relaciones entre las actividades y conjuntos de actividades que lo componen, siendo este el trayecto académico óptimo de un egresado.

En el modelado de las bases de datos orientadas a grafos se reduce la brecha entre el modelado lógico y físico. Esta disonancia semántica entre nuestra conceptualización del mundo (modelo lógico) y la instanciación de la base de datos (modelo físico) de ese modelo, se reduce en el modelado de la BDOG. El hecho de no requerir una definición explícita del esquema de la base de datos y que las propiedades de los nodos no sean estrictas facilita la extensión ad hoc de la base de datos, de manera que nuevos tipos de nodos y relaciones, así como nuevas propiedades, pueden ser añadidas al grafo actual directamente, sin necesidad de redefinir el esquema de la base de datos.

Otra característica a destacar de los modelos de grafos, es el hecho de que los grafos no solo comunican cómo pensamos que las cosas están relacionadas, sino que también el tipo de preguntas que queremos hacerle a nuestro dominio. Las relaciones en los grafos naturalmente crean caminos por lo que consultar el grafo implica recorrer caminos. Esta naturaleza orientada a caminos del modelo de datos implica que la mayoría de las operaciones de la BDOG están altamente alineadas con el modo en que los datos se disponen, haciendo esto extremadamente eficiente. Esto incide además en la escalabilidad del sistema dado que el tiempo de ejecución de las consultas no depende directamente del tamaño del grafo sino de la porción del grafo recorrido.

2 Objetivos

El objetivo general de nuestro trabajo se centra en alcanzar un modelo de grafo aplicable a las propuestas de grado de la universidad lo que implica el diseño de grafos tanto para los planes de estudio como para los trayectos académicos de los egresados. Este modelo permitirá a futuro mediante el análisis de los grafos resultantes buscar evidencias sobre problemáticas propias del currículo que inciden en la sobre-duración de las propuestas.

Para ello se definen los siguientes objetivos específicos:

- Identificar las características del dominio que lo hacen idóneo para ser almacenado en una base de datos orientada a grafos.

- Realizar el proceso de diseño de una base de datos orientada a grafos, desde la descripción de los datos hasta la implementación, pasando por la creación del modelo conceptual de datos y el diseño lógico.
- Determinar las consultas posibles a realizar sobre el dominio de los datos que validan el modelo.

3 Metodología

Los planes de estudios académicos de una propuesta en la Universidad, tienen una estructura naturalmente orientada a grafos; donde cada actividad en el mismo será correlativa o tendrá por correlativa a otra. Así mismo dichos planes pasan por distintas versiones, de las cuales existe una o varias activas, y una única vigente.

Las actividades que forman parte del mismo tienen un nombre y una carga horaria semanal y total en el período electivo o módulo del que forman parte, además de pertenecer a un área específica de conocimiento.

Teniendo claro las características de los datos y relaciones en el dominio en cuestión, nos encontramos en condiciones de realizar el diseño conceptual, el cual será independiente de la tecnología elegida para crear la base de datos. Definimos además las consultas que nos permitirán a posterior validar la transformación al modelo lógico elegido.

Dada la naturaleza de la estructura de grafo del dominio, un diseño lógico de almacenamiento de los datos en una base de datos orientada a grafos parece lo más apropiado para lograr la abstracción del mismo.

Una vez que se optado por un modelo lógico basado en grafos, el problema es: ¿cómo modelamos el grafo?

Podemos elegir entre tres tipos de modelos de diseño lógico de una base de datos orientada a grafos:

- Modelo grafo de propiedades: formado por nodos que pueden tener aristas de entrada, salida, propiedades y etiquetas; relaciones que tienen etiqueta, son dirigidas (un nodo de origen y uno de destino) y también pueden tener propiedades. Las propiedades son pares clave-valor.
- Modelo de hipergrafos: son grafos en el que una relación (denominada hiperarco) puede conectar cualquier número de nodos dados, y pueden ser útiles cuando sus datos incluyen un gran número de relaciones de muchos a muchos.
- Modelo de tripletas: almacenan los datos en un formato conocido como triple. Los triples consisten en una estructura de datos sujeto-predicado-objeto.

En nuestro caso, hemos elegido el modelo de grafo de propiedades, por lo que para transformar el modelo conceptual a este tipo de grafo, representamos los objetos de las entidades como nodos, y etiquetamos estos nodos con los nombres de dichas entidades.

Después de obtener el grafo inicial que se adapta a nuestro modelo conceptual, procedimos a ajustar el modelo lógico añadiendo relaciones y nodos, teniendo en cuenta que este debería responder a las consultas planteadas en el dominio. Posteriormente se

procedió al diseño físico que consiste en crear el grafo planteado en un Sistema de Gestión de BDOG, en este caso, Neo4j.

Neo4j es un Sistema de Gestión de Base de Datos (DBMS) multiplataforma, puramente orientado a grafos. Es nativo en almacenamiento y procesamiento de grafos. Actualmente es el sistema más expandido y mejor valorado por la comunidad de usuarios. EBay, Wal-Mart, Cisco y Hewlett-Packard, entre otras, han utilizado las cualidades de Neo4j para mejorar sus servicios.

El lenguaje declarativo con el que Neo4j trabaja es Cypher. Es potente y fácil de usar. Está basado en SQL y proporciona una forma visual y lógica para emplear nodos y relaciones. Muestra los resultados usando ASCII-Art. Algunos puntos importantes de su sintaxis:

- Cláusula create: para crear nodos y relaciones.
- Propiedades: se indica el conjunto, y sus valores en formato JSON: {clave: valor}.
- Etiquetas: clasifican los nodos dentro de la base de datos y en las consultas. Su sintaxis: (:Etiqueta)
- Cláusula match: busca en la base de datos uno o varios nodos, usando o no relaciones. Un ejemplo de su utilización:
- MATCH (nodo) – [:RELACION] -> (listaDeNodos)
- Cláusula return: retorna, mediante el uso de variables, el gráfico que contiene los nodos que éstas indican. Si dichos nodos tienen relaciones entre ellos, también se grafican.

4 Resultados

En una primera etapa con la información extraída de la base de datos de la universidad, se diseñó un modelo conceptual orientado a la implementación de los Planes de Estudio de Grado, recorrido óptimo de un egresado. En la figura 1 podemos ver el grafo correspondiente a este modelo.

Definimos además las consultas que nos permitirán a posterior validar la transformación al modelo lógico elegido:

- El Plan de Estudio de una propuesta académica
- Actividades correlativas de una actividad en un Plan de Estudio
- Actividades optativas de una actividad genérica de un Plan de Estudio
- Actividades en un período lectivo en un Plan de Estudio

Basados en este modelo y en las opciones expresadas y analizadas, se tomó como mejor opción el diseño de un modelo lógico orientado a un grafo de propiedades.

Para su realización se debe tener en cuenta que la versión del plan de estudios no es conocida por todo el grafo, sino que debe especificarse en cada relación para que su recorrido (desde el nodo que indica el plan hacia sus actividades) pueda llevarse a cabo. En una base de datos con múltiples planes, módulos y actividades, se debe discriminar cada recorrido para que no haya interferencias entre ellos. Para eso se crean las propiedades *Plan_versión* (Figura 2)

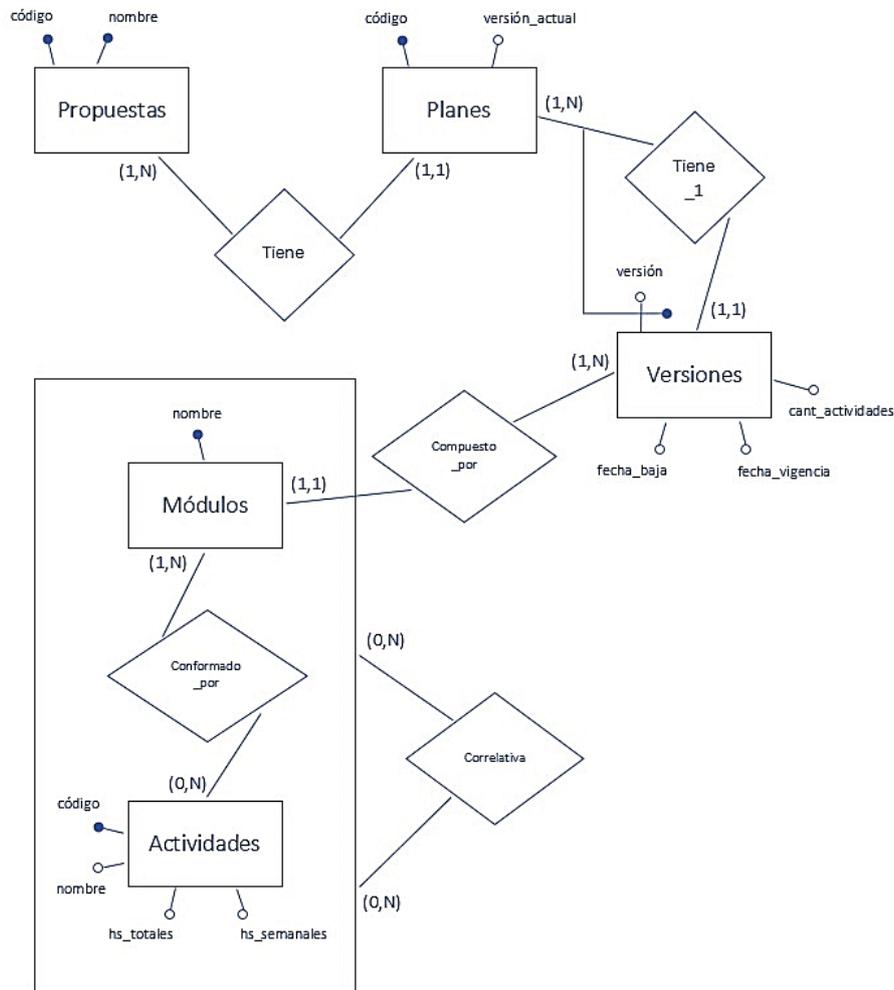


Fig. 1. Modelo Conceptual

Entre actividades se deben tener en cuenta dos tipos de correlatividades: de regularidad de cursada, y de aprobación de examen. Cada una de ellas tiene importancia en distintas etapas de la aprobación de una actividad.

Los planes de estudio cuentan con actividades genéricas: una lista de actividades optativas donde se eligen, cursan y aprueban una o varias. Se diseñó con su propia etiqueta y relaciones optativas (Figura 3).

Además de las relaciones de otros nodos con módulos, existen relaciones de correlatividad entre ellos, como por ejemplo la correlatividad n-4, donde para poder cursar un módulo cuatrimestral n, se debe tener aprobado el n-4 (Figura 4).

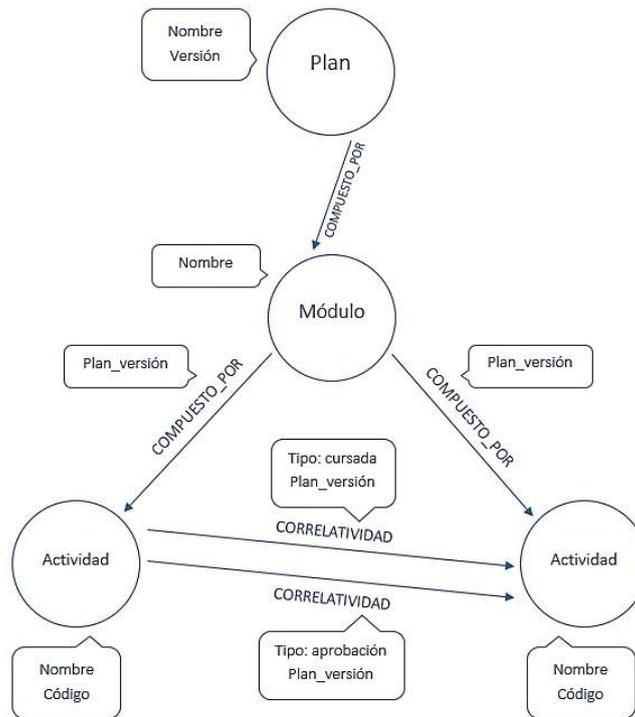


Fig. 2. Grafo de propiedades del Plan de Estudio.

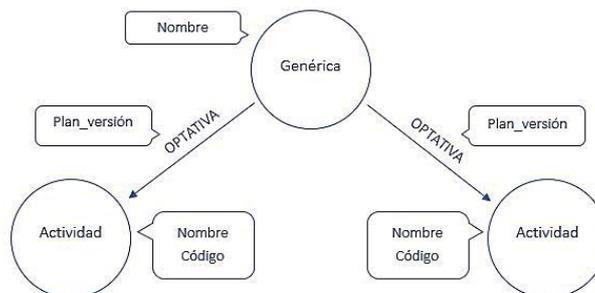


Fig. 3. Grafo de propiedades para Actividad Genérica.



Fig. 4. Grafo de propiedades del Módulo n-4

Una vez logrado este modelo lógico, se utiliza para la realización del modelo físico en la base de datos orientada a grafos elegida: Neo4j.

En su lenguaje, Cypher, no se declara la estructura y en base a ella los nodos, sino que los mismos nodos son los que dan la estructura. Es decir, para poder acceder a una generalización de la base de datos, se deben crear todos los nodos con las mismas etiquetas y propiedades.

Utilizando la sentencia Create, creamos todos los nodos y relaciones con sus propiedades. Un ejemplo reducido:

```
CREATE (lic:Plan {nombre:'Licenciatura en Sistemas',
version:415})
CREATE (cuatri6:Modulo {nombre:'Sexto Cuatrimestre'})
CREATE (cuatri10:Modulo {nombre:'Decimo Cuatrimestre'})
CREATE (bdd1:Actividad {nombre:'Base de Datos I',
codigo:867})
CREATE (bdd2:Actividad {nombre:'Base de Datos II',
codigo:387})
CREATE (cuatri6) - [:CORRELATIVIDAD {plan_version:415,
tipo:'n-4'}] -> (cuatri10), (cuatri6) - [:COMPUESTO_POR
{plan_version:415}] -> (bdd2), (bdd1) -
[:CORRELATIVIDAD {plan_version:415, tipo:' cursada'}] ->
(bdd2), (bdd1) - [:CORRELATIVIDAD {plan_version:415,
tipo:'aprobacion'}] -> (bdd2)
CREATE (optativa2:Generica {nombre:'Optativa II'})
CREATE (pweb:Actividad {nombre:'Programacion Web',
codigo:882})
CREATE (cuatri10) - [:COMPUESTO_POR {plan_version:415}]
-> (optativa2), (bdd2) - [:CORRELATIVIDAD
{plan_version:415, tipo:'aprobacion'}] -> (pweb),
(optativa2) - [:OPTATIVA {plan_version:415}] -> (pweb)
```

Luego, el modelo obtenido debería responder a las consultas iniciales planteadas a este grafo para su validación.

A continuación, presentamos algunos ejemplos detallando el código correspondiente a la consulta realizada y el grafo resultante como respuesta a la misma.

- El plan de estudio de una propuesta académica. Ejemplo: consulta sobre módulos y actividades correspondiente al Plan de Estudio versión 415

```
match (m:Modulo) - [:COMPUESTO_POR {plan_version:415}]
-> (mat) return m, mat
```

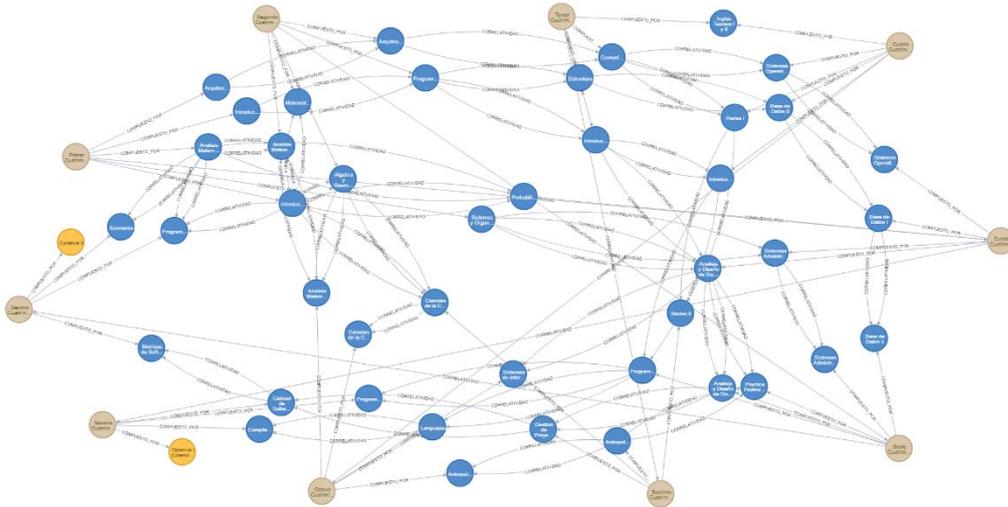


Fig. 5. Módulos y actividades del plan de estudio versión 415

- Actividades correlativas de una actividad en un plan de estudio. Ejemplo: consultamos las actividades correlativas a Análisis Matemático Básico cuyo código de actividad es 667.

```
match (amb:Actividad {codigo:667}) - [:CORRELATIVIDAD
{plan_version:415}] -> (mat:Actividad) return amb, mat
```



Fig. 6. Actividades correlativas de Análisis Matemático Básico.

- Actividades optativas de una genérica en un plan de estudio. Ejemplo: Actividades Optativas del plan 415 asociadas a la Actividad Genérica denominada 'Optativa I'

```
match (g:Generica {nombre:'Optativa I (Licenciatura)'})
- [:OPTATIVA {plan_version:415}] -> (mat:Actividad)
return g, mat
```



Fig. 7. Actividades Optativas de Genérica 'Optativa I'

- Actividades en un periodo lectivo en un plan de estudio. Ejemplo: actividades del Plan de Estudios versión 415, correspondientes al módulo Primer Cuatrimestre.

```
match (m:Modulo {nombre:'Primer Cuatrimestre'}) -
[:COMPUESTO_POR {plan_version:415}] -> (mat)
return m, mat
```

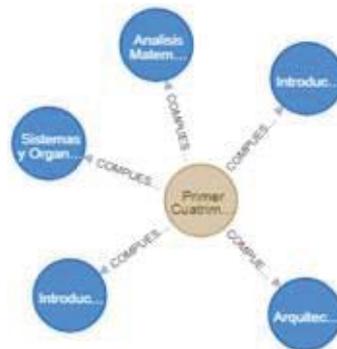


Fig. 8. Actividades del módulo Primer Cuatrimestre

5 Conclusiones y Trabajo Futuro

El modelo orientado a grafos alcanzado hasta el momento realza la importancia de los datos y principalmente, de las conexiones entre ellos.

Las consultas estudiadas son simples en comparación con el modelo relacional donde las 5 (cinco) entidades del modelo conceptual se convierten en tablas y para alcanzar los mismos resultados se requiere el join de todas estas. Esto evidencia un incremento en el rendimiento y la adaptabilidad del modelo en respuesta a los cambios del dominio.

Los datos de los planes de estudio utilizados son confiables ya que los mismos fueron extraídos directamente desde la base de datos de la universidad. La integridad del modelo en su primera etapa fue validada con las consultas planteadas al mismo.

El modelo planteado sobre los Planes de Estudio de las Propuestas de Grado de la Universidad corresponde al trayecto académico óptimo de los egresados. Actualmente se está trabajando en la implementación de los trayectos académicos propios de cada egresado, lo que permitirá a futuro, mediante técnicas de análisis de grafos buscar evidencias sobre problemáticas propias del currículum que inciden en la sobre-duración de las propuestas.

6 Referencias

1. Robinson, I., Webber, J. and Eifrem, E.: Graph Databases. Sebastopol, USA: O'Reilly (2015)
2. Angles R., Gutierrez C.: An Introduction to Graph Data Management. In: Fletcher G., Hidders J., Larriba-Pey J. (eds) Graph Data Management. Data-Centric Systems and Applications. Springer, Cham (2018)
3. Perez Sola C.: Diseño de una base de datos para analizar la actividad de usuarios en twitter. UOC
4. Pinilla, C.; Bello, M.; Peña, C.: Bases de datos orientadas a grafos. *TIA*, 5(2), pp. 153-160, . (2017).
5. Moneris, L.: Bases de datos orientadas a grafos aplicadas al estudio de informes radiológicos: utilizando entornos de computación en la nube para abordar estudios de gran dimensión. Tesis maestría, Valencia (2015)
6. Neo4j, <https://neo4j.com/graphacademy/>
7. Reconocimiento académico: pensar la formación con foco en el estudiante. Ministerio de Educación, Cultura, Ciencia y Tecnología (2018) <https://www.argentina.gob.ar/noticias/reconocimiento-academico-pensar-la-formacion-confoco-en-el-estudiante>
8. Habersfeld, L., Marquina M. y Morresi, S.: El Sistema universitario argentino. Situación, problemas y políticas. Centro de Estudios para el cambio estructural (2018) <http://fcece.org.ar/el-sistema-universitario-argentino/>

Solving All- k -Nearest Neighbor Problem without an Index

Edgar Chávez, Verónica Ludueña, and Nora Reyes

Departamento de Informática, Universidad Nacional de San Luis,
San Luis, Argentina

{vlud, nreyes}@unsl.edu.ar

Centro de Investigación Científica y de Educación Superior de Ensenada, México
elchavez@cicese.mx

Abstract. Among the similarity queries in metric spaces, there are one that obtains the k -nearest neighbors of all the elements in the database (*All- k -NN*). One way to solve it is the naïve one: comparing each object in the database with all the other ones and returning the k elements nearest to it (k -NN). Another way to do this is by preprocessing the database to build an index, and then searching on this index for the k -NN of each element of the dataset. Answering to the *All- k -NN* problem allows to build the k -Nearest Neighbor graph (kNNG). Given an object collection of a metric space, the Nearest Neighbor Graph (NNG) associates each node with its closest neighbor under the given metric. If we link each object to their k nearest neighbors, we obtain the k Nearest Neighbor Graph (kNNG). The kNNG can be considered an index for a database, which is quite efficient and can allow improvements.

In this work, we propose a new technique to solve the *All- k -NN* problem which do not use any index to obtain the k -NN of each element. This approach solves the problem avoiding as many comparisons as possible, only comparing some database elements and taking advantage of the distance function properties. Its total cost is significantly lower than that of the naïve solution.

1 Introduction

Similarity search has become a very important operation in applications that deal with unstructured data sources. It has applications in a large number of fields. Some examples are non-traditional databases; machine learning and classification; information retrieval; image quantization and compression; computational biology; text searching; and function prediction; between others. All those applications can be formalized with the *metric space model* [6]. A metric space is composed by a universe of objects \mathbb{U} and a distance function d . The distance function gives us a dissimilarity criterion to compare objects from \mathbb{U} . A database is a subset $S \subseteq \mathbb{U}$.

In the metric space model the similarity queries in S of any $q \in \mathbb{U}$ are usually of two types:

- *range query*: given $r \in \mathbb{R}^+$ it retrieves all the elements in S within distance r to q ($R(q, r)$), and
- *k -nearest neighbor*: given $k \in \mathbb{N}$ it retrieves the k closest elements to q in S - $\{q\}$ (k -NN(q)).

The k -NN(q) query is a building block for a large number of problems in a wide number of application areas. For instance, in pattern classification, the nearest-neighbor rule can be implemented with 1-NN(q) [8].

Formally, the Nearest Neighbor Graph (NNG) is a graph whose vertex set is S and with one edge from u to v whenever v is the nearest neighbor of u in S . The problem of determining the nearest neighbor of any element is often called the all-nearest neighbor problem. It could be generalized to obtain the k -NN of *all* elements of database: the *All- k -NN* problem. It is a useful operation for batch-based processing of a large distributed point dataset. Hence, it will be our focus.

Performing similarity queries on a database using a sequential scan can become impractical, either because of the database size or the cost of distance evaluations. As usually, the computation of distances represents the most significant cost in this type of searches. Hence, it is customary to use this cost as the complexity measure. For general metric spaces, there exist several methods to preprocess the database in order to reduce the number of distance evaluations [6], and then by performing n k -NN queries, avoiding the exhaustive search.

However, when the database is very large or the distance is very costly, building an index, and then performing a k -NN query for each database element could be too expensive. Therefore, in this work, we present a new method to solve the *All- k -NN* problem, which will allow us to effectively compute the k NNG. It is important to mention that this technique does not use an index to solve the problem. The cost of our approach is significantly lower than n^2 , which is the number of distance calculations used by the naïve solution. Our proposal computes some distances between database objects and ingeniously takes advantage of the properties that distance function satisfies.

This paper is organized as follows: Section 2 presents a brief description of some useful concepts. Section 3 introduces our proposal, and Section 4 contains the empirical evaluation of our proposed solution. Finally, in Section 5 we conclude and discuss about possible extensions for our work.

2 Previous Concepts

In this section, we briefly state the problem in a more formal way to continue the discussion. A metric space is composed of a universe of objects \mathbb{U} , and a distance function $d : \mathbb{U} \times \mathbb{U} \rightarrow \mathbb{R}^+$, such that for any $x, y, z \in \mathbb{U}$, $d(x, y) > 0$ (strict positiveness), $d(x, y) = 0 \iff x = y$ (reflexity), $d(x, y) = d(y, x)$ (symmetry), and obeying the triangle inequality: $d(x, z) + d(z, y) \geq d(x, y)$. The smaller the distance between two objects is, the more *similar* they are. We have a finite database S , which is a subset of \mathbb{U} and can be preprocessed. Later, given a new object q from \mathbb{U} used as a query, we must retrieve all elements in S close to q , using as few distance computations as possible. In the metric space model, similarity queries are usually of two types. For a given database S with size $|S| = n$, $q \in \mathbb{U}$ and $r \in \mathbb{R}^+$: $(q, r) = \{x \in S \mid d(q, x) \leq r\}$ is known as a *range query*; and k -NN(q), denotes the *k -nearest neighbors*, formally it retrieves the set $R \subseteq S$ such that $|R| = k$ and $\forall u \in R, v \in S - R, d(q, u) \leq d(q, v)$. This last primitive is a fundamental tool in cluster and outlier detection [3, 9], image segmentation [1], query or document recommendation systems [2], VLSI design, spin glass and other physical process simulations [4], pattern recognition [8], and so on.

As it is aforementioned, the distance is considered expensive to compute (think, for instance, in comparing two fingerprints). Thus, the ultimate goal is to build *offline* an index to speed up *online* queries. Different techniques to solve the problem of similarity queries have arisen to reduce these costs, usually based on data preprocessing. All those structures work based on discarding elements using the triangle inequality, and most of them use the classical divide-and-conquer approach.

A related version of the k -NN problem, perhaps less studied, is the *All- k -NN* problem. That is, if $|S| = n$, we solve the *All- k -NN* problem by efficiently retrieving the k -NN(u_i) for each u_i in S and performing less than $O(n^2)$ distance evaluations. It is a useful operation for batch-based processing of a large distributed point dataset. Consider, for example, a location-based service which recommends each user his or her nearby users, who may be the candidates of new friends. Given that locations of users are maintained by the underlying database, we can generate such recommendation lists by issuing an *All- k -NN* query on the database.

Most of the solutions that have been proposed and developed for this problem use indexes. Some of them, for general metric spaces [12, 13], are based on the construction of k -nearest neighbors graphs (k NNG). The k NNG is a weighted directed graph connecting each object from the metric space to its k nearest neighbors; that is, $G(S, E)$ such that $E = \{(u, v), u, v \in S \wedge v \in k\text{-NN}(u)\}$. G connects each element through a set of arcs whose weights are computed according to the distance of the corresponding space. Building the k NNG is a direct generalization of the *all-nearest-neighbor* (*All-1-NN*) problem, which corresponds to the 1NNG construction problem. The k NNG offers an indexing alternative which requires a moderately amount of memory, obtaining reasonably good performance in the search process. In fact, in low-memory scenarios, which only allow small values of k the search performance of k NNG is better than using classical pivot-based indexing alternative. In addition, graph-based techniques offer great potential for improvements, ranging from fully dynamic graph-based indexes to specific optimizations for metric space search.

The naïve algorithm for *All- k -NN* calculates the distance function d between each $u_i \in S$ and every element of S , so it has quadratic complexity.

3 Our proposal

As it was already mentioned, performing similarity queries on a database using a sequential scan can become impractical, either because of the size of the database or because of the cost of distance calculations. Under this cost model, the ultimate goal of any similarity search technique is to solve queries doing the fewest number of distance calculations. This objective was taken into mind in this new proposal. It is important to note that our proposal achieves its goal without using any index.

This technique allows to compute the k -nearest neighbors for all the elements from the database S , to compute the k NNG. To do this, some objects $x_i \in S$ are selected, and their distance to each of the other elements y in the database are calculated ($y \in S - \{x_i\}$). Since each selected element x_i knows its distance to the rest of the objects in S , it can be used as support by some elements close to it, to find their k -nearest neighbors.

After computing the $n-1$ distances from x_i to the other $n-1$ objects in S , we obtain the set of distances: $\{d(x_i, x_1), d(x_i, x_2), d(x_i, x_3), \dots, d(x_i, x_{n-1})\}$. The elements $\{x_1, x_2, x_3, \dots, x_{n-1}\}$ are sorted by distance to x_i , resulting in the sequence of objects $x_{j_1}, x_{j_2}, \dots, x_{j_{n-1}}$. The first k elements of this sequence, $x_{j_1}, x_{j_2}, \dots, x_{j_k}$, are the k objects more similar to x_i , therefore their k -nearest neighbors, and at this time they can be reported.

Next, we calculate the k -NN of the objects $x_{j_1} \in S$ that do not yet have them. For this purpose, we consider to solve the k -NN query by a range query with decreasing radius, and to use the known fact that the search will be more efficient while more quickly we can reduce the search radius. Assuming, furthermore, that very close objects possibly share some of their closest neighbors. That is, it takes the first neighbor of x_i , which is x_{j_1} , and starts the computation of its k -NN by comparing it with the k -NN(x_i) - $\{x_{j_1}\}$. It will be noticed that x_{j_1} already knows one distance; that is, its distance to x_i because of the symmetry of d . In this way, by adding x_i to this group, we get the first k candidates for being its closest neighbors, and an initial covering radius, which is the distance of x_{j_1} to its furthest neighbor:

$$r_k(x_{j_1}) = \max\{d(x_{j_1}, x_i), d(x_{j_1}, x_{j_2}), \dots, d(x_{j_1}, x_{j_k})\}$$

The knowledge acquired by x_i , when compared to all objects in the database, allows to estimate the distances between x_{j_1} and the other elements in S using triangular inequality. This estimate defines a lower bound of the actual distance between x_{j_1} and any other element of S ; that is, how close an object x_j could be to x_{j_1} . By knowing the cover radius and the estimate of distances it is possible to discard some objects of S , avoiding its actual comparison with x_{j_1} . If that lower bound is greater than the current radius enclosing the k candidates for k -NN(x_{j_1}), that is $|d(x_j, x_i) - d(x_i, x_{j_1})| > r_k(x_{j_1})$, this object will be discarded because it will not be closer to x_{j_1} than the current neighbors.

Otherwise, those objects x_j whose estimated distance is less than the current covering radius; that is, they satisfy $|d(x_j, x_i) - d(x_i, x_{j_1})| < r_k(x_{j_1})$, will be directly compared with x_{j_1} . Then, the elements that could not be discarded, will be sorted from highest to lowest estimate value. If those elements when they are compared with x_{j_1} verify $d(x_j, x_{j_1}) < r_k(x_{j_1})$; that is, their real distance is smaller than the covering radius, will become part of the k -NN(x_{j_1}). Hence, the element whose distance was $r_k(x_{j_1})$, which is the current farthest one from x_{j_1} , will be discarded and it implies that the covering radius $r_k(x_{j_1})$ should be updated. When we update the covering radius, reducing it, the estimates should be rechecked to determine if we can discard some other objects. If the elements that remained as promising ones when they are directly compared with x_{j_1} verify $d(x_j, x_{j_1}) > r_k(x_{j_1})$, they must be discarded. Otherwise, they can contribute to the set k -NN(x_{j_1}) and reduce again the covering radius $r_k(x_{j_1})$.

This process ends when all the objects in S were reviewed or when the next estimate analyzed is greater than the covering radius, because all elements of S from then on will be discarded. This is because they are sorted in ascending order by their estimated distance, there will no longer be items closer to x_{j_1} than neighbors that are already known. At the end of this process, k -NN(x_{j_1}) is reported. Then, the next object is taken in the sequence of x_i neighbors that do not already have their k -NN calculated, and proceed to calculate them in the same way.

As the elements to which k -NN are calculated move away from x_i , the number of objects that can be discarded from the list of promising ones, using triangular inequality, is decreasing. It causes an increase in the number of distances to be calculated. But, as our goal is to reduce this number, at that time we must look for a new element that “helps” in the calculation of the remaining k -NN. A parameter α helps us to decide when an object becomes useful in helping to discard candidates. Each time we finish the process of associating an object with its k -nearest neighbors, we check whether the element x_i used as support still produces a good discard. When an element x_i allows to reject less than α elements, its discard capability is considered insufficient. Therefore, it must be replaced by another element from the database. We select a new x_i randomly between the elements in S that do not have their k -NN already calculated. In that case, this new x_i replace the old one, so the $(n - 1)$ distances to the remaining members of S are computed. Hence, we obtain k -NN(x_i), and use x_i to assist in the calculation of k -NN(x_{j_i}) for some x_{j_i} for which x_i allows to discard more than α elements. Now, the process is repeated until all the elements in S have their k -NN.

4 Experimental Results

In order to evaluate our proposal, we perform some experiments over different metric databases. For each database we run the process over different database permutations. Besides, we test different values for the α parameter.

For the empirical evaluation of our proposal we consider a set of real-life metric spaces, all of them with a low intrinsic dimension, and with widely different histograms of distances: **Strings**: a dictionary of 69,069 English words. The distance is the *edit distance*, that is, the minimum number of character insertions, deletions and substitutions needed to make two strings equal.

NASA images: a set of 40,150 20-dimensional feature vectors, generated from images downloaded from NASA ¹. The Euclidean distance is used.

Color histograms: a set of 112,682 8-D color histograms (112-dimensional vectors) from an image database ². Any quadratic form can be used as a distance, so we chose Euclidean distance.

All these metric databases are available from www.sisap.org [10].

Besides, to analyze how the intrinsic dimensionality affects the behavior of our approach, we evaluated it over synthetic metric spaces, where we can control their intrinsic dimensionality. We use collections of 100,000 vectors, uniformly distributed in the unit hypercube, in dimensions 4, 8, 12, 16, 20, 24, 28 and 32. We consider the vectors in these spaces as metric objects. Thus, we do not use explicitly the information of the coordinates of each vector. In these spaces we also use Euclidean distance.

We test different values of the parameter α : $\frac{n}{10}$, $\frac{n}{5}$, $\frac{n}{4}$, $\frac{n}{3}$, and $\frac{n}{2}$. Besides, we evaluate several values of the number k of nearest neighbors to be obtained: 1, 2, 5, 10, 50, and 100. As it is aforementioned, we register the cost of our approach considering the number of calculations of distances done. However, for an easy understanding of the results we show the average number of distances needed by element.

¹ At <http://www.dimacs.rutgers.edu/Challenges/Sixth/software.html>

² At <http://www.dbs.informatik.uni-muenchen.de/~seidl/DATA/histo112.112682.gz>

The Figure 2 shows the costs of answering to *All-k-NN* in the space of NASA Images, Strings and Color Histograms, respectively. Each line represents the costs of obtaining k neighbors with a certain α . As it can be seen, better results are obtained when α is smaller (e.g. $\frac{n}{10}$). Possibly, in this case the distinguished objects x_i are not changed quickly and this allows reducing the number of distance calculations. However, as it can be noticed, all our alternatives need to perform much less than n distance evaluations in average by element. Therefore, the speedup obtained respect to the naive solution is very significant.

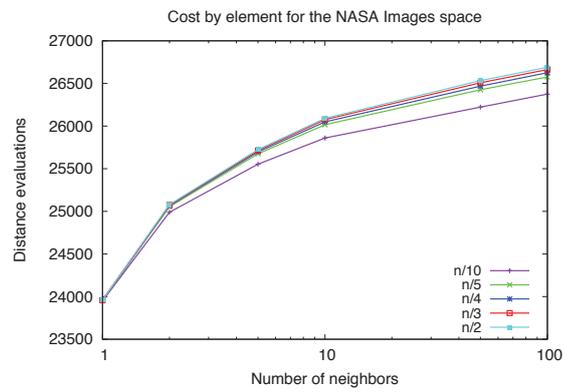
It can be regarded from these results that for higher values of α , for example $\frac{n}{2}$, as it difficult to find elements able to discarding one half of the database for each object whom estimates its distances through it. Higher values of α cause replacing more frequently the distinguished element owing to it does not satisfy the discard quality. Consequently, the new candidate must calculate again its $(n - 1)$ distances. Surprisingly, it do not occur in the String space. In this case, all costs are very similar, no matter the value of α .

It is remarkable that, throughout the process up to half of the database, the number of distinguished objects used is approximately the half of the elements that are obtained its k -NN. That is, each distinguished element is actually useful for calculating, on the average, the k -NN for two more objects, and then it loses its discard quality. When the process achieves the last database elements; that is, most of the objects already have their k -NN, the remaining objects to be used as distinguished elements are not so good for discard. Hence, they must be replaced quickly, increasing in consequence the number of distance calculations.

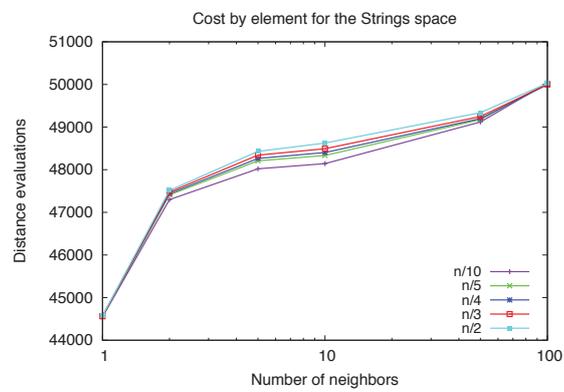
Comparison with others methods: We compare the performance of our technique with respect to several representative indexes: List of Clusters [5], Distal Spatial Approximation Trees [7], and a generic pivot index [6]. All of them have their implementation available from www.sisap.org [10]. As the other methods considered need to build the index to perform the searches, we compute their costs considering both the construction cost of the index and the search cost of all k -NN operations.

The Figure 2(a), Figure 2(b) and Figure 2(c) illustrate the comparison of the costs of our proposal and the other considered indexes. We use LC for the List of Clusters, DiSAT for the Distal Spatial Approximation Trees, and Piv for the generic pivot index to name the lines of these indexes. Particularly, the values that accompany LC indicate the different cluster sizes used to build the index, and for Piv the number of pivots. We choose to show for each index its better alternatives. For our approach we use the better values of α empirically determined: $\frac{n}{10}$ and $\frac{n}{5}$. We label each α option with the corresponding value.

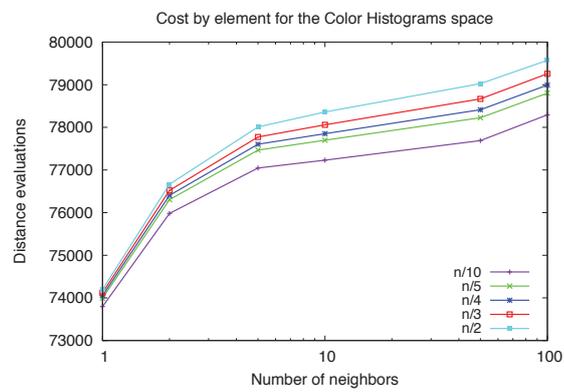
In the Strings space, only the pivot-based index is more expensive than our method, the other indexes beat us for all the k values. We only get closer to LC and DiSAT for $k = 100$. For the Nasa Images space, our technique is surpassed by all indexes and for all the values of k , we consider that it is because this space has the lowest intrinsic dimensionality. Moreover, in the Color Histograms space, we again beat pivot index but only for $k > 10$, and LC and DiSAT always are better than us. It is remarkable that all the Real spaces considered do not have high intrinsic dimensionality [11].



(a) Nasa Images space.

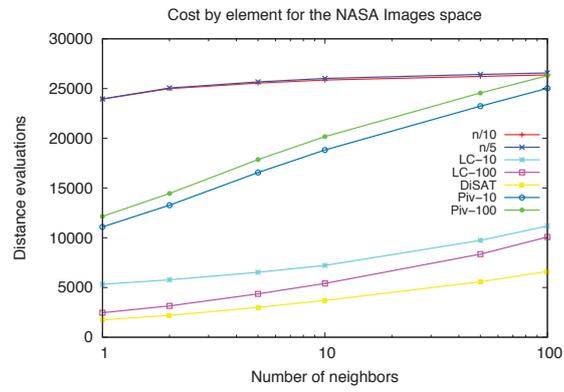


(b) Strings space.

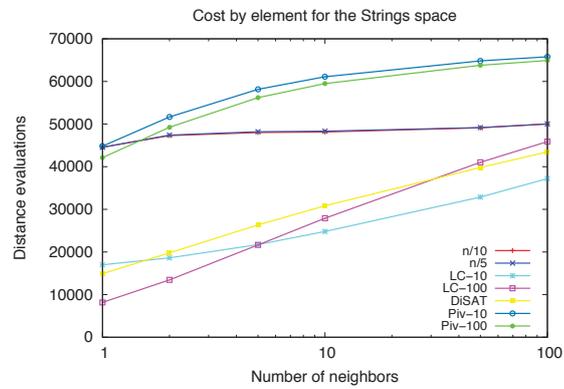


(c) Color histograms space

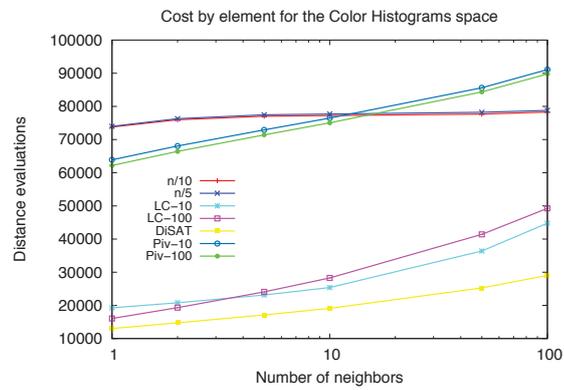
Fig. 1. Costs of *All-k-NN* in Real spaces.



(a) Nasa Images space.



(b) Strings space.



(c) Color histograms space

Fig. 2. Comparison of costs to solve All-k-NN on the Real spaces considered.

The Figure 3 depicts the same experiments on the synthetic spaces. For lack of space, we only show the results for the vector spaces in higher dimensions; that is, dimension from 20 to 32. We do not show the results for vector spaces in lower dimensions because the behavior of all the methods are similar to those of the already depicted for the real spaces. As it can be observed, as dimension grows our proposal becomes better than more and more of the other alternatives. For example, for dimensions from 24 (Figures 3(b), 3(c), and 3(d)), we can overcome all the other indexes, for almost all number of neighbors.

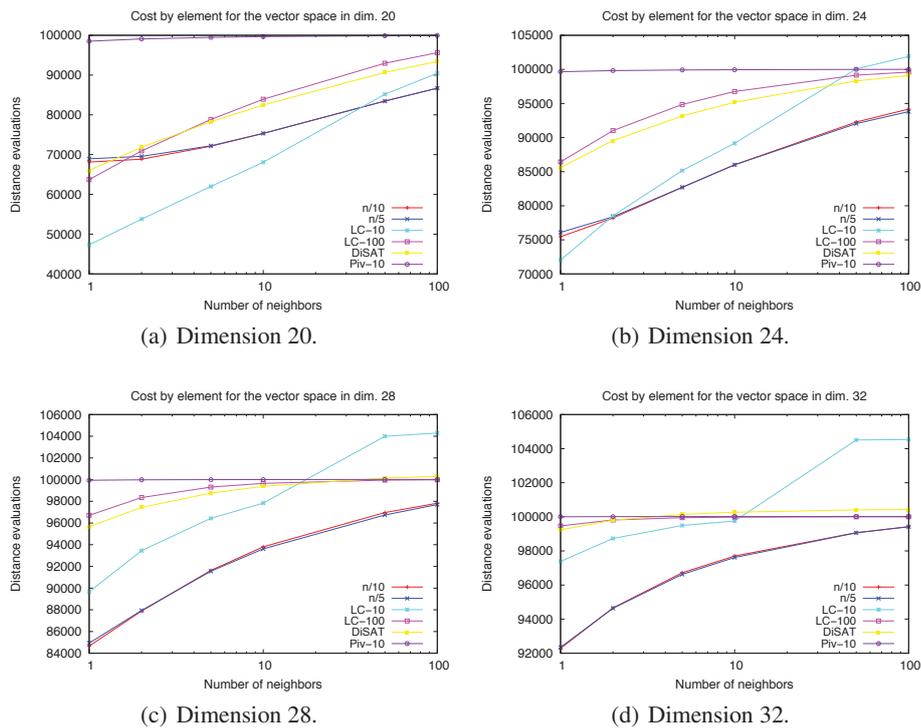


Fig. 3. Comparison of costs to obtain $All-k-NN$ on synthetic spaces.

Although LC and DiSAT are indexes that have good behavior on medium to high dimensional metric space, they also degrade their performance when dimension is higher than 24. Hence, it is important to remark that our approach seems more resistant to the curse of dimensionality.

5 Conclusions

An extended version of determining the $k-NN$ of an element, is to solve the $All-k-NN$ problem. Solving this problem is useful, for example, to build the metric index named the graph of k -nearest neighbors. Until this moment, the only way to solve this problem,

without using an auxiliary index, is via the naïve solution of comparing each database element with each other one. In this paper we propose and tested an alternative approach to solve this problem. Our proposal is very efficient with respect this naïve solution, because it needs to perform much less than n distance evaluations by element.

Our results are preliminary and encouraging. It can be affirmed that, on low dimensional metric spaces, it may be preferable for solving the *All-k*-NN problem to build an index and then perform the search for the k -NN of each edatabase lement. By the other hand, on medium to high dimensional spaces, the curse of dimensionality affects more clearly the performance of the different indexes than of our proposal. Therefore, as the dimension grows our proposal achieves a better performance than other alternatives that also solve this problem.

As future works, we consider analyzing different options to select the distinguished elements for each process step, and how this selection affects the method performance. Besides, we also plan to study how to take advantage of the available memory space to cache some distances calculated and avoid the recalculation of them.

References

1. Neculai Archip, Robert Rohling, Peter Cooperberg, Hamid Tahmasebpour, and Simon K. Warfield. Spectral clustering algorithms for ultrasound image segmentation. In James S. Duncan and Guido Gerig, editors, *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2005*, pages 862–869, Berlin, Heidelberg, 2005. Springer Berlin Heidelberg.
2. Ricardo Baeza-Yates, Carlos Hurtado, and Marcelo Mendoza. Query clustering for boosting web page ranking. In Jesús Favela, Ernestina Menasalvas, and Edgar Chávez, editors, *Advances in Web Intelligence*, pages 164–175, Berlin, Heidelberg, 2004. Springer Berlin Heidelberg.
3. M. Brito, E. Chávez, A. Quiroz, and J. Yukich. Connectivity of the mutual k -nearest neighbor graph in clustering and outlier detection. *Statistics & Probability Letters*, 35(4):33–42, 1996.
4. P. Callahan and R. Kosaraju. A decomposition of multidimensional point sets with applications to k nearest neighbors and n body potential fields. *JACM*, 42(1):67–90, 1995.
5. E. Chávez and G. Navarro. A compact space decomposition for effective metric indexing. *Pattern Recognition Letters*, 26(9):1363–1376, 2005.
6. E. Chávez, G. Navarro, R. Baeza-Yates, and J. Marroquín. Searching in metric spaces. *ACM Computing Surveys*, 33(3):273–321, September 2001.
7. Edgar Chávez, Verónica Ludeña, Nora Reyes, and Patricia Roggero. Faster proximity searching with the distal sat. *Information Systems*, 59:15–47, 2016.
8. R. Duda and P. Hart. Pattern classification and scene analysis. *John Wiley & Sons*, 1973.
9. David Eppstein and Jeff Erickson. Iterated nearest neighbors and finding minimal polytopes. In *An Int. Journal of Mathematics and Computer Science*, volume 11–3, pages 321–350, 1994.
10. Karina Figueroa, Gonzalo Navarro, and Edgar Chávez. Metric spaces library, 2007. Available at <http://www.sisap.org/MetricSpaceLibrary.html>.
11. G. Navarro, R. Paredes, N. Reyes, and C. Bustos. An empirical evaluation of intrinsic dimension estimators. *Information Systems*, 64:206–218, 2017.
12. R. Paredes. *Graphs for Metric Space Searching*. PhD thesis, University of Chile, 2008.
13. R. Paredes, E. Chávez, K. Figueroa, and G. Navarro. Practical construction of k -nearest neighbor graphs in metric spaces. In *Proc. 5th Workshop on Efficient and Experimental Algorithms (WEA)*, LNCS 4007, pages 85–97, 2006.

Towards Measuring the Severity of Depression in Social Media via Text Classification

Sergio G. Burdisso^{1,2}, Marcelo Errecalde¹, and Manuel Montes-y-Gómez³

¹ Universidad Nacional de San Luis (UNSL), Ejército de Los Andes 950, San Luis, San Luis, C.P. 5700, Argentina

{sburdisso, merreca}@unsl.edu.ar

² Consejo Nacional de Investigaciones Científicas y Técnicas (CONICET), Argentina

³ Instituto Nacional de Astrofísica, Óptica y Electrónica (INAOE), Luis Enrique Erro No. 1, Sta. Ma. Tonantzintla, Puebla, C.P. 72840, Mexico
mmontesg@inaoep.mx

Abstract. Psychologists have used tests or carefully designed survey questions, such as Beck’s Depression Inventory (BDI), to identify the presence of depression and to assess its severity level. On the other hand, methods for automatic depression detection have gained increasing interest since all the information available in social media, such as Twitter and Facebook, enables novel measurement based on language use. These methods learn to characterize depression through natural language use and have shown that, in fact, language usage can provide strong evidence in detecting depressive people. However, not much attention has been paid to measuring finer grain relationships between both aspects, such as how is connected the language usage with the *severity level* of depression. The present study is a *first step* towards that direction. First, we train a binary text classifier to detect “depressed” users and then we use its confidence values to estimate the user’s clinical depression level. In order to do that, our system has to fill the standard BDI depression questionnaire on users’ behalf, based only on the text of users’ postings. Our proposal, publicly tested in the eRisk 2019 T3 task, obtained promising results. This offers very interesting evidence of the potential of our method to estimate the level of depression directly from user’s posts in social media.

Keywords: Text Classification · Depression Level Estimation · Beck’s Depression Inventory · SS3 · CLEF eRisk 2019 · Reddit

1 Introduction

Depression is one of the leading cause of disability and one of the major contributor to the overall global burden of disease. Globally, in 2015 it was estimated that more than 332 million people suffered from this mental illness. Additionally, between 2005 and 2015 the total estimated number of people living with depression increased by 18.4%. Depressive disorders are ranked as the single largest contributor to non-fatal health loss and in extreme cases could lead to suicide[12].

Every 40 seconds a person dies due to suicide somewhere in the world, every year over 800.000 suicide deaths occur and it is the second leading cause of death in the 15-29 years-old range[11]. In 2015, suicide was among the top 20 leading causes of worldwide death[12]. Globally, 71% of all violent deaths in women, and 50% in men, are due to suicide[11]. Along with cancer, heart disease, stroke, and diabetes, suicide is among the 10 leading causes of death in the United States, as well as in other high-income countries. Additionally, the suicide rate increased by 3.7% from 2016 to 2017[10].

For many years, psychologists have used tests or carefully designed survey questions (such as BDI[2]) to identify the presence of depression and to assess its severity level. Nowadays, all the information available in social media, such as Twitter or Facebook, enabled novel methods for depression detection based on machine learning techniques to gain popularity. Even though multiple studies have attempted to predict or analyze depression using machine learning techniques, to the best of our knowledge, [6] was the first attempt to build a public dataset in which a large collection of social media users' posts leading to this disorder, was made available to the public. Thus, the main goal in [6] was to provide the first public collection to study the relationship between depression and language usage by means of machine learning techniques. This dataset was then used for the CLEF's eRisk 2017[7] and 2018[8] public tasks on early depression detection in social media.

Machine learning models learn to characterize depression through natural language use and have shown that, in fact, language usage can provide strong evidence in detecting depressive people. However, not much attention has been paid to measuring finer grain relationships between both aspects, such as how is connected the language usage with the *severity level* of depression. That is why the latest edition of this public challenge, CLEF's eRisk 2019[9], decided to shift the focus from early depression detection to trying to measure its severity. The present study is a first step towards that direction and describes how our team (UNSL) approached this task. First, in section 2, we describe the eRisk 2019 task and the used evaluation metrics in more details. Then, we introduce the approach we used to carry out this task in section 3 and the evaluation results are presented and discussed in section 4. Finally, the main conclusions derived from this study are summarized in section 4, along with suggestions for possible future work.

2 Measuring the Severity of Depression

As it is described in more details in [9], the CLEF's eRisk 2019 lab was divided into three different tasks, T1, T2 and T3, being only T3 related to depression. T3 task consisted of estimating the level of depression from a thread of user posts. For each user, the participants were given a history of postings and they had to fill a standard depression questionnaire (based on the evidence found in the history of postings). The questionnaires were defined from Beck's Depression Inventory (BDI)[2]. The BDI is a 21-question, self-report rating inventory that

Table 1: Summary of the test data

No. of subjects	20
No. of posts	10,941
Avg. No. of posts per subject	547
Avg. No. of days from first to last posts	881.2
Avg. No. of words per posts	46.4

measures characteristic attitudes and symptoms of depression. Each question has 4 possible answers, numbered from 0 to 3, and is useful to assess the presence of feelings like sadness, pessimism, loss of energy, etc. For example, the question 3 is as follows:

Question 3. Past Failure:

0. I do not feel like a failure.
1. I have failed more than I should have.
2. As I look back, I see a lot of failures.
3. I feel I am a total failure as a person.

Therefore, this task aimed at exploring the viability of automatically estimating the severity of multiple symptoms associated with depression. Given the user's history of postings, the algorithms had to estimate the user's response to each individual question.

It is worth mentioning that for this task, no training data was provided and therefore, only the raw (unlabeled) test set used to evaluate the performance of all participants was provided. To build this test set, questionnaires filled by Reddit⁴ users were collected together with their history of postings. User's posts were collected right after he/she filled the BDI questionnaire. The questionnaires filled by the users were then used as the ground truth to assess the quality of the responses given by the participating systems. The details of the built test set are presented in Table 1.

2.1 Evaluation Metrics

In order to assess the quality of questionnaires filled by the systems, four metrics were used:

- *Hit Rate (HR)*. This measure computes the ratio of cases where the automatic questionnaire has exactly the same answer as the real questionnaire. For example, an automatic questionnaire with 5 matches gets an HR equal to $\frac{5}{21}$.
- *Closeness Rate (CR)*. This measure takes into account that the answers of the BDI questionnaire represent an ordinal scale. For example, imagine that the

⁴ <https://www.reddit.com/>

⁵ Note this 21 here is the total number of questions.

real user answered option 0. A system, S1, whose answer was option 3 should be penalized more than a system S2 whose answer was 1. For each question i , the absolute difference (ad) between the real and the automated answer (e.g. $|0 - 3| = 3$ and $|0 - 1| = 1$ for S1 and S2, respectively) it is computed and next, this absolute difference is normalized as follows: $CR_i = \frac{3-ad_i}{3}$.⁶ Finally, the CR_i for each question is averaged to obtain the final effectiveness score, i.e. $CR = \frac{1}{21} \sum_{i=1}^{21} CR_i$.

- *Difference between Overall Depression Levels (DODL)*. The previous measures assess the systems' ability to answer each question. This measure, instead, does not look at question-level hits or differences but computes the overall depression level (sum of all the answers) for the real and automated questionnaire and next, the absolute difference ($ad_{overall}$) between the real and the automated score is computed. In the BDI, depression level is an integer between 0 and 63 and thus, DODL is normalized between 0 and 1 as follows: $DODL = \frac{63-ad_{overall}}{63}$.
- *Depression Category Hit Rate (DCHR)*. In the psychological domain, it is customary to associate depression levels with the following categories:

minimal (depression levels 0-9)
 mild (depression levels 10-18)
 moderate (depression levels 19-29)
 severe (depression levels 30-63)

This measure consists of computing the fraction of cases where the automated questionnaire led to a depression category that is equivalent to the depression category obtained from the real questionnaire.

Finally, for the first three measures, results were reported using the average over all the users and were referred to as AHR , ACR and $ADODL$.

3 Our approach

To carry out this task, we trained a binary text classifier to detect “depressed” users and then we use its confidence values to estimate the user’s clinical depression level by completing the BDI questionnaire. We decided to use a text classifier that has shown remarkable performance on early depression detection and was firstly introduced in [3]. Thus, subsection 3.1 briefly introduces the used classifier, called SS3, and then subsection 3.2 describes how questionnaires were actually filled in.

3.1 The SS3 text classifier

As it is described in more details in [3], SS3 first builds a dictionary of words for each category during the training phase, in which the frequency of each

⁶ Note that this 3 here is equal to the maximum possible answer.

word is stored. Then, using those word frequencies, and during the classification stage, it calculates a value for each word using a function $gv(w, c)$ to value words in relation to categories. gv takes a word w and a category c and outputs a number in the interval $[0,1]$ representing the degree of confidence with which w is believed to *exclusively* belong to c , for instance, suppose categories $C = \{food, music, health, sports\}$, we could have:

$$\begin{aligned} gv('sushi', food) &= 0.85; & gv('the', food) &= 0; \\ gv('sushi', music) &= 0.09; & gv('the', music) &= 0; \\ gv('sushi', health) &= 0.50; & gv('the', health) &= 0; \\ gv('sushi', sports) &= 0.02; & gv('the', sports) &= 0; \end{aligned}$$

Additionally, a vectorial version of gv is defined as:

$$\vec{gv}(w) = (gv(w, c_0), gv(w, c_1), \dots, gv(w, c_k))$$

where $c_i \in C$ (the set of all the categories). That is, \vec{gv} is only applied to a word and it outputs a vector in which each component is the gv of that word for each category c_i . For instance, following the above example, we have:

$$gv('sushi') = (0.85, 0.09, 0.5, 0.02); \quad gv('the') = (0, 0, 0, 0);$$

The vector $\vec{gv}(w)$ is called the “*confidence vector* of w ”. Note that each category c_i is assigned a fixed position in \vec{gv} . For instance, in the example above $(0.85, 0.09, 0.5, 0.02)$ is the *confidence vector* of the word “sushi” and the first position corresponds to *food*, the second to *music*, and so on. For those readers interested in how the gv function is actually computed, we highly recommend to read the SS3 original paper[3], since its equations are not given here to keep this paper shorter and simpler.

SS3 classification process can be thought of as a 2-phase process. In the first phase the input document is split into multiple blocks (e.g. paragraphs), then each block is in turn repeatedly divided into smaller units (e.g. sentences, words). Thus, the previously “flat” document is transformed into a hierarchy of blocks. In the second phase, the gv function is applied to each word to obtain the “level 0” *confidence vectors*, which then are reduced to level 1 *confidence vectors* by means of a level 0 *summary operator*, \oplus_0 . This reduction process is recursively propagated up to higher-level blocks, using higher-level *summary operators*, \oplus_j , until a single *confidence vector*, \vec{d} , is generated for the whole input. Finally, the actual classification is performed based on the values of this single *confidence vector*, \vec{d} , using some policy—for example, selecting the category with the higher *confidence value*. Note that using these *confidence vectors* in the hierarchy of blocks, it is quite straightforward for SS3 to visually justify the classification if different blocks of the input are colored in relations to their values, as can be seeing on an live demo available at <http://tworld.io/ss3> in which users can try out SS3 for topic categorization. This is quite relevant when it comes to health-care

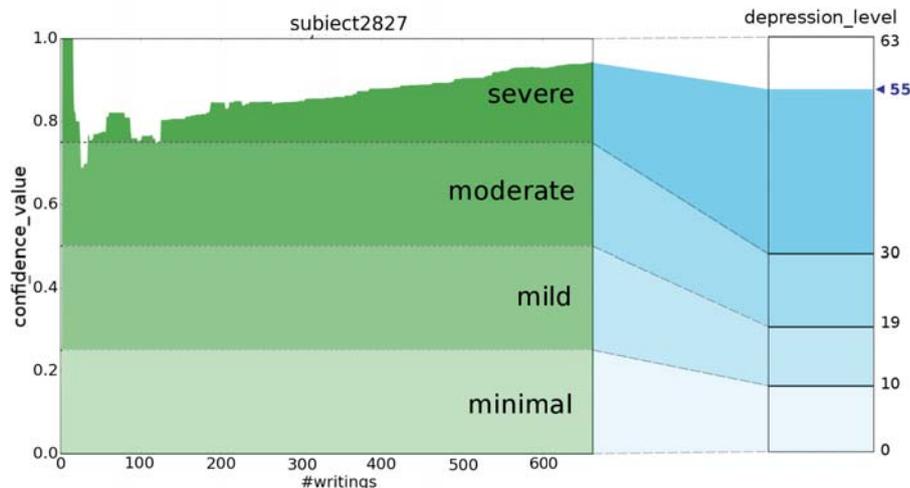


Fig. 1: Diagram of the *dep_level* computation process for subject 2827. As reader can notice, after processing all the subject's writings, the final *confidence_value* (0.941) was mapped into its corresponding *dep_level* (55).

systems, specialists should be able to manually analyze classified subjects and this type of visual tools could be really helpful.

We used the *addition* as the *summary operators* for generating the *confidence vectors* for all the levels, i.e $\oplus_j = \text{addition}$ for all j , which simplified the classification process to the summation of all words' $\vec{g}v$ vectors read so far, in symbols, for every subject s :

$$\vec{d}_s = \sum_{w \in WH_s} \vec{g}v(w) \quad (1)$$

where WH_s is the subject's writing history. Note that for this task, \vec{d}_s was a vector $\langle d_{pos}, d_{neg} \rangle$ with only two components, one for the "depressed" class, d_{pos} , and the other for the "non-depressed" one, d_{neg} . In [3], early classification of subjects was carried out by analyzing how this *confidence vector* changed over time (i.e. as more posts were processed).

3.2 Filling in the BDI questionnaires

As mentioned in section 2, the task was quite difficult, since it was not a single "yes or no" problem but a problem involving multiple decisions, one for each one of the 21 questions. To make things even harder, no training data was released either. Fortunately, early depression detection is a task we had some previous experience working with in previous CLEF's eRisk challenges[5][4][3]. Therefore,

Table 2: Summary of the eRisk 2018 depression task’s dataset

	Depressed	Control
No. of subjects	135	752
No. of posts	49,557	481,837
Avg. No. of posts per subject	367.1	640.7
Avg. No. of posting days	586.43	625
Avg. No. of words per post	27.4	21.8

we decided to train SS3 using the dataset for the eRisk 2018 depression detection task[8] (details are given in Table 2). Additionally, we used the same hyper-parameters used in [3], i.e. $\lambda = \rho = 1$ and $\sigma = 0.455$, for which SS3 showed to have state-of-the-art performance on depression detection.

However, the main problem was deciding how to turn this trained “yes or no” classifier into a classifier capable of filling BDI questionnaires. We came up with the idea of using the *confidence vector*, \vec{d} in Equation 1, to somehow infer a BDI depression level between 0 and 63. To achieve this, first, we converted the *confidence vector* into a single *confidence value* (cv) normalized between 0 and 1, by applying the following equation:

$$cv = \frac{d_{pos} - d_{neg}}{d_{pos}} \quad (2)$$

Then, after SS3 classified a subject, the obtained cv value was mapped into a region/category c , one for each BDI depression category ($c \in \{0, 1, 2, 3\}$). This was carried out by the following equation:

$$c = \lfloor cv \times 4 \rfloor \quad (3)$$

And finally, the subject depression level was predicted by mapping the percentage of cv left inside the predicted c region to its corresponding BDI depression level range (e.g. $(0.5, 0.75] \rightarrow [19, 29]$ for $c = 2 =$ “moderate depression”) by computing the following:

$$dep_level = min_c + \lfloor (max_c - min_c + 1) \times (cv \times 4 - c) \rfloor \quad (4)$$

Where min_c and max_c are the lower and upper bound for category c , respectively (e.g. 19 and 29 for “moderate depression” category).

In order to clarify the above process, we will illustrate it with the example shown in Figure 1. First, SS3 processed the entire writing history computing the *confidence value* (given by Equation 2) and then, the final cv value (0.941) was used to predict the depression category, “severe depression” ($c = 3$), by using the

Equation 3. Finally, the depression level was computed by the mapping given by Equation 4, as follows:

$$\begin{aligned}
 dep_level &= 30 + \lfloor (63 - 30 + 1) \times (0.941 \times 4 - 3) \rfloor \\
 &= 30 + \lfloor 34 \times (3.764 - 3) \rfloor \\
 &= 30 + \lfloor 34 \times 0.764 \rfloor \\
 &= 30 + 25 = \mathbf{55}
 \end{aligned} \tag{5}$$

At this point we have transformed the output of SS3 from a 2-dimensional vector, \mathbf{d} , into a BDI depression level (a value from 0 to 63). However we have not covered yet how to actually answer the 21 questions in the BDI questionnaire using this *depression level*. Regardless the method, we decided that for all those users whose *dep_level* was less or equal to 0, all the BDI questions were answered with 0. For the other users we applied different methods since every participating team was allowed to use up to five different models (called “runs”) to carry out the task. Thus, we use five different methods to accomplish this task, as described below:

- *UNSLA*: using the predicted *dep_level* our model filled the questionnaires answering the answer number $\lfloor \frac{dep_level}{21} \rfloor$ on each question. If this division had a remainder, the remainder points were randomly scatter so that the sum of all the answers always matched the predicted depression level given by SS3.
- *UNSLB*: this time, only the predicted category, c , was used. Our model filled the questionnaire randomly in such a way that the final depression level always matched the predicted category, c . Compared to the following three ones, these two models were the ones with the worst performance.
- *UNSLC*: this model, and the following, were more question-centered. Once again, as in UNSLA, our model filled the questionnaires answering the expected number derived from the predicted depression level ($\lfloor \frac{dep_level}{21} \rfloor$). But this time, answering this number only on questions for which a “textual hint” for a possible answer was found in the user’s writings, and randomly and uniformly answered between 0 and $\lceil \frac{dep_level}{21} \rceil$ otherwise. To find this “textual hint”, our model split the user’s writings into sentences and searched for the co-occurrence of the words “I” or “my” with at least one word matching a regular expression specially crafted for each question.⁷ This method obtained the best AHR (41.43%) and the second-best DCHR (40%).
- *UNSLD*: the same as the previous one, but not using the “textual hints”, i.e. always answering every question randomly and uniformly between 0 and $\lceil \frac{dep_level}{21} \rceil$. This model was mainly used only with the goal of measuring the actual impact of using these “textual hints” to decide which questions should be answered with the expected answer ($\lfloor \frac{dep_level}{21} \rfloor$).

⁷ e.g. “(sad)|(unhappy)” for question 1, “(future)|(work out)” for question 2, “fail\w*” for question 3, “(pleasure)|(enjoy)” for question 4, etc.

Table 3: Results for eRisk 2019 Task 3.

Run	AHR	ACR	ADODL	DCHR
BioInfo@UAVR	34.05%	66.43%	77.70%	25%
BiTeM	32.14%	62.62%	72.62%	25%
CAMH GPT nearest unsupervised	23.81%	57.06%	81.03%	45%
CAMH GPT supervised.181 features.58hr	35.47%	68.33%	75.63%	20%
CAMH GPT supervised.769 features.55hr	36.43%	67.22%	72.30%	20%
CAMH GPT supervised.949 features.75hr	36.91%	69.13%	75.63%	15%
CAMH LIWC supervised SVM	35.95%	66.59%	75.48%	25%
Fazl	22.38%	56.27%	72.78%	5%
Illinois	22.62%	56.19%	66.35%	40%
ISIKol multiSimilarity-5000-Dtac-Qtac	29.76%	57.94%	74.13%	25%
ISIKol-bm25-1.2-0.75-5000-Dtac-Qtac	29.76%	57.06%	72.78%	25%
ISIKol-lm-d-1.0-5000-Dtac-Qtac	30.00%	57.94%	73.02%	15%
Kimberly	38.33%	64.44%	66.19%	20%
UNSLA	37.38%	67.94%	72.86%	30%
UNSLB	36.93%	70.16%	76.83%	30%
UNSLC	41.43%	69.13%	78.02%	40%
UNSLD	38.10%	67.22%	78.02%	30%
UNSLE	40.71%	71.27%	80.48%	35%
Random (avg 1000 repetitions)	23.98%	58.55%	77.78%	33.55%

- *UNSLE*: the same as previous one, but this time not using a uniform distribution. More precisely, from the overall depression level predicted by SS3, once again the expected answer was computed ($\lfloor \frac{dep_level}{21} \rfloor$) and, depending on the value of the expected answer, actual answers were given using the following probability distributions:

$$P(0|0) = 0.9; P(1|0) = 0.1; P(2|0) = 0; P(3|0) = 0;$$

$$P(0|1) = 0.2; P(1|1) = 0.6; P(2|1) = 0.1; P(3|1) = 0.1;$$

$$P(0|2) = 0.15; P(1|2) = 0.25; P(2|2) = 0.5; P(3|2) = 0.1;$$

$$P(0|3) = 0.1; P(1|3) = 0.2; P(2|3) = 0.3; P(3|3) = 0.4;$$

where $P(A|B)$ means the “probability of answering A given that the expected answer is B”. Note that, unlike uniform distribution (used in UNSLD), when using these probability distributions the expected answer is more likely to be selected over the other ones. This model obtained the best ACR (71.27%) and the second-best AHR (40.71%) and ADODL (80.48%, best was only 0.54% above).

4 Evaluation Results

The task’s results are shown in Table 3. As mentioned above, we obtained the best AHR (41.43%) and ACR (71.27%), and the second-best ADODL (80.48%) and

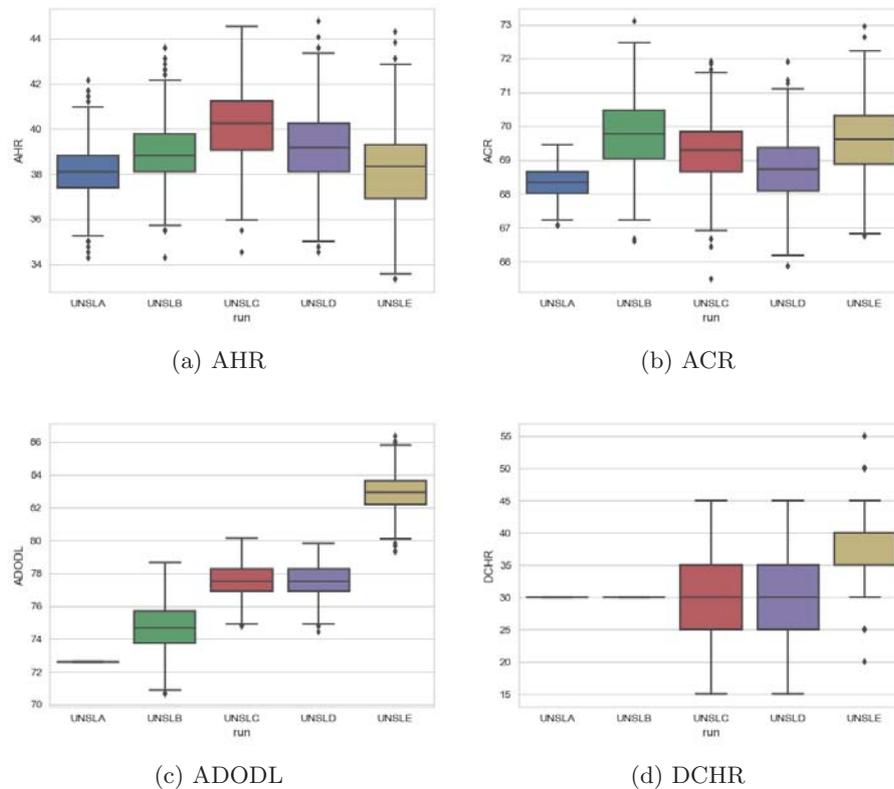


Fig. 2: Box plots for each measure and run.

DCHR (40%), best DCHR and ADODL were obtained by CAMH[1]. However, since most of our models' answers are stochastically generated, it implies that all of these measures are also stochastically generated.⁸ The natural question in cases like this is “How do we know these results properly represent our models performance and we did not obtained them just by pure chance?”. In order to clarify this, once the eRisk finished and the golden truth was released, we run each model 1000 times and calculated the values for AHR, ACR, ADODL and DCHR each time.⁹ After this process finished, we ended up with a sample of 1000 values for each measure and model, which we then used to produce the box plots shown in Figure 2. Analyzing both the values in the table and the box plots one can notice that, in fact, when we participated we had a little bit of bad luck, specially for UNSLE's ADODL, since one can see in Figure 2c that the actual

⁸ Only ADODL and DCHR for UNSLA and DHR for UNSLB are deterministically determined by $depression_{level}$ and c .

⁹ Just as if we had participated 1000 times in this task.

value we obtained (80.84%) is almost a lower bound outlier. Another important thing that can be seen in Figure 2a is that the use of “textual hints”, in UNSLC, really improved the Average Hit Rate (AHR) but did not impact on the other measures (as seen in the other figures). In Figure 2c we can see that UNSLE was considerably the best method to estimate the overall depression level since its ADODL takes values within a range that is quite above the others. Additionally, another important aspect is that, looking at the range of values each method takes, for the different measures, in Figure 2, we can see that the obtained values would be among the best ones, even in the worst cases (compared against the other participant’s).

5 Conclusions and Future Work

In previous scenarios, machine learning models have shown that, in fact, language usage can provide strong evidence in detecting depressive people, since these models have to learn to characterize depression through language use. The work presented in this article is a first step towards measuring finer grain relationships between these both aspects, namely, we studied how the language usage could be connected with the *severity level* of depression. We tested our proposal by participating in the eRisk 2019 T3 task. Obtained results were quite promising and showed us that could be a strong, and somewhat direct, relationship within these both aspects —i.e. it could really be a relationship between how subjects write, what words they use, and the actual depression level they have. Finally, since all the methods we used are based on the depression level predicted by SS3, results also showed us that SS3 correctly inferred the depression level (calculated by Equation 4) from the textual evidence accumulated while processing the user’s writings, i.e. SS3 correctly valued words in relation to each category (depressed and non-depressed).

For future work we will try to get access to a bigger test set, since, although results were quite promising, more data is needed to draw better and more robust conclusions. Additionally, since at the time this paper was written information regarding the other participant’s models were not yet given, once this information is released, we plan to make a better and more qualitative analysis of the results by comparing our models against the other ones in more details. In fact, once we have access to this information, we also plan to explore different model variations to improve our predicted depression level.

References

1. Abed-Esfahani, P., Howard, D., Maslej, M., Patel, S., Mann, V., Goegan, S., French, L.: Transfer learning for depression: Early detection and severity prediction from social media postings. In: Experimental IR Meets Multilinguality, Multimodality, and Interaction. 10th International Conference of the CLEF Association, CLEF 2019. Springer International Publishing, Lugano, Switzerland (2019)
2. Beck, A.T., Ward, C.H., Mendelson, M., Mock, J., Erbaugh, J.: An inventory for measuring depression. *Archives of general psychiatry* 4(6), 561–571 (1961)

3. Burdisso, S.G., Errecalde, M., y Gómez, M.M.: A text classification framework for simple and effective early depression detection over social media streams. *Expert Systems with Applications* **133**, 182 – 197 (2019). <https://doi.org/10.1016/j.eswa.2019.05.023>, <http://www.sciencedirect.com/science/article/pii/S0957417419303525>
4. Errecalde, M.L., Villegas, M.P., Funez, D.G., Ucelay, M.J.G., Cagnina, L.C.: Temporal variation of terms as concept space for early risk prediction. In: *Experimental IR Meets Multilinguality, Multimodality, and Interaction*. 8th International Conference of the CLEF Association, CLEF 2017. Springer International Publishing, Dublin, Ireland (2017)
5. Funez, D.G., Ucelay, M.J.G., Villegas, M.P., Burdisso, S.G., Cagnina, L.C., Montes-y Gómez, M., Errecalde, M.L.: UNSL's participation at erisk 2018 lab. In: *Experimental IR Meets Multilinguality, Multimodality, and Interaction*. 9th International Conference of the CLEF Association, CLEF 2018. Springer International Publishing, Avignon, France (2018)
6. Losada, D.E., Crestani, F.: A test collection for research on depression and language use. In: *International Conference of the Cross-Language Evaluation Forum for European Languages*. pp. 28–39. Springer (2016)
7. Losada, D.E., Crestani, F., Parapar, J.: erisk 2017: Clef lab on early risk prediction on the internet: Experimental foundations. In: *International Conference of the Cross-Language Evaluation Forum for European Languages*. pp. 346–360. Springer (2017)
8. Losada, D.E., Crestani, F., Parapar, J.: Overview of erisk: Early risk prediction on the internet. In: *International Conference of the Cross-Language Evaluation Forum for European Languages*. pp. 343–361. Springer (2018)
9. Losada, D.E., Crestani, F., Parapar, J.: Overview of eRisk 2019: Early Risk Prediction on the Internet. In: *Experimental IR Meets Multilinguality, Multimodality, and Interaction*. 10th International Conference of the CLEF Association, CLEF 2019. Springer International Publishing, Lugano, Switzerland (2019)
10. National Center for Health Statistics: Mortality in the United States, 2017. <https://www.cdc.gov/nchs/products/databriefs/db328.htm> (2019), [Online; accessed 13-April-2019]
11. World Health Organization: Preventing suicide: a global imperative. WHO (2014)
12. World Health Organization: Depression and other common mental disorders: global health estimates. WHO (2017)

Un estudio comparativo de Bases de Datos Relacionales y Bases de Datos NoSQL

Luciano Marrero¹ , Verena Olsowy¹ , Pablo Thomas¹ , Lisandro Delia¹ , Fernando Tesone¹ , Juan Fernandez Sosa¹ , Patricia Pesado¹ 

¹ Instituto de Investigación en Informática LIDI
Facultad de Informática - Universidad Nacional de La Plata – Argentina
Centro Asociado Comisión de Investigaciones Científicas de la Provincia de Buenos Aires

{lmarrero, volsowy, pthomas, ldelia, ftesone, jfernandez,
ppesado}@lidi.info.unlp.edu.ar

Resumen: las bases de datos relacionales han demostrado cubrir las necesidades de almacenamiento de información en una gran variedad de soluciones informáticas. No obstante, cuando estas bases de datos comenzaron a ser utilizadas, los esquemas y el crecimiento del volumen de datos era previsible en el tiempo. Actualmente, la realidad es muy distinta, existen aplicaciones que generan enormes volúmenes de información y de forma no estructurada. Ante esta realidad, se utilizan nuevas formas de almacenamiento denominadas bases de datos no relacionales o NoSQL. En este trabajo se compara y analiza el uso de bases de datos relacionales y NoSQL en diferentes casos de estudio con diferentes esquemas y volúmenes de información.

Keywords: Bases de Datos Relacionales, Bases de Datos NoSQL, Almacenamiento Documental, Almacenamiento de Familia de Columnas.

1 Introducción

El concepto de almacenamiento y procesamiento de datos relacionado con las ciencias informáticas se remonta al siglo XIX, al año 1884, donde Herman Hollerith [1], creó una máquina automática que perforaba una cinta de papel dividida en espacios, logrando así, persistir datos binarios [1]. Para fines del siglo XIX aparece el almacenamiento magnético, el cual dió origen en el siglo XX, a las cintas magnéticas (1950) y a los discos magnéticos (1960) [16]. No obstante, hasta entonces, los datos eran almacenados en sistemas de archivos secuenciales, complejos de administrar a medida que crecían en tamaño [15, 16]. El concepto de base de datos se utiliza por primera vez en el año 1963 (en un simposio, California, EEUU). En esa misma década, con el auge de los discos magnéticos, se da origen a la primera generación de bases de datos, las bases de datos de red y las bases de datos jerárquicas [11, 16]. En la década del 70, Edgar Frank Codd, publica un artículo (A Relational Model of Data for Large Shared Databanks) en el que aplicaba conceptos matemáticos de álgebra relacional para el tratamiento de grandes volúmenes de datos [16]. Este trabajo, dio

origen a la segunda generación de bases de datos, las bases de datos relacionales. En sus inicios, los sistemas relacionales fueron fáciles de utilizar pero poco prácticos por su lentitud. Esto cambió en la década del 80 con el desarrollo de hardware más potente, logrando que las bases de datos relacionales se consoliden en el mercado empresarial [13, 14]. Asociado al modelo relacional surge el lenguaje SQL (Structured Query Language), basado en álgebra y cálculo relacional, este lenguaje permite administrar y recuperar información de estos sistemas de base de datos. Posteriormente, para mediados de la década del 80, SQL se convierte en estándar.

Para fines de la década del 80 y principios de los 90, se comienza a utilizar un nuevo estilo de programación, llamado programación orientada a objetos (OOP, por sus siglas en inglés). Las estructuras de datos generadas por estos lenguajes son complejas y difíciles de almacenar en los motores de bases de datos relacionales. En consecuencia, surge una nueva categoría de sistemas de bases de datos, los sistemas de bases de datos orientados a objetos (ODBMS, por su sigla en inglés) [16].

Desde la década de los 90 hasta la actualidad, los cambios tecnológicos tienen un ritmo vertiginoso. Con el auge de la comunicación a través de internet, el volumen de datos a administrar ha crecido de manera exponencial. Actualmente, considerar las bases de datos solamente como un repositorio de información, puede llevar a una administración incorrecta de los datos. La dinámica en la generación de información ha cambiado, entidades y personas utilizan aplicaciones de software para compartir y difundir información. En respuesta a este avance, hace algunos años surgió un nuevo grupo de bases de datos, las bases de datos no relacionales o bases de datos NoSQL (no solo SQL).

En [29] se realizó un estudio comparativo de rendimiento entre 3 motores de Bases de Datos NoSQL (MongoDB, Cassandra y HBase), mientras que en [30] se realiza un estudio similar entre el motor de bases de datos relacional Oracle y el motor MongoDB.

Este trabajo se centra en 3 pruebas de rendimiento para 4 casos de estudio, en 3 motores de bases de datos distintos: MySQL como ejemplo de motor de base de datos relacional; MongoDB y Apache Cassandra como ejemplos de bases de datos no relacionales, totalizando así, un conjunto de 36 escenarios distintos.

A partir de la sección 2 el trabajo se organiza del siguiente modo, se detalla brevemente el panorama actual de las bases de datos relacionales, en la sección 3 se brinda una introducción al concepto de bases de datos NoSQL, en la sección 4 se explican características de los motores de bases de datos seleccionados; en la sección 5 se resumen los resultados obtenidos y finalmente se presentan las conclusiones y trabajo futuro.

2 Panorama actual de las bases de datos relacionales.

Los Sistemas de Gestión de Bases de Datos Relacionales (RDBMS, por sus siglas en inglés) hacen uso de transacciones con el fin de preservar la integridad de las bases de datos. El uso de transacciones afecta el rendimiento en un entorno concurrente debido al bloqueo de datos para preservar las propiedades ACID (Atomicidad,

Consistencia, Aislamiento, Durabilidad). [3, 10, 12]. Por otra parte, dado que se necesita administrar grandes volúmenes de información, los sistemas necesitan ser escalables. La escalabilidad en un sistema de base de datos, se refiere a la posibilidad de mejorar sus prestaciones agregando nuevos recursos. La escalabilidad vertical es económicamente costosa y tiene un límite. La escalabilidad horizontal es difícil de aplicar debido al modelo rígido y estructurado que presentan las bases de datos relacionales. Estas bases de datos se constituyen de un conjunto de tablas, en las cuales la consulta de información puede implicar relacionar varias tablas, y a medida que el volumen de datos crece, realizar operaciones de JOIN entre tablas afecta notablemente el rendimiento. [13, 14, 15].

3 Introducción a las Bases de Datos NoSQL.

Las bases de datos NoSQL responden a la evolución en el almacenamiento de la información y no son exactamente un tipo de bases de datos, sino que son un conjunto de tipos de bases de datos. NoSQL se distingue de los tradicionales sistemas de gestión de bases de datos relacionales en diversos aspectos, siendo el más notorio, el de no poseer un lenguaje de consulta estructurado (SQL) como lenguaje principal. Además, no requieren de una estructura fija y tabular, no soportan operaciones JOIN, no garantizan por completo las propiedades de ACID, y en general son adecuadas para la escalabilidad horizontal [2, 3]. Para NoSQL es necesario sacrificar algunas características de las bases de datos relacionales para obtener nuevas prestaciones o mejoras en algunos aspectos sin afectar el rendimiento del sistema. Para lograr esto, se propone una visión más liviana y optimista, aceptando que la consistencia en la base de datos esté en un estado de flujo o que la disponibilidad se logre mediante el apoyo a fallas. [2, 4].

NoSQL propone un sistema llamado “BASE (Básicamente Disponible, Estado Suave, Consistencia Eventual)” que es diferente a las propiedades de ACID de las bases de datos relacionales. En resumen, a través de las propiedades BASE se logra disponibilidad básica (Base Availability), esto significa que el sistema se encontrará disponible la mayoría del tiempo. Con el estado débil (Soft State) el sistema se vuelve más flexible en cuanto a consistencia y con la consistencia eventual (Eventual Consistency) se garantiza que el sistema eventualmente se volverá consistente.[3, 7]

Existen cuatro categorías de almacenamiento para bases de datos NoSQL:

Almacenamiento Clave/Valor: simples en cuanto a su implementación, almacenan datos como un conjunto de pares “clave/valor” (key-value). La clave representa un identificador único que puede retornar un objeto complejo y arbitrario de información, denominado valor (value). Por ejemplo, Redis y Amazon DynamoDB, entre otros, implementan este tipo de almacenamiento [23, 24].

Almacenamiento Documental: el concepto central de este tipo de almacenamiento es el documento. Una base de datos NoSQL documental, almacena, recupera y gestiona datos de documentos. Estos documentos encapsulan y codifican datos o información bajo algún formato estándar (XML, YAML, JSON, BSON). Por

ejemplo, MongoDB y Apache CouchDB, entre otros, son implementaciones de bases de datos documentales [20, 25].

Almacenamiento de Familia de Columnas: en este tipo de almacenamiento los datos se encuentran organizados por columnas, en lugar de por filas. Las bases de datos que utilizan esta forma de almacenamiento tienden a ser un híbrido entre las clásicas bases de datos relacionales y la tecnología orientada a columna. Por ejemplo, Cassandra y Apache HBase, entre otros, utilizan este tipo de almacenamiento [22, 26].

Almacenamiento de Grafos: en este tipo de almacenamiento se representa la base de datos bajo el concepto de un grafo. Permite almacenar la información como nodos de un grafo y sus respectivas relaciones con otros nodos, y se aplica la teoría de grafos para recorrer la base de datos; son muy útiles para almacenar información en modelos con muchas relaciones entre distintas entidades o nodos. Por ejemplo, Neo4j y OrientDB, entre otros, permiten este tipo de almacenamiento [27, 28].

4 Motores de Bases de Datos utilizados en los casos de estudio

Para realizar la experimentación presentada en este trabajo, se optó por MySQL (un motor de base de dato relacional), MongoDB (motor con almacenamiento documental) y Apache Cassandra (motor que combina un tipo de almacenamiento clave-valor con familia de columnas), estos últimos son los 2 motores de bases de datos NoSQL más populares [17, 19, 21, 22].

La elección de MySQL se debe a que es uno de los sistemas de base de datos relacional más reconocido y popular del mercado con licencia dual (licencia pública general y licencia comercial). MySQL es utilizado por numerosas empresas a nivel mundial, entre ellas se destacan, Facebook, Twitter, YouTube, GitHub, Uber, Latam Airlines, Booking.com, Spotify, entre otras. [17, 18, 19]

La elección de MongoDB se debe a que es la base de datos NoSQL documental con mayor popularidad. MongoDB es de código abierto, multiplataforma, escrita en C++, posee almacenamiento documental y prioriza la disponibilidad. Es una base de datos adecuada para contextos donde los datos no siempre poseen una estructura fija. En la actualidad, diferentes empresas (Coinbase, SEGA, Royal Bank of Scotland, Telefónica, Facebook, Nokia, entre otras) hacen uso de MongoDB en alguna parte de su infraestructura de datos. [17, 20, 21]

La elección de Apache Cassandra se basa en que es la base de datos NoSQL con almacenamiento en familia de columnas con mayor popularidad. Apache Cassandra es multiplataforma, diseñada para trabajar de forma distribuida con grandes volúmenes de datos. Su escalabilidad lineal, disponibilidad y su tolerancia a fallas la convierten en una plataforma adecuada para datos de misión crítica. Cassandra se origina en el año 2008, en Facebook, por una necesidad de atender búsquedas de su mensajería en tiempo real ante un volumen muy elevado de datos [17, 22].

5 Casos de estudio, pruebas de comparación y rendimiento.

Para representar la información de cada caso de estudio, se realizó previamente el modelo conceptual de datos utilizando un diagrama de entidad relación. Posteriormente, se realizó la derivación al esquema físico correspondiente a cada motor de base de datos.

En MySQL se generó el modelo relacional.

En MongoDB las entidades se representan mediante colecciones de documentos BSON y las relaciones se expresan mediante documentos embebidos, esto último se debe a que en MongoDB no existe el concepto de JOIN [4, 5, 20].

El caso de Apache Cassandra, los datos no se almacenan en términos de entidades, sino que se representan en términos de consultas, por lo tanto, es importante definir las consultas que se realizarán sobre el sistema.[6, 8, 22].

La ejecución de todas las pruebas se realizaron utilizando la misma infraestructura de hardware y sistema operativo. Se utilizó la versión 5.7 del motor de base de datos MySQL, la versión 4.0 del motor de base de datos MongoDB y la versión 3.11 del motor de bases de datos Apache Cassandra. Para cada caso de estudio el volumen de datos utilizado se generó de forma aleatoria.

A continuación, se detallan los resultados obtenidos de las pruebas realizadas en los 4 casos de estudio. Sólo en el primero de ellos se presenta el modelo conceptual, los esquemas físicos, el código de las consultas y los tiempos obtenidos; y en todos los casos solo se expresan los tiempos resultantes.

5.1 Caso de estudio 1

Se representa una empresa que nuclea reservas de viajes online para un conjunto de agencias de turismo. Actualmente, se llevan registradas 1.260.000 reservas online.

La figura 1 presenta el modelo entidad-relación del problema en cuestión.

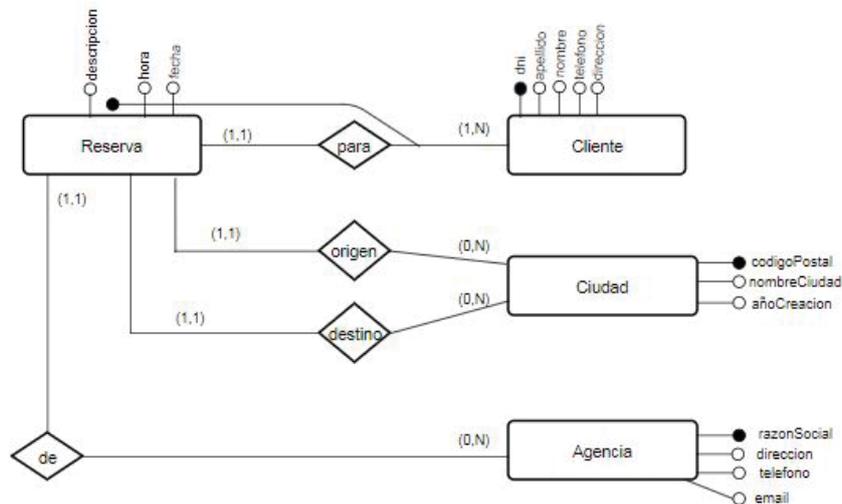


Figura 1. Esquema conceptual expresado mediante un Diagrama Entidad - Relación.

La figura 2 presenta el esquema relacional derivado del modelo conceptual expresado en la figura 1.

AGENCIA = (RAZONSOCIAL(PK), dirección, teléfono, e-mail)
CIUDAD = (CODIGOPOSTAL(PK), nombreCiudad, añoCreación)
CLIENTE = (DNI(PK), nombre, apellido, teléfono, dirección)
RESERVA = ((FECHA, HORA, DNI(FK))(PK), cpOrigen(FK), cpDestino(FK), razonSocial(FK), descripción)

Figura 2. Esquema Relacional.

La figura 3 muestra el formato de los datos en un documento para MongoDB.

```
Reservas:{
  _id,
  hora,
  fecha,
  agencia: { email, telefono, direccion, razonSocial },
  cliente : { dni, nombre, apellido, telefono, direccion},
  descripcion,
  ciudadOrigen : {anioCreacion, codigoPostal, nombreCiudad},
  ciudadDestino : {anioCreacion,codigoPostal, nombreCiudad}
}
```

Figura 3. Esquema para MongoDB.

En Cassandra, el esquema físico de información se expresa de acuerdo a las consultas a realizar. La figura 4 plantea dicho esquema físico.

RESERVAS = ((dni, fecha, hora)(PK), apellido, nombre, ciudaddestino, ciudadorigen, razonsocial)
PROBLEMAS = ((dni, fecha, hora)(PK), ciudaddestino, ciudadorigen, descripción, razonsocial)

Figura 4. Esquema físico de Apache Cassandra.

A continuación, se plantean 3 consultas para las pruebas de rendimiento en este caso de estudio.

“C1: Reportar los datos y la cantidad de reservas realizadas para cada cliente.”

“C2: Reportar los datos de reservas en los que se describe algún problema.”

“C3: Reportar las reservas realizadas por un cliente en particular.”

En la figura 5, 6 y 7 se muestran las soluciones de las 3 consultas para cada motor de base de datos.

MySQL:

```
select c.dni, c.nombre, c.apellido, count(r.dni) as cantidad
from cliente c inner join reserva r on (c.dni = r.dni)
group by c.dni, c.nombre, c.apellido, c.telefono, c.direccion
```

CASSANDRA:

```
select dni, nombre, apellido, count(dni) as cantidad
from reservas
group by dni
```

MONGO:

```
db.reservas.aggregate([ {$group : { _id:{ dni : "$cliente.dni",
nombre:"$cliente.nombre", apellido: "$cliente.apellido"},
"cantidad": { $sum: 1 } } } ])
```

Figura 5. Implementación de C1.

MySQL:

```
select r.fecha, r.hora, co.nombreCiudad, cd.nombreCiudad,
r.razonSocial, r.descripcion
from reservas r
inner join ciudad co on co.codigoPostal=r.cpOrigen
inner join ciudad cd on cd.codigoPostal=r.cpDestino
where descripcion like "%problema%"
```

CASSANDRA:

```
select * from problemas
```

MONGO:

```
db.reservas.find({descripcion:{$regex: /Problema/,
$options:"i"}}, {fecha:1, hora:1,
"ciudadOrigen.nombreCiudad":1, "ciudadDestino.nombreCiudad":1,
"agencia.razonSocial":1, descripcion:1, _id:0 })
```

Figura 6. Implementación de C2.

MySQL

```
select r.fecha, r.hora, co.nombreCiudad, cd.nombreCiudad, r.razonSocial
from reservas r
inner join ciudad co on co.codigoPostal=r.cpOrigen
inner join ciudad cd on cd.codigoPostal=r.cpDestino
where r.dni = 28764545
```

CASSANDRA

```
select fecha, hora, ciudadOrigen, ciudadDestino, razonSocial
from reservas
where dni = 28764545
```

MONGO

```
db.reservas.find({"cliente.dni":28764545}, {fecha:1, hora:1,
"ciudadOrigen.nombreCiudad":1, "ciudadDestino.nombreCiudad":1,
"agencia.razonSocial":1, _id:0 })
```

Figura 7. Implementación de C3.

En la tabla 1 se expresan los tiempos obtenidos en segundos de la ejecución de cada consulta.

Tabla 1. Resultados del caso de estudio 1 expresados en segundos

Consultas	MySQL	MongoDB	Cassandra
C1	6,9	34,31	9,83
C2	61,62	14,90	4,73
C3	0,52	0,85	0,06

5.2 Caso de estudio 2

En este caso de estudio se presentan los movimientos de cuentas en una entidad bancaria que ha alcanzado las 4.233.386 movimientos de cuenta.

En la tabla 2 se expresan los tiempos obtenidos en segundos de la ejecución de cada consulta.

Tabla 2. Resultados del caso de estudio 2 expresados en segundos

Consultas planteadas	MySQL	MongoDB	Cassandra
C1: Reportar cantidad de movimientos por cuenta.	22,5	77,28	53,41
C2: Reportar los movimientos donde el concepto contenga la cadena "alquiler de cochera".	224,29	51,52	11,6
C3: Reportar los movimientos realizados desde una cuenta en particular.	0,56	2,1	0,12

5.3 Caso de estudio 3

El tercer caso de estudio plantea un sistema de compras on-line a nivel nacional que ha alcanzado unas 10.473.392 compras.

La tabla 3 expresa los tiempos obtenidos en segundos de la ejecución de cada consulta.

Tabla 3. Resultados del caso de estudio 3 expresados en segundos

Consultas planteadas	MySQL	MongoDB	Cassandra
C1: Reportar la cantidad de compras por cliente.	52,60	145,12	78,6
C2: Reportar las compras que contienen la cadena "Excelente" en su opinión.	768,45	97,96	59,51
C3: Reportar las compras realizadas por un usuario en particular.	1,11	3,5	0,13

5.4 Caso de estudio 4

El cuarto y último caso presenta un registro de las líneas de teléfonos celulares activas para una empresa nacional que posee 20.120.254 líneas telefónicas.

La tabla 4 expresa los tiempos obtenidos en segundos de la ejecución de cada consulta.

Tabla 4. Resultados del caso de estudio 4 expresados en segundos

Consultas planteadas	MySQL	MongoDB	Cassandra
C1: Reportar la cantidad de líneas telefónicas para cada plan.	128,44	198,59	139,23
C2: Reportar las líneas telefónicas que contienen la cadena "Prepago con deuda" en la descripción de su estado.	19484,12	132,69	75,21
C3: Reportar las líneas telefónicas para una localidad en particular.	1,64	3,93	0,18

6 Discusión de los resultados

En la primera consulta para los 4 casos de estudio, se observa que MySQL obtiene un mejor tiempo de respuesta. No obstante, a medida que aumenta el volumen de datos en los distintos experimentos, los tiempos de respuesta se acercan a los obtenidos con Cassandra. Estas consultas requieren el uso de la función de agregación COUNT. MySQL optimiza el uso de las funciones de agregación en relación a los

otros 2 motores [4, 9, 20, 22]. Esta podría ser la justificación por la cual su desempeño es mejor.

Para la segunda consulta en los 4 casos de estudio, Cassandra logra el mejor tiempo de respuesta. Esto podría justificarse debido a que los esquemas físicos en Cassandra son planteados en base a las consultas del sistema. En este caso, Cassandra posee una tabla diseñada en términos de estas consultas. MongoDB está en segundo lugar, dado que realiza el filtro de documentos, pero sin necesidad de operaciones de JOIN. MySQL es el menos eficiente porque el filtro aplicado retorna un conjunto de datos considerable, a los cuales se les aplica operaciones JOIN, afectando de este modo el tiempo de respuesta.

En la tercera consulta, a pesar del incremento en el volumen de información, los tiempos de respuesta poseen una performance similar para todos los casos de estudio, independientemente del volumen de datos. Esto se debe a que en la implementación de la consulta en todos los motores de bases datos se aplica un filtro por un atributo que se encuentra indexado. Finalmente, los tiempo resultantes ubican a Cassandra en el mejor lugar, seguida por MySQL y MongoDB.

7 Conclusiones

Este trabajo se concentra en un estudio comparativo entre diferentes motores bases datos.

En primer lugar, se resume brevemente la evolución de las diferentes formas de almacenamiento de datos. Posteriormente, se plantean las restricciones existentes en las bases de datos relacionales para almacenar grandes volúmenes de información, entre ellos, el manejo de transacciones, las operaciones JOINS de SQL y la dificultad en la escalabilidad. Esta situación origina el análisis y estudio de una alternativa diferente para almacenar información, las bases de datos NoSQL.

Las bases de datos NoSQL no poseen un lenguaje de consulta estructurado, no requieren de una estructura fija y son adecuadas para la escalabilidad horizontal; por lo que mitigan algunos de los problemas que presentan las bases de datos relacionales a través de las propiedades BASE (básicamente disponible, estado flexible, consistencia eventual). Existen 4 tipos de almacenamientos para bases de datos NoSQL: Clave/Valor, Documental, Familia de Columnas y Grafos.

Se plantean 4 casos de estudios cada uno de ellos para diferentes dominios y con distintos volúmenes de datos. Para el primero se ha presentado el desarrollo en forma completa. En cada uno de ellos se plantean 3 modelos consultas para 3 motores de bases de datos MySQL (relacional), MongoDB (NoSQL documental) y Apache Cassandra (NoSQL familia de columnas). Se conformaron así, un conjunto de 36 pruebas, con el objetivo de obtener métricas de performance.

Finalmente, se realiza una discusión de los resultados obtenidos. En línea general se puede concluir que no hay un motor de base de datos que logre el mejor tiempo de respuesta para todas las consultas evaluadas. Los resultados obtenidos para los casos de estudio planteados, dependen del tipo de consulta planteada y el volumen de datos asociado.

Estas conclusiones tienen concordancia con las conclusiones obtenidas en [29] y [30].

8 Trabajo Futuro

Como trabajo futuro, se prevé incorporar nuevas pruebas y reforzar las existentes con nuevos motores de bases de datos relacionales, y las categorías de bases de datos no relacionales no contempladas en este trabajo (clave-valor, orientadas a grafos). Además, se buscará explorar la capacidad de escalamiento horizontal de las bases de datos NoSQL y de las bases de datos relacionales.

Finalmente se experimentará con bases de datos relacionales basados en NewSQL.

9 Bibliografía

1. Herman Hollerith. https://es.wikipedia.org/wiki/Herman_Hollerith. Accedido en agosto 2019.
2. Ajit Singh, Sultan Ahmad. Data Modeling with NoSQL Database. ISBN 978-1072978374 (2019).
3. Aspectos de ingeniería de software y bases de datos para el desarrollo de sistemas de software en escenarios híbridos. XXI Workshop de Investigadores en Ciencias de la Computación (WICC 2019, Universidad Nacional de San Juan). ISBN 978-987-3619-27-4. <http://sedici.unlp.edu.ar/handle/10915/77088>.
4. Jitender Kumar, Varsha Garg. Security analysis of unstructured data in NOSQL MongoDB database. International Conference on Computing and Communication Technologies for Smart Nation (IC3TSN) 2017. <https://ieeexplore.ieee.org/document/8284495>.
5. Benymol Jose, Sajimon Abraham. Exploring the merits of nosql: A study based on MongoDB. International Conference on Networks & Advances in Computational Technologies (NetACT) 2017. <https://ieeexplore.ieee.org/document/8076778>.
6. Shivendra Kumar Pandey, Sudhakar. Context based Cassandra query language. 8th International Conference on Computing, Communication and Networking Technologies (ICCCNT) 2017. <https://ieeexplore.ieee.org/document/8204142>.
7. Roshni Bajpayee, Sonali Priya Sinha y Vinod Kumar. Big Data: A Brief investigation on NoSQL Databases. International Journal of Innovations & Advancement in Computer Science (IJIACS). ISSN 2347. Volume 4, Issue 1. January 2015.
8. Deepak Vohra. NoSQL Web Development with Apache Cassandra. Cengage Learning, 2015. ISBN 9781305576773.
9. Guoxi Wang, Jianfeng Tang. The NoSQL Principles and Basic Application of Cassandra Model. International Conference on Computer Science and Service System 2012. <https://ieeexplore.ieee.org/document/6394574>.
10. Enrique J. Reinoso, Calixto A. Maldonado, Roberto Muñoz, Luis E. Damiano, Maximiliano A. Abrutsky. Base de Datos. Alfaomega. ISBN 978-987-1609-31-4 (2012).
11. Abraham Silberschatz, Henry F. Korth, S. Sudarshan. Fundamentos de Bases de Datos, Quinta Edición. McGraw Hill. ISBN 84-481-2021-3 (2006).

12. Ramez A. Elmasri, Shamkant B. Navathe. Fundamentos de Sistemas de Bases de Datos. Tercera Edición. Pearson (Addison Wesley). ISBN 84-782-9051-6 (2002).
13. C. J. Date. Introducción a los Sistemas de Bases de Datos. Séptima Edición. Pearson Educación. Prentice Hall. ISBN 968-444-419-2 (2001).
14. Carlo Batini, Stefano Ceri, Shamkant B. Navathe. Diseño Conceptual de Bases de Datos, un enfoque de entidades-interrelaciones. Addison-Wesley / Díaz de Santos. ISBN 0-201-60120-6 (1994).
15. Michael J. Folk y Bill Zoellick. Estructuras de Archivos, Un Conjunto de Herramientas Conceptuales. ISBN 0-201-62923-2
16. Burton Grad y Thomas J. Bergin. History of Database Management Systems. <https://ieeexplore.ieee.org/document/5370775>.
17. Ranking de bases de datos según su popularidad. <https://db-engines.com/en/ranking>. Accedido en Agosto 2019.
18. MySQL. <https://www.mysql.com/>. Accedido en agosto 2019.
19. Clientes de MySQL. <https://www.mysql.com/customers/>. Accedido en agosto 2019.
20. MongoDB. <https://www.mongodb.com/>. Accedido en agosto 2019.
21. Empresas que utilizan MongoDB. <https://www.mongodb.com/who-uses-mongodb>. Accedido en agosto 2019.
22. Apache Cassandra. <http://cassandra.apache.org/>. Accedido en agosto 2019.
23. Redis. <https://redis.io/>. Accedido en agosto 2019.
24. Amazon DynamoDB. <https://aws.amazon.com/es/dynamodb/>. Accedido en agosto 2019.
25. Apache CouchDB. <http://couchdb.apache.org/>. Accedido en agosto 2019.
26. Apache HBase. <http://hbase.apache.org/>. Accedido en agosto 2019.
27. Neo4j. <https://neo4j.com/>. Accedido en agosto 2019.
28. OrientDB. <https://orientdb.com/>. Accedido en agosto 2019.
29. Evaluating NoSQL performance: Which database is right for your data?. <https://jaxenter.com/evaluating-nosql-performance-which-database-is-right-for-your-data-107481.html>. Accedido en Septiembre 2019.
30. Una comparación de rendimiento entre Oracle y Mongodb. <http://www.scielo.org.co/pdf/cein/v26n1/v26n1a07.pdf>. Accedido Septiembre 2019.

Una Mejora en Tiempo del Trie de Sufijos

Jesica Cornejo, Darío Ruano, Norma Herrera

Departamento de Informática, Universidad Nacional de San Luis, Argentina,
jncornej@gmail.com, dmruano@unsl.edu.ar, nherrera@unsl.edu.ar

Abstract. Un *trie de sufijos* es un índice para bases de datos de texto que permite resolver eficientemente las operaciones de búsqueda pero que necesita en espacio 10 veces el tamaño del texto indexado. En [14] se propone una nueva representación compacta del trie de sufijos que resulta eficiente en espacio y que permite un posterior paginado del índice. En este artículo presentamos una mejora en tiempo de búsqueda de esta representación compacta del trie de sufijos manteniendo la competitividad en espacio.

Palabras claves: Bases de Datos de Texto, Índices, Trie de sufijos.

1 Introducción

La información disponible en formato digital aumenta día a día su tamaño de manera exponencial. Gran parte de esta información se representa en forma de texto, es decir, secuencias de símbolos que pueden representar no sólo lenguaje natural, sino también música, secuencias de ADN, secuencias de proteínas, etc. Debido a que no es posible organizar una colección de textos en registros y campos, las tecnologías tradicionales de bases de datos (modelo relacional) para almacenamiento y búsqueda de información no son adecuadas en este ámbito.

Una base de datos de texto es un sistema que mantiene una colección grande de texto y que provee acceso rápido y seguro al mismo. Sin pérdida de generalidad, asumiremos que la base de datos de texto es un único texto T que posiblemente se encuentra almacenado en varios archivos.

Una de las búsquedas más comunes en bases de datos de texto es la *búsqueda de un patrón*: el usuario ingresa un string P (*patrón de búsqueda*) y el sistema retorna las ocurrencias del patrón P en el texto T . Para resolver eficientemente estas búsquedas se hace necesario construir un índice que permita acelerar el proceso de búsqueda. Un índice debe dar soporte a dos operaciones básicas de búsqueda de patrón: *count* que consiste en contar el número de ocurrencias de P en T y *locate* que consiste en ubicar todas las posiciones de T donde P ocurre.

Mientras que en bases de datos tradicionales los índices ocupan menos espacio que el conjunto de datos indexados, en bases de datos de texto el índice ocupa más espacio que el texto en sí mismo, pudiendo necesitar de 4 a 20 veces el tamaño del mismo.

Algunos índices reducen el espacio ocupado restringiendo el tipo de búsqueda que se pueden resolver. Así, un *índice orientado a palabra*, permite solamente buscar palabras completas en el texto. Algunos índices orientados a palabras permiten también buscar patrones que sean inicios de palabras. Los índices que son capaces de encontrar

todas las ocurrencias de un patrón dentro del texto se denominan índices orientados a caracteres o *full text indexes*.

Otra alternativa para reducir el espacio ocupado por el índice es buscar una representación compacta del mismo, manteniendo las facilidades de navegación sobre la estructura. En los últimos años se han realizado grandes avances en este ámbito. Las nuevas técnicas de compresión de índices no sólo permiten reducir su tamaño, sino que procesan búsquedas más rápido que la versión sin comprimir [5, 11, 13].

Un *trie de sufijos* es un índice que permite resolver eficientemente las operaciones *count* y *locate* pero que necesita en espacio 10 veces el tamaño del texto indexado. En [14] y [15] se aborda el estudio de técnicas de representación del trie de sufijos con el objetivo de proponer una nueva representación compacta del mismo que resulte eficiente en espacio y que permita un posterior paginado del índice. Los autores muestran que con esta nueva representación se obtienen mejoras significativas en espacio manteniendo un buen desempeño en tiempo. Aun así, la representación clásica del trie de sufijos tiene una leve ventaja en tiempo con respecto a lo propuesto. Pero la nueva representación tiene la ventaja de permitir un posterior proceso de paginado, algo que no se puede hacer de manera eficiente con la representación clásica. En este artículo presentamos una mejora en tiempo de búsqueda de esta nueva versión del trie de sufijos.

Lo que resta del artículo está organizado de la siguiente manera. En la sección 2 presentamos el trabajo relacionado, dando los conceptos necesarios para comprender el artículo. En la sección 3 describimos la representación secuencial para el trie de sufijos en la que está basada este trabajo. La sección 4 y 5 presentan nuestra propuesta de mejora y la evaluación experimental de la misma. Finalizamos en la sección 6 dando las conclusiones y el trabajo futuro.

2 Trabajo Relacionado

2.1 Indexación en Bases de Datos de Texto

Dado un texto $T = t_1, \dots, t_n$ sobre un alfabeto Σ de tamaño σ , donde $t_n = \$ \notin \Sigma$ es un símbolo menor en orden lexicográfico que cualquier otro símbolo de Σ , definimos:

sufijo de T: cualquier string de la forma $T_{i,n} = t_i, \dots, t_n$ con $i = 1..n$.

prefijo de T: cualquier string de la forma $T_{1,i} = t_1, \dots, t_i$ con $i = 1..n$.

Cada sufijo $T_{i,n}$ se identifica unívocamente por i ; llamaremos al valor i *índice del sufijo* $T_{i,n}$. Un patrón de búsqueda $P = p_1 \dots p_m$ es cualquier string sobre Σ .

Entre los índices más populares para búsqueda de patrones encontramos el arreglo de sufijos [10], el árbol de sufijos [17] y el trie de sufijos [4]. Estos índices se construyen basándose en la observación de que un patrón P ocurre en el texto si es prefijo de algún sufijo del texto.

Un *trie de sufijos* [4] es un trie [3] construido sobre el conjunto de todos los sufijos de T . Cada rama está rotulada por un símbolo de Σ y cada hoja representa un sufijo de T . El conjunto de sufijos se obtiene recorriendo todos los caminos posibles desde la raíz hasta una hoja y concatenando los rótulos de las ramas que forman cada uno de esos caminos. En cada nodo hoja se mantiene el índice del sufijo que esa hoja representa. La figura 1 muestra un ejemplo de un texto y su correspondiente trie de sufijos.

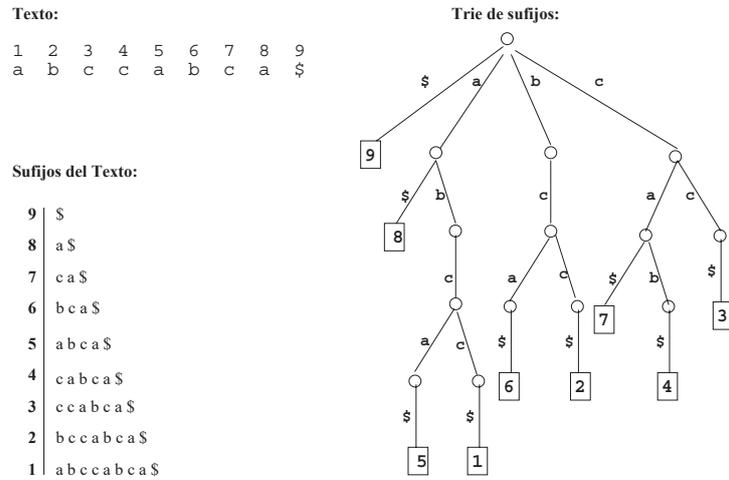


Fig. 1. Un texto, su correspondiente conjunto de sufijos y trie de sufijos.

Una forma de reducir el uso de espacio es reemplazar las ramas del árbol que han degenerado en una lista, por una única rama y agregar a cada nodo interno la longitud de la rama que se ha eliminado. Esta longitud se conoce con el nombre de *valor de salto*. La figura 2 muestra esta modificación para el trie de la figura 1. Esta última versión es la que utilizamos en este trabajo.

Para encontrar todas las ocurrencias de P en T , se busca en el trie utilizando los caracteres de P para direccionar la búsqueda. Se comienza por la raíz y en cada paso, estando en un nodo x con valor de salto j , avanzamos siguiendo la rama rotulada con el j -ésimo caracter de P . Durante este proceso se pueden presentar tres casos:

- que la longitud de P sea menor que j , por lo cual no hay caracter de P para seguir buscando en el árbol. En este caso se compara P con una de las hojas del subárbol con raíz x ; si esa hoja es parte de la respuesta todas las hojas de ese subárbol lo son, caso contrario ninguna lo es.
- que x sea una hoja del árbol y por lo tanto no tiene un valor de salto asignado sino el índice de un sufijo. En este caso se debe comparar P con el sufijo indicado por la hoja para saber si ese sufijo es o no la respuesta.
- que el nodo x no tenga ningún hijo rotulado con el j -ésimo caracter de P . En este caso la búsqueda fracasa.

En cada caso, si la búsqueda es una operación *locate* y es exitosa, hay que recuperar los índices de sufijos contenidos en las hojas que forman la respuesta a la consulta. Si la búsqueda es una operación *count* y es exitosa, basta con contar la cantidad de hojas que forman parte de la respuesta.

2.2 Representaciones Compactas para Árboles

Los árboles ordenados de búsqueda son una de las estructuras de datos más relevantes. La información asociada a cada nodo suele ser importante, e incluso dominante, en

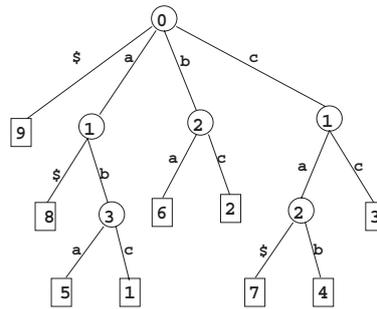


Fig. 2. Representación de Trie de Sufijos usando valores de salto.

el costo de almacenamiento total de un sistema. La cantidad de árboles binarios de n nodos puede ser calculada de la siguiente manera [2]:

$$C_n \equiv \binom{2n}{n} / (n + 1)$$

Una codificación compacta de un árbol binario requerirá entonces como mínimo $\log C_n$ bits. Usando la aproximación de *Stirling* se puede demostrar que $2n$ bits son suficientes para representar un árbol de n nodos [2]. En [7] y [9] se presentan varias técnicas de representación compacta de árboles que alcanzan esta cota de $2n$ bits, sin embargo ninguna de ellas provee las funcionalidades requeridas para navegar el árbol.

La representación de paréntesis [12] es una técnica de representación compacta de árboles que consiste en realizar un barrido preorden del árbol y colocar un paréntesis que abre (un bit en 1) cuando visitamos por primera vez un nodo x y un paréntesis que cierra (un bit en 0) cuando terminamos de barrer el subárbol izquierdo de x . Esta técnica de representación alcanza la cota de $2n$ bits para un árbol de n nodos y permite una implementación eficiente de las funciones requeridas para navegar el árbol.

Para poder navegar el árbol necesitamos, dado un nodo x , saber la posición donde se encuentra su hijo izquierdo (*leftchild*(x)), la posición donde se encuentra su hijo derecho (*rightchild*(x)) y la posición del padre (*parent*(x)). En la representación de paréntesis estas operaciones se pueden resolver a partir de las siguientes operaciones básicas:

***findclose*(i)** : retorna la posición del paréntesis que cierra que corresponde al paréntesis que abre en la posición i .

***findopen*(i)** : retorna la posición del paréntesis que abre que corresponde al paréntesis que cierra en la posición i .

***excess*(i)** : retorna la diferencia del número de paréntesis que abren y el número de paréntesis que cierra desde el principio a la posición i .

***enclose*(i)** : retorna la posición del paréntesis más cercano que encierra al nodo de la posición i

En el caso de árboles r -arios no completos, en el cual cada nodo puede tener un número indefinido de hijos, se realiza un recorrido preorden comenzando desde la raíz, se escribe un paréntesis que abre cuando encontramos por primera vez un nodo x y un paréntesis que cierra cuando se termina de barrer todos los subárboles de x .

Para poder navegar el árbol necesitamos, dado un nodo x , la posición del padre, la posición del i -ésimo hijo de x y la posición de su hermano izquierdo y derecho. Estas operaciones se pueden resolver realizando una adaptación para árboles generales de las operaciones básicas para árboles binarios: $findclose(i)$, $findopen(i)$, $excess(i)$ [12].

3 Representación Compacta de un Trie de Sufijos

La representación habitual de un trie consiste en mantener en cada nodo los punteros a sus hijos, junto con el rótulo correspondiente a cada uno de ellos. Existen distintas variantes de representación que consisten en organizar estos punteros a los hijos sobre una lista secuencial, sobre una lista vinculada o sobre una tabla de hashing [8]. Una de las propuestas de representación que mejor desempeño tiene en memoria principal es la de Kurtz, quien basándose en la idea de la representación sobre una lista vinculada, propuso que cada nodo mantenga un apuntador al primer hijo y almacenar los nodos hermanos en posiciones consecutivas de memoria. Esto permite durante una búsqueda, realizar una búsqueda binaria sobre los rótulos para decidir por cual hijo seguir.

La mayoría de las propuestas existentes mantienen explícitamente la forma del árbol con punteros, los que pueden ser punteros físicos (direcciones de memoria principal) o punteros lógicos (posiciones de un arreglo). En [14] se presenta una nueva representación de un trie de sufijos que permite reducir el espacio necesario para almacenar el índice, eliminando la necesidad de mantener los punteros explícitos a los hijos. Esta representación surge como una extensión a árboles r -arios de la técnica presentada en [6] y tiene la ventaja de permitir un posterior proceso de paginado para manejar eficientemente el trie de sufijos en memoria secundaria [16].

Notar que la información contenida en el trie está compuesta por: la topología del árbol, el rótulo de cada rama, el valor de salto de cada nodo, el grado de cada nodo y el índice del sufijo asociado a cada hoja. En [14] se propone representar de manera secuencial cada una de estas componentes, manteniendo la posibilidad de navegar eficientemente sobre el trie:

- Para la topología del trie de sufijos utilizamos un mapa de bits que mantiene la representación de paréntesis que explicamos en la sección 2.2. Llamaremos a esta secuencia RP .
- Para la representación de los rótulos de cada rama, de los valores de salto de cada nodo y del grado de cada nodo, se utilizan arreglos colocando los elementos que forman cada arreglo en el orden indicado por un barrido preorden del árbol. Esto permite durante la navegación del árbol moverse coordinadamente sobre todas las secuencias que conforman la representación del mismo. Estas secuencias a su vez se comprimen usando códigos DAC (Directly Addressable Variable-Length Code) [1], una técnica de compresión que permite acceso aleatorio y eficiente a cada código en una secuencia de códigos de longitud variable.

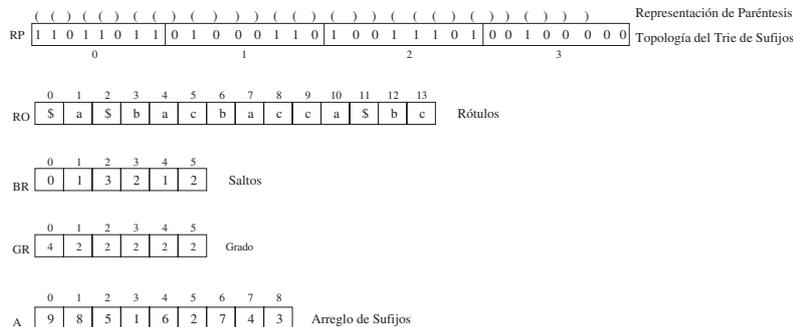


Fig. 3. Representación secuencial del trie de sufijos de la Figura 1.

- Para las hojas también se mantiene un arreglo de índices de sufijos, tomados en orden de izquierda a derecha del árbol.

La figura 3 muestra esta representación para el trie de sufijo de la Figura 1. Las secuencias se muestran sin la compresión DAC a fin de facilitar su entendimiento. Para poder navegar sobre esta representación los autores se basan en el algoritmo que permite navegar sobre una representación de paréntesis [12], adaptándolo para moverse coordinadamente sobre los 5 arreglos. Estos algoritmos necesitan realizar operaciones *findclose*, *excess* y *enclose* sobre secuencias binarias o mapa de bits [12].

4 Mejorando los Tiempos de Búsqueda

Para mejorar los tiempos de resolución de una búsqueda como primer paso se analizó cuánto tiempo insumían las operaciones involucradas en el proceso de navegación en la representación compacta del trie de sufijos. Para ello se realizaron experimentos que calcularan, del tiempo total del búsqueda, que porcentaje correspondía a cada operación sobre la representación de paréntesis.

Los resultados de estos experimentos permitieron determinar que la operación que mas tiempo consume es *findclose*. La Figura 4 muestra los resultados obtenidos para un texto que contiene secuencias de ADN de tamaño 4MB y 6MB. Sobre el eje *x* se han representado los distintos tamaños de patrones de búsqueda y sobre el eje *y* está representado el tiempo insumido. Se ha graficado el tiempo total de búsqueda y el tiempo de *findclose*. Como puede observarse un 99% del tiempo de búsqueda es insumido por la operación *findclose*.

Este comportamiento se debe a que, cuando el algoritmo de búsqueda navega la representación compacta del trie de sufijos y necesita pasar de un nodo *y* a su hermano derecho, la operación *findclose* debe realizar barridos grandes en *RP*, la representación de paréntesis del trie. Recordemos que entre un nodo *y* y su hermano derecho se encuentra toda la representación del subtrie que tiene como raíz a *y*. Mientras mas cerca de la raíz se encuentre *y* mas grande será el subárbol que depende de él.

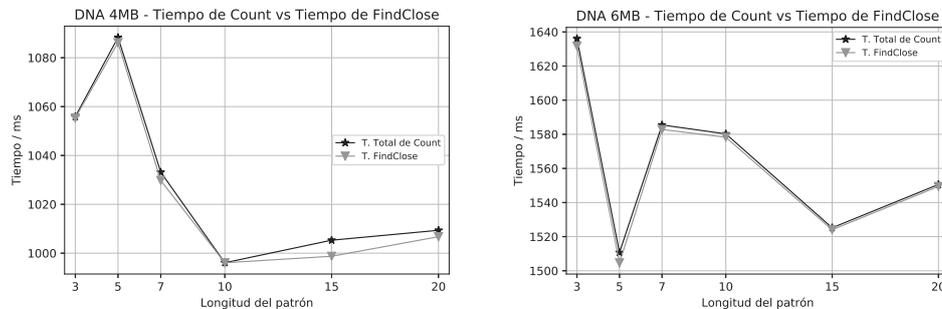


Fig. 4. Tiempo de *findclose* vs tiempo total de búsqueda.

Una forma de evitar estos barridos grandes sobre *RP*, es mantener para cada nodo *y* la posición donde se encuentra su paréntesis que cierra en *RP*. Esta idea es la que usamos para crear una nueva estructura que llamaremos *ParentClose*. Esta estructura tiene varios niveles y en cada nivel mantiene información que evita los barridos secuenciales grandes dentro de *RP*.

Sea *x* la raíz, *ParentClose* en el nivel 1 almacena para cada hijo *y* de *x* tres valores: la posición del paréntesis que cierra de *y*, la cantidad de nodos del subárbol con raíz *y* y la cantidad de hojas del subárbol con raíz *y*. Para poder mantener la navegación dentro de la representación compacta del trie de sufijos, esta información se ordena tomando los hijos de *x* de izquierda a derecha.

Esta misma información se puede mantener para los restantes niveles del trie. El problema es que si *ParentClose* se mantiene en todos los niveles del trie, el espacio que se necesita haría perder lo ganado con la representación compacta planteada. Además, a medida que bajamos en el trie los subárboles son cada vez más pequeños por lo que en algún punto será más conveniente (tanto en tiempo como en espacio) trabajar directamente sobre *RP* sin usar *ParentClose*.

En la siguiente sección presentamos los experimentos que permiten analizar el comportamiento de *ParentClose* y determinar empíricamente hasta que nivel conviene crear esta nueva estructura para que resulte competitiva.

5 Evaluación Experimental

Cabe señalar que este trabajo forma parte de un proyecto mayor cuyo objetivo principal es lograr una implementación eficiente en disco del trie de sufijos. Esto significa que como paso posterior a la creación del índice se realizará un paginado del mismo. El proceso de paginación de un índice consiste en dividir el mismo en partes, cada una de las cuales se aloja en una página de disco. Luego el proceso de búsqueda consiste en

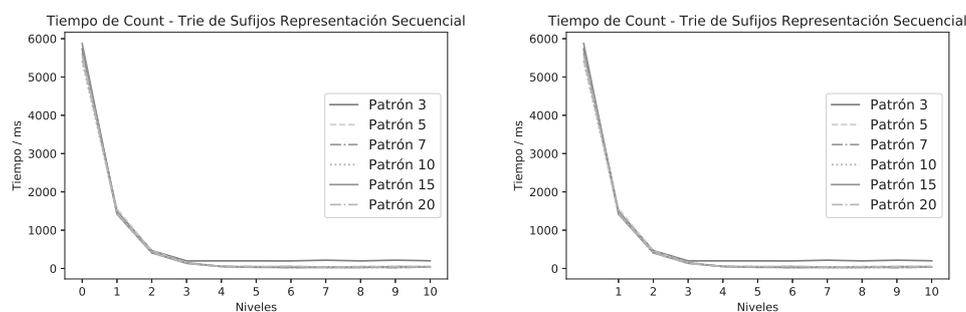


Fig. 5. Tiempo medio de *count* para distintos tamaños de patrones de búsqueda.

ir cargando en memoria principal una parte, realizar la búsqueda en memoria principal sobre esa parte, para luego cargar la siguiente y proseguir la búsqueda. Por esta razón los experimentos se han realizado sobre textos pequeños.

Para realizar esta evaluación hemos utilizado texto que contiene secuencias de ADN, tomados del sitio <http://pizzachili.dcc.uchile.cl>. Este sitio ofrece una amplia colección de índices comprimidos y de textos, que son los habitualmente usados por la comunidad que trabaja con índices sobre bases de datos textuales. Se han tomado textos de tamaño 4, 6, 8 y 10 MB. Los tamaños han sido establecidos tomando en cuenta que si paginamos el índice las búsquedas se realizarán sobre parte pequeñas del mismo.

Los resultados aquí mostrados fueron obtenidos realizando búsquedas con patrones de longitud 3, 5, 7, 10, 15 y 20. Utilizamos estas longitudes ya que por prácticas anteriores han demostrado ser representativas. Para cada longitud de patrón se generaron lotes de 500 patrones.

La Figura 5 muestra los resultados obtenidos para texto de tamaño 4MB (izquierda) y 6MB (derecha). Sobre el eje x se han representado los distintos niveles de *ParentClose* y sobre el eje y está representado el tiempo medio de realizar la operación *count* sobre un patrón, expresado en milisegundos. El nivel 0 corresponde a la representación del trie sin el uso de *ParentClose*; el nivel 1 corresponde al trie usando *ParentClose* solo en la raíz; en general el nivel i corresponde al trie mas *ParentClose* para todos los nodos hasta nivel $i - 1$. Como puede observarse el uso de *ParentClose* mejora drásticamente los tiempos de búsqueda, logrando reducciones de hasta el 26% con nivel 1. Estas mejoras van disminuyendo a medida que aumentamos la cantidad de niveles de *ParentClose* hasta prácticamente estabilizarse en nivel 4. Esto se debe al hecho de que en ese nivel del trie los subárboles son los suficientemente pequeños como para que *ParentClose* pueda mejorar el barrido secuencial. Con respecto a la longitud del patrón de búsqueda, no tiene efectos significativos sobre el comportamiento de *ParentClose*. Esto puede observarse más claramente en la figura 6 donde se han graficado los resultados por longitud

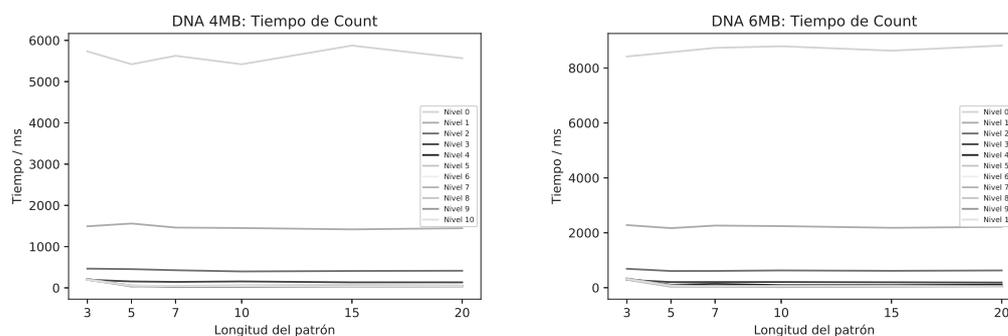


Fig. 6. Tiempo medio de *count* para distintos tamaños de patrones de búsqueda.

de patrón. Allí puede observarse que, para los niveles de *ParentClose* considerados, los tiempos de búsquedas son similares para las distintas longitudes de patrón.

Con respecto al espacio utilizado, la figura 7 muestra el espacio ocupado por *ParentClose*, el espacio ocupado por la representación compacta del trie y el espacio total ocupado (la suma de los dos anteriores). Se puede observar que del total de espacio ocupado solo un 0.03% corresponde a *ParentClose*, notándose un incremento significativo en el espacio a partir del nivel 7 llegando a ocupar un 9% del espacio total. Esto nos indica que con una pequeña sobrecarga en el espacio ocupado podemos lograr mejoras significativas en los tiempos de búsqueda.

Teniendo en cuenta lo analizado podríamos concluir que el nivel 4 es un punto de compromiso entre espacio y tiempo para *ParentClose*. Las conclusiones obtenidas se mantienen cuando el tamaño del texto crece; no se muestran las gráficas por razones de espacio.

6 Conclusiones y Trabajo Futuro

En este trabajo presentamos una mejora en tiempo del trie de sufijos compacto presentado en [14] [15]. La misma consiste en mejorar los tiempos de la operación *findclose* utilizando una estructura auxiliar con información útil para resolver esta operación. Mostramos que con poco espacio adicional, solo un 0.3%, se pueden conseguir reducciones de hasta un 26% en los tiempos de búsqueda. Con respecto al trabajo futuro nos proponemos integrar esta propuesta con técnicas de paginado para el trie de sufijos.

References

1. Nieves R. Brisaboa, Susana Ladra, and Gonzalo Navarro. Directly addressable variable-length codes. In *SPIRE*, pages 122–130, 2009.

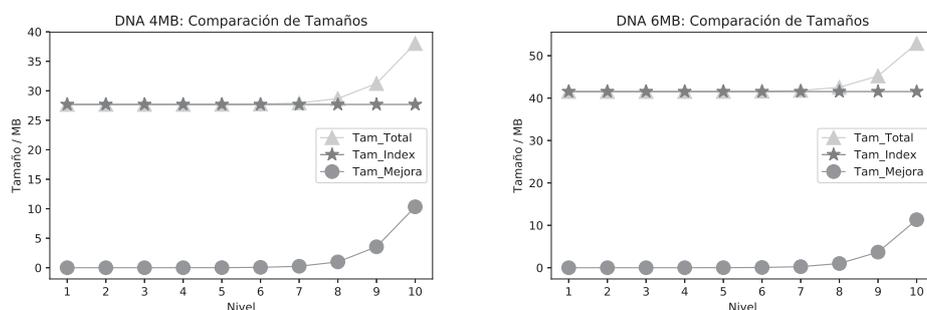


Fig. 7. Espacio total ocupado vs espacio ocupado por *ParentClose*.

2. David Clark. *Thesis: Compact Pat Trees*. PhD thesis, 1996.
3. G. H. Gonnet and R. Baeza-Yates. *Handbook of Algorithms and Data Structures*. Addison-Wesley, 1991.
4. G. H. Gonnet, R. Baeza-Yates, and T. Snider. *New indices for text: PAT trees and PAT arrays*, pages 66–82. Prentice Hall, New Jersey, 1992.
5. R. González and G. Navarro. Compressed text indexes with fast locate. In *Proc. 18th Annual Symposium on Combinatorial Pattern Matching (CPM)*, LNCS 4580, pages 216–227, 2007.
6. N. Herrera and G. Navarro. Árboles de sufijos comprimidos en memoria secundaria. In *Proc. XXXV Latin American Conference on Informatics (CLEI)*, Pelotas, Brazil, 2009.
7. J. Katajainen and E. Mäkinen. Tree compression and optimization with applications. *Int. Journal of Foundations of Computer Science*, 1(4):425–447, December 1990.
8. A. Thomo M. Barsky *, U. Stege. A survey of practical algorithms for suffix tree construction in external memory. In *Software: Practice and Experience*, 2010.
9. E. Mäkinen. A survey on binary tree codings. *The Computer Journal*, 34(5):438–443, 1991.
10. U. Manber and G. Myers. Suffix arrays: A new method for on-line string searches. *SIAM Journal of Computing*, 22(5):935–948, 1993.
11. E. Moura, G. Navarro, N. Ziviani, and R. Baeza-Yates. Fast and flexible word searching on compressed text. *ACM Transactions on Information Systems (TOIS)*, 18(2):113–139, 2000.
12. J. Ian Munro and Venkatesh Raman. Succinct representation of balanced parentheses and static trees. *SIAM J. Comput.*, 31(3):762–776, 2001.
13. G. Navarro and V. Mäkinen. Compressed full-text indexes. *ACM Computing Surveys*, 39(1):article 2, 2007.
14. D. Ruano and N. Herrera. Representación secuencial de un trie de sufijos. In *XX Congreso Argentino de Ciencias de la Computación*, Buenos Aires, Argentina, 2014.
15. D. Ruano and N. Herrera. Indexando bases de datos textuales: Una representación compacta del trie de sufijos. In *Congreso Nacional de Ingeniería Informática / Sistemas de Información*, Buenos Aires, Argentina, 2015.
16. J. Vitter. External memory algorithms and data structures: Dealing with massive data. *ACM Computing Surveys*, 33(2):209–271, 2001.
17. P. Weiner. Linear pattern matching algorithm. In *Proc. 14th IEEE Symposium Switching Theory and Automata Theory*, pages 1–11, 1973.



CACIC 2019
Congreso Argentino de Ciencias de la Computación

XVI Workshop Ingeniería de Software

Coordinadores

Pesado, Patricia (UNLP)
Estévez, Elsa (UNS)
Cechich, Alejandra (UNCOMA)
Kuna, Horacio (UNaM)

Comité Científico

Astudillo, Hernán (Chile)	Janowski, Tomasz (Polonia)	Olivas Varela, José Ángel (España)
Barbosa, Luis (Portugal)	Janssen, Marijn (Países Bajos)	Piatini, Mario (España)
Fillottrani, Pablo (Argentina)	Navarro Martín, Antonio (España)	Riesco, Daniel (Argentina)
Genero, Marcela (España)	Ojo, Adegboyega (Irlanda)	Rossi, Gustavo (Argentina)

Temas

Administración de Configuración de Software y Despliegue	Mantenimiento de Software
Administración de Proyectos de Software	Metodologías de Desarrollo de Software
Arquitecturas y Diseño de Software	Métricas de Software
Calidad de Software	Modelamiento de Software
Dependabilidad y Confiabilidad del Software	Modelamiento de Procesos de Negocio
Desarrollo Ágil de Software	Orientación a Aspectos
Desarrollo de Software basado en Componentes	Patrones y Frameworks
Entornos para el Desarrollo de Software	Plataformas como Servicios
Evolución del Software	Políticas para el Software y para las TICs
Estándares para el Desarrollo de Software	Privacidad
Herramientas de Software	Procesos de Software
Herramientas y Tecnologías de la Web 2.0	Reúso de Software
Ingeniería de Requerimientos de Software	Sistemas de Información
Ingeniería de Software Distribuido	Sistemas de Software basados en Cloud Computing
Ingeniería de Software Embebido y Tiempo Real	Sistemas de Software Sensibles al Contexto
Ingeniería de Software Empírica	Software como Servicio
Ingeniería de Software para Sistemas Ubícuos y Móviles	Soporte Computacional para Tareas Colaborativas
Ingeniería de Software para la Web	Testeo de Software
Ingeniería Reversa	Teoría y Métodos Formales
	Verificación y Validación de Software
	Workflows

Análisis del impacto en la performance de una web app utilizando un lenguaje compilado y un lenguaje interpretado

Brian Edgar Ré, Michele Philippe Ré, Agustín Olmedo

Facultad de ingeniería. Universidad Austral, Argentina
{brian.re,michele.re,agustin.olmedo}@ing.austral.edu.ar
<http://www.austral.edu.ar/ingenieria>

Resumen Actualmente se desarrollan muchas aplicaciones web y el tiempo de respuesta de las mismas es un factor muy importante. En este trabajo se presenta un análisis de dos stacks de tecnologías utilizando, por un lado, un lenguaje compilado (Golang) y, por otro lado, un lenguaje interpretado (Python). El aspecto a analizar será desde el punto de vista técnico aplicado a un caso de uso de un *e-commerce* (listar elementos aplicando un filtro). Se analizó principalmente el *throughput*. Las pruebas se realizaron en instancias EC2 de AWS que se encontraban dentro del mismo *availability zone*, haciendo peticiones al servidor a través de una IP privada. Además, se analizó el impacto que tiene considerar la latencia entre el cliente y el servidor al realizar peticiones. En conclusión, los resultados demuestran que Golang tiene un *throughput* mejor que Python con una latencia mínima, pero a medida que aumenta la latencia disminuye la diferencia entre ambos lenguajes.

Keywords: Lenguaje compilado, lenguaje interpretado, Golang, Python, AWS, test de performance, throughput, latencia.

1. Introducción

Cada lenguaje de programación posee ciertas características específicas con sus ventajas y desventajas. Por ejemplo, si el lenguaje está pensado para trabajar eficientemente de forma paralela y concurrente, si utiliza un garbage collector, si compila a un lenguaje intermedio, etc. Es por ello que es de gran utilidad comparar las características entre distintos lenguajes de programación y conocer en que es bueno cada lenguaje para poder tomar una decisión fundamentada a la hora de tener que elegir un lenguaje de programación. Una de las principales diferencias que se presenta entre los lenguajes de programación es que pueden ser compilados o interpretados.

La principal diferencia entre un lenguaje compilado y uno interpretado es que el compilado, al no tener que traducir el programa en tiempo de ejecución, se ejecuta más rápido que el interpretado.

Hoy en día, el enfoque que se tiene a la hora de elegir las tecnologías necesarias para llevar a cabo un proyecto de software es el de elegir las tecnologías de acuerdo a las necesidades del proyecto. Por ejemplo, si se va a desarrollar un servicio de mensajería en tiempo real, se va a elegir un lenguaje que sea concurrente, eficiente y a prueba de fallos, mientras que si se va a desarrollar una aplicación web para una *startup*, se va a priorizar un lenguaje que sea fácil de usar y que pueda realizarse una primera versión funcional en poco tiempo. En este último caso, la performance puede ser o no un requerimiento indispensable. En muchas ocasiones, las empresas eligen como tecnologías el stack de Django para desarrollar aplicaciones web debido a la facilidad que se tiene para desarrollar nuevas funcionalidades en poco tiempo (ya que el framework tiene ciertas configuraciones por defecto que ayudan a reducir el tiempo de desarrollo). Sin embargo, en muchas ocasiones, las empresas que eligen este stack deciden migrar sus aplicaciones a un stack que utilice un lenguaje compilado debido a los problemas de performance que tienen. Estos problemas comienzan a impactar fuertemente en el negocio, lo que se traduce en un impacto en los ingresos de la empresa (se tiene que pagar un elevado precio por el uso de servidores, la aplicación funciona con lentitud, etc.).

Por lo tanto, si las empresas deciden migrar sus aplicaciones web de un lenguaje interpretado a uno compilado, la pregunta que se presenta es: ¿cuánta diferencia de performance hay entre una aplicación web escrita en un lenguaje interpretado respecto a una aplicación escrita en un lenguaje compilado? Para responder a esta pregunta compararemos una aplicación web escrita en Go[6] (lenguaje compilado) y la misma aplicación web escrita en Python[7] (lenguaje interpretado).

2. Trabajos relacionados

Existen muchos estudios evaluando y analizando el rendimiento de servidores web. T. Lance et al. [1] evaluaron experimentalmente el impacto en el rendimiento del servidor web de tres tecnologías de contenido dinámico diferentes (Perl, PHP, Java). Los resultados mostraron que los costos de la generación del contenido dinámico puede reducir la tasa máxima de peticiones admitida por un servidor web hasta 8 veces. Mientras tanto, los resultados indicaron que las tecnologías de servidor Java suelen superar a Perl y PHP para la generación dinámica de contenido. T. Suzumura et al. [2] realizaron un estudio exhaustivo de la capacidad de PHP como motor de servicios web tanto en aspectos cualitativos como cuantitativos mientras lo compararon con otros motores de servicios web implementados en Java y C. Los resultados mostraron que PHP como motor de servicio web funciona de manera competitiva con Axis2 Java para servicios web que involucran pequeñas cargas útiles, y lo supera en gran medida para cargas grandes de 5 a 17 veces. Axis2 C funciona mejor que PHP, pero los resultados experimentales demuestran que el rendimiento de PHP está más cerca de Axis2 C con cargas útiles más grandes. N. Togashi et al. [3] implementaron

programas simples de multiplicación de matrices tanto en Go como en Java para analizar las características de concurrencia. La implementación de Java usa Java Thread, y la implementación de Go usa Goroutine y Channel. En los resultados de los experimentos se observó que Go obtuvo un mejor rendimiento que Java en tiempo de compilación y concurrencia. K. Lei et al. [4] compararon Node.js, Python y PHP realizando pruebas sistemáticas objetivas y pruebas realistas de comportamiento de los usuarios especialmente tomando Node.js como punto central a discutir. Llegando a la conclusión que Node.js puede manejar muchos más peticiones que Python y PHP en un periodo de tiempo determinado.

Y. Ueda et al. [5] compararon el lenguaje Go, compilado estáticamente, con dos lenguajes compilados dinámicamente, JavaScript y Java. Los resultados experimentales demostraron que la implementación de Go logró un rendimiento 3.8x y 2.4x mayor que las implementaciones de JavaScript y Java respectivamente, después de un simple ajuste en la configuración del servidor. El análisis detallado indicó que esto se debe principalmente a que Go sufre menos el polimorfismo que JavaScript debido a la escritura estática, y porque el framework web para Go produce menos sobrecarga para procesar los requests de servicio web RESTful que Java. Además argumentan que los lenguajes compilados estáticamente jugarán un papel más importante para las aplicaciones web debido a sus ventajas de rendimiento y también a las nuevas metodologías de integración e instalación continuas que eliminan algunas de las deficiencias en los lenguajes compilados estáticamente sobre los lenguajes compilados dinámicamente y de secuencias de comandos.

Las investigaciones anteriores se centran en la comparación de las tecnologías de desarrollo web. Pero hasta donde sabemos, todos estos estudios no consideran la latencia. Este documento es el primero en comparar el rendimiento de estas dos tecnologías web populares, Golang y Python teniendo en cuenta también el impacto de la latencia.

3. Entorno de pruebas

3.1. Configuración de Hardware

En cada prueba se utilizaron dos máquinas: una como servidor que alojará la aplicación web de uno de los stacks; y otra como cliente que alojará la herramienta K6, la cual realizará las peticiones al servidor.

Las pruebas serán ejecutadas en el cloud de Amazon (AWS) en el mismo *Availability Zone* para que la latencia entre ambas instancias sea mínima, y que las instancias no sean del tipo *burstable* (para evitar que la CPU sea el cuello de botella en las pruebas).

Se utilizaron las siguientes instancias:

- 1 instancia c5.large para el servidor.
- 1 instancia c5.xlarge para el cliente.

Todos los servidores correrán el sistema operativo Ubuntu 18.04. Las instancias que actuarán de servidor ejecutarán una imagen de Docker¹ que contendrá la aplicación web. Cada stack correrá en servidores y clientes distintos.

3.2. Configuración de Software

Este estudio se centra en comparar dos stacks de tecnologías para aplicaciones web. Una utilizando el lenguaje Go y otra utilizando el lenguaje Python. A continuación se describen cada uno de los stacks.

Stack de Go

- Go (v1.12): es un lenguaje compilado, fácil de utilizar y de instalar en producción. Fue desarrollado pensando en la concurrencia, por lo que a la hora de realizar las pruebas se podría llegar a obtener una performance mucho mayor que con un lenguaje interpretado. Se utilizó el servidor HTTP que trae Go en la librería estándar. Además, no se utilizó ningún framework ya que no existe alguno lo suficiente maduro que se utilice para el desarrollo de páginas web (como puede ser Django para el stack de Python). Según la encuesta² realizada por JetBrains en el año 2018, el 49% de los desarrolladores no utiliza frameworks web.

Stack de Python

- Python (v3.6): es un lenguaje interpretado ampliamente utilizado en la industria debido a la baja curva de aprendizaje y a la gran comunidad que tiene.
- Django (v2.1.1): es uno de los frameworks de Python más utilizados para desarrollar aplicaciones web debido a la gran cantidad de funcionalidades que ofrece y la productividad que se tiene trabajando con el mismo.
- Gunicorn (v19.9.0): es el servidor de aplicación utilizado para Django (WSGI server) que servirá para procesar las peticiones que son recibidas.

Además para ambos stacks se utilizará la base de datos PostgreSQL (v11)³ (que será ejecutado dentro de un Docker) y también un agrupador de conexiones de base de datos (PGBouncer v1.7.2)⁴.

¹ <https://docs.docker.com>

² <https://www.jetbrains.com/research/devecosystem-2018/go/>

³ <https://www.postgresql.org/>

⁴ <https://pgbouncer.github.io/>

3.3. Herramienta utilizada para obtener las métricas

En cuanto a la herramienta utilizada para realizar las mediciones, se utilizó K6⁵, que es una herramienta escrita en Go y que utiliza Javascript para escribir las pruebas.

4. Metodología de las pruebas

Las métricas utilizadas para medir la performance de las aplicaciones web son las siguientes:

- **Throughput:** cantidad de peticiones que el servidor responde por segundo.
- **Throughput teórico:** cálculo estimado de la cantidad de peticiones que el servidor responde por segundo teniendo en cuenta la latencia de red.

El **Throughput teórico** lo definimos como:

$$\textit{Tiempo de procesamiento de respuesta} = \frac{1000 \textit{ milisegundos}}{(\textit{throughput} / \textit{cantidad de virtual users})}$$

$$\textit{Tiempo de procesamiento con latencia} = \textit{Tiempo de procesamiento de respuesta} + \textit{latencia}$$

$$\textit{Throughput teórico} = \left(\frac{1000 \textit{ milisegundos}}{\textit{Tiempo de procesamiento con latencia}} \right) * \textit{cantidad de virtual users}$$

Si se quiere estimar el *throughput real*, se deberá definir previamente un determinado valor de latencia y hacer el cálculo del *throughput* basándonos en este tiempo adicional. En la vida real, los clientes (usuarios de la aplicación) se encuentran por fuera del *datacenter* en donde se encuentra alojado el servidor (que aloja la web app). Como los clientes pueden ser de cualquier parte del mundo (y pueden a veces encontrarse a muchos kilómetros de distancia del servidor), la latencia puede variar entre unos pocos milisegundos (10 ms) hasta varias decenas de milisegundos (200 ms). Para poder calcular el *throughput teórico*, tomaremos como valor de latencia 100 ms.

En el caso del *throughput* la prueba se realiza utilizando dos servidores que se encuentran en el mismo *Availability Zone* y se utiliza la ip privada de la red para realizar los peticiones, por lo que el cliente y el servidor se encuentran separados a muy poca distancia. En este caso, se asumirá que la latencia es cero y el tiempo de respuesta registrado en la prueba será considerado como el tiempo que el servidor demora en procesar y responder a una petición.

⁵ <https://k6.io/>

5. Resultados

5.1. Test N°1 - Test de performance de los stacks sin sistema de templating ni base de datos

Esta prueba consiste en realizar peticiones al servidor HTTP y que este devuelva en la respuesta un *string* fijo, sin utilizar ningún tipo de sistema de *templating*. El objetivo de la prueba es la de conocer la cantidad máxima de respuestas que puede servir cada servidor HTTP. De esta forma, se tendrá una buena aproximación de las capacidades que tiene cada servidor HTTP sin utilizar componentes de software adicionales.

Throughput del servidor

En la prueba notamos que, a medida que el tamaño de la respuesta aumenta, la diferencia de *throughput* de ambos servidores se reduce, haciendo que la diferencia de performance entre ambos stacks pase del 65 % a un 10 %. Esto se debe a que, para respuestas pequeñas, el tiempo que tarda en transmitir la información por la red (desde el primer bit que se envió hasta el último) es pequeño respecto al tiempo que tarda el servidor en procesar la respuesta. Por lo tanto, del tiempo total que tarda el servidor en responder la petición recibida, la mayor parte está compuesta por el tiempo que tardó el servidor en procesar la respuesta, y el tiempo restante está compuesto por la transmisión de la información por la red. Por el contrario, para respuestas grandes, el tiempo que tarda en transmitir la información por la red es grande respecto al tiempo que tarda el servidor en procesar la petición. Como el tiempo que tarda en transmitir la información es aproximadamente la misma en ambos stacks (ya que depende de la red y no de los stacks), entonces el tiempo total que demora en responder un request va a ser similar. Por ende, el *throughput* de ambos servidores va a ser parecido.

Además, se puede observar que hay mucha diferencia de *throughput* cuando se hace la prueba con varios usuarios virtuales (también llamados *virtual users* o VU, que es el equivalente a cantidad de peticiones simultáneas) respecto a cuando se hace la prueba con un solo usuario virtual. Por ejemplo, para el caso de 1 VU y 1000 caracteres, la diferencia de *throughput* entre el stack de Go y el de Python es de aproximadamente 2.75 a 1. En el caso en que se realice la prueba con 128 VUs, la diferencia de *throughput* entre ambos stacks se incrementa a 8 a 1 (Python tuvo un rendimiento inferior que Go).

Si bien se podría pensar que el motivo de esta diferencia se deba a que el lenguaje interpretado es más lento, luego de investigar el caso se concluyó que se debe a que Python tiene un *Global Interpreter Lock* (o GIL) [8] que es un *mutex* que protege el acceso a los objetos de Python, previniendo que múltiples *threads* ejecuten *bytecode* de Python simultáneamente. Este candado, según la

documentación de Python⁶, es necesario porque la gestión de la memoria de Python no es *thread-safe*. El problema que resuelve el GIL es la condición de carrera que se puede producir en el procesador cuando varios procesos quieren acceder al mismo sector de memoria al mismo tiempo. Como consecuencia del uso del GIL, la performance del sistema disminuirá.

Por lo tanto se puede concluir que el factor que más peso tiene en el *throughput* del servidor es el modelo de concurrencia que tiene el lenguaje y no debido a que Python sea interpretado.

Throughput teórico del servidor

Considerando el **throughput teórico** definido en la sección 4, se pudo observar que la diferencia de *throughput* para 1 VUs era menos del 1%, independientemente de la longitud de la respuesta. Para el caso de 128 VUs, la diferencia de *throughput* fue del 26% (Python tuvo un rendimiento menor que Go) cuando la longitud de la respuesta era pequeña, mientras que para respuestas con mayor cantidad de caracteres (millón de caracteres) la diferencia fue del 5%.

Por lo tanto, si se tuviera que elegir entre un stack u otro, no van a haber diferencias significativas de performance. La única variación que puede haber es en el uso de recursos en el lado del servidor.

5.2. Test N°2 - Test de performance de los stacks utilizando un sistema de templating y realizando operaciones dentro del mismo

Esta prueba consiste en realizar peticiones al servidor HTTP y que éste devuelva en la respuesta un determinado contenido en donde se haya realizado una serie de operaciones dentro del *template*, haciendo uso del sistema de *templating*. Para ello, se listan 50 strings y se hará uso de un condicional para simular alguna condición dentro de la aplicación. El objetivo es conocer la cantidad de respuestas que puede servir cada servidor HTTP realizando operaciones dentro del *template*. De esta forma, se podrá comparar esta prueba con el **Test N°1** y saber si existe algún *overhead* en el caso en que se utilice un sistema de *templating*.

Throughput del servidor

Al realizar la comparación de *throughput* entre ambos stacks, se observó que, para 1 VUs y para 1000 caracteres, la diferencia de Go vs Python fue de 4.5 a 1. Para una respuesta de 1 millón de caracteres, Python superó el *throughput* de

⁶ <https://wiki.python.org/moin/GlobalInterpreterLock>

Go, realizando un throughput 50 % mayor que Go.

En el escenario de 128 VUs, para 1000 caracteres, la diferencia fue de 8 a 1 (Go vs Python), mientras que para el millón de caracteres, la diferencia de Python vs Go fue aproximadamente un 50 % mayor.

Si realizamos la comparación de los stacks con respecto al **Test N°1**, observamos que para Python, tanto para 1 VUs como para 128 VUs, para 1000 caracteres, la performance disminuyó un 65 %, mientras que para el millón de caracteres y 1 VUs disminuyó un 50 %, y para 128 VUs disminuyó un 65 %.

Para el caso de Go, para 1 VU y 1000 caracteres, hubo una reducción del 45 % en el *throughput* del servidor, mientras que para tamaños de respuesta grandes, el *throughput* disminuyó aproximadamente un 70 %. En el caso de 128 VUs, la performance se redujo por lo menos un 70 %.

En este test concluimos que, por más pequeña que sea la operación que se realiza en el *template*, el impacto que tiene es alto. Por lo tanto, cuanto menos operaciones deba realizar el servidor para renderizar una respuesta, mejor. Es por ello que hoy en día se intenta desacoplar el *front-end* del *back-end* (utilizando frameworks front-end como ser React, Vue.js, etc.), ya que el *front-end* se encarga de renderizar la información desde el lado del cliente, haciendo que el servidor se libere de esta responsabilidad y que el procesamiento de la información que se muestra la lleve a cabo el cliente y no el servidor.

Throughput teórico del servidor

Al comparar Python y Go para 1 VUs, la diferencia entre ambos stacks es mínima (tanto para respuestas con poca cantidad de caracteres como para mucha cantidad de caracteres, la diferencia entre ambos stacks es menor al 5 %).

En el caso de 128 VUs, la diferencia entre ambos stacks es menor que si no hubiera latencia. Aún así, podemos ver que, para poca cantidad de caracteres, Python tiene la mitad de *throughput* que Go. Mientras que para respuestas más largas, Python supera a Go en un 60 %.

A partir de los datos recolectados, se puede concluir que utilizar un sistema de *templating* y realizar operaciones dentro del mismo impacta enormemente en el *throughput* del servidor. Aún así, se puede observar que la latencia juega un rol muy importante en la performance general de ambos stacks.

Por lo tanto, para obtener la mayor performance posible de un servidor, es necesario realizar la menor cantidad de operaciones con el sistema de *templating*. Además, es recomendable reducir al máximo la latencia entre el cliente y el servidor y, de ser posible, delegar la tarea de renderizado al cliente.

5.3. Test N°3 - Test de performance simulando un entorno real (utilizando base de datos y sistema de templating)

Esta prueba consiste en realizar peticiones al servidor HTTP y que éste devuelva en la respuesta una determinada cantidad de elementos que fueron buscados en la base de datos (filtrado por algún criterio) y renderizados utilizando un sistema de *templating*. El objetivo de esta prueba es el de conocer la cantidad de respuestas que puede servir cada servidor HTTP, el cual renderiza resultados que fueron buscados en la base de datos. Lo que se busca es realizar las mismas operaciones que realiza un *e-commerce* cuando se muestran productos según un criterio (a nivel *templating* y base de datos, no en cuanto a estética). Dicha prueba consta de buscar 50 elementos de una base de datos (simulando ser publicaciones de productos), renderizar el resultado (*template*), y devolver dicha página al usuario que realizó la petición. El tamaño de la respuesta a devolver será de 1.000.000 de caracteres (cantidad de caracteres promedio que devuelve una página de Amazon.com).

Throughput del servidor

Luego de haber realizado las pruebas, observamos que:

- Cuanto más grande sea el *template*, más tarda en generar la respuesta. Para ambos stacks, a medida que la cantidad de VUs aumenta, la performance del servidor va disminuyendo.
- El uso de una base de datos degrada enormemente el *throughput* de ambos servidores.
- Para los casos en que se probaron con 64 y 128 VUs, el consumo de recursos de Python era máximo, por lo que comenzaba a afectar la performance del servidor y comenzaba a responder de forma no satisfactoria a las peticiones.
- Para el caso de Go, también se detectaron anomalías. Se cree que el motivo por el cual ocurre este comportamiento es debido a cómo Go gestiona la concurrencia al renderizar templates. Por lo tanto, se excluirán del análisis los resultados obtenidos para 64 y 128 VUs.

Si comparamos Python vs Go para tamaños de respuestas grande (1 millón de caracteres) para 1 VU, la diferencia es de un 42 %, mientras que para 32 VUs es de un 58 %.

Throughput teórico del servidor

Al comparar Python y Go, observamos que la diferencia de *throughput* es mínima. Para el caso en que se haga la prueba con 1 VU, la diferencia de *throughput* es menor al 2 %. En cambio, si se hacen de a 32 VUs, la diferencia de *throughput* llega aproximadamente al 30 %. Esto se debe a que la latencia reduce drásticamente la performance del servidor. Por ende, a medida que se tenga más latencia, menor será el *throughput* del mismo. Esto se corrige teniendo el servidor lo más cerca posible del cliente.

6. Conclusión

A partir de los resultados obtenidos de las pruebas realizadas en el presente trabajo, se puede llegar a la conclusión de que si bien un lenguaje compilado es más rápido que un lenguaje interpretado, esta diferencia de performance no es tan grande si consideramos la latencia.

Por lo tanto, se puede concluir que el problema de performance de una aplicación web termina siendo mayormente un problema de arquitectura y no un problema del tipo del lenguaje utilizado. Es decir, que un lenguaje sea compilado o interpretado no será determinante para la performance de una aplicación web.

Es importante destacar que, si bien podría ser mejor utilizar un lenguaje compilado en lugar de uno interpretado, es más importante tener una buena arquitectura. A continuación mencionaremos algunos puntos a tener en cuenta:

- Si se utiliza una base de datos en el sistema, tratar de acceder a la misma la menor cantidad de veces posible (de ser posible, utilizar una caché).
- Separar el front-end del back-end, ya que de no hacerlo, la performance del servidor se va a ver afectada enormemente.
- Tener el servidor lo más cercano posible al cliente, ya que la latencia hace que la performance de la aplicación web se degrade enormemente.

Referencias

1. T.Lance, A.Martin and W.Carey, Performance Comparison of Dynamic Web Technologies, ACM SIGMETRICS Performance Evaluation Review, Volume 31 Issue 3, December 2003.
2. Suzumura, Toyotaro & Trent, Scott & Tsubori, Michiaki & Tozawa, Akihiko & Onodera, Tamiya. (2008). Performance Comparison of Web Service Engines in PHP, Java and C. 385-392.
3. N. Togashi and V. Klyuev, "Concurrency in go and java: Performance analysis," in 2014 4th IEEE International Conference on Information Science and Technology, pp. 213–216, April 2014.
4. K. Lei, Y. Ma, and Z. Tan, "Performance comparison and evaluation of web development technologies in php, python, and node.js," in Computational Science and Engineering (CSE), 2014 IEEE 17th International Conference on, pp. 661–668, Dec 2014.
5. Y. Ueda and M. Ohara, "Performance competitiveness of a statically compiled language for server-side Web applications," 2017 IEEE International Symposium on Performance Analysis of Systems and Software (ISPASS), Santa Rosa, CA, 2017, pp. 13-22.
6. The Go Programming Language. <https://golang.org/>.
7. <https://www.Python.org/>.
8. <https://wiki.python.org/moin/GlobalInterpreterLock>

Aspect Oriented Behavioral Synthesis

Fernando Asteasuain^{1,2}, Federico Calonge¹, and Pablo Gamboa²

¹ Universidad Nacional de Avellaneda, Avellaneda, Argentina

² Universidad Abierta Interamericana -Centro de Altos Estudios CAETI, CABA, Argentina

`fasteasuain, fcalonge@undav.edu.ar, pgamboa.uai.edu.ar`

Abstract. Modern modularization techniques such as Aspect Orientation require powerful and expressive enough specification languages in order to conceive the development of a system as the combination of the different views it is composed of. In this work we present FVS as an aspect oriented language where the composition of individual aspects is achieved employing behavioral synthesis. As a distinctive feature, our approach can handle properties denoted by non deterministic Büchi automata. A case of study is introduced to show our approach in action.

Keywords: Aspect Orientation, Behavioral Synthesis

1 Introduction

In the recent years modern modularization techniques such as Aspect Orientation [19, 15] or Feature Oriented Programming [1, 25] have draw the attention of the Software Engineering Community and their application on several domains such as software protocols, software architectures and others keep growing and growing [1, 11, 21]. In few words, these approaches conceive systems as different views, perspectives or features that combined together produce the expected behavior of the system. For example, in the classic ATM example the system is built combining diverse views such as security, availability, efficiency, data integrity or transaction management, just to mention a few.

The main tasks to be addressed when dealing with these approaches are to specify individual views for one side, and for the other side, to specify how to compose and combine each view to produce the expected behavior of the system. This last item is clearly the ultimate challenge to be tackled because the different views rarely behave separately but rather interact with each other. In addition, feature or view interaction might not be trivially resolved. In some occasions one feature might depend on the behavior of another feature, sometimes two or more features can be in conflict with each other, or sometimes the behavior of the system might be different according to the order in which the views are combined. For example, in the ATM system the encryption view must act (ie, must encrypt the message) before any message is sent from the ATM to the bank. Similarly, the encryption perspective may constitute a menace to achieve successfully the efficiency view, since delays are introduced when performing transactions.

In the Aspect Orientation approach each view is named an *Aspect* and the process of combining aspects behavior is named *Weaving*. Several approaches aim to address aspects' interaction trying to identify and resolve possible conflicts between aspects [26, 8, 13, 22] tackling problems known as the Aspect Interference [8] problem and the Aspect Oriented Paradox [27]. However, most of them are based in operational notations inspired in finite state machines or labeled transition systems (e.g., statecharts) [18, 17]. Many aspect oriented approaches boil down into providing syntactic weaving mechanisms, usually with non-clear semantics counterpart [17]. Thus, unlike other well-established modularization mechanisms as procedures, parallel composition, or logical conjunction (in declarative approaches) aspect orientation, though attractive in principle, is still a second class citizen, holding just the status of a hacking or dynamic instrumentation mechanism where semantics impact is not neatly characterized.

An interesting and novel way of addressing aspect interaction is given in [24]. In this approach the system is built in a fashion that its behavior satisfies its specification by construction. This is achieved by employing behavioral synthesis [9, 12]. Behavioral synthesis can be seen as an automated procedure to obtain a correct-by-construction reactive system from its temporal logic specification [23]. In the case of reactive synthesis, an implementation is typically given as an automaton that accepts input from the environment (e.g., from sensors) and produces the system's output (e.g., on actuators). By construction the input and output assignments of every infinite run of the automaton satisfy the specification it was synthesized from [23]. In order to reduce the time complexity of the algorithms involved work in [9] suggested the General Reactivity of Rank 1 (GR(1)) fragment of LTL, which has an efficient polynomial time symbolic synthesis algorithm. GR(1) is a strict assume/guarantee subset of LTL, comprised of constraints for initial states, safety propositions over the current and successor state and assertions about what should hold infinitely often also known as justice constraints. A GR(1) synthesis problem is defined as a game between a system player and an environment player [12]. Efficient symbolic algorithms for GR(1) realizability checking and controller synthesis have been presented in [9].

However, the synthesis technique introduced in [24] only works if the behavior can be denoted using only Deterministic Büchi automata. This is because if an LTL formula can be expressed by Deterministic Büchi automaton then it can be expressed in the GR(1) fragment of LTL [12]. Therefore, there exists the need to further extend the expressive power of the specification language used to synthesize behavior.

Given this context in this work we present the FVS (Feather Weight Visual Scenarios) specification language [4, 3] as a powerful tool to denote, compose and synthesize aspect oriented behavior. FVS is a declarative language based on graphical scenarios and features a flexible and expressive notation with clear and solid language semantics. FVS expressivity is a distinguished characteristic among declarative approaches since it is able to denote omega-regular properties, being for example, more expressive than LTL (Linear Temporal Logic) [4]. In addition, in [5] we introduced a version of FVS where behavioral synthesis it is

not limited to behavior properties expressible by deterministic Büchi automata since properties which are not expressible by deterministic Büchi automata can also be handled. FVS was also thoroughly explored in previous work as an aspect oriented specification language [7, 2], and a tool named GTxFVS was developed giving support to all of the FVS features [6]. Taken all these points into consideration, *in this work we combine FVS expressive power with its aspect oriented flavour to provide an attractive tool to handle aspects interactions by employing behavioral synthesis as the weaving mechanism to build the system.*

The rest of the paper is structured as follows. Section 2 presents some background concepts for a more comprehensive reading of this work. Section 3 shows our approach in action whereas Section 4 presents related and future work. Finally, Section 5 enumerates the conclusions of the work.

2 Background

In this section we present some preliminaries concepts introducing our language, its usage on aspect orientation and the synthesis procedure which play the role of the weaver.

Feather weight Visual Scenarios: In this section we will informally describe the standing features of FVS [3, 4]. The reader is referred to [4] for a formal characterization of the language. FVS is a graphical language based on scenarios. Scenarios are partial order of events, consisting of points, which are labeled with a logic formula expressing the possible events occurring at that point, and arrows connecting them. An arrow between two points indicates precedence of the source with respect to the destination: for instance, in figure 1-(a) *A*-event precedes *B*-event. We use an abbreviation for a frequent sub-pattern: a certain point represents the next occurrence of an event after another. The abbreviation is a second (open) arrow near the destination point. For example, in figure 1-b the scenario captures the very next *B*-event following an *A*-event, and not any other *B*-event. Conversely, to represent the previous occurrence of a (source) event, there is a symmetrical notation: an open arrow near the source extreme. Events labeling an arrow are interpreted as forbidden events between both points. In figure 1-c *A*-event precedes *B*-event such that *C*-event does not occur between them. Finally, FVS features aliasing between points. Scenario in 1-d indicates that a point labeled with *A* is also labeled with $A \wedge B$. It is worth noticing that *A*-event is repeated on the labeling of the second point just because of FVS formal syntaxis.

We now introduce the concept of FVS rules, a core concept in the language. Roughly speaking, a rule is divided into two parts: a scenario playing the role of an antecedent and at least one scenario playing the role of a consequent. The intuition is that whenever a trace “matches” a given antecedent scenario, then it must also match at least one of the consequents. In other words, rules take the form of an implication: an antecedent scenario and one or more consequent scenarios. Graphically, the antecedent is shown in black, and consequents in grey.

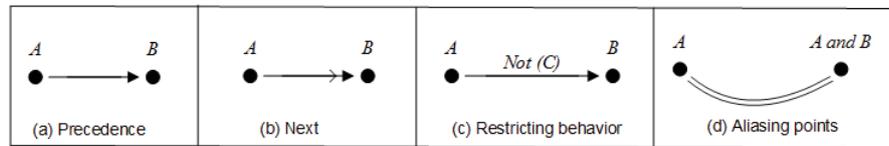


Fig. 1. Basic Elements in FVS

Since a rule can feature more than one consequent, elements which do not belong to the antecedent scenario are numbered to identify the consequent they belong to. An example is shown in Figure 2 describing the circumstances under which writing in a pipe is valid. For every write in the pipe it must be the case that either the pipe did not reach its maximum capacity since it was open (Consequent 1) or the pipe did reach its capacity, but another component performed a read over the pipe (making the pipe available again) afterwards and the pipe capacity did not reach again its maximum (Consequent 2).

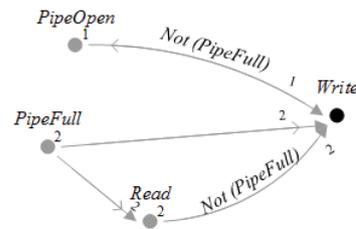


Fig. 2. An FVS rule example

FVS as an aspect oriented specification language: Aspect orientation can be described as a modularization technique which extends a base system at certain points of interest introducing new behavior specified in the so called *aspects*. Aspects are described as a twofold: a *pointcut*, which selects where the aspect's behavior is to be introduced, and an *advice*, which details what behavior in particular is to be added. FVS rules fits into the aspect oriented perspective: rules' antecedents play the role of *pointcuts*, whereas consequents play the role of *advices*. The reader is referred to [2] for a more comprehensive description of FVS as an aspect oriented language. Weaving is the process which inserts into the base system the behavior described by the aspects in those points of the base system that were indicated.

Behavioral Synthesis in FVS: FVS specifications can be used to automatically obtain a controller employing a classical behavioral synthesis procedure. We now briefly explains how this is achieved while the complete description is available in [5]. Using the tableau algorithm detailed in [4] FVS scenarios are translated into

Büchi automata. Then, if the obtained automata is deterministic, then we obtain a controller using a technique [23] based on the specification patterns [14] and the GR(1) subset of LTL. If the automaton is non deterministic, we can obtain a controller anyway. Employing an advanced tool for manipulating diverse kinds of automata named GOAL [28] we translate these automata into Deterministic Rabin automata. Since synthesis algorithms are also incorporated into the GOAL tool using Rabin automata as input, a controller can be obtained. Although this gain in expressiveness come with an cost in terms of performance due to the size of the involved automata we believe its crucial being able to express all type of behavioral properties. In the next section we illustrate the power of FVS's behavioral synthesis as a weaving process specifying an interesting case study.

3 Case of Study

In this section we analyze the case of study of this work. Section 3.1 models a first iteration of the system to be developed. Section 3.2 aggregates more sophisticated functionality whereas Section 3.3 enumerates some final considerations about the weaving process.

3.1 The Exam System

In order to show in action FVS's synthesis procedure as an aspect oriented weaving mechanism we studied the case study introduced in [24]. The system basically provides a service for student exams. Following the strategy in [24] we start off with a base system, and use it as a minimal basis on which to add aspects. This base system behaves as follows: when a new student comes, the service may leave the waiting state, show a welcome screen, and start the exam. The student may fail or pass the exam, after which the service moves to the exit state and back to waiting for a new student. The environment controls the input variables such as *evalExam* and *newStudent*. The system itself has another variable, *state*, specifying its current state (for example, *inExam*, *failed*, *passed* or *wait*). Note that the base system is not deterministic: when it is waiting and a new student comes, it can either stay in waiting state or move to the welcome state.

We start extending this base functionality adding new behavior in the shape of aspects. The first aspect to be introduced, called *Tuition*, handles a security concern: it prevents new students who have not paid their tuition from taking an exam and instead redirects them to the exit state where it shows a message with information about tuition payments. This behavior is shown in Figure 3. The FVS rule in the top of the figure dictates that a welcome screen will only be shown if and only if the tuition was paid before (that is, if an *PaidTuition* event occurred in the past). Similarly, the rule in the bottom of the figure says that in those cases where the student did not paid the tuition the system must show an allusive message and exit.

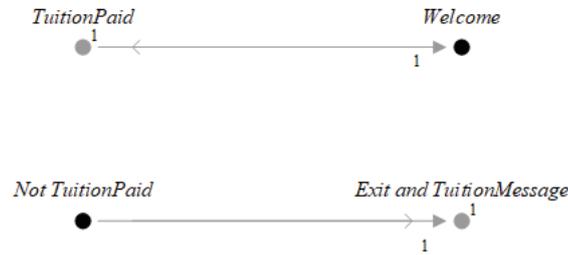


Fig. 3. Tuition Aspect in FVS

The second behavior to be introduced by an aspect handles the services availability concern: it specifies that whenever a new student is present when the system is waiting, eventually the system will welcome the student. As in [24], we named this aspect *Availability* which addresses a typical liveness constraint. This behavior is modeled in FVS as shown in Figure 4. The rule establishes that when the system is in the *wait* state and a new student arrives, then the welcome screen will be eventually displayed.



Fig. 4. A liveness property in the Exam System

Having specified these two aspect, we weave them into the base system by employing the synthesis procedure described earlier in Section 2. However, a winning strategy for the system is not found. As explained in [24], there is a winning move for the environment since the initial wait state is not winning for the system: an environment that sends infinitely many students such that only a finite number of them have paid their tuition, will force the system not to visit the welcome state infinitely often, and thus to violate its specification. We solve this functionality bug following the solution presented in [24]: forcing the system to leave the wait state but not necessarily visit the welcome state. This is shown in Figure 5.

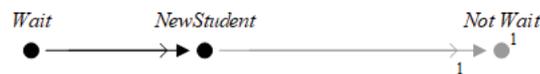


Fig. 5. A more general Availability Aspect

When performing the behavioral synthesis considering the *Tuition Aspect* and the revisited *Availability Aspect* a controller can be automatically obtained for the *Exam System*.

3.2 Adding new behavior

We now add some three extra features to the system. The first one gives the possibility to the student to retry the exam in the case she failed. Given this new feature, we consider an aspect that checks that every *retry* must occur while the system is in the *inExam* state. This is illustrated in Figure 6. In other words and considering the concrete events of the system, the aspects checks that once the system has presented the *Welcome* screen every occurrence of the event *retry* must be preceded by the occurrence of the *InExam* event.

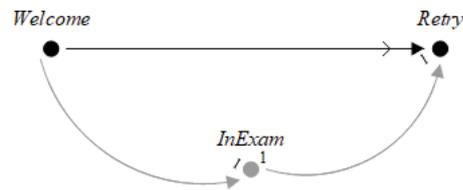


Fig. 6. Validating Retry occurrences

A security aspect is added next. It persists and logs every failed exam in case a student's complaint is raised in the future. The requirement for this aspect is the following: In each student's valid session (between the *Welcome* state and until the *Exit* state) every failed exam must be persisted and logged. This is reflected in Figure 7. During a valid session, the occurrence of the event *Fail* must trigger the occurrence of the events *Persist* and *Log*.

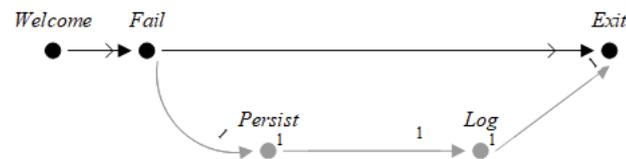


Fig. 7. Capturing information about failed exams

Finally, we revisit this last requirement (persisting and logging failed exams information) considering that a new feature is added: the student can quit the exam and leave the system. However, every failed exam must be persisted and logged anyway. Therefore, the *quit* event must not occur until the exam is persisted and logged. Taking this fact into consideration, we modified previous rule

in Figure 7 as it is shown in Figure 8. The only difference is that quitting is not allowed until the failed exam is persisted and logged.

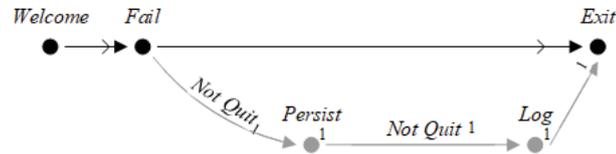


Fig. 8. Failed exams and the Quit feature

3.3 Case Study: Remarks and Observations

A controller was built upon FVS specifications for every version of the Exam System. This was achieved by employing the behavioral synthesis procedure described in Section 2. It is worth mentioning that the three aspects introduced in Section 3.2 can not be represented by Deterministic Büchi automata. As it was described in Section 2 FVS is still capable of obtaining a controller in those cases by translating non Deterministic Büchi automata into Deterministic Rabin automata. In particular, the three aspects considered in Section 2 corresponds to three particular specification patterns [14]. The aspect that validates retry occurrences corresponds to the pattern *Precedence pattern with After Q scope*; the first version of the audition aspect for failed exams corresponds to the *Response Chain pattern (with one stimuli and two responses) with After q until r scope* and second version of this latter aspect corresponds to the *Constrained Chain pattern (responses s, t without z responds to the p stimuli event) with After q until r scope*. The time consumed to obtain this new controller (including these three new aspects) took eight times more than the previous one. However, we believe it is desirable being able to express these kind of properties when they arise despite the fact they are time consuming.

4 Related and Future Work

FVS was previously explored in the aspect oriented world [7, 2]. While this previous work was focused on modeling aspect's behavior this work provides the means to perform behavior synthesis. Other several approaches aim to formalize aspects specification and interaction. For example, work in [16] employs model checking techniques to prove the correctness of aspects application and weaving into the base system. Aspects are specified using state machines. Similarly, work in [20] validates CTL properties when weaving aspects into the base system. Also notations based on state machines are employed. We rely on a different weaving approach based on behavioral synthesis. In this sense, AspectLTL [24] also employs behavioral synthesis for weaving aspects. However, their approach

is limited to the usage of Deterministic Büchi automata while our approach handles also Non Deterministic Büchi automata.

Work in [22] employs constructors based on implementations languages named *gummy* modules to denote aspect behavior and resolve aspects interaction in a modular way. Also more related to the implementation phase, work like [26] provides an interesting tool to detect aspect interference in the AspectJ language. Work in [10] extends aspects definition to tackle high level behavior with a Finite State Machine based notation denominated Dependency State Machines. The focus of this work is to improve automatic verification and optimization of aspects. An AspectJ extension is also presented in that work. We rely on a more declarative way of defining aspects and the implementation view is built using behavioral synthesis.

Regarding future work we would like to improve the performance of our approach. This line of research involves trying to optimize the size of the automata involved since the complexity of the algorithms involved are heavily influenced by the size of the automata. Other similar future direction is to analyze the trade off between expressivity and performance and provide stronger empirical evidence in this topic. Finally, we would like to explore automatic code generator tools using as input the controllers as in [23].

5 Conclusions

In this work we propose FVS a powerful tool to denote, compose and synthesize aspect oriented behavior. Aspects are defined using graphical rules and the weaving process is achieved using behavioral synthesis. As a distinctive feature, our approach is not limited to Deterministic Büchi automata since Non Deterministic Büchi can also be input to the synthesis procedure. A case study is presented illustrating the main points of our approach.

6 Acknowledgements

This work was partially funded by UNDAVCYT 2014 and UAI-CAETI.

References

1. S. Apel, D. Batory, C. Kästner, and G. Saake. *Feature-oriented software product lines*. Springer, 2016.
2. F. Asteasuain and V. Braberman. FVS: A declarative aspect oriented modeling language. *EJS - Electronic Journal SADIO*, 10(1):20–37, 2011.
3. F. Asteasuain and V. Braberman. Specification patterns: formal and easy. *IJSEKE*, 25(04):669–700, 2015.
4. F. Asteasuain and V. Braberman. Declaratively building behavior by means of scenario clauses. *Requirements Engineering*, 22(2):239–274, 2017.
5. F. Asteasuain, F. Calonge, and M. Dubinsky. Exploring specification pattern based behavioral synthesis with scenario clauses. In *CACIC*, 2018.

6. F. Asteasuain and F. Tarulla. Exploring architectural model checking with declarative specifications. In *CACIC*, 2017.
7. F. Asteasuain, F. Tarulla, and P. Gamboa. Using the power of abstraction to express high-level behavior in aspect oriented approaches. In *CONAIIISI*, 2017.
8. L. M. Bergmans. Towards detection of semantic conflicts between crosscutting concerns. *Analysis of Aspect-Oriented Software (ECOOP 2003)*, 2003.
9. R. Bloem, B. Jobstmann, N. Piterman, A. Pnueli, and Y. Sa'Ar. Synthesis of reactive (1) designs. 2011.
10. E. Bodden. Specifying and exploiting advice-execution ordering using dependency state machines. In *FOAL*, volume 151, 2010.
11. T. Cerny. Aspect-oriented challenges in system integration with microservices, soa and iot. *Enterprise Information Systems*, 13(4):467–489, 2019.
12. N. DiPpolito, V. Braberman, N. Piterman, and S. Uchitel. Synthesising non-anomalous event-based controllers for liveness goals. *ACM Tran*, 22(9), 2013.
13. C. Disenfeld and S. Katz. A closer look at aspect interference and cooperation. In *AOSD*, pages 107–118. ACM, 2012.
14. M. Dwyer, M. Avrunin, and M. Corbett. Patterns in property specifications for finite-state verification. In *ICSE*, pages 411–420, 1999.
15. R. Filman, T. Elrad, S. Clarke, and M. Aksit. *Aspect-oriented software development*. Addison-Wesley Professional, 2004.
16. M. Goldman, E. Katz, and S. Katz. Maven: modular aspect verification and interference analysis. *Formal Methods in System Design*, 37(1):61–92, 2010.
17. S. Katz. Aspect categories and classes of temporal properties. In *Transactions on aspect-oriented software development I*, pages 106–134. Springer, 2006.
18. S. Katz and H. Israel. Diagnosis of harmful aspects using regression verification. *FOAL: Foundations Of Aspect-Oriented Languages*, pages 1–6, 2004.
19. G. Kiczales, J. Lamping, A. Mendhekar, C. Maeda, C. Lopes, J.-M. Loingtier, and J. Irwin. Aspect-oriented programming. In *European conference on object-oriented programming*, pages 220–242. Springer, 1997.
20. S. Krishnamurthi, K. Fisler, and M. Greenberg. Verifying aspect advice modularly. In *ACM SIGSOFT*, volume 29, pages 137–146. ACM, 2004.
21. J. Krüger. Separation of concerns: experiences of the crowd. In *ACM Symposium on Applied Computing*, pages 2076–2077. ACM, 2018.
22. S. Malakuti and M. Aksit. Event-based modularization: how emergent behavioral patterns must be modularized? *FOAL*, pages 7–12, 2014.
23. S. Maoz and J. O. Ringert. Synthesizing a lego forklift controller in gr (1): A case study. *arXiv preprint arXiv:1602.01172*, 2016.
24. S. Maoz and Y. Sa'ar. Aspectltl: an aspect language for ltl specifications. In *AOSD*, pages 19–30. ACM, 2011.
25. M. Mezini and K. Ostermann. Variability management with feature-oriented programming and aspects. In *ACM SEN*, volume 29, pages 127–136. ACM, 2004.
26. S. Sandra I. Casas, J. J. Baltasar García Perez-Schofield, and C. Claudia A. Marcos. MEDIATOR: an AOP Tool to Support Conflicts among Aspects. *International Journal of Software Engineering and Its Applications (IJSEIA)*, 3(3):33–44, 2009.
27. T. Tourwé, J. Brichau, and K. Gybels. On the existence of the aosd-evolution paradox. *SPLAT*, 2003.
28. Y.-K. Tsay, Y.-F. Chen, M.-H. Tsai, K.-N. Wu, and W.-C. Chan. Goal: A graphical tool for manipulating büchi automata and temporal formulae. In *TACAS*, pages 466–471. Springer, 2007.

Coreografía de dispositivos ubicuos basada en SOA: ¿Cuánto de limitados pueden ser los dispositivos?

Oscar A. Testa, Efraín R. Fonseca C., Germán Montejano, and Oscar Dieste

Universidad Nacional de La Pampa
Facultad de Ciencias Exactas y Naturales
La Pampa - Argentina
Universidad de las Fuerzas Armadas ESPE
Departamento de Ciencias de la Computación
Quito - Ecuador
Universidad Nacional de San Luis
Facultad de Ciencias Físico Matemáticas y Naturales
San Luis - Argentina
Universidad Politécnica de Madrid
Escuela Técnica Superior de Ing. Informática
Madrid - España
otesta@exactas.unlpam.edu.ar
erfonseca@espe.edu.ec
gmonte@unsl.edu.ar
odieste@fi.upm.es

Resumen Actualmente nos encontramos inmersos en ambientes donde los dispositivos ubicuos forman parte de nuestras tareas diarias. Los dispositivos ubicuos no proporcionan servicios de forma aislada, sino que deben cooperar con otros dispositivos. Existen desafíos para la composición de dispositivos ubicuos, como el caso de tolerancia a fallas, escaso nivel de procesamiento, etc. Se han desarrollado distintos frameworks para dar solución a la composición de dispositivos y los problemas asociados, sin embargo estos frameworks no están basados sobre mecanismos estándares y abiertos. Ante esto hemos desarrollado nuestro propio framework, el cual cumple con estos requisitos, además de ser interoperable, simple y extensible, basándose sobre la especificación WS-CDL de SOA. El desafío es poder incluir en el framework dispositivos con capacidades menores (memoria y procesamiento). Presentamos un análisis de memoria que nos indique los requerimientos mínimos para la aplicación de los mecanismos de SOA en la coordinación e inclusión de estos dispositivos.

Keywords: Ingeniería de Software; SOA; Dispositivos ubicuos; Servicios; Coreografía de servicios, WS-CDL

1. Introducción

Los avances tecnológicos han permitido que los dispositivos ubicuos ¹ sean al mismo tiempo generadores y consumidores de servicios, lo que se traduce en un ambiente de cooperación entre dispositivos, permitiendo aumentar o mejorar las funcionalidades en base a la composición. Por composición, entendemos la forma en que los dispositivos ubicuos se pueden combinar para realizar una tarea determinada. Esto implica que se produzcan desafíos tales como: tolerancia a fallas, escaso nivel de procesamiento, problemas de conectividad, por mencionar algunos ejemplos [2].

La composición de dispositivos ubicuos presenta desafíos adicionales tales como: la heterogeneidad de los mismos, las contingencias de los dispositivos, y la personalización de los mismos (ej: provisión de servicios de acuerdo a las preferencias del usuario). Dado que los dispositivos ubicuos poseen limitaciones de recursos (ej. poca memoria y batería), se deben hacer consideraciones especiales respecto a la eficiencia y rendimiento de la composición de dispositivos.

Las características descritas hacen que la composición de dispositivos ubicuos se constituya un área de investigación muy atractiva, donde los avances aún no son claros al día de hoy [2].

Actualmente existen algunos frameworks de coordinación de dispositivos ubicuos, como **Aura**[3] de la Universidad de Carnegie Mellon; **Gaia** [4] de la Universidad de Illinois; **Oxigen** [5] perteneciente al MIT (Massachusetts Institute of Technology); **Amigo**[6] proveniente de Amigo Consortium; **OSGi**; **Android Studio** y otros menos específicos. Todos estos frameworks tienen como objetivo principal compartir y proveer de servicios a los usuarios.

La academia también ha aportado estudios relacionados a la composición de dispositivos ubicuos, como es el caso de Viroli [7]; Najjar & otros [8]; Loke [9]; Palmieri [10]. Estos trabajos van desde ambientes auto organizados hasta descubrimiento de dispositivos de manera automática.

Si bien, se han realizado esfuerzos y avances en la composición de dispositivos ubicuos, éstos esfuerzos no han logrado que los dispositivos se coordinen entre sí sin la necesidad de un coordinador central y sin basarse sobre mecanismos estandarizados y abiertos.

Nuestro framework [11] propone ventajas alternativas en la composición de dispositivos ubicuos, como es el caso de la estandarización, la extensibilidad, etc. Para ello, construimos un framework que nos permite ejecutar coreografías definidas en el lenguaje en WS-CDL (de las siglas del inglés Web Service Choreography Description Language), con dispositivos ubicuos. Por coreografía entendemos que los dispositivos se comunican de a pares sin un coordinador central para llevar adelante una tarea determinada [12]. El framework desarrollado puede interpretar prácticamente la totalidad de las descripciones realizadas en WS-CDL, además de contar con la posibilidad de trabajar con transacciones distribuidas

¹ Dispositivos electrónicos que tienen capacidad de procesamiento y comunicación, y pueden ser encontrados en cualquier lugar: la oficina, el auto, la casa, o la misma ropa con la que vestimos [1]

al estilo de WS-Transaction ². A su vez, también tiene la posibilidad de realizar adecuaciones en la ejecución de la coreografía ante la falla o desaparición de un dispositivo, logrando que la ejecución se recupere y pueda continuar.

Este desarrollo inicial del framework de ejecución de coreografías, ahora tiene como desafío incorporar nuevos dispositivos ubicuos, con menores prestaciones a nivel de capacidad de procesamiento, memoria y otros mecanismos de comunicación. Es decir, la versión actual del framework no puede añadir a dichos dispositivos, debido a las características mencionadas.

Nuestra contribución consiste en lograr la inclusión de dispositivos ubicuos con capacidades cada vez menores, a un ambiente de cooperación y coordinación como es el caso del framework que hemos desarrollado. Para ello realizaremos un análisis de memoria que nos indique cuáles son los requerimientos mínimos para que se puedan aplicar los mecanismos de SOA en la coordinación de los dispositivos.

Lo que resta del artículo se estructura de la siguiente forma: en la Sección 2 se describirá el framework desarrollado junto a determinados conceptos de SOA. En la Sección 3 presentamos las limitaciones que presentan algunos dispositivos en la utilización del framework. El análisis de memoria propiamente dicho, que es en sí la principal contribución del trabajo, es descrita en la Sección 4. Finalmente, en la Sección 5 presentamos las conclusiones y trabajos futuros.

2. Antecedentes

2.1. Computación orientada a servicios

Los mecanismos de composición de servicios, como las orquestaciones y coreografías, son aspectos bien conocidos en SOA (Service Oriented Architecture) que permiten construir sistemas de negocio complejos y aplicaciones a partir de una gran cantidad de servicios heterogéneos, simples y distribuidos. Estos conceptos podrían ser aplicables a ambientes ubicuos, en especial las coreografías, las cuales pueden fácilmente interpretarse en términos de dispositivos y ambientes ubicuos. Sin embargo, en determinados ambientes donde los servicios son dinámicos, móviles, menos fiables y dependientes del dispositivo, los mecanismos de composición establecidos para servicios web no son directamente aplicables [14].

Adicionalmente, la composición de múltiples dispositivos ubicuos presenta nuevos desafíos que no son compatibles con la composición de servicios web. En particular, los mecanismos de composición en ambientes masivos ³ necesitan hacer frente a distintas contingencias que pueden ocurrir con estos elementos, así como también contemplar la heterogeneidad de los mismos. Estos dispositivos tienen distintas limitantes como son la cantidad de memoria disponible, la

² El framework implementa una capa de transacciones distribuidas con servicios REST, de la forma definida en [13]

³ Por ejemplo el de dispositivos móviles

durabilidad de la batería, la disponibilidad de acuerdo a la red del lugar donde se encuentren en un momento determinado, etc. En ambientes ubicuos, la disponibilidad y confiabilidad de los dispositivos no puede ser garantizada. No obstante, a pesar de todas las dificultades, existen ventajas que podría brindar la adaptación de los conceptos de SOA para la composición de dispositivos ubicuos, como son la estandarización tanto de los protocolos como de los mecanismos de comunicación entre los dispositivos, y la compatibilidad con otras plataformas de servicios ya existentes.

Las similitudes entre la composición de servicios web y la coordinación de dispositivos ubicuos es sorprendente. Si pensamos que cada dispositivo ubicuo en un ambiente pervasivo ⁴ es proveedor, o, consumidor de un servicio, la coordinación de dispositivos encaja perfectamente con la composición de servicios. Es también sorprendente que esta similitud no haya sido apenas explorada con anterioridad, salvo en el trabajo de Sheng [2] se hace mención a ello.

2.2. Framework

El framework de ejecución de coreografías desarrollado[11] se construyó siguiendo los lineamientos de la metodología Design Science, de forma evolutiva. Cada nuevo ciclo se basó en la evaluación de los resultados obtenidos en el ciclo previo, aplicando las mejoras necesarias o pertinentes. Los lenguajes de programación utilizados son PHP y C++ y fueron seleccionados en función de los dispositivos que se utilizaron para la prueba de concepto.

La base de la ejecución de coreografías se centra en una serie de clases que, en principio, leen la descripción en WS-CDL de la coreografía, y en base al dispositivo que lo ejecuta; determina en primer lugar en qué posición de la ejecución de la coreografía se encuentra, esto puede verse en la Figura 1 como el punto inicial.

Luego se determina cuál o cuáles son los pasos siguientes en la ejecución, se puede ver en la Figura 1 en el estado "Determinar siguiente".

Una vez determinados los pasos que se deben ejecutar, se hacen las invocaciones a otros dispositivos, teniendo en cuenta las actividades descritas en la definición de la coreografía en el lenguaje WS-CDL. Esta ejecución se hace de manera controlada, en el sentido de examinar que no se produzcan contingencias provenientes de las características de los dispositivos (desapariciones, latencia en la respuesta por falta de capacidad de procesamiento, etc.). En base a esta verificación se realizan las tareas correctivas correspondientes, según se han expresado en la definición de la coreografía. La estructura de ejecución del framework la podemos apreciar en la Figura 1.

Durante el desarrollo del framework se trabajó con un escenario de pruebas donde se planteó un ambiente de autopistas inteligentes para dar solución a problemas referidos al tránsito vehicular, donde se simulaban vehículos, balizas, Centrales de Emergencias, etc.

⁴ Ambientes pervasivos son entornos poblados por varios dispositivos (sensores, actuadores, etc) y aplicaciones de software integrados de forma transparente [1]

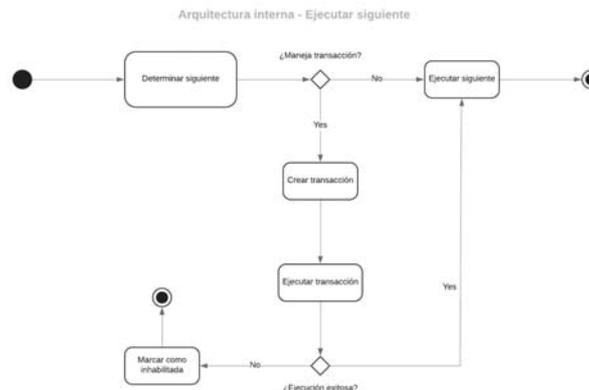


Figura 1. Diagrama de arquitectura de ejecución

El framework como vemos se encuentra desarrollado y funcionando⁵, pero aún falta que el mismo permita dispositivos con menores capacidades (memoria, procesamiento, etc.) puedan participar de la ejecución de una coreografía.

3. Limitaciones derivadas de los dispositivos

Algunas características de los dispositivos ubicuos son específicas y perfectamente identificables, como es el caso de la movilidad o volatilidad de su conexión; no obstante, hay otras que no resultan tan claras (capacidad de procesamiento, memoria disponible, batería, etc.), lo que implica que se precisa hacer consideraciones especiales para poder contemplarlas dentro del framework desarrollado.

A continuación mostramos el cuadro 1 con un lista de algunos dispositivos y sus características, especialmente aquellos que tienen menos capacidad tanto de memoria como de procesamiento. Estos dispositivos representan únicamente algunos ejemplos de los que hemos trabajado, sin pretender hacer una lista exhaustiva.

En el caso de prueba seleccionado, el rol VehiculoAccidentado fue representado por una placa Arduino NANO, el rol Baliza fue caracterizado por una placa Arduino MEGA y el de CentralBalizas por un dispositivo Raspberry Pi B+, que se corresponden con los presentados en el Cuadro 1.

En este caso particular, la placa Arduino NANO, tiene tan solo el 12% de la capacidad de memoria flash (almacenamiento de programa) de la que cuenta Arduino Mega, y el 25% de la capacidad de almacenamiento de variables en memoria SRAM, esto se puede apreciar en la Figura 2. En un principio el código fuente del framework de ejecución de coreografías pudo ser almacenado en la memoria flash de la placa Arduino NANO, ocupando casi la totalidad de la

⁵ El código fuente del framework desarrollado, puede ser accedido a través de esta dirección: https://github.com/GRISE-UPM/ml_server_rest.

Dispositivo	Características distintivas
Raspberry Pi B+	- Escaso nivel de procesamiento - Escasa capacidad de memoria
Arduino Mega	- Bajo nivel de procesamiento - Baja capacidad de memoria - Escasa conectividad - Dependencia de batería
Arduino NANO	- Escasísimo procesamiento - Escasísima capacidad de memoria - Escasa conectividad - Dependencia de batería
Dispositivos RFID	- Nula capacidad de procesamiento - Capacidad de memoria casi nula - Escasa conectividad - Dependencia de batería

Cuadro 1. Características de los dispositivos ubicuos

misma ⁶, por lo que no era aconsejable su ejecución con tan poca memoria libre disponible. Hubo que hacer adaptaciones para que finalmente ocupara el 81 % de la memoria flash disponible para programas.

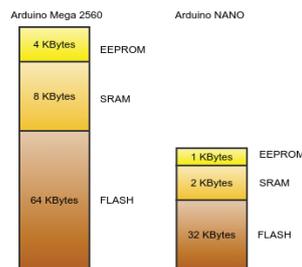


Figura 2. Comparación de memoria de dispositivos Arduino

Luego el otro inconveniente con el que hubo que realizar ajustes fue la memoria SRAM disponible y la necesaria para la ejecución del framework codificado. En una primera instancia el framework ocupaba el 85 % de la memoria disponible, por lo que su ejecución no era factible en absoluto, ya que esta memoria inicial es la que ocupan las variables globales, strings literales, objetos, etc., dejando únicamente un 15 % máximo de memoria libre para el almacenamiento de variables globales y punteros, esto puede verse en el Listado 1.1. Realizando un análisis exhaustivo del código fuente en C++, con el fin de reducir el tamaño de memoria utilizada por variables globales, strings y demás, se logró hacer una op-

⁶ En este caso solamente son 30k, ya que 2k son utilizados por el bootloader.

timización del código de tal manera que se llegó a una utilización del 61 % (1260 bytes) de la totalidad de la memoria SRAM. Por lo tanto, quedan disponibles para utilización de variables locales y punteros un 39 %, es decir unos 788 bytes, puede observarse en el Listado 1.2

```
El Sketch usa 24888 bytes (81%) del espacio de almacenamiento de
programa
El maximo es 30720 bytes
Las variables Globales usan 1740 bytes (85%) de la memoria dinamica ,
dejando 308 bytes para las variables locales
El maximo es 2048 bytes
```

Listado 1.1. 1ra. Compilación en entorno Arduino

```
El Sketch usa 24888 bytes (81%) del espacio de almacenamiento de
programa
El maximo es 30720 bytes
Las variables Globales usan 1260 bytes (69%) de la memoria dinamica ,
dejando 788 bytes para las variables locales
El maximo es 2048 bytes
```

Listado 1.2. 2da. Compilación en entorno Arduino

Esta nueva versión del framework se cargó en la placa Arduino NANO y se intentó una ejecución del mismo, resultando insuficiente la memoria disponible para variables locales y punteros. Para superar esta situación, se realizaron simplificaciones del código para que pueda ser ejecutado en este tipo de placa. Estas simplificaciones implican que se tengan que introducir decisiones específicas para dicha placa de desarrollo.

Luego de realizadas estas pruebas, se pudo determinar ciertos límites para la ejecución de coreografías en dispositivos ubicuos, lo que se detalla a continuación:

- Los dispositivos deben contar con al menos 32 kbytes de memoria flash para poder almacenar de manera adecuada el código de ejecución del framework y con al menos 4 kbytes de memoria SRAM para el almacenamiento de las variables y punteros de programa.
- El motor de ejecución de coreografías puede ser ejecutado en placas Arduino Mega, equipos Raspberry Pi B+, pero no así en la placa Arduino NANO (cuenta solamente con 2 kbytes de memoria SRAM).

4. Análisis de las limitaciones de procesamiento y memoria

Las limitaciones derivadas de los dispositivos nos permitieron encontrar que la coreografía descrita en WS-CDL no puede ser utilizada en ciertos dispositivos, ya que para su interpretación se requiere de un cierto tamaño de memoria y una capacidad de procesamiento determinada. Ahora bien, los factores que producen que la descripción en WS-CDL necesite determinadas capacidades de los dispositivos ubicuos radica principalmente en el formato en el cual se encuentra especificada; esto es, XML. La especificación XML consume un espacio adicional de memoria, el cual justo es escaso en este tipo de dispositivos; además, el

procesamiento e interpretación de la estructura XML necesita un rendimiento de procesador considerable. A modo de ejemplo, la especificación completa de la coreografía utilizada en el escenario de pruebas tiene una longitud de 8 kbytes, lo que representa cuatro veces más de memoria de lo que cuenta una placa Arduino Nano y el doble de los 4 kbytes necesarios para la ejecución del framework. Es preciso recordar que el escenario propuesto es reducido respecto de otras especificaciones que se pueden dar en ambientes de producción, o en el mismo caso del escenario planteado pero en un ambiente real.

Para poder implementar en el framework la ejecución de la coreografía definida en el lenguaje WS-CDL, es necesario que se hagan simplificaciones de la especificación, ya que la misma es muy extensa. Una alternativa es que éstas simplificaciones sean realizadas con anterioridad y pre-cargadas en el dispositivo (de forma estática); y por otro lado, que las mismas se hagan a demanda en tiempo real. Hemos implementado en el framework la simplificación pre-cargada, pero para una mayor independencia tanto de los dispositivos como de la coreografía en su conjunto, se debe implementar la segunda alternativa, que es la lectura a demanda de la especificación. En el primero de los casos, la simplificación pre-cargada, el ahorro de memoria ha llegado a más de un 50%, ya que de los 8 kbytes necesarios solo por la especificación en XML, se necesitan 4 kbytes para la totalidad de la ejecución (esto incluye variables y punteros que hacen al procesamiento propiamente dicho), lo cual puede ser visualizado en el Listado 1.3 donde podemos apreciar el resultado de una compilación en el entorno Arduino. En el segundo caso, de solicitar la especificación o parte de la misma a demanda, los valores se reducirían aún más, pero con la necesidad de hacer mayor uso de las comunicaciones para poder solicitar a un nodo centralizador o a otros dispositivos con mayor capacidad los trozos de especificación necesarios para la ejecución.

Además de las limitaciones de memoria planteadas hasta este punto, nos encontramos con aquellos dispositivos que no tienen capacidad de procesamiento, como es el caso de tarjetas RFID. Éstos dispositivos también deben ser considerados para ser parte de la ejecución de una coreografía, dentro del framework desarrollado, pero para ello se deben hacer consideraciones especiales. Estos dispositivos al contar únicamente con la posibilidad de almacenar información y no poseer capacidad de procesamiento no pueden ejecutar de manera convencional la coreografía, pero sí pueden ser el vínculo para que otros dispositivos puedan encontrar información que no pueden almacenar en su memoria debido a la escasa cantidad como planteamos anteriormente. Para ser más específicos, estos dispositivos que solamente tienen la capacidad de almacenar información podrían contener la definición de la coreografía (o parte de ella) en formato XML para que otros dispositivos con escasa capacidad de memoria puedan solicitar dicha información para poder continuar con la ejecución, tal como se plantea en el párrafo anterior.

```
El Sketch usa 25368 bytes (9%) del espacio de almacenamiento de
programa
El maximo es 253952 bytes
Las variables Globales usan 2055 bytes (25%) de la memoria dinamica ,
dejando 6137 bytes para las variables locales
```

El máximo es 8192 bytes

Listado 1.3. Compilación en entorno Arduino

5. Conclusiones y Futuras Líneas

El framework para dispositivos ubicuos en ambientes pervasivos obtenido cumple con la propuesta inicial de trabajo, el cual está basado en los conceptos de SOA, que implementa la especificación WS-CDL. Esta solución a su vez cuenta con ciertas ventajas por sobre el resto de los frameworks existentes, siendo éste abierto, estándar, extensible e interoperable. En contrapartida tiene ciertas desventajas como es el caso de las limitaciones de memoria y de requerimientos de capacidad de procesamiento que hacen que el framework deba ser mejorado y ampliado para dar soporte a estas falencias.

Más específicamente, determinamos que nuestra propuesta requiere un mínimo de 4 Kbytes de memoria SRAM y 32 Kbytes de memoria Flash, para ser ejecutada en dispositivos ubicuos y una capacidad de procesamiento de al menos 8 bits a 16Mhz. Estos requerimientos necesarios para la ejecución del framework permiten que el mismo pueda ser utilizado en placas del estilo Arduino Mega o equipos como Raspberry Pi B+. No obstante, estas mismas características mínimas limitan el uso de por ejemplo placas de desarrollo Arduino Nano (y todas aquellas que posean igual o menor capacidad) y dispositivos RFID.

Sin embargo, en base a las adaptaciones propuestas en la Sección 4, donde hablamos de reducciones de la especificación podríamos incluir los dispositivos que han quedado excluidos por escasa capacidad de memoria o procesamiento como puede ser el caso de placas Arduino Nano, dispositivos RFID, wearables, etc.

Como futuras líneas de trabajo avizoramos la necesidad de ahondar en la búsqueda de alternativas de mejora para las soluciones propuestas (reducciones ad-hoc y peticiones a demanda) sobre la interpretación de la especificación en XML de la coreografía. De este modo, nuestro framework puede ser utilizado por dispositivos de menor capacidad antes mencionados y similares.

Agradecimientos

Esta investigación fue financiada en parte por el rubro de la Corporación Ecuatoriana para el Desarrollo de la Investigación y la Academia (CEDIA) para el Grupo de Trabajo: "Internet de las Cosas y Ciudades Inteligentes" y en parte por el rubro de la Universidad de las Fuerzas Armadas ESPE del Ecuador para estancias de investigación.

Referencias

1. M. Weiser, "Hot topics-ubiquitous computing," *Computer*, vol. 26, pp. 71–72, Oct 1993.

2. Q. Z. Sheng, X. Qiao, A. V. Vasilakos, C. Szabo, S. Bourne, and X. Xu, "Web services composition: A decade's overview," *Information Sciences*, vol. 280, no. 0, pp. 218–238, 2014.
3. J. Harkes, T. Farbacher, and N. Miller, "Project Aura Distraction-free Ubiquitous Computing." Available at <http://www.cs.cmu.edu/~./aura/people.html>, Last visited: Aug 15th, 2002.
4. R. H. Campbell, D. M. Mickunas, D. Reed, and K. Nahrstedt, "Active Spaces for Ubiquitous Computing." Available at <http://gaia.cs.illinois.edu/>, Last visited: Aug 15th, 2002.
5. M. L. f. C. Science and M. A. I. Laboratory, "Pervasive Human-Centered Computing." Available at <http://oxygen.csail.mit.edu/>, Last visited: Aug 15th, 2002.
6. "Amigo Project." Available at <http://gforge.inria.fr/projects/amigo/>.
7. M. Viroli, "On competitive self-composition in pervasive services," *Science of Computer Programming*, vol. 78, no. 5, pp. 556–568, 2013. Special section: Principles and Practice of Programming in Java 2009/2010 & Special section: Self-Organizing Coordination.
8. S. Najjar, M. K. Pinheiro, and C. Souveyet, "A New Approach for Service Discovery and Prediction on Pervasive Information System," *Procedia Computer Science*, vol. 32, pp. 421–428, 2014. The 5th International Conference on Ambient Systems, Networks and Technologies (ANT-2014), the 4th International Conference on Sustainable Energy Information Technology (SEIT-2014).
9. S. W. Loke, "Supporting ubiquitous sensor-cloudlets and context-cloudlets: Programming compositions of context-aware systems for mobile users," *Future Generation Computer Systems*, vol. 28, no. 4, pp. 619–632, 2012.
10. F. Palmieri, "Scalable service discovery in ubiquitous and pervasive computing architectures: A percolation-driven approach," *Future Generation Computer Systems*, vol. 29, no. 3, pp. 693–703, 2013. Special Section: Recent Developments in High Performance Computing and Security.
11. G. M. Oscar A. Testa, Efraín R. Fonseca C. and O. Dieste, "Coordination of ubiquitous devices in pervasive environments: A proposal based on ws-cdl," in *XXXVIII International Conference of the Chilean Computer Science Society (SCCC 2019)*, p. en proceso de publicación, November 2019.
12. M. Rosen, B. Lublinsky, K. T. Smith, and M. J. Balcer, *Applied SOA: Service-Oriented Architecture and Design Strategies*. Wiley Indianapolis, 2008.
13. G. Pardon and C. Pautasso, "Atomic distributed transactions: a restful design," (Seoul, Korea), ACM, April 2014.
14. G. Cassar, P. Barnaghi, W. Wang, S. De, and K. Moessner, "Composition of services in pervasive environments: A Divide and Conquer approach," in *Computers and Communications (ISCC), 2013 IEEE Symposium on*, pp. 000226–000232, July 2013.

Detección de objetos espurios en generación automática de entradas

Simón Gutiérrez Brida, Pablo Ponzio, Valeria Bengolea, y Nazareno Aguirre

Universidad Nacional de Río Cuarto,
Río Cuarto, Argentina

{sgutierrez,pponzio,vbengolea,naguirre}@dc.exa.unrc.edu.ar

Resumen En el contexto de testing, el uso de entradas generadas de manera automática, sobre las cuales se va a ejecutar el programa bajo prueba, es una práctica cada vez más común. En el caso particular de tipos de datos complejos, como los encontrados en programación orientada a objetos, una de las técnicas utilizadas se basa en el uso de mecanismos de reflexión provistos por el lenguaje, y especificaciones para evitar la generación de entradas inválidas. Sin embargo, cuando las especificaciones son débiles pueden llevar a construir objetos espurios, es decir, inválidos o no construibles por la API asociada al tipo del objeto. Estos objetos pueden llevar a falsos negativos: tests que fallan cuando no existe un bug e incrementan el trabajo del tester que deberá filtrar los tests que efectivamente evidencian un bug de aquellos que fallan solo por una entrada inválida; y falsos positivos: tests que deberían fallar pero que no lo hacen debido a que la entrada inválida enmascara el bug. En este trabajo evidenciaremos el problema mediante un ejemplo y delinearemos los componentes y pasos necesarios para construir una técnica que detecte cuando un objeto es inválido con respecto al API.

1. Introducción

Garantizar que un producto de software realiza correctamente las tareas para las cuales fue desarrollado corresponde a una de las tareas más importantes en ingeniería de software. Corresponde a un problema indecidible que puede ser atacado mediante enfoques analíticos (ej.: cálculo de precondition más debil) no completamente automatizables, o por un enfoque empírico el cual puede ser completamente automatizado.

Testing es uno de los enfoques más utilizados para garantizar que un producto de software realiza correctamente las tareas para las cuales fue desarrollado [6,12,8]. Siendo el testing un enfoque empírico, las garantías que ofrece son débiles en comparación con enfoques analíticos. Aunque dentro de sus ventajas se encuentra su facilidad de uso y su escalabilidad, este enfoque consiste en ejecutar el programa bajo evaluación sobre un conjunto (finito) de entradas y contrastar el comportamiento del programa obtenido con el esperado [1]. Evidentemente cómo se elige el conjunto de entradas sobre las cuales evaluar al programa, tiene un gran impacto en la garantía de que el programa es correcto, cuando los tests pasan.

Generar tests de manera automática es actualmente un área importante de investigación en ingeniería de software. Para comprender las dificultades de este problema es necesario considerar las tres partes que constituyen un test. **Preparación**, la combinación de los argumentos del programa a evaluar y el estado del sistema sobre el que se va a ejecutar al mismo; **Ejecución**, la ejecución del programa sobre el escenario anterior; **Evaluación**, la evaluación del comportamiento obtenido de la ejecución del programa, incluyendo resultados y cambios de estado del sistema, con respecto al comportamiento esperado. En este trabajo nos centraremos en la generación automática de entradas.

Para tipos de datos básicos como tipos primitivos, arreglos de tipos primitivos, y algunas estructuras de datos, la generación automática de valores es relativamente simple. Sin embargo, ciertas estructuras de datos requieren combinación de métodos con determinados valores y en un orden particular para producir una instancia específica de dicha estructura. Un caso simple es una lista simplemente encadenada con cache de nodos (*NodeCachingLinkedList* [2]). Esta estructura funciona como una lista simplemente encadenada con la diferencia de que al eliminar un elemento de la lista, el nodo asociado se guarda en una cache en lugar de dejar que el *garbage collector* lo elimine de la memoria, o para evitar la llamada de métodos de destrucción como en *C++*, luego al insertar un elemento se reutilizan los nodos almacenados en la cache. Para esta estructura, producir una instancia de una lista vacía con dos nodos en la cache, requiere llamar dos veces al método *insertar* y luego dos veces al método *eliminar*. En la sección 2 se mencionarán las dos metodologías más utilizadas para generar entradas para tipos complejos de datos.

Una de las técnicas utilizadas para generar entradas de tipos complejos se basa en el uso de mecanismos de reflexión e invariantes de representación. Cuando estos invariantes son débiles, es decir, fallan en considerar aspectos cruciales sobre la correcta implementación de un tipo, estas técnicas pueden llevar a la construcción de entradas inválidas. Nuestra propuesta es mostrar en detalles este problema y sus efectos negativos en el contexto de testing, y esbozar una técnica que permita la detección de estas entradas inválidas.

2. Preliminares

Como mencionamos, la generación automática de entradas para datos complejos no es una tarea trivial. Los dos enfoques utilizados actualmente para la generación de entradas para estructuras de datos complejas son:

Generación basada en reflexión y especificaciones, es utilizada en lenguajes que proveen los mecanismos apropiados para modificar valores de un objeto en tiempo de ejecución, ignorando los modificadores de acceso al mismo (ej.: modificar directamente el valor de un campo *private* en *Java*). No utilizar la *API* para construir las entradas, lleva a que sea posible generar instancias inválidas, por esto es que este tipo de generación requiere que se provea una especificación del invariante del tipo de la entrada, comúnmente provista por un método *repOK* que no toma parámetros y retorna si una instancia satisfac-

ce o no el invariante del tipo. Herramientas basadas en este enfoque incluyen *Korat* [3] para generación de entradas, y *TACO* [5] como herramienta de verificación automática de especificaciones que utiliza reflexión y especificaciones para construir contraejemplos¹.

Generación basada en secuencias de métodos, que utiliza la *API* del tipo de la entrada a generar para construir instancias particulares de la misma. Este enfoque tiene la ventaja de no generar instancias inválidas si la implementación es correcta, mientras que la generación basada en reflexión puede utilizarse incluso si los métodos de la *API* están incorrectamente implementados o directamente no están provistos. *EvoSuite* [4] y *Randoop* [11] son ejemplos de herramientas que generan entradas² mediante el uso de secuencias de métodos.

3. Objetos espurios en generación automática de entradas

En este trabajo nos vamos a centrar en la generación de entradas de tipos de datos complejos basada en reflexión y especificaciones, cuando éstas últimas son débiles, más precisamente, cuando las instancias generadas no pueden ser construidas mediante los métodos asociados al tipo de la entrada (*API*).

Es necesario, antes de continuar, introducir la noción de invariante de representación y que significa que éste sea débil: *Un invariante de representación establece las propiedades que debe cumplir un objeto para ser considerado válido con respecto al tipo abstracto que representa*³ [9].

Si la especificación utilizada es muy débil habrá instancias de un tipo concreto que sean erróneamente consideradas como representaciones válidas del tipo abstracto. Por ejemplo, que el invariante de *NodeCachingList* no exija que un nodo no pertenezca al mismo tiempo a la cache y a la lista.

A modo de motivación, consideremos un grafo dirigido *DirectedGraph* implementado mediante listas de adyacencias. La implementación se basa en un arreglo de vértices donde cada uno posee un identificador y una lista de vértices adyacentes. El grafo de la Figura 1a debería estar representado por un arreglo de 3 vértices con, etiqueta 1 y la lista $L1(2,3)$ para el primero, etiqueta 2 y lista $L2(2,3)$ para el segundo, y finalmente etiqueta 3 y lista $L3()$ para el tercero. Si bien los métodos de grafos sobre listas de adyacencias deberían asegurar la representación para la instancia anterior, una construcción basada en reflexión podría utilizar la misma lista para los vértices 1 y 2, tal como se muestra en la Figura 1b, si la especificación del invariante no verifica que cada vértice debe estar asociado con una lista única. Al utilizar *Korat* para generar instancias de *DirectedGraph* una de las obtenidas es justamente la que acabamos de presentar.

¹ Entradas particulares que cumplen con la precondition del programa a evaluar pero causan que se viole la postcondition del mismo.

² Como parte de generación automática de tests

³ Desde otra perspectiva, un invariante de representación define el subconjunto de todas las instancias de un tipo concreto que representan a un tipo abstracto.

⁴ A modo de simplificación solo se muestran los vértices del grafo con sus correspondientes listas de adyacentes.

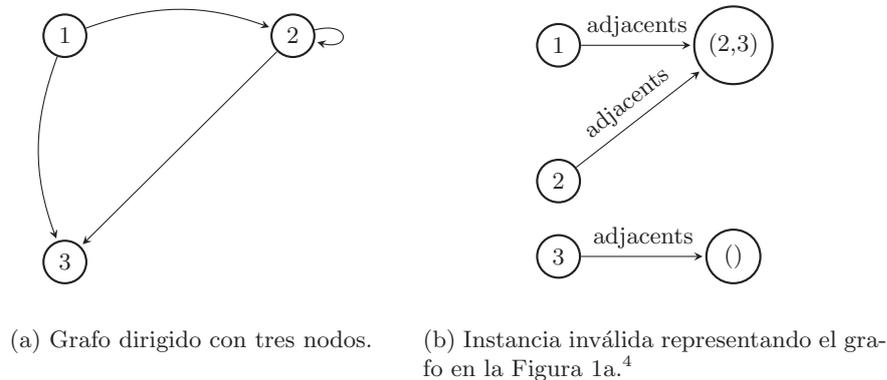


Figura 1: Ejemplo de objeto espurio.

En el contexto de testing, la segunda instancia podría llevar a falsos negativos, es decir, un test que falla porque la entrada es inválida; y a falsos positivos cuando existe un bug pero no es detectado porque la entrada inválida hace que éste no sea visible. Consideremos la API de *DirectedGraph* de la Figura 2 y los dos tests de la Figura 3.

El primer test de la Figura 3 falla porque al usar la misma lista de adyacentes para los vértices 1 y 2, al conectar 1 con 1, también se conecta 2 con 1. El segundo test utiliza la misma instancia del anterior, y el método *disconnectVertex* que está incorrectamente implementado, sin embargo el test pasa porque al estar compartida la lista de adyacentes entre los vértices 1 y 2, eliminar el vértice 3 de los adyacentes del vértice 1 logra el mismo efecto para el vértice 2.

4. Detectando objetos espurios

Dado un objeto generado mediante reflexión, y la API asociada al tipo del mismo, detectar si este objeto es inválido consiste en verificar que éste no pertenece al conjunto de todos los objetos construibles mediante la API. Considerando que generar todo el conjunto de objetos construibles resulta imposible o sumamente ineficiente en el mejor caso, es necesario considerar el siguiente problema:

Dado un objeto y una API, buscar si existe una secuencia de operaciones provistas por la misma que permita construir el objeto.

Esto es consistente con un problema de búsqueda, en el sentido general ya que el problema consiste en buscar una solución en un conjunto de soluciones candidatas, a continuación analizaremos todos los subproblemas que requieren ser resueltos antes de poder desarrollar una técnica para detectar si un objeto generado mediante reflexión es inválido (no construible mediante la API) o si es posible su construcción, en cuyo caso se debería proveer una secuencia de operaciones que lo construye.

```

public class DirectedGraph {
    /*
     * Los vértices del grafo, cada Vertex tiene una
     * etiqueta asociada y una lista de adyacentes.
     */
    private Vertex[] vertices;

    /**
     * Crea un nuevo grafo con n vértices
     */
    public DirectedGraph(int n) {...}

    /**
     * Conecta el vértice a con el vértice b
     */
    public void addEdge(int a, int b) {...}

    /**
     * Remueve la conexión del vértice a al b
     * (no remueve la conexión inversa)
     */
    public void removeEdge(int a, int b) {...}

    /**
     * Remueve todas las conexiones hacia un nodo
     */
    public void disconnectVertex(int a) {
        for (Vertex v : vertices) {
            if (connected(v.label, a)) {
                removeEdge(v.label, a);
                break; //bug
            }
        }
    }

    /**
     * Retorna true si y solo si existe un camino
     * desde el vértice a al vértice b
     */
    public boolean connected(int a, int b) {...}
}

```

Figura 2: Implementación de la clase *DirectedGraph*.

4.1. Representación

Representar a cada elemento en el espacio de búsqueda directamente como un objeto construido mediante la API es posible pero lleva acarreado varios problemas: dificultad en obtener la secuencia de operaciones que lo construye, para obtener a la misma sería necesario llevar un historial de que operaciones (y con que valores) fueron aplicadas; y dificultad en la implementación de enfoques asociados a algoritmos genéticos, teniendo en cuenta que los elementos dentro de estos algoritmos se deben representar en forma de genes para las operaciones de *crossover* en donde los genes de dos candidatos son *entrecruzados* para generar un nuevo candidato [10]. A partir de estas observaciones creemos que la mejor representación sería utilizar la secuencia de operaciones (junto a sus argumentos) que construye cada objeto en el espacio de búsqueda.

```

@Test
public void addEdgeTest() {
    //Falso negativo, el test falla pero no existe un bug
    DirectedGraph graph = invalidInstance();
    graph.addEdge(1,1);
    assertTrue(graph.connected(1,1));
    assertFalse(graph.conneted(2,1)); //falla
}

@Test
public void disconnectNodeTest() {
    //Falso positivo, el test pasa cuando existe un bug
    DirectedGraph graph = invalidInstance();
    graph.disconnectVertex(3);
    assertFalse(graph.connected(1,3)); //no falla
    assertFalse(graph.connected(2,3)); //no falla
    assertFalse(graph.connected(3,3)); //no falla
}

```

Figura 3: Dos tests para *DirectedGraph* en donde el tester piensa utilizar el grafo de la Figura 1a pero obtiene la representación inválida de la Figura 1b.

4.2. Exhaustividad del espacio de búsqueda

El problema tal como fue planteado anteriormente requiere acotar el espacio de búsqueda, dentro de las cotas provistas es posible considerar de manera exhaustiva a todas las secuencias de operaciones a analizar, o puede considerarse un subconjunto. La ventaja de no hacer un análisis exhaustivo es principalmente la disminución en los recursos requeridos (tiempo y memoria principalmente), la desventaja es que no ser exhaustivo puede llevar a no encontrar una secuencia de operaciones que existe dentro de las cotas consideradas, aumentando las chances de dar una respuesta negativa y que se utilicen cotas más grandes que las necesarias para intentar encontrar una secuencia que construya al objeto buscado. Esto lleva a una observación importante, si el objeto no pudo ser construido no necesariamente significa que éste sea inválido, solo que dentro del espacio de candidatos evaluado, no existe una secuencia de operaciones que lo construya.

A modo de ejemplo consideremos un tipo *List* con solo el método *add(int)*, que agrega un valor al final de la lista, y la instancia $(3,3)$. Las cotas que se deben establecer son la longitud máxima de las secuencias y que valores se van a utilizar (en este caso por el método *add(int)*). Definir que valores asignar a estas cotas no es un problema trivial, cotas muy restrictivas llevan a no poder encontrar la secuencia que construye el objeto, incluso cuando ésta existe; mientras que cotas muy relajadas tienen asociadas un crecimiento del espacio de búsqueda, considerando una solución por búsqueda basada en recorrido de árboles (*Depth First Search*, *Breadth First Search*, *Best First Search*, etc), el tamaño del espacio de búsqueda está dado por la fórmula $\sum_{i=0}^h m^i$, con h la cota establecida para el tamaño máximo de las secuencias de operaciones a considerar, y m por cuantas operaciones y valores utilizar. Específicamente para el ejemplo de *List*, h es el tamaño máximo de las listas, y m la cantidad de valores a utilizar para el método

$add(int)$. En la Figura 4 se muestra el árbol de búsqueda para el ejemplo de anterior con las cotas mencionadas.

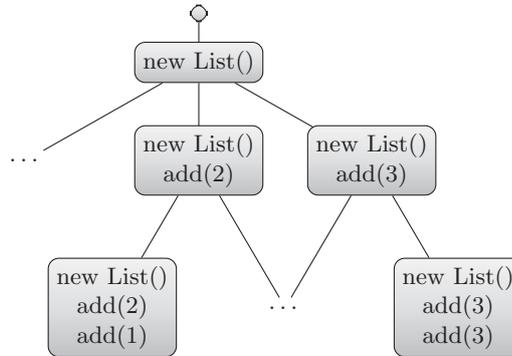


Figura 4: Búsqueda para la lista $(3,3)$ dentro de un espacio acotado de todas las listas de tamaño 0 a 3 con valores a insertar del 1 al 3 y solo utilizando el constructor vacío y el método $add(int)$.

4.3. Operaciones redundantes y observadoras

Métodos observadores: No todas las operaciones asociadas a un tipo realizan modificaciones al estado, muchas permiten observar datos del objeto (ya sean datos computados a partir del estado, ej.: $List\#contains(int)$, o acceso de lectura a valores específicos, ej.: $List\#size()$). Estas operaciones nunca deberían ser necesarias para construir un objeto particular, y a su vez incrementan el espacio de búsqueda.

Detectar cuando una operación es observadora corresponde a un problema indecidible, aunque es posible dar soluciones aproximadas. En [7] los autores presentan un sistema de tipos junto a verificación estática para identificar métodos observadores (llamados también *puros*).

Operaciones redundantes: No es raro encontrar APIs que definen operaciones similares donde una se puede definir en base a la otra, como por ejemplo las operaciones de $add(int)$ y $add(int, int)$ de $List$ que agregan un elemento al final de la lista o en un índice particular respectivamente, es posible definir la primera a partir de la segunda como $add(i) = add(i, size())$. Este tipo de operaciones no son equivalentes pero generan un conjunto de secuencias representando al mismo objeto lo que a su vez lleva a incrementar el espacio de búsqueda. Si bien este también es un problema indecidible, resulta incluso más complejo dar una solución aproximada.

4.4. Evaluación de un candidato

Finalmente un punto muy importante en todo problema de búsqueda es poder determinar cuando la solución es finalmente encontrada. Para el problema planteado en este trabajo, esta evaluación deberá constatar si el objeto de entrada fue construido por alguna secuencia. La comparación se deberá hacer entre el objeto a buscar y el objeto generado por la secuencia bajo evaluación. Comúnmente funciones como *equals* o *compareTo* podrían ser utilizadas pero tal como la funciones de hash o de representación (al estilo *toString()* en *Java*) muchas veces ignoran ciertos valores de una estructura, o utiliza valores que no hacen a la misma, particularmente en *Java* el hash por defecto contiene valores aleatorios, esto lleva a que objetos distintos puedan ser considerados iguales o que objetos iguales puedan ser considerados distintos, ambos casos son perjudiciales para el proceso de búsqueda. También es necesario considerar que estas funciones pueden no estar provistas, o aplicarlas sobre un objeto inválido puede llevar a excepciones.

5. Delineamientos para detectar un objeto espurio

En base a lo discutido en la sección 4 delinearemos a continuación una técnica para detectar objetos espurios. El objetivo es dar un punto de inicio para el desarrollo de una técnica eficaz, siempre dentro del contexto de búsqueda acotada, y eficiente.

Nuestro primer enfoque se basa la generación exhaustiva acotada de objetos mediante la API, para la construcción del espacio de búsqueda. El resultado de esta etapa es un conjunto de objetos, la ventaja de este enfoque inicial es que es posible cambiar la herramienta de generación del espacio de búsqueda sin tener que modificar la búsqueda en el espacio de candidatos. La evaluación de candidatos, es decir, dado un candidato c evaluar si se cumple $equal(c, o)$ donde o representa el objeto objetivo, requiere una representación de los objetos a comparar, anteriormente en 4.1 se mencionó el uso de funciones provistas por el mismo objeto (*equals*, *compareTo*, *hashCode*, y *toString*), todas con problemas asociados. Creemos que una representación basada en recorrer la estructura de un objeto siguiendo un orden determinado de los campos (como recorrido Depth first search) permite independizarse de la existencia y correcta implementación de los métodos anteriores evitando que información necesaria no sea considerada e información irrelevante (como el hash por defecto de cada objeto) lo sea.

La generación exhaustiva acotada basada exclusivamente en el uso de operaciones de la API presenta dos problemas importantes: pueden existir varias secuencias de operaciones que generen el mismo objeto; y acotar el tamaño de las secuencias generadas no necesariamente acota de la misma forma a la estructura, a modo de ejemplo consideremos una *NodeCachingList*, para construir la lista que tiene un solo elemento y dos nodos en la cache es necesario 6 operaciones: la construcción de la lista, agregar 3 elementos y eliminar 2; mientras que la estructura solo contiene 2 nodos en la cache y 1 en la lista. Por lo tanto es necesario que la herramienta permita utilizar cotas sobre la estructura. Obtener

automáticamente las cotas a utilizar representa un subproblema muy complejo de resolver, nuestro enfoque considera que las cotas a utilizar son provistas por el usuario. A su vez, la exhaustividad de la generación estará basada en los objetos a generar, en lugar de las secuencias a generar. Para esto consideraremos todas las relaciones entre campos y valores de acuerdo a las cotas provistas y el objetivo consiste en “cubrir” todas las relaciones durante la generación del espacio de candidatos.

Finalmente la búsqueda es inicialmente lineal, tal como se puede apreciar en el algoritmo de la Figura 5, en donde *generateCandidates* representa la herramienta de generación exhaustiva, basada exclusivamente en el uso de operaciones de la API, y *fieldBasedRepresentation* es el método que retorna la representación discutida anteriormente.

Para retornar una secuencia de operaciones que construye a un objeto solo es necesario considerar los candidatos como listas de operaciones y disponer de un método que retorne un objeto a partir de estas operaciones.

```

isValid(Object o, Bounds b) {
    List<Object> candidates = generateCandidates(b);
    Representation oRep = fieldBasedRepresentation(o);
    for (Object c : candidates) {
        Representation cRep = fieldBasedRepresentation(c);
        if (oRep.equals(cRep)) {
            return true;
        }
    }
    return false;
}

```

Figura 5: Algoritmo inicial para detección de objetos inválidos.

6. Conclusiones

Hemos presentado el problema de objetos inválidos en la generación automática de entradas basada en mecanismos de reflexión y especificaciones. Inicialmente presentamos un ejemplo de un tipo para el cual la herramienta *Korat* genera un objeto inválido. Luego presentamos el problema de detección de objetos inválidos como un problema de búsqueda detallando que subproblemas surgen. Finalmente dimos un esbozo de una técnica para detectar estos casos. Es importante destacar que el ejemplo mostrado fue satisfactoriamente detectado con una implementación inicial del algoritmo propuesto. Sin embargo es claro ver que este enfoque no escala al considerar objetos “más grandes” que requieren aumentar las cotas utilizadas y por ende realizar la búsqueda sobre un espacio de candidatos mucho mayor. Esto abre una vía de investigación para resolver este problema de manera más eficiente, considerando el objetivo final

de obtener la secuencia que construye (si existe dentro de las cotas provistas) un objeto particular.

Referencias

1. Paul Ammann and Jeff Offutt. *Introduction to Software Testing*. Cambridge University Press, New York, NY, USA, 2nd edition, 2016.
2. Apache Software Foundation. Commons collection. <https://commons.apache.org/proper/commons-collections>.
3. Chandrasekhar Boyapati, Sarfraz Khurshid, and Darko Marinov. Korat: automated testing based on java predicates. In Phyllis G. Frankl, editor, *Proceedings of the International Symposium on Software Testing and Analysis, ISSTA 2002, Roma, Italy, July 22-24, 2002*, pages 123–133. ACM, 2002.
4. Gordon Fraser and Andrea Arcuri. Evosuite: automatic test suite generation for object-oriented software. In Tibor Gyimóthy and Andreas Zeller, editors, *SIGSOFT/FSE'11 19th ACM SIGSOFT Symposium on the Foundations of Software Engineering (FSE-19) and ESEC'11: 13th European Software Engineering Conference (ESEC-13), Szeged, Hungary, September 5-9, 2011*, pages 416–419. ACM, 2011.
5. Juan P. Galeotti, Nicolás Rosner, Carlos Gustavo López Pombo, and Marcelo F. Frias. TACO: efficient sat-based bounded verification using symmetry breaking and tight bounds. *IEEE Trans. Software Eng.*, 39(9):1283–1307, 2013.
6. Carlo Ghezzi, Mehdi Jazayeri, and Dino Mandrioli. *Fundamentals of software engineering*. Prentice Hall, 1991.
7. Wei Huang, Ana Milanova, Werner Dietl, and Michael D. Ernst. Reim & reiminfer: checking and inference of reference immutability and method purity. In Gary T. Leavens and Matthew B. Dwyer, editors, *Proceedings of the 27th Annual ACM SIGPLAN Conference on Object-Oriented Programming, Systems, Languages, and Applications, OOPSLA 2012, part of SPLASH 2012, Tucson, AZ, USA, October 21-25, 2012*, pages 879–896. ACM, 2012.
8. Pankaj Jalote. *An Integrated Approach to Software Engineering*. Springer-Verlag, Berlin, Heidelberg, 2nd edition, 1997.
9. Barbara Liskov and John V. Guttag. *Program Development in Java - Abstraction, Specification, and Object-Oriented Design*. Addison-Wesley, 2001.
10. Melanie Mitchell. *An introduction to genetic algorithms*. MIT Press, 1998.
11. Carlos Pacheco and Michael D. Ernst. Randoop: feedback-directed random testing for java. In Richard P. Gabriel, David F. Bacon, Cristina Videira Lopes, and Guy L. Steele Jr., editors, *Companion to the 22nd Annual ACM SIGPLAN Conference on Object-Oriented Programming, Systems, Languages, and Applications, OOPSLA 2007, October 21-25, 2007, Montreal, Quebec, Canada*, pages 815–816. ACM, 2007.
12. Roger S. Pressman. *Software Engineering: A Practitioner's Approach*. McGraw-Hill Higher Education, 5th edition, 2001.

Directrices para mejorar la usabilidad de los frameworks web

Marcela A. Constanzo y Sandra I. Casas,

GISP - Instituto de Tecnología Aplicada
Universidad Nacional de la Patagonia Austral
Río Gallegos, Argentina, mconst1978@hotmail.com

Resumen. Los frameworks web son herramientas para construir rápidamente aplicaciones web a través de diferentes mecanismos (componentes, clases, plantillas, descriptores, archivos de configuración, etc.). La documentación proporcionada a los usuarios es un aspecto clave para la usabilidad del framework, ya que su objetivo es facilitar y ayudar en su uso y optimizar el proceso de desarrollo, reduciendo el tiempo, el esfuerzo y los costos. La identificación y clasificación de los problemas en el uso de estas herramientas, provenientes de los usuarios de frameworks, puede aportar información para la elaboración de directrices y guías que permitan mejorar la usabilidad del framework web en general y la documentación en particular. Debido a la escasez de propuestas de usabilidad (métodos y/o modelos de evaluación, directrices y/o guías) específicos para el framework web, este estudio propone identificar, caracterizar y analizar los problemas de usabilidad de los recursos y la documentación que actualmente están disponibles para los desarrolladores, y a partir de estos resultados, presentar un conjunto de directrices para facilitar el uso de los frameworks web.

Palabras Claves: Usabilidad, Framework web, Documentación, Directrices

1 Introducción

Un framework web, es un conjunto de componentes con un diseño reutilizable que facilita y agiliza el desarrollo de aplicaciones Web. Se basan principalmente en el Modelo Vista Controlador (MVC), ofreciendo mecanismos para mejorar la integración con otras herramientas para la implementación de las capas de negocio y presentación [1].

Los frameworks son particularmente difíciles de aprender y comprender por los usuarios principiantes, especialmente si no son acompañados con la documentación apropiada [2] [3]. La documentación de buena calidad es un factor de éxito fundamental para la reutilización del framework ya que ayuda a la comprensión, guía a los desarrolladores en el proceso de personalización, explica sus principios de diseño y detalles y evita errores [2] [4].

La complejidad resulta de los diferentes públicos que lo utilizan, desarrolladores de aplicaciones, desarrolladores de frameworks, encargados de proyectos, de los diferentes estilos de documentos y diferentes notaciones. La identificación y clasificación de

los problemas en el uso de estas herramientas, provenientes de los usuarios de frameworks, puede aportar información para la propuesta de métodos de evaluación de usabilidad.

La usabilidad es una propiedad inherente a la calidad del software debido a que expresa la relación entre el software y su dominio de aplicación [3] [5]. La usabilidad es una de las características principales que utiliza el Modelo de Calidad de ISO/IEC 25010 y puede ser valorada en término de 4 subcaracterísticas, Comprensibilidad, Aprendibilidad, Operabilidad y Atractividad [6] [7]. La usabilidad aplicada a los frameworks web se refiere a la capacidad de un framework de ser entendido, aprendido, usado y protegido de errores por el usuario, es de interés tanto para desarrolladores de aplicaciones web como para los desarrolladores de frameworks web.

La evaluación de la usabilidad ha aportado diversos métodos, modelos y lineamientos/directrices, principalmente enfocados a aplicaciones web, escritorio y móviles. Sin embargo en el contexto de herramientas de desarrollo de software, particularmente para el caso de frameworks web, no se encuentran claramente sistematizaciones de métodos, modelos, lineamientos y guías de usabilidad.

El presente trabajo expone resultados de estudios realizados a los recursos disponibles en los sitios oficiales de un grupo de frameworks web, que permitieron identificar los aspectos que presentan mayor dificultad en su uso y que características debe cumplir la documentación que acompaña a los mismos con el objeto de presentar un conjunto de directrices para mejorar la usabilidad de los frameworks web. Algunos de estos estudios ya han sido documentados en [8] y [9] y en los mismos ya se han presentado otros aportes.

Este estudio continúa de la siguiente manera, en la Sección 2 se resumen los trabajos relacionados, en la Sección 3 se describe la metodología utilizada, en la Sección 4 los resultados obtenidos, en la Sección 5 se presentan las directrices y en la Sección 6 se detallan las conclusiones y futuros trabajos.

2 Trabajos Relacionados

La relevancia de la usabilidad, ha dado lugar a numerosos estudios, métodos y modelos para evaluarla, con especial énfasis en aplicaciones de usuario final (web-escritorio, móvil, etc.). Se encuentran trabajos en usabilidad de aplicaciones [10], [11], [12] y [13] que proponen métodos y técnicas para la evaluación y/o mejora de la usabilidad, pero éstos no son apropiados o adecuados para frameworks, por no considerar variables propias de los mismos.

El desarrollo de software depende en gran medida del uso de bibliotecas o librerías, APIs, frameworks y componentes reutilizables, entre otras herramientas. Los problemas de usabilidad de estas herramientas a menudo aumentan el costo de desarrollo y reducen la calidad del código. Se encuentran algunas propuestas para APIs [14], [15], [16], en las cuales se presentan estudios que identifican problemas que reflejan errores de usuarios y factores de usabilidad. Estos trabajos estudian la usabilidad de las APIs, a partir de criterios como documentación, dependencia, adaptabilidad, capacidad de aprendizaje y complejidad, y aplican técnicas de encuestas, segui-

miento a errores y bugs, e identifican diversos problemas que dificultan el uso de las APIS (ubicación, documentación, conflictos, semántica, parámetros etc.).

El trabajo de [5] estudia la usabilidad de Componentes COTS, utiliza diversos factores de análisis, entre los que se encuentran comprensibilidad, operabilidad, aprendibilidad y sobre criterios aplicados a los manuales (efectividad, contenido, calidad, etc.) y varios criterios aplicados a la complejidad.

3 Metodología

Con el objeto de identificar qué tipo de documentación es la más utilizada y cuáles son las características más relevantes que permiten a los usuarios facilitar el uso de frameworks web, se realizó un estudio exploratorio y descriptivo sobre el contenido de los distintos tipos de soporte y recursos disponibles y publicados en los sitios oficiales de los frameworks web populares entre los desarrolladores. Se buscó analizar todo tipo de soporte y recurso que se provea y sirva para el uso del framework, independientemente de su formato (documento textual/ visual/ etc.).

El desarrollo del trabajo constó de tres estudios. Se aplicaron técnicas de investigación y fuentes de información diversas para llevar a cabo la identificación y caracterización de problemas de usabilidad de framework mediante la recolección de datos. Primero, se realizó un estudio exploratorio y descriptivo mediante la observación de diversos framework que disponen de sitios web oficiales, ya presentado en [8], en particular se relevó y analizó todos los recursos y soportes de documentación que ofrecen a los desarrolladores.

Luego se analizaron manualmente las entradas a los foros existentes de estos framework con el objeto de obtener información sobre las problemáticas planteadas y/o consultadas por los desarrolladores, descripto en [9]. Finalmente se realizó una encuesta a desarrolladores del sector de desarrollo de software en Argentina para confirmar los resultados obtenidos en los estudios precedentes y para completar y ampliar la información de la problemáticas identificadas.

Los resultados de estos tres estudios permitieron redactar un conjunto de directrices para mejorar la documentación y facilitar el uso de los frameworks web en relación a su aprendizaje, como también obtener mejoras en el sitio oficial.

4 Resultados obtenidos de los estudios realizados

Este trabajo se centra en la usabilidad y se basa en la Norma ISO/IEC 25010/11, adaptándola particularmente a los frameworks web, así la Usabilidad de un framework web es la capacidad del framework para ser entendido, aprendido, usado, cuando se usa bajo determinadas condiciones.

Se valorará en base a las siguientes características: *comprensibilidad* (capacidad del framework que permite al usuario entender si es adecuado para sus necesidades, bajo condiciones de uso), *aprendibilidad* (se refiere a la facilidad de aprender la funcionalidad básica, para que el usuario realice la tarea), *operabilidad* (capacidad del framework para permitir al desarrollador operar con él y controlarlo) y *protección*

contra errores de usuario (capacidad del framework para proteger a los usuarios de cometer / incurrir errores).

4.1 Identificación y análisis de los recursos y soportes de documentación

Tal lo desarrollado en [9], el relevamiento consistió en un examen exhaustivo de los sitios oficiales de los frameworks web identificando y analizando todos y cada uno de los recursos y soportes disponibles. Se consideró como recurso, aquellos materiales en soporte de tipo documental y audiovisual, como así también todo servicio que preste el sitio para asistir a los desarrolladores en el uso del framework. Se identificaron los componentes prioritarios de cada uno de los recursos y las características comunes a todos los frameworks. También a partir del análisis de sus contenidos y propósitos se estableció el factor de la usabilidad que cubren.

Las Tabla 1 y Tabla 2, aportan fundamentalmente dos datos: a) la cantidad de recursos de documentación por framework y b) la cantidad de frameworks que usan los distintos Recursos.

Tabla 1. Cantidad de Recursos disponibles por Framework web.

FRAMEWORK	F1	F2	F3	F4	F5	F6	F7	F8	F9	F10	F11	F12	F13	F14	F15	F16	F17
RECURSOS	6	10	6	8	9	7	9	9	6	7	6	7	8	7	6	6	4
F1	Laravel			F4	Ruby on Rails		F7	Symfony	F10	Zend			F13	React		F16	Vue
F2	Yii			F5	AngularJS		F8	Ember.js	F11	Backbone.js			F14	Aurelia.js		F17	Mercury
F3	CodeIgniter			F6	Meteor		F9	CakePHP	F12	Knockout.js			F15	Polymer			

Tabla 2. Cantidad de Framework web por Recursos.

RECURSOS	F	B	C	L	O	G	M	I	T	U	E	V	A	H
FRAMEWORK	7	12	3	5	5	14	1	6	9	1	15	10	17	16
F	Foro				O	Comunidad	G	GetStarted	U	Guías	A	API		
B	Blog				M	Manuales	I	Libros	E	Ejemplos	H	GitHub		
L	Lista de Mails				C	Chat	T	Tutoriales	V	Videos				

Estos recursos se pueden relacionar con la usabilidad del framework, ejemplos, tutoriales, get started, videos, con la aprendibilidad, debido a que esta característica se refiere a la facilidad de aprender la funcionalidad básica, para que el usuario realice la tarea correctamente y estos soportes, correctamente documentados, permiten eso.

Los foros, blogs, github, chats, comunidades, presentan relación directa con la operabilidad, esta capacidad de usabilidad que permite al usuario operarlo y controlarlo con facilidad, este recurso ayuda, mediante la interacción de los usuarios, a operar los diferentes componentes que lo integran y evitar o resolver los errores que puedan presentarse; también se asocia con la aprendibilidad ya que permiten aprender las funcionalidades que ofrece el framework.

Las APIs, tutoriales y github son los recursos que permiten comprender si el framework se adapta a las necesidades que el usuario necesita, en el caso de las APIs por ser código que forma parte del mismo, y tutoriales y github por tener información que explica el contenido y la estructura del frameworks y permite conocer de una manera correcta los elementos que lo componen. Los foros, blogs, get started, manuales, libros, tutoriales, guías, ejemplos, videos, APIs y github se pueden asociar con la

protección contra errores porque son recursos que permiten cierta protección a los usuarios sobre los errores que puedan presentarse al usar el framework.

4.2 Análisis de Foros

En este análisis el enfoque es particular sobre los foros, un foro es una aplicación web, generalmente organizada en categorías, que da soporte a discusiones u opiniones en línea. Son contenedores en los que se abren temas de discusión y en los cuales los usuarios cumplen el rol más importante, por la interacción y el dinamismo, ya que participan con consultas, preguntas y también responden con sus opiniones. Esta interacción de los usuarios permite obtener un resumen de los temas consultados por los usuarios del framework web, categorizarlos y organizarlos de acuerdo a su importancia en relación a la usabilidad [9].

Se trabajó sobre un subconjunto de los frameworks seleccionados anteriormente, en razón que solo 6 de estos disponen de foros en sus sitios (Knockout.js, Yii, Codelgniter, Meteor, React y Laravel).

El proceso de análisis se realizó en varios pasos, considerando que estos foros presentan distintas estructuras, categorías, secciones y en algunos casos ninguna organización o estructura. Las categorías temáticas que se aplicaron son aquellas relacionadas con las actividades o tareas que requiere el uso del framework y los componentes del mismo, y se considera que tienen influencia en la usabilidad de la herramienta. Estas categorías son, APIs, Instalación, Configuración, Plantillas, Testing, Código y Versiones. Luego, sobre estas temáticas se clasificó el tipo de asistencia requerida por el usuario, en consulta (cómo hacer una tarea o usar un componente) o error (reporte de error, cómo se resuelve y/o cual es la causa del mismo). El total de entradas analizadas fue de 20104, a las cuales el tipo de entrada se representa en la Figura 4. La proporción de consultas (42%) y errores (22%) corresponden a la categorías definidas. En cuanto al tipo otras (36%), las mismas refieren a consultas o a errores, pero que no se ha podido clasificar en una de las categorías definidas por ser muy específicas de los framework.

El estudio de los foros de 6 frameworks web, sobre un total de 20104 entradas, indica que el principal problema de los usuarios son los errores, principalmente los errores relacionados con el código, la conexión a las bases de datos y el uso de las APIs. Esto podría estar orientado a la escasa documentación existente en los sitios oficiales en relación a estos temas o a cómo realizar determinadas tareas tales como la instalación o configuración de los componentes y sobre todo la implementación de APIs.

La revisión realizada sobre los recursos existentes tales como tutoriales, ejemplos, get started, videos, documentación de APIs, no cuenta con información sobre los errores que puedan surgir con el uso del framework, por eso se indica que la documentación es escasa, insuficiente, incompleta y se complementa con el análisis sobre los foros.

4.3 Aportes de los usuarios de frameworks web

Con el objeto de contrastar y validar los resultados obtenidos en los estudios anteriores y también obtener más información referente a los problemas de la usabilidad de

frameworks web, se realizó una encuesta a desarrolladores web. Los cuestionamientos sobre los que se pretenden indagar son, los recursos y soporte disponible en los sitios oficiales de los frameworks web más utilizados, el rol que cumple la documentación, los motivos por los cuales es complejo usar los frameworks web, los errores más habituales al utilizar los frameworks web, las características o componentes que pueden incidir en el grado de usabilidad de frameworks web y de qué manera se puede reducir la complejidad en el uso de un framework web.

La encuesta consistió en 12 preguntas, de las cuales 9 fueron respuestas cerradas y concretas basadas en los resultados obtenidos del trabajo desarrollado y las otras 3 con respuestas abiertas orientadas a recolectar opiniones y experiencias personales relacionadas con la usabilidad de los frameworks web. La encuesta estuvo disponible durante 30 días, desde 15/03/2017 hasta el 15/04/2017. Los encuestados corresponden a desarrolladores de empresas y organizaciones adheridas a la Cámara de la Industria Argentina de Software (CESSI)

En relación a los encuestados, de un total de 51, el 68% residen en la Ciudad Autónoma de Buenos Aires (CABA), y el 32 % restante corresponden a las provincias de Santa Cruz, Salta, Córdoba, Misiones, Santa Fé. El 60% desarrolla la actividad en el sector privado, el 29% en el sector público y el 11% en el sector académico. En relación a los años de experiencia en el desarrollo de aplicaciones con frameworks web de los encuestados, el 62,7% cuenta con más de 5 años de experiencia en la utilización de frameworks para este tipo de desarrollo, el 27, 5% con 2 a 5 años de experiencia y solo el 9,8% son novatos en el tema.

En relación a las preguntas abiertas, los encuestados opinaron sobre problemas más comunes. Con frecuencia aparece la documentación incompleta para el caso del código y uso de las APIs, los ejemplos y la falta de información de los errores.

En cuanto a las propuestas de mejora en relación a la documentación de los frameworks, las opiniones de los encuestados va en la misma línea, indicando la necesidad de documentación de mejor calidad, más completa, clara y estructurada, incorporar y mejorar ejemplos, incorporar videos y comunidades organizadas y con estructura.

En varias preguntas abiertas, tanto en las consignas que requieren identificar problemáticas centrales y/o propuestas, siempre en relación a la usabilidad de framework, aparece la complejidad inherentes al diseño interno de framework y/o de sus componentes.

5 Directrices para mejorar la usabilidad de los frameworks web

A partir de los estudios desarrollados se identificaron diversos problemas que afectan la usabilidad de los frameworks web, asociados a la aprendibilidad, comprensibilidad, operabilidad y protección contra errores. Entre los problemas más comunes se encuentra la falta de documentación o documentación incompleta o desactualizada, como así también inadecuada para solucionar los errores planteados por los usuarios al utilizar los framework. De igual forma, se identificaron recursos y características de los mismos que resultan valiosos para el uso de estas herramientas.

Se propone un conjunto de directrices orientadas a los desarrolladores de frameworks, como también a los diseñadores de los sitios web oficiales, con el objeto de

mejorar la documentación y así facilitar el uso de los frameworks web en relación a su aprendizaje, como también obtener mejoras en el sitio oficial.

Se presentan las directrices clasificadas según los siguientes objetivos: A-Mejorar los sitios oficiales de los frameworks; B-Mejorar la interacción entre usuarios; C-Mejorar la documentación para reducir los errores. La Tabla 3 describe las directrices planteadas, indicando el propósito, la justificación y clasificación según su propósito.

Tabla 3. Directrices propuestas

C	Directriz	Propósito	Justificación
A	Adecuar los recursos de documentación.	Proveer información de relevancia para mejorar la aprendibilidad del framework y reducir el nivel de errores	El estudio de los 17 frameworks web arrojó que existen diversos recursos que permiten documentar el framework, entre los más utilizados se encuentran los Get started, Tutoriales, Ejemplos, Videos y principalmente la documentación que acompaña a las APIs.
A	Proveer recursos de documentación en formatos variados	Facilitar la aprendibilidad, operabilidad y reducir el nro. de errores que puedan ocurrir.	Como resultado de los estudios anteriores se puede determinar que la diversidad de recursos, en cuanto a su tipo y presentación, facilita al usuario del mismo su utilización y reduce el nro. de errores.
A, C	Incluir ejemplos.	Simplificar la utilización del framework, y reducir el nro. de errores que se puedan producir durante su uso.	Esto permite visualizar la funcionalidad del framework, mejorar su uso, reducir la complejidad y facilitar la instalación, configuración e implementación.
C	Documentar de forma correcta las APIs.	Facilitar el aprendizaje y la implementación de las APIs que provee el framework mejorando la aprendibilidad, operabilidad y favoreciendo la reducción del nro. de errores	Esto se evidencia al visualizar, en los estudios realizados, un alto porcentaje de consultas sobre funcionamiento o peticiones de solución de errores sobre estas herramientas que proporciona el framework. Al proporcionar un información detallada de las APIs se podrá reducir el nro. de errores detectados.
A	Extender la interacción mediante el uso de plataformas colaborativas.	Mejorar la aprendibilidad, operabilidad y comprensibilidad del framework y reducir los errores mediante el uso de plataformas colaborativas.	La plataforma colaborativa GitHub fue el recurso más identificado, según el estudio realizado a los sitios web oficiales y descriptos.
A	Acceder en forma directa, a los recursos disponibles en el sitio web.	Ingresar en forma rápida y mediante un solo click a los recursos disponibles.	El acceso a los recursos de manera fácil y directa permite a los usuarios disponer de la información rápidamente, facilitando la resolución de problemas en el uso y evacuando todas aquellas consultas que puedan surgir.

B, C	Estructurar los foros.	Proveer foros con estructura organizada para facilitar el acceso a la información y reducir el nro. de errores que se presentan al utilizar el framework.	Los errores representan uno de los mayores problemas de usabilidad, el presentar una estructura organizada en los foros permitiría mejorar la interacción con los desarrolladores y así reducir el nro. de errores.
B	Interaccionar con usuarios expertos.	Permitir acceder a usuarios expertos en el uso del framework que puedan interactuar con otros usuarios menos experimentados para facilitar la aprendibilidad, comprensibilidad, operabilidad y reducir los errores.	El análisis realizado de los foros permitió observar que existe un número importante de entradas generadas por usuarios novatos, principalmente errores de uso. El contar con acceso a una subcomunidad de usuarios expertos permite agilizar la evacuación de consultas.
C	Proveer documentación específica de errores.	Incorporar documentación que permita identificar los errores, causas del error, que lo provocó, acción en la que se presenta, y formas de resolverlo.	La documentación sobre errores es un recurso no hallado en el estudio descripto, sin embargo se debe tener en cuenta, ya que surge la necesidad de esta documentación en el análisis de los foros y de las opiniones vertidas por los desarrolladores usuarios de frameworks web, es por ello que requiere de un tratamiento específico.
A, C	Detallar en Get started	Contar con el recurso get started permite a los usuarios tener un conocimiento preliminar sobre el framework y su uso.	Es un recurso muy utilizado, porque provee una mirada compacta de cómo usar el framework, es por ello que debe contar mínimamente entre sus ítems con instalación, configuración, conexión a la base de datos y algunas aplicaciones básicas, tal como surge del estudio de los recursos.
A, C	Incluir en Tutoriales.	Proveer información completa de todos los componentes que forman parte del framework.	Este recurso debe contener toda la información necesaria para que los usuarios conozcan en profundidad al framework, cubriendo todas las características y componentes que lo conforman. Del relevamiento realizado sobre la documentación de los frameworks web seleccionados, se obtuvieron los contenidos mínimos que deben formar parte de este recurso.
A, C	Mantener la documentación actualizada	Contar con documentación actualizada y acorde a las versiones del framework.	Gran parte de los errores detectados en las entradas a los foros, están asociados a la falta de actualización de la documentación a las diferentes versiones del framework.

6 Conclusiones

Este trabajo permitió identificar los recursos de documentación más utilizados y las características más relevantes de cada uno de ellos, asociando los atributos de usabilidad. Mediante el estudio de los foros, se identificaron los aspectos que presentan mayor dificultad y los errores más comunes consultados por los usuarios de los frameworks web, categorizados y organizados de acuerdo a su importancia. Se recolectó información directa de los usuarios que utilizan frameworks web.

Se propuso un conjunto de directrices para mejorar la usabilidad de los frameworks web, ofrecer documentación de calidad con contenido y presentación necesaria para optimizar los sitios oficiales, mejorar la interacción con y entre usuarios y reducir los errores que se puedan presentar con el uso.

A la vez, la aplicación de las directrices permite acrecentar la usabilidad dado que las mismas presentan una relación directa con las características de usabilidad asociadas a los frameworks web: aprendibilidad, operabilidad, comprensibilidad y protección contra errores.

En relación a los trabajos futuros los mismos se orientan a profundizar el estudio realizado a los foros, concentrándose en los errores, con el fin de proponer documentación específica para los mismos. Además realizar estudios sobre la estructura interna de los frameworks web que permitan ampliar la identificación de problemas de usabilidad.

Referencias

1. Gutiérrez, J. J.: ¿Qué es un framework web? Escuela Técnica de Ingeniería Informática – Departamento de Lenguajes y Sistemas Informáticos, sitio web: http://www.lsi.us.es/~javierj/investigacion_ficheros/Framework.pdf (2018)
2. Taligent, P.: Building Object-Oriented Framework, Addison-Wesley (1994)
3. Butler, G. Denomme, P., Documenting frameworks. Department of Computer Science Concordia University Montreal, Quebec. Canadá. (2006)
4. Fayad, M., Schmidt, D. Object-oriented application frameworks. Communications of the ACM -October 1997/Vol. 40, Nro. 10, PP 32-38. (1997)
5. Bertoa, M.F., Vallecillo, A.: Medidas de Usabilidad de Componentes Software, IEEE, Latin America Transactions, Vol. 4, Nro. 2 (2006)
6. International Standard ISO/IEC 25010 – First editions (2011)
7. Alfonso, P.L.: Revisión de modelos para evaluar la calidad de productos Web, Experimentación en portales bancarios del NEA, Universidad Nacional de la Plata – Facultad de Informática (2012)
8. Constanzo, M. A., Casas, S.: Usability Evaluation of Web Support Frameworks, XLII Conferencia Latinoamericana de Informática. Simposio Latinoamericano de Ingeniería de Software. ISBN: 978-1-5090-1632-7 (USB) (2016)
9. Constanzo, M. A., Casas, S.: Usabilidad de Framework Web: identificación de problemas y propuesta de evaluación, XV Workshop de Ingeniería de Software. Congreso Argentino de Ciencias de la Computación 2018, Tandil. (2018)
10. Microsoft [https://msdn.microsoft.com/enus/library/windows/desktop/ff728829\(v=vs.85\)](https://msdn.microsoft.com/enus/library/windows/desktop/ff728829(v=vs.85)). Acceso: Junio 2018. (2017)

11. Perurena Cancio, L. Morázquez Bergues, M., Usabilidad de los sitios Web, los métodos y las técnicas para la evaluación. (Revista Cubana de Información en Ciencias de la Salud) PP 176-194. (2013)
12. Hussain, A., Kutar, M., Usability Metric Framework for Mobile Phone Application. PGNet, ISBN: 978-1-902560-22-9. (2009)
13. Enriquez, J. G., Casas, S., Usabilidad en Aplicaciones Móviles. Informes Científicos - Técnicos UNPA- ISSN: 1852 – 4516. Vol. 5, Nro. 2, PP. 25-47. (2014)
14. Minhaz, F., Zibran Farjana, Z., Eishita Chanchal, K., Factors Affecting the Usability of APIs, Department of Computer Science, University of Saskatchewan, Saskatoon, SK, Canada S7N 5C9 (2011)
15. Hou, D., Li, L., Obstacles in Using Frameworks and APIs: An Exploratory Study of Programmers'Newsgroup Discussions. Department of Electrical and Computer Engineering Clarkson University Potsdam, NY USA. (2011)
16. Uddin, G., Robillard, How, M. P., API Documentation Fails. McGill University, IEEE Software, Vol. 32, Issue 4, DOI: 10.1109/MS.2014.80. (2015)

Estudio de la Capacidad de Mantenimiento de las Aplicaciones Móviles Híbridas

Daniel Díaz¹, Susana Herrera¹, Federico Rosenzvaig¹

¹ Instituto de Investigación en Informática y Sistemas de Información, Universidad Nacional de Santiago del Estero, 1912 Av. Belgrano (S), Santiago del Estero, Argentina
{danieldiaz,sherrera}@unse.edu.ar, fedvaig@gmail.com

Abstract. El presente artículo muestra los resultados de una investigación aplicada sobre la capacidad de mantenimiento de las aplicaciones móviles híbridas multiplataforma. Se utilizó el *framework Ionic* para generar aplicaciones Android y iOS a partir de un entorno web basado en el núcleo de HTML5, CSS3 y Javascript. Para la evaluación, se desarrolló la aplicación *Mi día en la UNSE*, usada en el evento de difusión de la oferta académica de la Universidad Nacional de Santiago del Estero. El desarrollo se basó en la metodología Mobile-D. La capacidad de mantenimiento de las aplicaciones móviles generadas por Ionic fue evaluada en el marco de la norma ISO/IEC 25.000, usando la estrategia de medición y evaluación *Goal Oriented Context Aware Measurement and Evaluation (GOCAME)*. Los resultados obtenidos conducen a afirmar que el framework Ionic permite generar aplicaciones multiplataforma híbridas con alto nivel de mantenibilidad.

Keywords: Desarrollo híbrido de aplicaciones móviles, Ionic, Mantenimiento de aplicaciones móviles, Mobile-D.

1 Introducción

A pesar de sus ventajas, las aplicaciones móviles requieren ser desarrolladas atendiendo características especiales y limitadas de los dispositivos donde se ejecutan: sistema operativo (SO), capacidad del procesador, tamaño de pantalla, duración de la batería, conectividad, entre otros. Considerando el SO, las aplicaciones móviles requieren ser desarrolladas en forma separada para cada uno, lo cual implica un alto costo de desarrollo y mantenimiento, ya que involucra formar equipos de trabajo con competencias profesionales diferentes.

Sin embargo, han surgido alternativas que permiten que estas aplicaciones se desarrollen bajo el núcleo de las tecnologías web y se ejecuten de forma nativa en los diversos SO móviles, mediante un explorador web. A este tipo de aplicaciones se las denomina aplicaciones móviles híbridas [1]. Esto da la posibilidad de diseñar una única instancia, que sea adaptable a las diferentes plataformas del mercado, optimizando la eficiencia del desarrollo. Sin embargo, este tipo de desarrollo podría arrastrar problemas en el mantenimiento de las aplicaciones generadas. Existen varios

frameworks que permiten generar aplicaciones híbridas. En este artículo se muestran resultados de un estudio sobre la capacidad de mantenimiento (o mantenibilidad) de aplicaciones híbridas usando el *framework* Ionic.

Para llevar adelante la evaluación, se construyó una aplicación móvil híbrida denominada Mi día en la UNSE. La misma fue utilizada en el evento “La UNSE para todos”, en el cual la Universidad Nacional de Santiago del Estero presenta su oferta académica a alumnos de nivel secundario de toda la provincia. La aplicación posee información relacionada con la difusión de las carreras de todas las facultades, así como también brinda actividades de transmisión en vivo y de recreación (concurso fotográfico).

La calidad de un sistema puede ser vista desde un enfoque interno o externo. Desde el modelo de calidad interna, la calidad es evaluada durante la revisión del producto, considerando sus características, siendo la mantenibilidad una de las principales según el estándar ISO IEC 25000 [2].

Cabe considerar que el mantenimiento, representa más esfuerzo que cualquier otra actividad de la Ingeniería del software. Es la facilidad con la cual se puede corregir un programa si se encuentra un error, adaptarlo si su entorno cambia, o mejorarlo si el cliente desea un cambio en los requisitos.

Para evaluar la característica de mantenibilidad se tomó como referencias los pasos sugeridos por la estrategia GOCAME [3].

Este trabajo forma parte de un proyecto más amplio donde se estudian diversos aspectos de las aplicaciones móviles multiplataforma [4].

El artículo se estructura de la siguiente manera. En el apartado 2 se presentan los conceptos básicos que sustentan la investigación: aplicaciones móviles híbridas, calidad del software, mantenibilidad, estrategias para la evaluación de la mantenibilidad. En el apartado 3 se presenta el desarrollo de la aplicación multiplataforma usando la metodología Mobile-D y el *framework* Ionic. Posteriormente, en el apartado 4, se muestra la estrategia seguida para la evaluación de la mantenibilidad y los resultados obtenidos. Finalmente, en el apartado 5 se presentan las conclusiones alcanzadas.

2 Marcos referenciales

2.1 Aplicaciones móviles híbridas

Las aplicaciones móviles son aplicaciones informáticas diseñadas para ser ejecutadas en teléfonos inteligentes, tabletas y otros dispositivos móviles [5]. Se encuentran disponibles a través de plataformas de distribución, operadas por las compañías propietarias de los SO móviles como Android, iOS, BlackBerry OS, Windows Phone, entre otros. En la actualidad, el mercado de aplicaciones móviles está dominado por dos grandes empresas: Google y Apple, con sus SO móviles Android y iOS, respectivamente [6].

Las aplicaciones móviles son diferentes para cada SO. Son aplicaciones nativas que se desarrollan en el lenguaje nativo del propio dispositivo. Son dependientes del SO donde residen. Las aplicaciones no son portables, hay que desarrollar una por cada

SO. El lenguaje de programación para aplicaciones Android es Java. Mientras que las aplicaciones iOS se programan con Objective C y Swift.

En los últimos años, ha surgido como alternativa el desarrollo móvil multiplataforma, que trata de compartir la misma codificación para los diferentes SO. Entre sus ventajas sobresalen: menor tiempo y costo de desarrollo, permite acceso al hardware del dispositivo, disponibilidad de entornos potentes de desarrollo (Delphi, Visual Studio, etc.), utilización de tecnologías web (HTML5, Javascript y CSS). Sin embargo, el rendimiento de las aplicaciones y sus interfaces de usuario, pueden afectar la experiencia de usuario [5].

Según Lisandro Delía [1, 7] las aplicaciones móviles multiplataforma pueden ser de diferentes tipos:

- Aplicaciones web responsivas. Se ejecutan dentro de un navegador, no dependen de ninguna plataforma y su ejecución es rápida y fácil. Una desventaja con respecto al desarrollo nativo, además de limitar accesos a todos los recursos de un dispositivo, se hace necesario acceder a internet para su funcionamiento. Existen *frameworks* que son marcos de desarrollo que hacen más simple el desarrollo de aplicaciones web con diseño adaptable, entre éstos están CodeIgniter [8], Symphony [9,10] y Laravel [11].
- Aplicaciones híbridas. Usan tecnologías web pero se ejecutan en un “contenedor web (*webview*)”, es parte de una aplicación nativa, desde la cual se puede acceder a todos los recursos del dispositivo. Existen diversos *frameworks* que permiten es tipo de aplicación: PhoneGap [12], CocoomJS, Ionic [13], Sench Touch.
- Aplicaciones interpretadas. Usan un lenguaje base que traduce gran parte a código nativo, y el resto se interpreta cuando se ejecuta. Por un lado, esto es una ventaja, pero se depende en forma exclusiva de una plataforma.
- Aplicaciones generadas por compilación cruzada. Las aplicaciones se compilan en forma nativa para cada plataforma que se va a usar. Entre las plataformas de desarrollo se encuentran Xamarin [14], Embarcadero Delphi 10 Seattle, Ruby Motion.

2.2 Calidad del software y mantenibilidad

La norma ISO/IEC 25000 [2], conocida como SQuaRE (*System and Software Quality Requirements and Evaluation*), es una familia de normas que tiene por objetivo la creación de un marco de trabajo común para evaluar la calidad del software. Este estándar define tres vistas diferenciadas en el estudio de la calidad del producto:

- Vista interna: se ocupa de las propiedades del software como tamaño, complejidad o la conformidad con las normas de orientación a objetos.
- Vista externa: analiza el comportamiento del software en la producción y estudia sus atributos.
- Vista en uso: mide la productividad y efectividad del usuario final al utilizar el software.

La norma ISO/IEC 25010 contempla los modelos: modelo de calidad de producto y modelo de calidad en uso.

El Modelo de Calidad de Producto (vista interna/externa) categoriza las propiedades de la calidad del producto o sistema en ocho características:

funcionalidad, eficiencia, compatibilidad, usabilidad, confiabilidad, seguridad, **mantenibilidad** (o capacidad de mantenimiento) y portabilidad. Cada característica está compuesta por un conjunto de subcaracterísticas.

El Modelo de Calidad en Uso (vista en uso). Define cinco características referidas a las salidas de la interacción con el sistema: efectividad, eficiencia, satisfacción, libre de riesgos y contexto. Cada característica puede ser asignada a diferentes actividades de los *stakeholders*.

Este trabajo se enmarca en el Modelo de Calidad del Producto. Y, teniendo en cuenta el estándar, la mantenibilidad se define en función de las siguientes subcaracterísticas:

- Modularidad: Propiedad que permite subdividir una aplicación en partes más pequeñas (llamadas módulos), cada una de las cuales debe ser tan independiente como sea posible de la aplicación en sí y de las restantes partes.
- Reusabilidad: Es el grado en que un programa (o partes de este) se puede reusar en otras aplicaciones.
- Analizabilidad: Facilidad con la que se puede evaluar el impacto de un determinado cambio sobre el resto del software, diagnosticar las deficiencias o causas de fallos en el software, o identificar las partes a modificar.
- Cambiabilidad: Capacidad del producto que permite que sea modificado de forma efectiva y eficiente sin introducir defectos o degradar el desempeño.
- Capacidad de ser probado: Facilidad con la que se pueden establecer criterios de prueba para un sistema o componente y con la que se pueden llevar a cabo las pruebas para determinar si se cumplen dichos criterios.

3 Desarrollo de aplicación híbrida con Ionic

El análisis, diseño, codificación y prueba de la aplicación híbrida multiplataforma “Mi día en la UNSE” se hizo con Mobile-D [15]. Esta metodología se compone de diversas fases por las cuales pasa el producto a realizarse: exploración, inicialización, producción, estabilización y pruebas. Cada una de las fases posee un día de planificación y un día de entregas.

Una vez finalizada todas las fases, se obtuvo un prototipo de aplicación móvil distribuido para múltiples plataformas, en particular, Android, iOS y Windows Phone.

En la fase de exploración se definieron los requisitos iniciales, el alcance y los recursos a usar. Teniendo en cuenta las necesidades del Centro de Orientación de la UNSE, la aplicación debía ser usada en el evento “La UNSE abre sus puertas”, evento en el cual se muestra la oferta académica de la UNSE a los aspirantes de las escuelas secundarias. Las principales funcionalidades de la aplicación serían:

- Gestión de usuarios: Permitir el registro de usuarios para monitorear sus interacciones con el prototipo y para la generación de reportes estadísticos solicitados por el área del COEP.
- Difusión de carreras: utilizar la cámara del dispositivo para leer datos por medio de códigos QR y visualizar en pantalla su resultado. Este código posee información específica de los eventos que ocurren en la jornada de difusión de carreras.

- Concurso fotográfico: Subir una fotografía tomada con la cámara del dispositivo. El usuario que captó dicha fotografía participa de un sorteo por un premio sorpresa.
- Transmisión en vivo: Posibilidad de seguir el evento vía streaming por el canal de Youtube de UNSE Tevé.

En cuanto a los recursos tecnológicos necesarios, se definieron los siguientes:

- Entorno de desarrollo: Ionic *Framework*
- Lenguaje de programación: Java, TypeScript
- Tecnologías web: HTML5, SASS, ANGULAR JS, JS, PHP
- IDE de desarrollo: Visual Studio Code
- IDE de diseño y maquetado: Adobe Photoshop CS6
- Gestor de Base de Datos: Firebase
- Bibliotecas: Node JS v8.11.2, Apache Cordova, Java SE Development Kit 8, Android Studio 3.1.2, Xcode 7, SDK WP8, Microsoft Visual Studio 2015

En la fase de inicialización, se prepararon los recursos tecnológicos (instalación y configuración) y humanos. Además, se definió la arquitectura de la aplicación y se realizó la planificación de las fases y de sus iteraciones.

En la fase de Producción se llevó a cabo toda la implementación de la aplicación. Se definieron 7 historias de usuario: registro de usuarios, acceso de usuarios, escanear código QR, registro y captura de imagen, transmisión en directo, programación de actividades. A partir de ello, se confeccionaron las *Task Card* en las cuales se asignaron las tareas a los desarrolladores. Posteriormente, se diseñó el modelo de la Base de Datos y se lo implementó en Firebase. Luego, se diseñaron las pantallas de usuario. Finalmente, se llevaron a cabo la codificación y las pruebas de aceptación (a cargo de los desarrolladores). En las figuras 1 y 2 se muestran pantallas correspondientes a algunas de las funcionalidades de la aplicación.

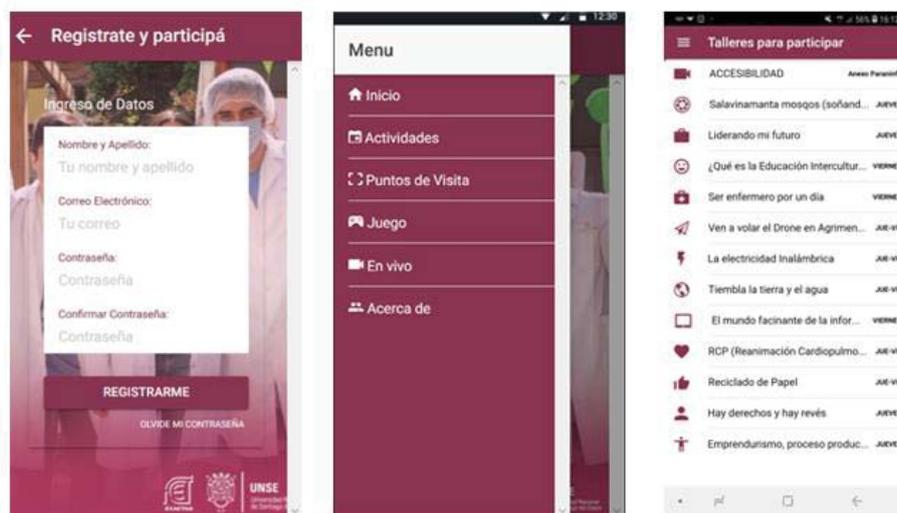


Figura 1. Principales pantallas de la aplicación Mi día en la UNSE: registro de usuarios, menú principal, actividades.

En la fase de estabilización, se llevaron a cabo las últimas acciones de integración para asegurar que el sistema completo funcione correctamente. Una vez integrado el sistema, se procedió a generar las aplicaciones nativas para los sistemas operativos móviles Android y iOS, mediante el empaquetador proporcionado por *Ionic Framework*. Este proceso es iterativo. Por cada modificación o integración que se realizara, se repitió el proceso de compilación y liberación de la aplicación nativa resultante.

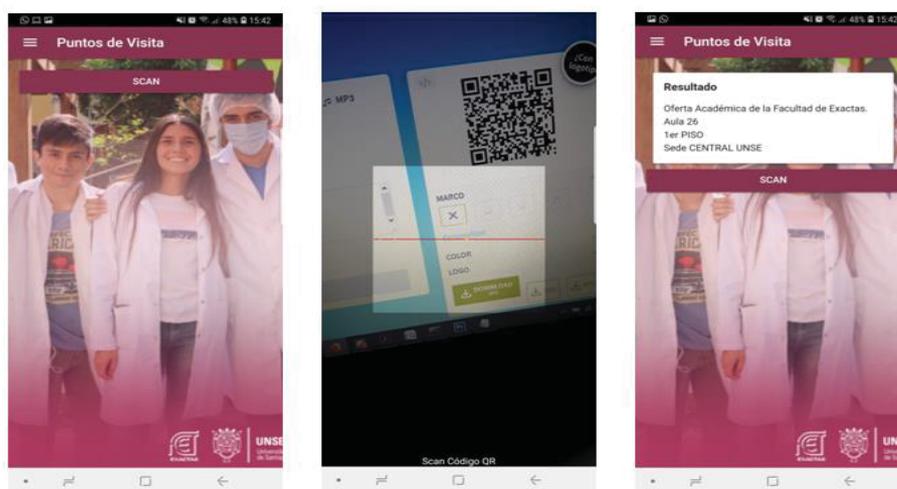


Figura 2. Pantallas de “Puntos de vista”. Los usuarios obtienen información leyendo un código QR.

Por último, en la fase de prueba, se validan las funcionalidades del producto terminado, considerando los requisitos. Se utilizó la técnica de observación directa [16] sobre una muestra aleatoria de usuarios.

4 Evaluación de la capacidad de mantenimiento

Como se mencionó en 2.2, la calidad de un sistema puede ser vista desde un enfoque interno o externo. Desde el modelo de calidad interna, la calidad es evaluada durante el desarrollo del producto, considerando sus características, siendo la mantenibilidad una de las principales.

Para efectuar la medición y evaluación de la mantenibilidad de las aplicaciones móviles generadas a partir de la utilización de *Ionic*, se seleccionó la estrategia GOCAME. Para la elección se tuvo en cuenta sus numerosos casos de aplicación, la base conceptual, la definición precisa del proceso de medición y evaluación y la solidez de sus métodos y herramientas. GOCAME es una estrategia para la medición y evaluación de proyectos de cualquier especie basada en propósitos y objetivos [3]. Fue elaborada sobre las bases del *Framework Contextual Information Need, Concept Model, Attribute, Metric, Indicator* (C-INCAMI) y utiliza la metodología *Web Quality*

Evaluation Method (WebQEM) para el proceso de construcción de métricas e indicadores y proceso de ejecución de mediciones. Sus principales etapas son: Definición de los requerimientos no funcionales, Diseño de la evaluación, Implementación de la medición y evaluación, Análisis y Recomendaciones.

4.1 Definición de requerimientos no funcionales

Como lo indica la estrategia, en esta etapa se definió el Árbol de Requerimientos. Partiendo de la característica de calidad que se desea evaluar, capacidad de mantenimiento, se definieron las sub-características y los atributos (Ver Tabla 1). Se utilizaron las subcaracterísticas especificadas en la norma ISO IEC 25000 para la mantenibilidad: analizabilidad, modificabilidad, capacidad de ser probado y reusabilidad.

Tabla 1. Árbol de requerimientos (etapa 1 de GOCAME).

Mantenibilidad
1.1 Analizabilidad
1.1.1 Complejidad ciclomática
1.1.2 Código repetido
1.1.3 Densidad Comentarios
1.2 Modificabilidad
1.2.1 Densidad de ciclos
1.2.2 Nombres de las variables
1.3 Capacidad de ser probado
1.3.1 Densidad de pruebas
1.4 Reusabilidad
1.4.1 Estructura del programa

Complejidad ciclomática: se basa en el análisis del diagrama de flujo determinado por las estructuras de control de un determinado código. Cuanto más compleja sea la lógica del código, más difícil será de entender, mantener y probar.

Código Repetido: mide la relación entre la cantidad de código repetido de un producto y su tamaño (sin contar comentarios). Cuando un código tiene errores y este es repetido, también se duplican los errores.

Densidad de comentarios: mide la relación entre la cantidad de comentarios (documentación) de un producto y su tamaño (incluyendo comentarios). Cuanto mayor sea la densidad de comentarios, más fácil de mantener será el software examinado.

Densidad de ciclos: mide la relación entre el número de dependencias cíclicas que presenta el producto y la cantidad de módulos existentes.

Nombre de variables: los nombres de variables, funciones, constantes y procedimientos son omnipresentes deberían ser nombres significativos que tengan lógica y sean explicativos.

Densidad de pruebas: mide la relación entre el número de pruebas unitarias y la complejidad ciclomática, es decir, el número de pruebas unitarias codificadas en relación con el número de pruebas unitarias deseable.

Estructura del programa: el código fuente de una aplicación debe estar escrito de tal manera que un componente pueda leerse fácilmente desde lo general a lo particular. Esto reduce el tiempo de comprensión por parte de las personas que lean el código fuente y por ende resulta ser más mantenible.

4.2 Diseño de la medición

La medición del software se logra aplicando métricas. El diseño de las mediciones se llevó a cabo considerando: la definición de requisitos no funcionales (el árbol) las métricas existentes: la complejidad ciclomática, el total de líneas de un programa (LOC) y la densidad de comentarios. A partir de la definición de ésta última, se determinaron las restantes medidas, las cuales siguen una especificación similar.

4.3 Diseño de la evaluación

En esta etapa se definieron los indicadores para cada uno de los atributos y los valores considerados para evaluar el resultado (Ver Tabla 2).

Luego se diseñaron los indicadores globales para cada una de las subcaracterísticas, para ello se utilizó la sumatoria de los indicadores multiplicada por su peso. Ver Formula 1.

$$IG = \sum_{i=1}^n p_i * IE_i \quad (1)$$

Donde,

IG es cada indicador global.

n es la cantidad de indicadores elementales de una subcaracterística.

pi es el peso de cada atributo.

IEi es cada indicador elemental.

Se asignaron “pesos” a cada uno de los atributos de acuerdo a la relevancia que tiene cada uno de ellos, teniendo en cuenta la calidad del producto y la experiencia de desarrolladores. Sumados todos los de una subcaracterística, dan 1 (ver Tabla 3).

Luego, se definieron los indicadores globales (IG) y los umbrales para establecer un criterio de decisión. Dada la fórmula presentada anteriormente, el indicador global siempre puede tomar un valor perteneciente al rango real [1-100] y no se requiere mapeo alguno. En la Tabla 4 se presentan los indicadores globales.

IG para la característica de mantenimiento se define de la siguiente manera:

$NCM = 0,20 * NA + 0,50 * NM + 0,05 * NCP + 0,25 * NR$
--

Siendo los criterios de decisión los siguientes:

10 ≤ NCM ≤ 40

Revisar código

40 < NCM ≤ 50

Marginal

NCM > 50

Nivel de mantenimiento satisfactorio

Tabla 2. Indicadores y sus valores de decisión.

Indicador	Criterios de toma de decisión
CC = E-N +2 CC: Complejidad ciclomática E: número de aristas N: número de nodos	Criterios tabulados (McCabe,T., 1976) 1-10 Programa simple, sin mucho riesgo 11-20 Más complejo, riesgo moderado 21-50 Complejo, Programa de alto riesgo >=50 Programa con muy alto riesgo
DCR= (1-(LC/LT)) *100 DCR: Densidad de código repetido LC: Líneas de código duplicadas LT: Líneas totales de código – líneas comentadas	2<=DCR<=20 Insatisfactorio. Programa con mucho código repetido. 20<DCR<=50 Moderado, revisar código DCR>50 Satisfactorio.
DCOM= (LC/LOC) *100 DCOM: Densidad de comentarios LC: Líneas comentadas LOC: Líneas totales de código	10<=DCOM<20 Programa con pocos comentarios 21<=DCOM<=35 Programa con comentarios moderado DCOM>35 Programa correctamente documentado
CYCD= (1-(CEM/M))*100 CYCD: Densidad de ciclos CEM: Ciclos entre módulos M: Total de módulos	10<=CYCD<=25 Programa con muchos ciclos 26<CYCD<=50 Programa con ciclos moderado CYCD>50 Programa con pocos ciclos.
DNCS= NVCS/NVT *100 DNCS: Densidad de nombres con sentido NVCS: Número de variables con sentido NVT: Número de variables totales	15<=DNCS<50 Variables con poco sentido 51<=DNCS<=70 Variables con sentido moderado DNCS>70 Nombres correctos con sentido
UTD= (1-((UN/CC)/100))*100 UTD: Densidad de pruebas unitarias UN: Número de pruebas unitarias CC: Complejidad ciclomática	1<=UTD <=25 Insatisfactorio 26<UTD<50 Caminos de prueba moderado UTD>=50 Fácil de ser probado
EP=NMBE/M*100 EP: Estructura del programa NMBE: Número de módulos bien escritos M: Cantidad de módulos	15<=EP<=25 Programa con dificultad de entender 26<EP<=50 Programa moderado EP>50 Programa bien estructurado

4.4 Implementación de la medición y evaluación

Los proyectos desarrollados bajo el entorno Ionic generan una gran cantidad de archivos y directorios. El proyecto de la aplicación “Mi día en la UNSE” tiene el siguiente tamaño: 9 descriptores de acceso (número de funciones “get” y “set”), 9 clases, 4117 directorios, 35851 archivos, 537 líneas de código físicas, 16 métodos.

Los archivos que se tomaron para realizar la medición fueron los que contienen la mayor parte del código del proyecto: Registro.ts (funcionalidades relacionadas con el registro de usuarios), Login.ts (permite el acceso o denegación de usuarios en la

aplicación móvil), List.ts (despliega la lista de actividades), Reset.ts (implementa los mecanismos para la recuperación de contraseñas).

Tabla 3. Pesos asignados a los indicadores de los atributos.

Subcaracterística	Atributos	Peso de atributos	Peso de Subcaracterística
Analizabilidad	Complejidad ciclomática	0,30	0,20
	Código repetido	0,40	
	Densidad de comentarios	0,30	
Modificabilidad	Densidad de ciclos	0,25	0,50
	Densidad de nombres con sentido	0,75	
Capacidad de ser probado	Densidad de pruebas	1	0,05
Reusabilidad	Estructura del programa	1	0,25

Tabla 4. Indicadores Globales y sus valores de decisión.

Indicador Global	Criterios de toma de decisión	
Nivel de Analizabilidad (NA) $NA=0,30 * CC + 0,40 * DCR + 0,30 * DCOM$	$1 \leq NAM \leq 20$	No es analizable
	$20 < NAM \leq 50$	Analizable moderado
	$NAM > 50$	Satisfactorio
Nivel de Modificabilidad (NM) $NM=0,25 * DYCD + 0,75 * DNCS$	$15 \leq NM \leq 40$	Insatisfactorio
	$41 < NM \leq 70$	Moderado
	$NM > 70$	Satisfactorio
Nivel de Capacidad de ser probado (NCP) $NCP=1 * UPD$	$1 \leq NCP \leq 20$	Difícil de probar
	$20 < NCP \leq 50$	Moderado
	$NCP > 50$	Programa fácil de probar
Nivel de Reusabilidad (NR) $NR=1 * EP$	$15 \leq NM \leq 40$	Insatisfactorio
	$41 < NM \leq 70$	Moderado
	$NM > 70$	Satisfactorio

En la Tabla 5 se presentan los valores obtenidos a partir de la medición de los atributos de la capacidad de mantenimiento, teniendo en cuenta el conjunto de archivos seleccionados del proyecto Ionic. Estos valores están mapeados en un rango de 1 a 100 y expresados en valor de porcentaje (%), ya que se aplica una función de proporción a cada uno de ellos, a excepción de la complejidad ciclomática, cuyos valores ya están predeterminados, según la literatura. En la misma tabla se muestra, además, los resultados de la evaluación de los indicadores globales y también el valor final obtenido para la Capacidad de Mantenimiento. En todos los casos se indica con verde cuando los indicadores adoptan un valor positivo, con amarillo cuando adoptan un valor marginal (intermedio). En ningún caso se obtuvieron resultados negativos.

El valor de la Capacidad de Mantenimiento obtenido es 87,84.

Tabla 5. Mediciones obtenidas y evaluación.

Árbol de Requerimientos	Archivos del Proyecto IONIC			
	Registro.ts	Login.ts	List.ts	Reset.ts
Mantenibilidad	87,84			
Analizabilidad	44,87			
Complejidad ciclomática (CC)	4			
Densidad de Código Repetido (DCR)	92,92			
Densidad de Comentarios (DCOM)	21,53			
Modificabilidad	97,91			
Densidad de Ciclos (CYCD)	91,66			
Nombres de Variables con Sentido (DNCS)	100			
Capacidad de ser probado	98,5			
Densidad de Pruebas (UTD)	98,5			
Reusabilidad	100			
Estructura del Programa (EP)	100			

5 Conclusiones

A partir de la aplicación de la estrategia GOCAME, se pudo medir y evaluar la mantenibilidad de la aplicación multiplataforma desarrollada con Ionic. Para ello, se ha invertido un esfuerzo considerable en la definición de las métricas e indicadores propuestos y en el proceso de medición manual, ya que no se utilizaron herramientas automatizadas para el análisis estático del código fuente. Como se mencionó anteriormente, la medición y evaluación se realizó en base a un grupo de cuatro archivos que contienen la mayor parte del código del proyecto Ionic.

Como se mencionó en el marco teórico, el desarrollo híbrido de las aplicaciones permite, utilizando un único *framework* (en este caso Ionic), generar aplicaciones móviles para diversos SO móviles. Durante el proceso de desarrollo, se utilizó el mismo código fuente para la liberación de aplicaciones compatibles con las plataformas Android y iOS. Al compilar la aplicación para cada plataforma, los componentes de Ionic se adaptaron estéticamente a las reglas de cada una. Al finalizar la compilación se obtuvieron los ejecutables. Esto se realizó cada vez que se lanzaba una nueva versión.

De acuerdo con las mediciones efectuadas, la capacidad de mantenimiento de la aplicación multiplataforma “Mi Día en la UNSE” fue evaluada usando el indicador global NCM, igual a 87,84%. Este valor indica que la capacidad de mantenimiento es satisfactoria, ya que supera el valor establecido como objetivo, 50% (apartado 4.3).

Teniendo en cuenta que el valor del NCM representa la medición de un único desarrollo individual (por cada plataforma), el coste y tiempo de desarrollo implicaría

el aumento del esfuerzo de mantenimiento dos veces más, ya que se debería de disponer de los recursos necesarios para programar en entornos específicos de cada plataforma. Por lo tanto, si se requiere que el proceso de mantenimiento sea corto y menos costoso, entonces lo mejor es desarrollar una aplicación multiplataforma híbrida.

En síntesis, si bien este estudio recae sobre el desarrollo de una única aplicación (“Mi día en la UNSE”), el resultado obtenido conduce a la siguiente afirmación: **el desarrollo híbrido usando Ionic permite obtener aplicaciones móviles multiplataforma con capacidad de mantenimiento satisfactorio.**

Asimismo, se observaron algunas desventajas en cuanto el uso del *framework* Ionic, entre las que se destaca la dificultad para trabajar con bibliotecas nativas que requieren configuraciones diferentes para cada SO móvil. En este caso, como solución se optó por dejar de lado bibliotecas de terceros que no ofrezcan compatibilidad con los SO móviles trabajados en este proyecto. Se utilizaron bibliotecas estándares del repositorio de Ionic.

Referencias

1. Delía, L. Trabajo final de especialista en Ingeniería Web: *Desarrollo de aplicaciones móviles multiplataforma*. Facultad de Informática, Universidad Nacional de la Plata. 2017.
2. International Standards Organization. ISO 25000. <https://iso25000.com/index.php/normas-iso-25000> [Consultado 20/08/2018].
3. Becker P., Lew P., Olsina L.: *Strategy to Improve Quality for Software Applications: A Process View*. 2011.
4. Herrera, S., Fennema, C., Morales, M., Maldonado, M., Palavecino, R., Rosenzvaig, F., Macedo, A., Suárez, C., Pellicer, R., Villavicencio, R. *Sistemas móviles multiplataforma con realidad aumentada. Aplicaciones en educación y salud*. XXI Workshop de Investigadores en Ciencias de la Computación. ISBN: 978-987-3619-27-4. San Juan. Disponible en: <http://sedici.unlp.edu.ar/handle/10915/77150>. 2019.
5. Santiago, R. et al. *Mobile learning: nuevas realidades en el aula*. Grupo Océano. pp. 8-26-27, 22-29. ISBN 9788449451454. 2015.
6. Statcounter, Globalstats, <http://gs.statcounter.com/os-market-share/mobile/worldwide> (2019)
7. Delía, L. N., L. Galdamez, P. Corbalán, P. Pesado and P.Thomas. *Approaches to Mobile Application Development: Comparative Performance Analysis*. Computing Conference 2018 18-20 July 2018 | London, UK. 2018.
8. EllisLab, CodeIgniter, <https://codeigniter.com>. 2002.
9. Symfony SAS, Symfony, <https://symfony.com>. 2019.
10. Hernández, M. T. *Symfony Framework. Desarrollo Rápido de Aplicaciones Web*. IT Campus Academy. 2015.
11. Taylor Otwell, Laravel, <https://laravel.com>. 2019.
12. PhoneGap. <https://www.phonegap.com> [Consultado 01/07/2018]. 2018.
13. Sperry, B. Lynch, M. Ionic Framework. <https://ionicframework.com/docs/v2/getting-started/concepts>. 2012.
14. Xamarin. (2018). Sitio Web: <https://www.xamarin.com>. [Consultado 11/08/2018].
15. Abrahamsson, P., Hanhineva, A., Hulkko, H., Ihme, T., Jääliñoja, J., Korkala, M., Koskela, J., Kyllönen, P., Salo, O. *Mobile D: An Agile Approach for Mobile Application Development*. OOPSLA 2004, Canadá. Online: <http://agile.vtt.fi/mobiled.html>. 2004.
16. Hernández Sampieri, R. *Metodología de la Investigación*. México. Mc Graw-Hill. Quinta edición. ISBN: 978-607-15-0291-9. 2010.

Evaluación de herramientas libres para la gestión de requerimientos

Sonia R. Santana¹, Lucrecia R. Perero¹, Amalia G. Delduca¹,
Gladys N. Dapozo²,

⁽¹⁾ Facultad de Ciencias de la Administración - Universidad Nacional de Entre Ríos

⁽²⁾ Facultad de Ciencias Exactas y Naturales y Agrimensura - Universidad Nacional del Nordeste

sr.santana675, rmlperero, amadel.agd [at]gmail.com]

Abstract. La gestión de los requerimientos es una tarea fundamental en cualquier proyecto de Ingeniería de Software. Debe ser un proceso continuo en el ciclo de vida del desarrollo del sistema. Los requerimientos son volátiles debido a los constantes cambios en las necesidades, procesos y tecnologías, esto hace que la gestión de los requerimientos en forma manual sea una tarea difícil. Existen herramientas software que facilitan la gestión de requerimientos. En este trabajo se abordará el estudio y profundización de herramientas de software libre orientadas a la gestión de requerimientos. El software libre presenta características y modos de usos que lo hacen accesible y apropiado para esta tarea. Se realizará un análisis comparativo de las herramientas para determinar la performance de su aplicación en base a determinados criterios de evaluación establecidos previamente.

Keywords: Ingeniería de requerimientos, validación de requerimientos, herramientas para la ingeniería de requerimientos, software libre.

1 Introducción

El desarrollo de software no es solo un desafío tecnológico sino también un proceso social complejo. En este proceso, la comunicación efectiva entre los interesados desempeña un papel vital en el desarrollo del proyecto de software que conduce a recopilar requerimientos de manera correcta y consistente.

Los errores de requerimientos tales como inconsistencia, incompletitud e incorrección pueden llevar al retrabajo extenso y fallas irre recuperables [1]. Además, la detección de estos errores en una etapa posterior de un proyecto de desarrollo de software es más difícil, lento y costoso [2] [3].

Es importante lograr que los errores se detecten, se corrijan y minimicen lo antes posible. La prevención de errores es una cuestión de buenas prácticas en la Ingeniería de Software, sin embargo, es prudente suponer que se producirán errores. Afortunadamente, este tipo de errores innecesarios pueden evitarse con una adecuada validación de requerimientos.

El principal objetivo de la validación de los requerimientos es certificar que los requerimientos elicitados son representaciones exactas de las necesidades y

expectativas de los usuarios [4] [5] [6] y deben ser completos, correctos y consistentes [7].

Esta actividad clave puede ayudar a identificar, prevenir los defectos y errores, mejorando la calidad de los requerimientos, reduciendo el tiempo de desarrollo, el costo y los riesgos para el desarrollo del software de alta calidad que cumpla con las expectativas de los usuarios [2] [8].

La validación de requerimientos debe considerarse más como una actividad de "debugging" en lugar de probar "lo correcto". En efecto, es una actividad "siempre presente" que tiene lugar cada vez que se solicita que un requerimiento sea adquirido, analizado o integrado con el resto del modelo de requerimientos [9].

La gestión de requerimientos es fundamental para el éxito de entrega de software y ciclo de vida del proyecto. Proporciona asistencia para muchas actividades de ingeniería de software como análisis de impacto, análisis de responsabilidad, validación de requerimientos y pruebas de repercusión.

Se define gestión de requerimientos como: *“Proceso de administrar los cambios a los requerimientos para asegurarse de que los cambios realizados se analizan adecuadamente y se rastrean a lo largo del sistema”* [6].

Las herramientas de Ingeniería de Requerimientos (ER por sus siglas en inglés Requirements Engineering) forman parte integral de la mayoría de las soluciones de gestión de requerimientos. Son necesarias por varias razones, permiten: administrar un número creciente de información, mantener la trazabilidad entre los requerimientos, la solución y las pruebas y evaluar el impacto de un cambio de requerimientos en la solución y las pruebas.

Las herramientas de ER se utilizan cada vez más para facilitar los procesos de ER y permitir un manejo más sistemático y formal de los requerimientos, la gestión de cambios y la trazabilidad.

El proyecto PID 7057 denominado Evaluación de Herramientas Libres para la Validación de Requerimientos se divide en dos etapas, la primera esta presentada en este trabajo donde se evaluaron diversas herramientas libres de RE para la gestión de requerimientos comparando evidencia empírica, capacidades y funcionalidades de las mismas.

La próxima etapa del PID, la cual se encuentra en desarrollo, tiene como objetivo definir una metodología para la validación de requerimientos en las herramientas libres donde se evaluarán las funcionalidades deseables y el comportamiento de los requerimientos frente a las técnicas para la validación. Esta comparación permitirá responder cuestiones como puntos fuertes y débiles, aseguramiento de la trazabilidad y disponibilidad de datos de las herramientas libres en la validación de requerimientos.

2 Metodología

La metodología adoptada para este trabajo contempla tres etapas:

En la primera etapa se realiza una revisión de documentación en diferentes fuentes bibliográficas con el objetivo de detectar la existencia de herramientas libres relacionadas con la gestión de requerimientos que acompañan el desarrollo de un proyecto de software. Luego, se clasifican las herramientas y se describen sus

principales características, funcionalidades y limitaciones, identificando las principales entradas de información que requieren para su funcionamiento.

Analizadas las herramientas, en la segunda etapa se identifican funcionalidades de las mismas según distintas fuentes de datos que permitan asegurar la evaluación de las herramientas.

Finalmente, como última etapa, se evalúan según las funcionalidades varias de las herramientas identificadas para conocer su funcionamiento, necesidades de información y restricciones.

2.1 Caracterización de las herramientas

Para realizar la revisión de documentación se comienza con la pregunta ¿Cuáles son las herramientas libres disponibles en la literatura para gestionar requerimientos, qué funcionalidades y alcance poseen? En función de lo señalado, se presenta en la Tabla 1 un resumen de la evaluación global de las herramientas libres de ER encontradas en la cual se indica con el vocablo SI que la herramienta contribuye a la funcionalidad, mientras que el vocablo NO indica que la herramienta no contribuye con la funcionalidad. El guion (-) representa que no se obtuvieron datos en la literatura o documentación oficial.

El: Elicitación, **An:** Análisis, **Es:** Especificación, **Vv:** Validación y Verificación, **Tr:** Trazabilidad, **Do:** Documentación, **Rg:** Representación gráfica, **Ih:** Integración de la herramienta

Tabla 1. Funcionalidades de las herramientas libres de ER

Herramienta	El	An	Es	Vv	Tr	Do	Rg	Ih
aNimble [10]	SI	SI	SI	SI	SI	SI	NO	-
Controla [11]	SI	NO						
Heler [12]	SI	NO	SI	SI	SI	SI	NO	-
Jeremia [13]	SI	SI	SI	SI	SI	SI	NO	-
OpenOME [14]	NO	SI	NO	SI	NO	SI	SI	SI
OSRMT [15]	SI	SI	SI	SI	SI	SI	NO	-
Rambutan [16]	SI	SI	SI	NO	NO	SI	NO	NO
REM [17]	SI	SI	SI	SI	NO	SI	SI	NO
Remas[18]	SI	SI	SI	SI	SI	SI	NO	-
Rmtoo [19]	SI	NO	SI	NO	NO	SI	SI	NO
ReqHeap [20]	SI	SI	NO	SI	NO	SI	NO	-
ReqMan [21]	SI	SI	NO	NO	NO	SI	NO	-
ReqT [22]	SI	SI	NO	NO	NO	NO	NO	NO
Retro [23]	NO	NO	NO	SI	SI	SI	NO	NO
TestMeReQ [24]	NO	NO	NO	SI	NO	SI	NO	-

La revisión realizada permitió determinar no solo las capacidades básicas de las herramientas, sino además aspectos como la trazabilidad, integración con otras plataformas, documentación y representaciones gráficas. Todos los aspectos relevantes para la investigación, ya que permiten dar idea de la completitud con la que un sistema de gestión de requerimientos debe operar.

2.2 Definición de funcionalidades de las herramientas

En este apartado se presentan las funcionalidades requeridas de las herramientas para asegurar la evaluación de las mismas, para ello se consideran dos fuentes de datos:

- Marco ISO / IEC TR 24766: 2009 “Guía para las capacidades de las herramientas de ingeniería de requerimientos” [25].
En [27] se ofrece una lista de capacidades requeridas para las herramientas de ER, las mismas están organizadas de acuerdo a las actividades y requerimientos de un sistema de software mencionados en la norma ISO / IEC 12207: 2008, ISO / IEC 15288: 2008 e ISO / IEC TR 19759 (SWEBOK).
- Información recopilada en la primera etapa, analizando cuales son las entradas solicitadas por las herramientas y las funcionalidades que debe brindar una herramienta en la gestión de requerimientos durante todo el ciclo de vida del desarrollo del sistema.

A continuación, en la Tabla 2 se listan las funcionalidades identificadas para evaluar las herramientas libres.

Tabla 2. Funcionalidades requeridas de las herramientas.

Identificador	Descripción	Identificador	Descripción
F000	Crear proyecto	F036	Describir requerimiento
F001	Eliminar proyecto	F037	Asignar requerimiento al usuario
F002	Modificar proyecto	F038	Priorizar requerimiento
F003	Consultar proyecto	F039	Asignar riesgo requerimiento
F004	Crear fecha-hora proyecto	F040	Complejizar requerimiento
F005	Asignar fecha-hora proyecto	F041	Asignar dependencias requerimiento
F006	Crear múltiples proyectos	F042	Asignar conflicto requerimiento
F007	Crear subproyecto	F043	Importar requerimiento
F008	Eliminar subproyecto	F044	Exportar requerimiento
F009	Modificar subproyecto	F045	Clasificar requerimientos
F010	Consultar subproyecto	F046	Comentar requerimientos
F011	Crear fecha-hora subproyecto	F047	Crear fecha-hora requerimiento
F012	Asignar fecha-hora subproyecto	F048	Asignar fecha-hora requerimiento
F013	Crear usuario	F049	Consultar reporte requerimiento
F014	Eliminar usuario	F050	Ayudar a redactar requerimiento
F015	Modificar usuario	F051	Escalar requerimientos
F016	Consultar usuario	F052	Versionar requerimientos
F017	Asignar usuario	F053	Generar especificación requerimientos
F018	Permitir permisos usuarios	F054	Administrar diagramas
F019	Crear stakeholder	F055	Administrar prototipos
F020	Eliminar stakeholder	F056	Administrar modelos
F021	Modificar stakeholder	F057	Revisión de requerimientos
F022	Consultar stakeholder	F058	Inspección de requerimientos
F023	Asignar stakeholder	F059	Realizar testcase
F024	Crear release	F060	Inspección testcase
F025	Eliminar release	F061	Realizar estadísticas
F026	Asignar release	F062	Trazabilidad de requerimientos
F027	Crear keyword	F063	Emitir reportes

F028	Eliminar keyword	F064	Representación gráfica
F029	Asignar keyword	F065	Importar diagramas/modelos
F030	Versionar proyecto	F066	Exportar diagramas/modelos
F031	Enviar mensajes	F067	Portabilidad
F032	Crear requerimiento	F068	Seguridad
F033	Eliminar requerimiento	F069	Integridad
F034	Modificar requerimiento	F070	Usabilidad
F035	Consultar requerimiento		

2.3 Evaluación de las herramientas

En esta etapa se describe la evaluación de varias herramientas libres. En la Tabla 3 se muestra las herramientas excluidas por diferentes problemas encontrados:

Tabla 3. Herramientas excluidas en la evaluación.

Herramienta	Problema
aNimble	No se logró su funcionamiento.
Heler y Controla	Enlace de descarga fuera de servicio.
Jeremía	No se logró su funcionamiento.
OpenOME	No fue calificada en la evaluación debido a que solo realiza análisis de modelos.
Rambutan	Error al trabajar en la herramienta.
Rmtoo	No fue calificada en la evaluación debido a que está destinada a los desarrolladores no posee interfaz gráfica de usuario.
ReqMan	No se logró su funcionamiento.
Retro y TestMeReq	Las herramientas no fueron calificadas en la evaluación debido a que solo realizan validación y verificación de requerimientos.

La Tabla 4 muestra la evaluación realizada a las herramientas **OSRMT**, **REM**, **Remas** y **ReqHeap** según las funcionalidades definidas en la Tabla 2 y el checklist de a Tabla 5.

Tabla 5. Categorías de cumplimiento de la condición evaluada en las herramientas.

Acrónimo	Significado	Descripción
SI	Adecuado	El ítem se cumple adecuadamente
NO	No adecuado	El ítem no se cumple o se cumple solo en parte
N/M	Necesita mejora	El ítem se cumple pero puede optimizarse
N/A	No acepta	El ítem no aplica a la herramienta.

Tabla 4. Evaluación de las herramientas.

Identificador	Descripción	OSRMT	REM	Remas	ReqHeap
F000	Crear proyecto	SI	SI	SI	SI
F001	Eliminar proyecto	NO	NO	SI	SI
F002	Modificar proyecto	NO	SI	SI	SI
F003	Consultar proyecto	SI	SI	SI	SI
F004	Crear fecha-hora proyecto	N/M	N/M	N/M	SI
F005	Asignar fecha-hora proyecto	N/M	N/M	N/M	SI

F006	Crear múltiples proyectos	SI	SI	N/A	SI
F007	Crear subproyecto	N/A	N/A	N/A	SI
F008	Eliminar subproyecto	N/A	N/A	N/A	SI
F009	Modificar subproyecto	N/A	N/A	N/A	SI
F010	Consultar subproyecto	N/A	N/A	N/A	SI
F011	Crear fecha-hora subproyecto	N/A	N/A	N/A	SI
F012	Asignar fecha-hora subproyecto	N/A	N/A	N/A	SI
F013	Crear usuario	SI	SI	N/M	SI
F014	Eliminar usuario	NO	NO	N/M	SI
F015	Modificar usuario	SI	SI	N/M	SI
F016	Consultar usuario	SI	SI	N/M	SI
F017	Asignar usuario	SI	SI	N/A	SI
F018	Permitir permisos usuarios	SI	N/A	N/A	SI
F019	Crear stakeholder	SI	SI	N/A	SI
F020	Eliminar stakeholder	N/A	SI	N/A	SI
F021	Modificar stakeholder	SI	SI	N/A	SI
F022	Consultar stakeholder	SI	SI	N/A	SI
F023	Asignar stakeholder	SI	SI	N/A	SI
F024	Crear release	N/A	SI	N/A	SI
F025	Eliminar release	N/A	N/A	N/A	SI
F026	Asignar release	N/A	SI	N/A	SI
F027	Crear keyword	N/A	SI	N/A	SI
F028	Eliminar keyword	N/A	SI	N/A	SI
F029	Asignar keyword	N/A	SI	N/A	SI
F030	Versionar proyecto	N/A	SI	N/A	SI
F031	Enviar mensajes	N/M	N/A	N/A	SI
F032	Crear requerimiento	SI	SI	SI	SI
F033	Eliminar requerimiento	SI	SI	SI	SI
F034	Modificar requerimiento	SI	SI	SI	SI
F035	Consultar requerimiento	SI	SI	SI	SI
F036	Describir requerimiento	SI	N/M	N/M	SI
F037	Asignar requerimiento al usuario	SI	SI	N/A	SI
F038	Priorizar requerimiento	SI	SI	N/A	SI
F039	Asignar riesgo requerimiento	N/A	SI	N/A	SI
F040	Complejizar requerimiento	SI	N/A	N/A	SI
F041	Asignar dependencias requerimiento	SI	SI	N/M	SI
F042	Asignar conflicto requerimiento	N/A	SI	N/A	SI
F043	Importar requerimiento	N/M	N/A	N/A	N/A
F044	Exportar requerimiento	N/M	N/A	N/A	N/A
F045	Clasificar requerimientos	N/A	SI	N/M	SI
F046	Comentar requerimientos	SI	SI	SI	SI
F047	Crear fecha-hora requerimiento	SI	N/M	N/M	SI
F048	Asignar fecha-hora requerimiento	SI	N/M	N/M	SI
F049	Consultar reporte requerimiento	SI	SI	SI	SI
F050	Ayudar a redactar requerimiento	N/A	N/A	N/A	SI
F051	Escalar requerimientos	N/A	SI	N/A	SI

F052	Versionar requerimientos	SI	SI	N/A	SI
F053	Generar especificación requerimientos	N/M	N/M	N/M	N/A
F054	Administrar diagramas	N/A	N/M	N/M	N/A
F055	Administrar prototipos	N/A	N/A	N/A	N/A
F056	Administrar modelos	N/A	N/A	N/A	N/A
F057	Revisión de requerimientos	N/M	N/M	N/M	SI
F058	Inspección de requerimientos	N/M	N/M	N/M	SI
F059	Realizar testcase	N/M	N/M	N/A	SI
F060	Inspección testcase	N/M	N/M	N/A	SI
F061	Realizar estadísticas	N/A	N/A	N/A	SI
F062	Trazabilidad de requerimientos	N/M	N/A	N/M	N/A
F063	Emitir reportes	N/M	N/M	N/M	N/M
F064	Representación gráfica	N/A	N/M	N/M	N/A
F065	Importar diagramas/modelos	N/A	N/A	N/A	N/A
F066	Exportar diagramas/modelos	N/A	N/A	N/A	N/A
F067	Portabilidad	N/A	N/A	N/A	SI
F068	Seguridad	N/M	NO	N/A	SI
F069	Integridad	N/M	N/M	N/M	N/M
F070	Usabilidad	N/M	N/M	N/M	N/M

3. Resultados

Del análisis comparativo realizado entre las herramientas seleccionadas surge cuanto cumplen los distintos procesos de la ingeniería de requerimientos, que se muestran en la Fig. 1

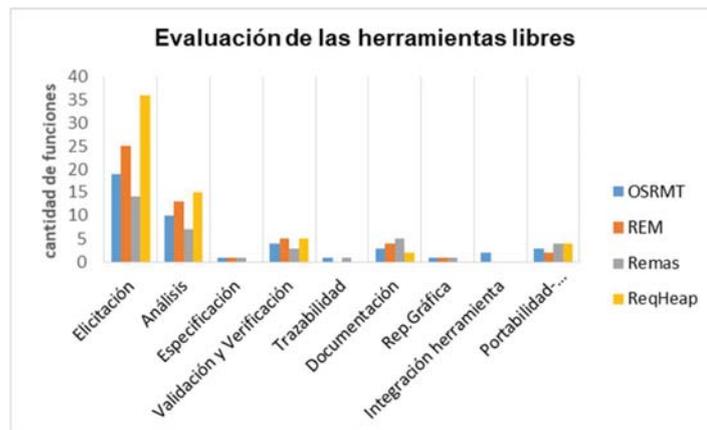


Fig.1. Aporte de cada herramienta en la gestión de requerimientos.

Según la Fig. 1 se extraen los siguientes resultados.

- La mayoría de las herramientas libres tienen funcionalidades importantes a la hora de elicitar los requerimientos. La herramienta ReqHeap presenta el mayor número de funcionalidades para la elicitación de los requerimientos.
- En el análisis de requerimientos la herramienta ReqHeap presenta el mayor número de funcionalidades. OSRMT y REM necesitan agregar nuevas funcionalidades y mejorar las existentes y Remas necesita incorporar nuevas funcionalidades.
- OSRMT, REM y Remas generan la especificación de requerimientos pero no cumplen con el estándar IEEE 830 y su exportación no se realiza en un formato estándar para que pueda ser accesible y revisable. ReqHeap no genera la especificación de requerimientos.
- La validación y verificación de requerimientos en las herramientas evaluadas presentan muy pocas funcionalidades y con necesidad de mejora. La herramienta ReqHeap y OSRMT realizan revisión e inspección de requerimientos y test case. ReqHeap confecciona estadísticas de requerimientos para mostrar el seguimiento de los cambios. Remas modela diagramas de casos de uso, pero ninguna de las herramientas implementa prototipos.
- La trazabilidad de los requerimientos, solamente es aplicada por las herramientas OSRMT y Remas a través de la confección de la matriz de trazabilidad, la misma presenta necesidad de mejora en ambas herramientas.
- Todas las herramientas generan muy poca documentación, necesitan mejorar incorporando documentos relacionado al modelado de los requerimientos.
- Las herramientas REM y Remas poseen representaciones gráficas a través del modelado de diagramas de casos de uso.
- Las herramientas no importan y exportan modelos desde y hacia herramientas de modelado, en consecuencia, no tienen integración con otras herramientas. Solo la herramienta OSRMT importa y exporta requerimientos.

Otras capacidades evaluadas de las herramientas libres:

- ReqHeap es la única herramienta portable de las evaluadas. OSRMT, REM y Remas funcionan en plataformas antiguas.
- ReqHeap y OSRMT implementan funcionalidades para la seguridad en cuanto al control de usuario.
- Las herramientas evaluadas mantienen la integridad de los datos, pero OSRMT, REM y Remas necesitan mejora debido a que funcionan en plataformas antiguas.
- Las herramientas evaluadas no son fáciles de usar debido a la falta de información suministrada al usuario es escasa en cuanto a la instalación y manejo de las mismas.



Fig.2. Grado de cumplimiento de las funcionalidades requeridas

Luego de haber realizado la caracterización de algunas de las herramientas libres en la gestión de requerimientos, en la Fig.2 se muestra el grado de cumplimiento de las funcionalidades requeridas para una completa gestión de requerimiento. Se puede observar que ninguna de las herramientas libres satisface totalmente las funcionalidades definidas en la Tabla 2, ReqHeap cumple el 32% de las funcionalidades, Rem el 26%, OSRMT el 23% y finalmente Remas con el 19%.

Los resultados indican que el desarrollo de herramientas libres para la gestión de requerimientos está más orientado a la elicitación y análisis de los requerimientos que al modelado de los mismos, esto puede ser debido a que las herramientas tradicionalmente son orientadas más hacia el lenguaje natural que al modelado.

En este trabajo se ha demostrado que el desarrollo de herramientas libres para la gestión de requerimientos es una tarea que aún tiene mucho por explorar, pues requiere optimizar y definir funcionalidades para el aseguramiento de la calidad de los requerimientos.

Referencias

1. J. Zhou: An Observer-Based Technique with Trace Links for Requirements Validation in Embedded Real-Time Systems, International Conference on Research in Engineering and Technology, no. 177, pp. 1–52 (2014).
2. D. Aceituna, H. Do and S.-W. Lee: SQ² E: An Approach to Requirements Validation with Scenario Question, Asia Pacific Software Engineering Conference, pp. 33–42 (2010).
3. L. Kof, R. Gacitua, M. Rouncefield, and P. Sawyer: Ontology and Model Alignment as a Means for Requirements Validation, IEEE Fourth International Conference on Semantic Computing, pp. 46–51 (2010).
4. P. A. Laplante: Requirements Engineering for Software and Systems, CRC Press (2019).
5. B. H. C. Cheng and J. M. Atlee: Current and Future Research Directions in Requirements Engineering, Design Requirements Engineering A Ten-Year Perspective, Lecture Notes in Business Information Processing, vol. 14, pp. 11–43 (2019).

6. G. Kotonya and I. Sommerville: Requirements Engineering: Processes and Techniques, Wiley (1998).
7. S. L. Pfleeger: Software Engineering – Theory and Practice, Prentice Hall (1998).
8. M. Kamalrudin, N. A. Moketar, J. Grundy, and J. Hosking: “Automatic Acceptance Test Case Generation From Essential Use Cases, in 13th International Conference on Intelligent Software Methodologies, Tools and Techniques, pp. 246–255 (2014).
9. Loucopoulos, P., Karakostas, V.: System Requirements Engineering, McGraw-Hill, London, ISBN 0-07-707843-8 (1995).
10. aNimble, Platform, <https://sourceforge.net/projects/nimble/>.
11. C. V. Fraga Filho, J. M. dos Reis: CONTROLA: Herramienta de apoyo al proceso de desarrollo de software en las pequeñas compañías, Revista Ingeniería Informática, Ed.12 (2006).
12. Callejas Cuervo, Mauro, Castillo Estupiñán, Luz Yadira, Fernández Álvarez, Ruby Mónica: HELER: una herramienta para la ingeniería de requisitos automatizada, ISSN 1900-3803 (2010).
13. Jeremia, Sistema de Gestión de Requerimientos, <http://jeremia.sourceforge.net/>.
14. Horkoff, Jennifer, Yijun Yu and Eric Kai-Hsiang Yu: OpenOME: An Open-source Goal and Agent-Oriented Model Drawing and Analysis Tool, iStar (2011).
15. OSRMT Open Source Requirements Management Tool, <https://sourceforge.net/projects/osrmt/>.
16. Rambutan, Requirements Management Tool for Busy System Analysts, <http://rambutan.sourceforge.net/>.
17. Durán Toro A.: Herramienta REM, Universidad De Sevilla, http://www.lsi.us.es/descargas/descarga_programas.php?id=3&lang=en.
18. J. Hayes, A. Dekhtyar, S. Sundaram, E. Holbrook, S. Vadlamudi, A. April: Requirements tracing on target (retro): improving software maintenance through traceability recovery, Innovations in Systems and Software Engineering, vol.3, no.3, pp.193–202 (2007).
19. Rmtoo, Requirements Management Tool, <http://rmtoo.florath.net/>.
20. ReqHeap, The Requirement Heap, <https://sourceforge.net/projects/reqheap/>.
21. ReqMan, Requirements Manager, <https://sourceforge.net/projects/reqman/>.
22. Regnell B.: reqT.org – Towards a Semi-Formal, Open and Scalable Requirements Modeling Tool, In: Doerr J., Opdahl A.L. (eds) Requirements Engineering: Foundation for Software Quality, vol 7830, Springer, Berlin, Heidelberg, (2003).
23. Hayes, J.H., Dekhtyar, A., Sundaram, S.K.: REquirements TRacing On target (RETRO): improving software maintenance through traceability recovery, et al. Innovations Syst Softw Eng, 3: 193 (2007).
24. N. A. Moketar, M. Kamalrudin, S. Sidek, and M. Robinson: TestMEReq: Automated Acceptance Testing Tool For Requirements Validation, in International Symposium on Research in Innovation and Sustainability, vol. 2014, pp. 15–16 (2014).
25. ISO/IEC TR 24766:2009: Information technology, Systems and software engineering, Guide for requirements engineering tool capabilities, <https://www.iso.org/standard/51041.html>.

Evaluación de la calidad en patrones de arquitecturas software para sistemas críticos ferroviarios: un enfoque basado en la mantenibilidad

Joaquín Acevedo Duprat¹, Emanuel Irrazábal¹,

¹ Facultad de Ciencias Exactas y Naturales y Agrimensura.
Universidad Nacional del Nordeste. Corrientes, Argentina.

{cruz95ad, emanuelirrazabal}@gmail.com

Resumen. El desarrollo de software crítico para sistemas ferroviarios necesita garantizar la seguridad a la vez que la mantenibilidad; un fallo puede ocasionar graves consecuencias para el entorno, especialmente pérdida de vidas humanas. En particular, en la normativa EN 50128 de software crítico ferroviario se especifican las buenas prácticas al construir la arquitectura del software. Pero los patrones de arquitectura que cumplen con las buenas prácticas antes mencionadas pueden ser complejos y, a su vez, disminuir aspectos relacionados con la calidad del producto software, especialmente la mantenibilidad. En este trabajo se seleccionaron un conjunto de patrones arquitecturales, se realizaron pruebas de concepto para cada arquitectura evaluándose los resultados en cuanto a complejidad y características del código fuente resultante. De esta manera se pudo obtener un resultado comparable entre los diferentes patrones: el nivel de seguridad asociado, la complejidad de su construcción y métricas relacionadas con la mantenibilidad. Los resultados muestran que existen patrones con baja complejidad y buena mantenibilidad que pueden ser utilizadas en la construcción de productos software según la norma EN 50128.

Palabras clave: EN 50128, mantenibilidad, arquitectura, sistemas críticos.

1 Introducción

Los sistemas críticos son aquellos compuestos por software y hardware dedicado en los cuales un fallo o mal funcionamiento puede ocasionar graves consecuencias para el entorno, especialmente pérdidas materiales, medioambientales o de vidas humanas [1]. Por esta razón dichos sistemas tienen una serie de propiedades para garantizar un comportamiento esperado con niveles de seguridad, fiabilidad, mantenibilidad y disponibilidad altos [2].

En particular, los sistemas ferroviarios son complejos, compuestos por distintos componentes software, hardware y humanos, que interactúan con su entorno de maneras muy variadas. Un fallo en uno de estos componentes o subsistemas puede llegar a tener asociados distintos niveles de peligros, pudiendo causar pérdidas financieras, daño al equipamiento, daños ambientales, lesiones a personas o en los

peores casos pérdidas de vidas humanas [3]. Por estos motivos dichos sistemas se encuentran regulados con normativas cuyo fin es preservar los recursos anteriormente mencionados.

Una normativa importante para el desarrollo de software crítico ferroviario es la norma EN 50128 [4]. En la misma se trata la calidad del software en términos del ciclo de vida del desarrollo, especificando los procedimientos y requisitos técnicos. Abarca los aspectos de seguridad en cinco niveles desde el 0 al 4, de acuerdo a la gravedad de las consecuencias del fallo, siendo el nivel 4 el que tiene asociado una mayor cantidad de técnicas y características a cumplir. A estos niveles de seguridad se los conoce con la sigla en inglés SSIL derivada de la expresión *software safety integrity level*.

En particular, en el apartado 7.4 de la norma EN 50128 se especifican las buenas prácticas al construir la arquitectura del software en términos de actividades, documentación, especificación integral de cada módulo de la arquitectura y uso de buenas prácticas de programación.

En este sentido, los patrones de arquitectura que cumplen con las buenas prácticas antes mencionadas pueden ser complejos [5] y, a su vez, disminuir aspectos relacionados con la calidad del producto software, especialmente la mantenibilidad [6].

En este artículo se ha llevado adelante una revisión de la bibliografía para seleccionar las arquitecturas software habituales al desarrollar sistemas críticos. Se diseñó una prueba de concepto para cada arquitectura y se evaluaron los resultados en cuanto a complejidad y características del código fuente resultante. De esta manera se pudo obtener un resultado comparable entre los diferentes patrones de arquitectura para software crítico, el nivel de seguridad asociado, la complejidad de su construcción y el grado de mantenibilidad resultante.

El trabajo está organizado de la siguiente manera: además de esta introducción la sección 2 se refiere a la selección de las herramientas y las arquitecturas software escogidas. En la sección 3 se detalla el diseño del experimento llevado adelante para evaluar la construcción de ejemplos con las arquitecturas seleccionadas. En la sección 4 se presentan los resultados de cada desarrollo y el análisis estático del código fuente. Finalmente, en la sección 5 se describen las conclusiones y los trabajos futuros.

2 Selección de las arquitecturas y las herramientas de análisis

A continuación se presentan otros trabajos relevantes relacionados con el desarrollo de arquitecturas software para sistemas críticos ferroviarios que han sido adecuadamente comprobados. En la tabla 1 se identifican los patrones tratados en cada publicación.

Las arquitecturas encontradas fueron las siguientes: programación de N versiones (PNV), aceptación por votos con redundancia de hardware (AV), bloque de recuperación (BR), maestro-esclavo (ME) y redundancia de hardware (RH).

Tabla 1. Selección de patrones de arquitectura software utilizados en sistemas críticos.

Publicación / patrones	PNV	AV	BR	ME	RH
[7]	X	X	X		
[8]			X		X
[9]			X	X	
[10]			X		
[11]			X	X	
[12]			X		
[13]	X		X		
[14]	X	X	X		X

En la arquitectura programación de N versiones se trabaja sobre versiones equivalentes relacionadas a un mismo módulo funcional que estarán desarrolladas por distintos grupos. Cada versión envía la salida a un módulo votador que procesa las entradas y selecciona el valor final de salida (ver Fig. 1).

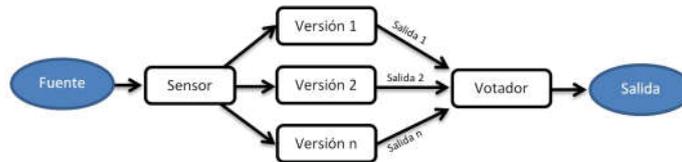


Fig. 1. Programación de N versiones.

En la arquitectura de aceptación por votos con redundancia de hardware se agrega un módulo de test en la salida de cada sensor para identificar fallos en las mediciones o enviar el valor aceptado al votador, dicho módulo es dinámico ya que puede tener diferente cantidad de entradas.

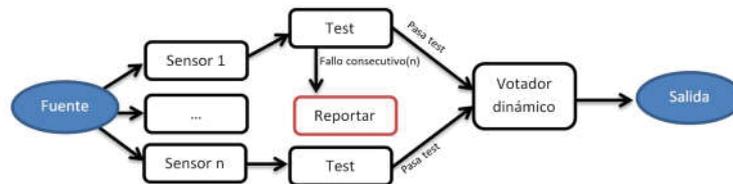


Fig. 2. Arquitectura de aceptación por votos con redundancia de hardware.

En la arquitectura de bloque de recuperación se establecen tres instancias. Una primera instancia define el test de aceptación, el mismo sirve de parámetro para

evaluar si una entrada es correcta o no. En una segunda instancia, se captura y almacena una medición, esta última cumple el rol de punto de verificación o *checkpoint*. Esta segunda medición capturada se procesa con la primera versión del programa y se evalúa con el test. En caso de que sea aceptada, la medición se envía a la salida. En caso de que falle el test, se envía el *checkpoint* a la siguiente versión y se repite la misma evaluación. Si todos los puntos de test fallan se considera una falla total y la medición *checkpoint* es errónea.

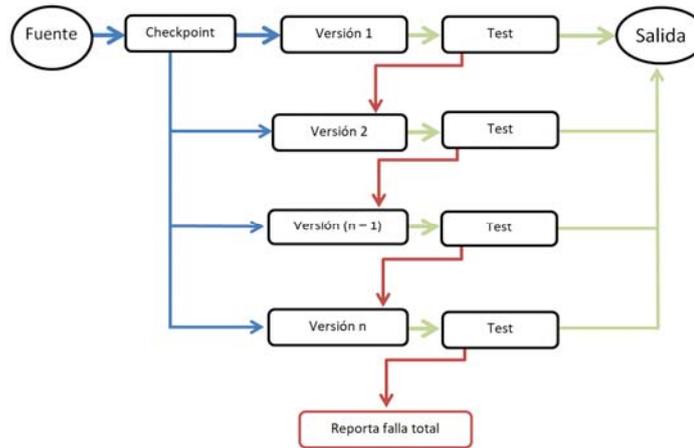


Fig. 3. Arquitectura de bloque de recuperación.

La arquitectura maestro-esclavo con *hot-swap* cuenta con módulos que cumplen dos roles diferenciados. El módulo esclavo toma la señal de la fuente y la procesa, cada esclavo además se da de baja luego de un número definido de fallos consecutivos. El módulo de maestro solicita esa información y trata cada una en un votador para enviar una salida correcta.



Fig. 4. Arquitectura de maestro esclavo.

La arquitectura de redundancia de hardware es de tipo convencional ejecutando una sola versión de software. Simplemente cuenta con la captura de la señal del sensor y un solo tratamiento software para cada entrada redundada.



Fig. 5. Arquitectura de redundancia de hardware.

2.1 Selección de las herramientas de análisis estático

Como se indicó al inicio, la normativa EN 50128, que se encuentra bajo la normativa general de sistemas críticos ferroviarios EN 50126, busca garantizar la fiabilidad, disponibilidad, mantenibilidad y seguridad del sistema. En particular, según el estándar EN 50128 sección 9.2.4.4, la mantenibilidad debe diseñarse como un aspecto inherente del software, y en vinculación con normas de calidad del producto software, como, por ejemplo la ISO/IEC 9126.

La selección de un patrón de arquitectura tiene consecuencias en la calidad del código fuente, y, por lo tanto en la mantenibilidad del producto software resultante [15]. Es decir, la selección de una arquitectura solamente basada en la seguridad es un enfoque parcial. Y ante diferentes patrones que pueden cumplir un nivel SSIL alguno de ellos podrá generar código fuente más mantenible que otro.

En este sentido, existen diferentes características a tener en cuenta. De acuerdo con los anexos A.8, A.10 y A.19 de la normativa 50128 es necesario realizar un análisis estático del código fuente asistido mediante herramientas. Para la elección del lenguaje de programación se tuvieron en cuenta las restricciones de arquitectura hardware de la unidad de procesamiento del microcontrolador y el apartado D.54 de la normativa EN 50128 en la cual se describen los requisitos que debe cumplir el lenguaje de programación. La elección del lenguaje C cumple con los requisitos de rigor mandatorio, esperado y deseable.

Teniendo en cuenta esto se han seleccionado las principales métricas relacionadas con la mantenibilidad a partir de las cuales poder comparar los patrones de arquitectura previamente identificados (ver tabla 2). Las métricas seleccionadas fueron las relacionadas con tamaño y complejidad, teniendo en cuenta desarrollos realizados anteriormente por el grupo de investigación [15]–[17]. Asimismo se tuvieron en cuenta los resultados de herramientas relacionadas con el uso de buenas prácticas de programación.

Junto con cada métrica se seleccionan las herramientas para el análisis estático del código fuente escrito en lenguaje C. Las herramientas se enumeran en la tercera columna.

Asimismo, se sumaron dos herramientas de análisis estático de código fuente orientadas a las normas de codificación. En especial, se incorporó el análisis de las buenas prácticas MISRA C [18], un estándar de facto para el desarrollo de software

crítico. Se eligió más de una herramienta por cada atributo para, de manera secundaria, contrastar las diferencias de medición.

Tabla 2. Selección de métricas y herramientas

Métrica	Descripción	Herramienta
LOC	Líneas de código fuente	CCCC[19] RSM [20] SONAR [21]
MVG	Complejidad ciclomática de McCabe. Es la cuenta de caminos linealmente independientes.	CCCC RSM SONAR
COM	Líneas de texto comentadas	CCCC RSM SONAR
RULES	Reglas de codificación orientadas al buen estilo, para aumentar la analizabilidad del código fuente. O reglas orientadas al uso correcto de las estrategias de programación	CPPCHECK[22] SONAR

3 Diseño del experimento

Habiendo seleccionado las arquitecturas y las herramientas de calidad se realizó el diseño del experimento con el cual se midieron las diferentes características de construcción de los patrones de arquitectura.

El experimento consistió en la construcción del código fuente que representa al patrón de arquitecturas en entornos controlados y homogéneos por parte de cinco grupos diferentes de trabajo. A partir de ello se pudieron medir tanto las características del proceso de desarrollo como las métricas del código fuente.

Resumiendo, el experimento tuvo los siguientes puntos:

- Contexto de trabajo: alumnos de la universidad homogeneizados habiendo realizado una prueba de habilidades en programación con lenguaje C.
- Capacitación previa: se realizó una capacitación previa a los diferentes equipos tanto en programación C como en aspectos de las arquitecturas.
- Desarrollo de los equipos: los equipos fueron elegidos al azar teniendo en cuenta las características de los alumnos. En base al estudio previo y la capacitación se dividieron a los alumnos en dos grandes grupos. Los equipos de trabajo fueron seleccionados con un integrante de cada grupo.
- Asignación de la arquitectura: se trabajó con cinco equipos de dos desarrolladores. cada uno asignado a una arquitectura determinada.

Los equipos tuvieron una semana de trabajo en los que se les proveyó del material necesario para la construcción de los prototipos hardware y el desarrollo del software de acuerdo con las características de cada patrón arquitectural. Tuvieron acceso al material de cada arquitectura para que puedan centrarse en el desarrollo propiamente de la solución.

A lo largo de la semana cada integrante incluía la cantidad de horas que estuvo trabajando en el desarrollo del software (esto incluía tiempo de codificación, tiempo de diseño o tiempo de análisis). Asimismo, al final de la semana cada integrante del equipo pudo evaluar el nivel de complejidad del desarrollo.

Una vez terminados los desarrollos y habiendo controlado el funcionamiento adecuado de cada solución se procedió a analizar el código fuente resultante con las herramientas indicadas en la tabla 2.

4 Resultado

A continuación se resumen los principales resultados y se establece una comparación en cuanto a nivel SSIL. Se ha seguido el apartado A.3 de la norma EN 50128 como criterio de selección, lo indicado en la publicación que presenta el uso de esta arquitectura y el resultado del experimento (ver tabla 3).

La segunda columna identifica que tan recomendable es este patrón para cada nivel de SSIL, siendo:

- DR. Débilmente recomendado para el nivel de seguridad.
- R. Recomendado para el nivel de integridad.
- AR. Altamente recomendado para el nivel de seguridad.

La tercera columna resume el tiempo de desarrollo estimado de acuerdo a las características del patrón. La cuarta columna promedia el grado de dificultad (Dif.) encontrado por parte de los miembros del equipo al construir el patrón (escala de 5 tipo lickert, 1 = muy sencillo, 5 = muy difícil). La quinta y sexta columna define la cantidad de horas invertidas por el equipo en análisis (TA) y desarrollo (TD).

Así, el patrón BR puede ser utilizado para niveles altos de seguridad y su complejidad es menor respecto de AV y PNV.

Tabla 3. Evaluación de los patrones de arquitectura.

Patrón	SSIL				Pasos de construcción	Dif.	TA	TD
	1	2	3	4				
PNV	DR	R	R	AR	Desarrollo de N versiones	4,2	25	30
AV	DR	R	R	AR	Desarrollo de N versiones más el desarrollo de los tests	4,5	30	35
BR	R	R	R	AR	Desarrollo de N versiones más desarrollo de los tests de forma secuencial	3,3	20	33
ME	DR	R	R	R	Desarrollo del maestro (control) y el esclavo (rutina para verificación del funcionamiento correcto)	3,5	20	22
RH	DR	R	R	R	Desarrollo de solo una versión	2,5	10	15

La arquitectura más sencilla de realizar fue la arquitectura de redundancia de hardware, en cuanto a desarrollo y número de componentes, la versión del programa es una sola y se sirve de la replicación de módulos de sensores. El único grado de dificultad se presenta en el control sobre estos, sin embargo se reduce a la gestión de conexión de los mismos.

La arquitectura más difícil de realizar fue la arquitectura AV. Se debió trabajar con implementaciones alternativas en cada versión para un mismo procesamiento, de manera que estas sean independientes. A su vez, el desarrollo de pruebas de aceptación aumentó la complejidad en cuanto al análisis. En cuanto a las métricas de código fuente, la arquitectura AV fue la de mayor cantidad de líneas de código y mayor complejidad ciclométrica

La arquitectura que resultó con la peor relación entre grado de nivel integral de seguridad y dificultad requeridos, fue la arquitectura ME. Se trabajó sobre las funciones delegadas a cada módulo y la dificultad se presenta al trabajar sobre dos equipos físicos distintos sumado al problema de comunicar de forma eficiente a ambas partes.

A continuación se pueden ver los resultados del análisis estático del código fuente para cada arquitectura (ver tabla 4). En general todas las herramientas coincidieron en cuanto a las mediciones. Respecto de la cantidad de líneas de código fuente AV y ME fueron los que presentaron mayor cantidad de líneas y mayor complejidad. Asimismo, AV y ME requirieron mayor cantidad de comentarios (la estrategia de desarrollo resultó más compleja). Respecto de las reglas de estilo y buenas prácticas de codificación, PNV y AV son las que concentraron mayor cantidad de violaciones.

Tabla 4. Evaluación de las métricas y atributos asociados a la mantenibilidad.

Métricas / Herramientas	PNV	AV	BR	ME	RH
RSM (LOC)	286	476	255	505	219
SONAR(LOC)	283	451	247	520	227
CCCC (COM)	62	317	245	299	136
RSM (COM)	69	306	284	298	129
SONAR(COM)	60	308	280	287	124
CCCC (MVG)	54	102	39	85	54
RSM (MVG)	57	106	39	87	52
SONAR(MVG)	55	105	38	84	53
CPPCHECK (RULES propias de estilo)	11	12	6	7	3
SONAR (RULES MISRA C de estilo)	0	1	0	4	2
CPPCHECK (RULES propias tipo "buena práctica")	0	1	0	0	0
SONAR (RULES MISRA C tipo "buena práctica")	3	1	0	3	1

5 Conclusiones y Trabajos Futuros

En el trabajo presentado se ha realizado una evaluación de diversas arquitecturas para software crítico ferroviario. Se han presentados los criterios de selección previos, empleando los estándares requeridos. Se realizó el diseño de un experimento para la construcción de las arquitecturas seleccionadas y su posterior evaluación en cuanto a complejidad. También se tuvieron en cuenta métricas de calidad de código fuente para cada implementación realizada, como está previsto en la sección 9.2.4.4 y el anexo A.19 de la norma EN 51280. De esta manera se pudo analizar la relación entre seguridad y mantenibilidad de las arquitecturas.

De las cinco arquitecturas analizadas se destaca la arquitectura PNV para niveles SSIL 3 y 4. Y la arquitectura RH para niveles SSIL 1 y 2. En ambos casos

Las arquitecturas más recomendables para los niveles altos de seguridad, como PNV y AV mantienen una mayor complejidad en cuanto al uso de buenas prácticas de codificación. Y por lo tanto, que pueda ser necesario un equipo de desarrollo más experimentado. En este sentido la arquitectura AV resultó ser mucho más compleja y menos mantenible.

Como trabajo futuro se pretende mejorar el diseño del experimento asegurando homogeneidad en el mismo. Asimismo se buscará aumentar la cantidad de arquitecturas y métricas.

Referencias

- [1] D. J. Smith y K. G. Simpson, *The Safety Critical Systems Handbook: A Straightforward Guide to Functional Safety: IEC 61508 (2010 Edition), IEC 61511 (2015 Edition) and Related Guidance*. Butterworth-Heinemann, 2016.
- [2] M. A. Lundteigen, M. Rausand, y I. B. Utne, «Integrating RAMS engineering and management with the safety life cycle of IEC 61508», *Reliability Engineering & System Safety*, vol. 94, n.º 12, pp. 1894–1903, 2009.
- [3] J.-L. Boulanger, *CENELEC 50128 and IEC 62279 standards*. John Wiley & Sons, 2015.
- [4] «Norma UNE-EN 50128:2012». [En línea]. Disponible en: <https://www.une.org/encuentra-tu-norma/busca-tu-norma/norma/?c=N0049040>. [Accedido: 18-jul-2019].
- [5] W. Wu y T. Kelly, «Safety tactics for software architecture design», en *Proceedings of the 28th Annual International Computer Software and Applications Conference, 2004. COMPSAC 2004.*, 2004, pp. 368–375.
- [6] L. Bass, P. Clements, y R. Kazman, *Software architecture in practice*. Addison-Wesley Professional, 2003.
- [7] M. S. Durmuş, O. Eriş, U. Yildirim, y M. T. Söylemez, «A new voting strategy in Diverse programming for railway interlocking systems», en *Proceedings 2011 International Conference on Transportation, Mechanical, and Electrical Engineering (TMEE)*, 2011, pp. 723–726.

- [8] S. Mockel, F. Scherer, y P. F. Schuster, «Multi-sensor obstacle detection on railway tracks», en IEEE IV2003 Intelligent Vehicles Symposium. Proceedings (Cat. No. 03TH8683), 2003, pp. 42–46.
- [9] M. Wiest, E. Kassa, W. Daves, J. C. Nielsen, y H. Ossberger, «Assessment of methods for calculating contact pressure in wheel-rail/switch contact», *Wear*, vol. 265, n.º 9-10, pp. 1439–1445, 2008.
- [10] A. Watanabe y K. Sakamura, «MLDD (multi-layered design diversity) architecture for achieving high design fault tolerance capabilities», en European Dependable Computing Conference, 1994, pp. 336–349.
- [11] J. P. J. Kelly, T. I. McVittie, y W. I. Yamamoto, «Implementing design diversity to achieve fault tolerance», *IEEE Software*, vol. 8, n.º 4, pp. 61–71, 1991.
- [12] I. I. Courtright, V. William, y G. A. Gibson, «Backward error recovery in redundant disk arrays», CARNEGIE-MELLON UNIV PITTSBURGH PA DEPT OF COMPUTER SCIENCE, 1994.
- [13] U. Voges, *Software diversity in computerized control systems*, vol. 2. Springer Science & Business Media, 2012.
- [14] J.-C. Laprie, «Dependable computing and fault-tolerance», *Digest of Papers FTCS-15*, pp. 2–11, 1985.
- [15] E. Irrazábal, J. Garzás, y E. Marcos, «Alignment of Open Source Tools with the New ISO 25010 Standard-Focus on Maintainability.», en ICISOFT (2), 2011, pp. 111–116.
- [16] E. Irrazábal, «Mejora de la mantenibilidad con un modelo de medición de la calidad: resultados en una gran empresa», en XXI Congreso Argentino de Ciencias de la Computación (Junín, 2015), 2015.
- [17] E. Irrazábal y J. Garzás, «Análisis de métricas básicas y herramientas de código libre para medir la mantenibilidad», *REICIS. Revista Española de Innovación, Calidad e Ingeniería del Software*, vol. 6, n.º 3, pp. 56–65, 2010.
- [18] L. Hatton, «Safer language subsets: an overview and a case history, MISRA C», *Information and Software Technology*, vol. 46, n.º 7, pp. 465–472, 2004.
- [19] «C and C++ Code Counter download | SourceForge.net». [En línea]. Disponible en: <https://sourceforge.net/projects/cccc/>. [Accedido: 18-jul-2019].
- [20] «M Squared Technologies LLC - RSM for C/C++, Java and C#», *msquaredtechnologies*. [En línea]. Disponible en: <http://msquaredtechnologies.com/index.html>. [Accedido: 18-jul-2019].
- [21] «Continuous Code Quality | SonarSource». [En línea]. Disponible en: <https://www.sonarsource.com/>. [Accedido: 18-jul-2019].
- [22] «Cppcheck - A tool for static C/C++ code analysis». [En línea]. Disponible en: <http://cppcheck.sourceforge.net/>. [Accedido: 18-jul-2019].

Guía para evaluar calidad de datos basada en ISO/IEC 25012

Julieta Calabrese¹ , Silvia Esponda , Ariel Pasini , Marcos Boracchia ,
Patricia Pesado 

Instituto de Investigación en Informática LIDI (III-LIDI)*
Facultad de Informática - Universidad Nacional de La Plata
50 y 120 - La Plata - Buenos Aires

*Centro Asociado Comisión de Investigaciones Científicas de la Pcia. de Bs. As. (CIC)
¹Becaria UNLP

{jcalabrese, sesponda, apasini, marcosb,
ppesado}@lidi.info.unlp.edu.ar

Abstract: En la actualidad, las organizaciones manejan un gran volumen de datos. Poder fomentar y mantener la calidad de los mismos es uno de los grandes retos a los que se enfrentan dichas organizaciones hoy en día. Existen normas, tales como ISO/IEC 25012, destinadas a poder medir la calidad de los datos basándose en un conjunto de características. Realizar dichas mediciones puede resultar costoso y complejo para quien cuente con poca experiencia en el área. En este contexto, el presente artículo propone una guía de medición basada en la ISO/IEC 25012, con el fin de brindar un estado actual de los datos en diferentes aspectos definidos en la norma. Asimismo, la guía propuesta es utilizada en una evaluación de producto bajo el proceso definido en ISO/IEC 25040, tomando como caso de estudio una base de datos de acceso libre.

Keywords: Calidad de datos - Guía de medición - ISO/IEC 25012 - ISO/IEC 25040

1. Introducción

Desde los inicios de la Ingeniería de Software, siempre estuvieron presentes los problemas relacionados con alcanzar niveles óptimos de calidad en diferentes aspectos del software. La calidad de un sistema informático depende esencialmente de los procesos utilizados para su producción junto con la posibilidad de adecuarse a determinados factores, tales como el tiempo de desarrollo y el costo de soluciones. Dichos factores suelen afectar el trabajo a realizar, siendo la calidad la primera variable de ajuste.

La gestión de la calidad es una parte fundamental dentro de cualquier organización. Éstas manejan una gran variedad de información, aspecto clave para la toma de decisiones que dependen directamente de los datos con los que operan. Sin embargo, no suelen disponer de recursos que evalúen la calidad de dichos datos, siendo uno de los principales problemas a los que se enfrentan. No contar con datos de calidad puede generar grandes consecuencias en la organización al momento de querer destacarse en el mercado actual y de prestar un mejor servicio que cumpla con las exigencias del cliente. Dentro de este contexto, es de sumo interés evaluar diferentes características asociadas a los datos utilizados.

Esto nos lleva a centrar nuestra atención en los estándares definidos por ISO. La producción de software tiene a su disposición un conjunto de normas específicas destinadas a evaluar diferentes aspectos del software. La familia ISO/IEC 25000 (mayormente conocida como SQuaRE: Software Product Quality Requirements and Evaluation) nace en el año 2005 para dar respuesta a las necesidades actuales de las organizaciones. Su objetivo es la creación de un marco de trabajo común para evaluar la calidad de un producto software, sustituyendo a las anteriores ISO/IEC 9126 [1] e ISO/IEC 14598 [2]. Dentro de la familia ISO/IEC 25000, se presenta la norma ISO/IEC 25012 - “Data Quality Model” [3] y la ISO/IEC 25040 - “Evaluation process” [4].

La norma ISO/IEC 25012 – “Data Quality Model” especifica un modelo general de calidad para aquellos datos que se encuentran definidos en un formato estructurado dentro de un sistema informático. Puede utilizarse para detallar requisitos, establecer medidas, como así también para planificar y realizar evaluaciones de calidad de datos. La norma clasifica los atributos de calidad en quince características analizadas desde dos puntos de vista: inherentes y dependientes del sistema. Dichas características tendrán una importancia y prioridad diferente para cada persona, siendo ésta la encargada de definir sus propias necesidades.

Las características definidas en la norma poseen ejemplos de métricas que permiten obtener valores para establecer los niveles de dichas características mediante fórmulas. De la misma manera que en la norma ISO/IEC 25024 - “Measurement of data quality” [6], en donde se definen métricas para realizar la medición de la calidad de datos. Estas métricas requieren cierto nivel de análisis, ya que se componen de valores que deben ser de conocimiento por parte del evaluador. Esta tarea suele ser dificultosa para alguien con poca experiencia en el área de calidad e incluso costosa para empresas que estén interesadas en medir niveles de calidad y no posean un sector especializado en el tema..

Por otra parte, la norma ISO/IEC 25040 proporciona requisitos, recomendaciones y guías para llevar a cabo el proceso de evaluación del producto de software. Define un modelo de referencia para la evaluación, considerando las entradas, restricciones y recursos necesarios para obtener las salidas correspondientes. El proceso definido está formado por cinco secciones.

El presente artículo propone una guía de medición para aquellos interesados en reconocer el nivel de ciertas características de calidad asociadas a sus datos, empleando su uso en el contexto de una evaluación de producto de software. De esta forma, una organización podrá obtener una retroalimentación del estado de los datos con los que opera, pudiendo especificar previamente cuáles son los valores esperados. Cabe destacar que la aplicación de la guía en una evaluación no se corresponde con una certificación.

En la siguiente sección se describen en profundidad las estructuras de las normas ISO/IEC 25012 e ISO/IEC 25040, haciendo hincapié en las características asociadas a los datos. En la tercera sección se define la guía de medición propuesta para medir la calidad de los mismos, tomando como referencia lo definido en ISO/IEC 25012. En la cuarta sección se realiza una evaluación de producto guiada por ISO/IEC 25040, utilizando la guía definida y

tomando como caso de prueba un conjunto de datos de acceso libre. Por último, se exponen las conclusiones y la bibliografía.

2. ISO/IEC 25000

La familia ISO/IEC 25000 (SQuaRE) propone un conjunto de normas que tiene por objetivo la creación de un marco de trabajo común para evaluar la calidad del producto de software. En base a la problemática definida, es de interés destacar en el marco de este artículo las normas ISO/IEC 25012 e ISO/IEC 25040.

2.1 ISO/IEC 25012

La calidad de los datos es un factor clave en cualquier organización, ya que los mismos representan a la información de una manera formal y adecuada para la comunicación, la interpretación o el procesamiento. ISO/IEC 25012 define un modelo general de calidad para aquellos datos que están representados en un formato estructurado dentro de un sistema informático, y tiene como objetivo presentar una visión integrada de los mismos para garantizar la interoperabilidad de los sistemas. La norma está compuesta por un conjunto de quince características, las cuales se clasifican en dos grandes grupos:

Calidad de Datos Inherente: Grado con el que las características tienen el potencial de satisfacer las necesidades establecidas y necesarias cuando los datos son utilizados bajo condiciones específicas. Desde el punto de vista inherente, la calidad de datos hace referencia a:

- Valores de datos para el dominio y sus posibles restricciones (ej., Reglas de negocio con la calidad requerida por las características en una aplicación)
- Relaciones entre valores de datos (ej., Consistencia)
- Metadatos (ej., Autor, Formato, etc.)

Calidad de Datos Dependiente del Sistema: Grado con el que la calidad de datos es alcanzada y preservada a través de un sistema informático cuando los datos son utilizados bajo condiciones específicas. Desde este punto de vista, la calidad de datos depende del dominio tecnológico en el que los datos se utilizan, y se alcanza mediante las capacidades de los componentes del sistema informático tales como: dispositivos hardware (ej., respaldo para alcanzar la Recuperabilidad) y software (ej., herramientas de migración para alcanzar la Portabilidad). Este punto de vista suele ser responsabilidad de los técnicos del sistema.

La siguiente tabla grafica la clasificación de las características:

Característica	Inherente	Dependiente del sistema
Exactitud	X	
Compleitud	X	
Consistencia	X	
Credibilidad	X	
Actualidad	X	
Accesibilidad	X	X
Conformidad	X	X
Confidencialidad	X	X

Eficiencia	X	X
Precisión	X	X
Trazabilidad	X	X
Comprensibilidad	X	X
Disponibilidad		X
Portabilidad		X
Recuperabilidad		X

Tabla 1. Clasificación de características definidas en ISO/IEC 25012.

El conjunto de características que conforman la *Calidad de Datos Inherente* está formado por: **Exactitud** (dividida en Semántica y Sintáctica), que especifica el grado en el que los datos representan correctamente el valor deseado en un contexto específico; **Compleitud**, donde se espera que los datos obligatorios estén completos; **Consistencia**, haciendo referencia a los datos libres de contradicción y coherentes en un contexto específico; **Credibilidad**, la cual incluye el concepto de autenticidad, que define el grado en el que los datos se consideran ciertos y creíbles en un contexto específico; y **Actualidad**, la cual define el grado en el que los datos se encuentran actualizados.

Por otra parte, las características que conforman la *Calidad de Datos Inherente y Dependiente del Sistema* hacen referencia a la **Accesibilidad**, donde se especifica el grado en que los datos pueden ser accedidos en un contexto específico (por ejemplo, personas que necesiten tecnologías de apoyo por algún tipo de discapacidad); **Conformidad**, donde se verifica que los datos correspondientes cumplen con estándares, convenciones o normativas vigentes; **Confidencialidad** (asociada a la seguridad de la información), donde se asegura que los datos solo son accedidos e interpretados por usuarios específicos autorizados; **Eficiencia**, donde se analiza el grado en el que los datos pueden ser procesados y proporcionados con los niveles de rendimiento esperados; **Precisión**, donde los datos requieren de valores exactos o con discernimiento en un contexto específico; **Trazabilidad**, donde se analiza si los datos proporcionan un registro de los acontecimientos que los modifican; y **Comprensibilidad**, en donde los datos son expresados utilizando lenguajes, símbolos y unidades apropiados y pueden ser leídos e interpretados por cualquier tipo de usuario.

Por último, las características que conforman la *Calidad de Datos Dependiente del Sistema* hacen hincapié en la **Disponibilidad**, que define el grado de los datos para ser obtenidos por usuarios y/o aplicaciones autorizadas; **Portabilidad**, donde se analiza si los datos pueden ser copiados, reemplazados o eliminados al realizar un cambio de un sistema a otro, preservando el nivel de calidad; y **Recuperabilidad**, donde se comprueba que los datos mantienen y preservan un nivel de operaciones en caso de fallos.

2.2 ISO/IEC 25040

El uso de normas y estándares para lograr mayor competitividad es cada vez más habitual en las PyMEs desarrolladoras de software, ya sea para la obtención de una certificación o simplemente para lograr una mejora en el desarrollo del software a través del uso de buenas prácticas. ISO/IEC 25040 define el proceso para llevar a cabo la evaluación del producto de software mediante un modelo de referencia, considerando las entradas, restricciones y recursos necesarios para obtener las salidas correspondientes. El proceso para llevar a cabo la evaluación consta de cinco actividades:

- Actividad 1: *Establecer los requisitos de la evaluación*. Esta actividad consiste en establecer el propósito de la evaluación, identificando las partes interesadas en el producto, los riesgos posibles si los hubiese y el modelo de calidad a utilizar.
- Actividad 2: *Especificar la evaluación*. Dentro de esta actividad se especifican los módulos de evaluación (métricas, herramientas y técnicas) junto con los criterios de decisión a aplicar.
- Actividad 3: *Diseñar la evaluación*. En esta actividad se define el plan con las tareas que se deben realizar en la evaluación.
- Actividad 4: *Ejecutar la evaluación*. Actividad destinada a la ejecución de la evaluación, obteniendo las métricas de calidad y aplicando los criterios de decisión.
- Actividad 5: *Concluir la evaluación*. En esta última actividad se culmina la evaluación de la calidad, realizando un informe de resultados finales y conclusiones en base a los valores obtenidos.

3. Guía de medición basada en características definidas en ISO/IEC 25012

Uno de los elementos claves en cualquier proceso de ingeniería es la medición. Particularmente en SQuaRE, las medidas se emplean para analizar el nivel de diferentes características asociadas a lo que se desea evaluar. El proceso de medición puede resultar complejo para personas con poca experiencia en el área de calidad ya que se debe llevar a cabo una evaluación de producto aplicando métricas, quizás sin contar con la experiencia y los conocimientos necesarios para ello.

Dentro de este contexto, se presenta una guía que tiene como objetivo medir las características definidas en ISO/IEC 25012. Cabe mencionar que en el presente artículo sólo se presentan tres de ellas, pero la misma incluye todas las características de la norma.

La guía define para cada una de las características diferentes secciones:

- Tipo: Inherente | Dependiente del sistema.
- Documentación previa requerida: Documentación/especificación necesaria para poder realizar la medición. Se indica si es necesaria para cada uno de los atributos¹ a evaluar o genérica.
- Método para la medición: En base a la documentación previa requerida, se indica cuál es la forma de realizar la medición.
- Variables: Valores a completar en base a la medición.
- Fórmula: Fórmula a resolver para obtener el valor final.
- Escala: Opción a seleccionar luego de aplicar el método para la medición.

La *Tabla 2* especifica las secciones anteriormente definidas para las características Exactitud (dividida en Semántica y Sintáctica), Completitud y Actualidad.

Semántica (Exactitud)	Tipo	Inherente
	Documentación previa	Por cada atributo a evaluar: Especificar reglas semánticas.

¹ Hace referencia a las características de una entidad en la base de datos.

	<i>requerida</i>	
	<i>Método para la medición</i>	Por cada atributo, verificar si los datos cumplen con las reglas semánticas definidas.
	<i>Variables</i>	X = Porcentaje de datos del atributo que cumplen con las reglas semánticas definidas.
	<i>Fórmula</i>	Valor = X/100%
	<i>Escala</i>	1) Casi ningún dato se corresponde con las reglas definidas ($X \leq 10\%$). 2) Algunos datos se corresponden con las reglas, pero se observan muchos errores ($X > 10\% \& X \leq 45\%$). 3) Muchos datos se corresponden con las reglas definidas y se observan algunos errores ($X > 45\% \& X \leq 85\%$). 4) La gran totalidad de los datos se corresponden con las reglas definidas ($X > 85\%$).
Sintáctica (Exactitud)	<i>Tipo</i>	Inherente
	<i>Documentación previa requerida</i>	Por cada atributo a evaluar: Especificar reglas sintácticas.
	<i>Método para la medición</i>	Por cada atributo, verificar si los datos cumplen con las reglas sintácticas definidas.
	<i>Variables</i>	X = Porcentaje de datos del atributo que cumplen con las reglas sintácticas definidas.
	<i>Fórmula</i>	Valor = X/100%
	<i>Escala</i>	1) Casi ningún dato se corresponde con las reglas definidas ($X \leq 10\%$). 2) Algunos datos se corresponden con las reglas, pero se observan muchos errores ($X > 10\% \& X \leq 45\%$). 3) Muchos datos se corresponden con las reglas definidas y se observan algunos errores ($X > 45\% \& X \leq 85\%$). 4) La gran totalidad de los datos se corresponden con las reglas definidas ($X > 85\%$).
Completitud	<i>Tipo</i>	Inherente
	<i>Documentación previa requerida</i>	Por cada atributo a evaluar: Indicar si es obligatorio.
	<i>Método para la medición</i>	Por cada atributo, verificar que no haya datos vacíos/en blanco.
	<i>Variables</i>	X = Porcentaje de datos del atributo que se encuentran completos.
	<i>Fórmula</i>	Valor = X/100%
	<i>Escala</i>	1) Casi ningún dato se encuentra completo ($X \leq 10\%$). 2) Algunos datos se encuentran completos, pero se observan muchos incompletos ($X > 10\% \& X \leq 45\%$). 3) Muchos datos se encuentran completos y se observan algunos incompletos ($X > 45\% \& X \leq 85\%$).

		4) La gran totalidad de los datos se encuentran completos ($X > 85\%$).
Actualidad	<i>Tipo</i>	Inherente
	<i>Documentación previa requerida</i>	Fecha de la última actualización de la base de datos.
	<i>Método para la medición</i>	Analizar la fecha en la cual los datos fueron actualizados por última vez.
	<i>Variables</i>	-
	<i>Fórmula</i>	-
	<i>Escala</i>	1) Los datos fueron actualizados en el corriente mes. 2) Los datos fueron actualizados en el transcurso de este año. 3) Los datos fueron actualizados en el transcurso de los últimos tres años. 4) Desconozco cualquier dato relacionado a la fecha de actualización de los datos.

Tabla 2. Guía de medición.

En la siguiente sección se lleva a cabo una evaluación de producto tomando como base el proceso definido en ISO/IEC 25040, utilizando la ISO/IEC 25012 para la selección de características. Asimismo, se utiliza la guía definida para la medición de dichas características explicando paso a paso su modo de aplicación.

4. Evaluación de calidad de datos aplicando la guía de medición

Para llevar a cabo la evaluación se utilizó, a modo de guía, el proceso y la estructura definida en ISO/IEC 25040. Asimismo, se seleccionó una base de datos de acceso libre que contiene información relacionada a farmacias que dispensan medicamentos de PAMI. Los atributos que conforman la base de datos utilizada son: Código, Nombre, Dirección, Provincia, Localidad, Código Postal (CP), CUIT, Institución y Entidad [5].

4.1 Establecer los requisitos de la evaluación

En primer lugar, se define el propósito de la evaluación.

El propósito de la evaluación es determinar si los datos considerados como obligatorios están presentes, analizar si los campos cumplen con los formatos esperados y considerar qué tan actualizados se encuentran los datos.

Basándose en el propósito, se seleccionan las características de interés para la evaluación, definidas en la norma ISO/IEC 25012.

Para el propósito planteado, se seleccionan las características: Exactitud (Semántica y Sintáctica), Completitud y Actualidad.

4.2 Especificar la evaluación

Como se mencionó anteriormente, la guía definida en la sección 3 se utiliza como método de medición para las características definidas en ISO/IEC 25012. Este último define una

métrica de ejemplo por cada característica, a excepción de la Exactitud que define dos (correspondientes a la Semántica y a la Sintáctica). La guía de medición mantiene este mismo lineamiento.

En esta etapa de la evaluación, es necesario definir criterios de decisión² para cada una de las características y para la evaluación final. Para ello, debe analizarse la escala de valores y la fórmula brindada en la guía, definiendo los rangos para los posibles resultados.

A continuación, se definen los criterios de decisión por cada característica seleccionada para el propósito de la evaluación.

Semántica (Exactitud):

Atributo: Código	Inaceptable	Si Valor ≥ 0 y Valor $< 0,4$
	Mínimamente aceptable	Si Valor $\geq 0,4$ y Valor $< 0,6$
	Rango Objetivo	Si Valor $\geq 0,6$ y Valor $< 0,9$
	Excede los Requerimientos	Si Valor $\geq 0,9$
Atributo: Localidad	Inaceptable	Si Valor ≥ 0 y Valor $< 0,3$
	Mínimamente aceptable	Si Valor $\geq 0,3$ y Valor $< 0,6$
	Rango Objetivo	Si Valor $\geq 0,6$ y Valor $< 0,7$
	Excede los Requerimientos	Si Valor $\geq 0,7$
Atributo: Entidad	Inaceptable	Si Valor ≥ 0 y Valor $< 0,3$
	Mínimamente aceptable	Si Valor $\geq 0,3$ y Valor $< 0,6$
	Rango Objetivo	Si Valor $\geq 0,6$ y Valor $< 0,7$
	Excede los Requerimientos	Si Valor $\geq 0,7$
Semántica	Inaceptable	Código: Mínimamente aceptable. Localidad: Inaceptable. Entidad: Inaceptable.
	Mínimamente aceptable	Código: Rango Objetivo. Localidad: Mínimamente aceptable. Entidad: Mínimamente aceptable.
	Rango Objetivo	Código: Rango Objetivo. Localidad: Rango Objetivo. Entidad: Rango Objetivo.
	Excede los Requerimientos	Código: Excede los requerimientos. Localidad: Excede los requerimientos. Entidad: Rango Objetivo.

Tabla 3. Criterios de decisión para la semántica.

Sintáctica (Exactitud):

Atributo: Código	Inaceptable	Si Valor ≥ 0 y Valor $< 0,4$
	Mínimamente aceptable	Si Valor $\geq 0,4$ y Valor $< 0,6$

² Rangos de valores clasificados en Inaceptable, Mínimamente Aceptable, Rango Objetivo y Excede los Requerimientos.

	Rango Objetivo	Si Valor $\geq 0,6$ y Valor $< 0,9$
	Excede los Requerimientos	Si Valor $\geq 0,9$
Atributo: CP	Inaceptable	Si Valor ≥ 0 y Valor $< 0,1$
	Mínimamente aceptable	Si Valor $\geq 0,1$ y Valor $< 0,3$
	Rango Objetivo	Si Valor $\geq 0,3$ y Valor $< 0,6$
	Excede los Requerimientos	Si Valor $\geq 0,6$
Atributo: Entidad	Inaceptable	Si Valor ≥ 0 y Valor $< 0,1$
	Mínimamente aceptable	Si Valor $\geq 0,1$ y Valor $< 0,3$
	Rango Objetivo	Si Valor $\geq 0,3$ y Valor $< 0,6$
	Excede los Requerimientos	Si Valor $\geq 0,6$
Sintáctica	Inaceptable	Código: Mínimamente aceptable. CP: Inaceptable. Entidad: Inaceptable.
	Mínimamente aceptable	Código: Rango Objetivo. CP: Mínimamente aceptable. Entidad: Rango objetivo.
	Rango Objetivo	Código: Excede los requerimientos. CP: Rango objetivo. Entidad: Rango objetivo.
	Excede los Requerimientos	Código: Excede los requerimientos. CP: Rango objetivo. Entidad: Excede los requerimientos.

Tabla 4. Criterios de decisión para la sintáctica.

Exactitud	Inaceptable	Semántica: Inaceptable. Sintáctica: Inaceptable.
	Mínimamente aceptable	Semántica: Mínimamente aceptable. Sintáctica: Mínimamente aceptable.
	Rango Objetivo	Semántica: Rango Objetivo. Sintáctica: Mínimamente aceptable.
	Excede los Requerimientos	Semántica: Excede los requerimientos. Sintáctica: Excede los requerimientos.

Tabla 5. Criterios de decisión para la característica "Exactitud".

Compleitud:

Atributo: CP	Inaceptable	Si Valor ≥ 0 y Valor $< 0,3$
	Mínimamente aceptable	Si Valor $\geq 0,3$ y Valor $< 0,6$
	Rango Objetivo	Si Valor $\geq 0,6$ y Valor $< 0,7$
	Excede los Requerimientos	Si Valor $\geq 0,7$
Atributo: Nombre	Inaceptable	Si Valor ≥ 0 y Valor $< 0,4$
	Mínimamente aceptable	Si Valor $\geq 0,4$ y Valor $< 0,7$

	Rango Objetivo	Si Valor $\geq 0,7$ y Valor $< 0,9$
	Excede los Requerimientos	Si Valor $\geq 0,9$
Completitud	Inaceptable	CP: Mínimamente aceptable. Nombre: Mínimamente aceptable.
	Mínimamente aceptable	CP: Rango objetivo. Nombre: Rango objetivo.
	Rango Objetivo	CP: Excede los requerimientos. Nombre: Excede los requerimientos.
	Excede los Requerimientos	CP: Excede los requerim. Nombre: Excede los requerim.

Tabla 6. Criterios de decisión para la característica "Completitud".

Actualidad:

Actualidad	Inaceptable	Opción 4
	Mínimamente aceptable	Opción 3
	Rango Objetivo	Opción 2
	Excede los Requerimientos	Opción 1

Tabla 7. Criterios de decisión para la característica "Actualidad".

Por último, se definen los criterios de decisión para la evaluación final. Cabe mencionar que cada característica debe tomar COMO MÍNIMO el valor definido (a modo de ejemplo, la evaluación se considera en un "Rango Objetivo" si la Exactitud posee como mínimo un rango objetivo, si la Completitud como mínimo excede los requerimientos, y así sucesivamente).

Evaluación final:

Resultado evaluación	Inaceptable	Exactitud: Inaceptable. Completitud: Inaceptable. Actualidad: Inaceptable.
	Mínimamente aceptable	Exactitud: Min aceptable. Completitud: Rango Objetivo. Actualidad: Min aceptable.
	Rango Objetivo	Exactitud: Rango Objetivo. Completitud: Excede los requerimientos. Actualidad: Rango Objetivo.
	Excede los Requerimientos	Exactitud: Excede los requerimientos. Completitud: Excede los requerimientos. Actualidad: Excede los requerimientos.

Tabla 8. Criterios de decisión para la evaluación final.

4.3 Diseñar la evaluación

Al momento de diseñar la evaluación, debe especificarse la documentación previa requerida en la guía para cada una de las características seleccionadas. La *Tabla 9* ilustra esta situación.

Característica	Documentación previa requerida
Semántica (Exactitud)	Atributos sobre los cuales se analizará la semántica: Código -> Se esperan números que comiencen con 9. Localidad -> Se esperan ciudades de Argentina. Entidad -> Se esperan "FACAF, FEFARA, COFA, SINDICAL".
Sintáctica (Exactitud)	Atributos sobre los cuales se analizará la sintáctica: Código -> Deben tener 9 dígitos. CP -> 4 dígitos. Entidad -> Deben escribirse en mayúscula.
Complejidad	CP -> Dato obligatorio. Nombre -> Dato obligatorio.
Actualidad	Fecha de actualización de los datos: 19 de Julio de 2019.

Tabla 9. Documentación previa requerida.

4.4 Ejecutar la evaluación

Para ejecutar la evaluación, por cada una de las características se debe analizar la guía de medición y llevar a cabo un conjunto de pasos:

- 1) Aplicar el método para la medición.
- 2) Seleccionar un valor de la escala, definiendo el valor de X en el caso que sea necesario.
- 3) Resolver la fórmula planteada.
- 4) Analizar los valores resultantes en conjunto con los criterios de decisión definidos, obteniendo un resultado final.

Se ejecutó la evaluación según lo planificado y se obtuvieron los siguientes resultados:

Exactitud:

Atributo: Código	Opción 3 (X = 85%)	Valor = $85/100 = 0,85$ (Rango Objetivo)
Atributo: Localidad	Opción 4 (X = 90%)	Valor = $90/100 = 0,90$ (Excede los requerimientos)
Atributo: Entidad	Opción 3 (X = 80%)	Valor = $80/100 = 0,80$ (Excede los requerimientos)
Semántica		Rango Objetivo
Atributo: Código	Opción 3 (X = 80%)	Valor = $80/100 = 0,80$ (Rango Objetivo)
Atributo: CP	Opción 4 (X = 100%)	Valor = $100/100 = 1$ (Excede los requerimientos)
Atributo: Entidad	Opción 3 (X = 55%)	Valor = $55/100 = 0,55$ (Rango Objetivo)
Sintáctica		Mínimamente aceptable

Tabla 10. Valores obtenidos en la medición de la semántica y sintáctica.

Semántica	-	Rango Objetivo
Sintáctica	-	Mínimamente aceptable
Exactitud		Rango Objetivo

Tabla 11. Valores obtenidos en la medición de la Exactitud.

Compleitud:

Atributo: CP	Opción 3 (80%)	Valor = $80/100 = 0,80$ (Excede los requerimientos)
Atributo: Nombre	Opción 4 (100%)	Valor = $100/100 = 1$ (Excede los requerimientos)
Compleitud		Excede los requerimientos

Tabla 12. Valores obtenidos en la medición de la Compleitud.

Actualidad:

-	Opción 1	Los datos fueron actualizados en el corriente mes (Excede los requerimientos)
Actualidad		Excede los requerimientos

Tabla 13. Valores obtenidos en la medición de la Actualidad.

4.5 Finalización de la evaluación

En base a los valores obtenidos y a los criterios de decisión definidos, se determina el resultado de la evaluación.

Exactitud	-	Rango Objetivo
Compleitud	-	Excede los requerimientos
Actualidad	-	Excede los requerimientos
Resultado Evaluación		Rango Objetivo

Tabla 14. Valores obtenidos para la evaluación final.

5. Conclusiones

Se diseñó y presentó una guía de medición para cada una de las características definidas en ISO/IEC 25012, cuya medición actualmente requiere de un análisis que puede resultar complejo y costoso para una persona con poca experiencia en el área. Se hizo hincapié en que sea de fácil aplicación y lectura por cualquier tipo de usuario.

Como caso de prueba, se llevó a cabo una evaluación basada en el proceso definido en ISO/IEC 25040 utilizando la guía de medición. Para realizar la evaluación fue necesario contar con una base de datos, la cual se obtuvo de un sitio de Datos Abiertos de Argentina.

6. Bibliografía

[1] ISO/IEC 9126. Software engineering -- Product quality.

[2] ISO/IEC 14598. *Information technology -- Software product evaluation.*

[3] ISO/IEC 25012:2008. *Software engineering -- Software product Quality Requirements and Evaluation (SQuaRE) -- Data quality model.*

[4] ISO/IEC 25040:2011. *Systems and software engineering -- Systems and software Quality Requirements and Evaluation (SQuaRE) -- Evaluation process.*

[5] Datos obtenidos en <http://datos.gob.ar>

[6] ISO/IEC 25024:2015. *Systems and software engineering -- Systems and software Quality Requirements and Evaluation (SQuaRE) -- Measurement of data quality*

Hacia la construcción de drivers eficientes en bounded model checking mediante detección automática de builders

Mariano Politano^{1,2}, Valeria S. Bengolea¹, Pablo Ponzio^{1,2}, and Nazareno Aguirre^{1,2}

¹ Universidad Nacional de Río Cuarto, Río Cuarto, Argentina.

{mpolitano, vbengolea, pponzio, naguirre}@dc.exa.unrc.edu.ar

² Consejo Nacional de Investigaciones Científicas y Técnicas (CONICET), Argentina

Resumen Las técnicas que permiten mejorar la calidad del software producido son de vital importancia, sobre todo en sistemas críticos. Entre ellas, contamos con técnicas de verificación acotada de software, como el model checking de software, que permiten explorar exhaustivamente todas las ejecuciones posibles del software con entradas de tamaño acotado, y reportar fallas encontradas durante el proceso.

Para llevar a cabo la verificación acotada, los model checkers de software se basan en la definición de *drivers*: combinaciones de métodos que permiten construir las entradas con las que se ejecutará el programa. En este trabajo se observa que la selección de los métodos empleados en la definición del *driver* es de vital importancia para la verificación. Intuitivamente, es deseable seleccionar un conjunto de métodos tan pequeño como sea posible (para mayor eficiencia en el análisis), cuyas combinaciones permitan construir todas las estructuras acotadas para el módulo (para analizar el software con todas las entradas posibles). Esta selección de métodos, que usualmente se lleva a cabo de forma manual, no es una tarea fácil: requiere un análisis exhaustivo de las rutinas disponibles en el módulo y una comprensión profunda de la semántica de las mismas.

En este trabajo se propone utilizar una herramienta automática para seleccionar un subconjunto de métodos relevantes de un módulo para la construcción de *drivers* eficientes para *bounded model checking*. Además, se evalúa el enfoque propuesto en el análisis de una propiedad particular del módulo Apache NodeCachingLinkedList, empleando el *model checker* Java PathFinder (JPF). Los resultados muestran que el enfoque de construcción de drivers presentado permite incrementar la eficiencia y la escalabilidad a estructuras de mayor tamaño en el análisis usando JPF.

1. Introducción

El uso de software está en continuo crecimiento gracias al avance de la tecnología, lo que significa que garantizar el correcto funcionamiento del software es más crucial que nunca. Esto es central sobre todo en los denominados sistemas críticos, como lo son los sistemas de equipamiento médico, de control de aviones,

etc, cuyas fallas pueden tener consecuencias fatales. Por lo tanto, un área de investigación de creciente importancia es la de análisis automático de software, cuyo objetivo es ayudar a los desarrolladores de software a detectar fallas en el software mediante herramientas que funcionan de manera autónoma, es decir, sin asistencia por parte del desarrollador. La generación automática de tests [4,1], la verificación de software [3,10], y los análisis estáticos [2], entre muchos otros, son enfoques destacados en esta línea de investigación.

Java PathFinder (JPF) es una herramienta de verificación automática de programas, que implementa diversas técnicas de análisis [10]. Uno de las posibles aplicaciones de JPF, en la que estamos interesados en este artículo, es realizar *bounded model checking*. Esto es explorar sistemáticamente el conjunto de todas las ejecuciones del software con entradas de tamaño acotado en busca de fallas. Los límites en los tamaños de las entradas deben ser proporcionados por el usuario de la herramienta (ej. cantidad máxima de nodos en una lista, rango de enteros a considerar, etc.). En caso de encontrar una falla, JPF la reporta al usuario para facilitar el debugging.

Para llevar a cabo la verificación acotada, JPF se basa en la definición de *drivers*: combinaciones de métodos que permiten construir todas las entradas acotadas para ejecutar el código bajo análisis (que usualmente requieren del uso de construcciones no deterministas del *model checker*). Para tipos de datos complejos (ej. estructuras dinámicas alocadas en el *heap* como listas doblemente encadenadas) usualmente la creación de estructuras requiere de la utilización de varias de las rutinas de la API del módulo que implementa la estructura (ej. add, remove, etc.), y por lo tanto, la construcción del *driver* depende de estas rutinas.

Sin embargo, para módulos ricos con respecto a la cantidad de métodos, la selección de métodos para construir el *driver* es una tarea no trivial, ya que como se detalla en la sección 5 de resultados experimentales, esto puede tener un impacto importante en el análisis posterior. La intuición manifiesta que es deseable seleccionar un conjunto de métodos tan pequeño como sea posible cuyas combinaciones permitan construir todas las estructuras acotadas para el módulo. Por un lado, es importante no incluir rutinas superfluas con respecto a la construcción de objetos, ya que la cantidad de combinaciones posibles de un conjunto de métodos crece exponencialmente con el número de métodos en el conjunto (por lo tanto, utilizar un subconjunto más pequeño de métodos implica una mejora de la eficiencia del análisis). El ejemplo más simple de rutinas superfluas son los métodos puros [6], que nunca modifican las estructuras sobre las que operan. Por otro lado, cuanto mayor es el número de estructuras diferentes que se pueda crear con el subconjunto de rutinas seleccionado, mayor será la probabilidad de encontrar errores en el código. Es decir, es importante seleccionar subconjuntos de métodos suficientes, que pueden combinarse para construir todas las instancias acotadas posibles. Esta selección de métodos, que usualmente se lleva a cabo de forma manual, no es una tarea fácil: requiere un análisis exhaustivo de las rutinas disponibles en el módulo y una comprensión profunda de la semántica de las mismas. Esta tarea es muy tediosa sobre todo

para módulos con APIs ricas, donde hay muchas rutinas y mucha redundancia entre ellas. En un trabajo previo [8], se propuso un enfoque automático para seleccionar un subconjunto de rutinas suficientes y no superfluas de la API de un módulo, de manera automática (*Builders*).

En este trabajo, se observa que el enfoque mencionado puede ser de utilidad para la construcción de *drivers* eficientes para *bounded model checking*, ya que permite reducir la cantidad de métodos a utilizar en los *drivers* (evitando métodos superfluos), mejorando su eficiencia, y manteniendo la capacidad de construir todos los objetos acotados posibles (por la suficiencia de los métodos elegidos). Todo esto, sin requerir trabajo adicional al usuario de la herramienta. Así, se propone una construcción de *drivers* que utilice los métodos suficientes y no superfluos seleccionados por la herramienta, y se lo compara experimentalmente con el enfoque de utilizar todos los métodos de la API (que representa el caso en que no se tiene información adicional de cuáles métodos son importantes para la construcción del *driver*). Se evaluaron ambos *drivers* en un caso de estudio (Apache NodeCachingLinkedList) para el análisis de una propiedad particular de este módulo. Los resultados muestran que aplicando solamente los métodos llamados *builders* a los *drivers* utilizados por JPF se obtiene una ganancia en tiempo y en la cantidad de las estructuras generadas para la verificación del software.

2. Ejemplo motivador

En esta sección describiremos un ejemplo que nos ayudara a explicar el trabajo realizado. Nos centramos en la estructura de datos Apache NodeCachingLinkedList (NLC) [11]. Es una implementación de listas que intenta reducir la creación de objetos y el posterior uso del *garbage collector*, manteniendo una lista de los nodos borrados en una *cache*. Es una lista doblemente encadenada y circular, esto quiere decir que cada nodo tiene una referencia al nodo siguiente y al anterior. Además cuenta con una *cache*, que es una lista simplemente encadenada que se forma con los nodos que se remueven de la lista principal. Estos nodos pueden ser reutilizados insertándolos nuevamente en la lista principal.

NCL tiene una gran cantidad de métodos en su API, pero solo unos pocos son suficientes para generar cualquier instancia finita de NCL. En la tabla 1 se encuentran todos los métodos que existen en dicha API. En primer lugar, siempre se necesita un método constructor de objetos, luego se puede observar que al menos una de las variantes del método **add** es suficiente para agregar elementos a la lista *principal*. Sin embargo, si queremos generar instancias donde la lista *cache* es no vacía, debemos utilizar alguna variante del método **remove**. El método **remove** elimina nodos de la lista *principal* y los inserta en la lista *cache*. Estos métodos (**remove** y **add**) son suficiente para crear cualquier estructura posible. También, son minimales dado que cualquiera de estos 3 métodos que no esté, implicaría que haya alguna instancia que no se va a lograr construir. Decimos que una rutina es un observador si nunca modifica los parámetros que toma. En la API de NLC hay varios métodos que son observadores, como por

No.	Return type	Method name	No.	Return type	Method name
0		NCL()	17	boolean	isEmpty()
1		NCL(int)	18	Iterator	iterator()
2		NCL(Collection)	19	int	lastIndexOf(Object)
3	boolean	add(Object)	20	ListIterator	listIterator()
4	void	add(int,Object)	21	ListIterator	listIterator(int)
5	boolean	addAll(Collection)	22	Object	remove(int)
6	boolean	addAll(int,Collection)	23	boolean	remove(Object)
7	boolean	addFirst(Object)	24	boolean	removeAll(Collection)
8	boolean	addLast(Object)	25	Object	removeFirst()
9	void	clear()	26	Object	removeLast()
10	boolean	contains(Object)	27	boolean	retainAll(Collection)
11	boolean	containsAll(Collection)	28	Object	set(int,Object)
12	boolean	equals(Object)	29	int	size()
13	Object	get(int)	30	List	subList(int,int)
14	Object	getFirst()	31	Object[]	toArray()
15	Object	getLast()	32	Object[]	toArray(Object[])
16	int	indexOf(Object)	33	String	toString()

Tabla 1. API de Apache NodeCachingLinkedList

```
(0) NodeCachingLinkedList ()
(7) addFirst ( Object )
(25) removeFirst ()
```

Figure 1.1. driverNLC

ejemplo el método `size` que no modifica la estructura del módulo. Por lo tanto, los observadores son siempre superfluos y nunca deben incluirse en un conjunto de constructores mínimos. El enfoque presentado en [8] trata de reconocerlos de antemano y los descarta de la búsqueda para reducir significativamente el espacio de búsqueda.

Se observa que cuanto más simples son los parámetros de una rutina, más fácil es usar la rutina para generar entradas en el contexto de un análisis de programa. Además, existen varios subconjuntos de métodos equivalentes para obtener los generadores de objetos. Por ejemplo, se puede lograr utilizando diferentes versiones de `add` o `remove`. El enfoque implementado en el trabajo previo le da prioridad a aquellos métodos que contienen menos cantidad de parámetros y menos complejidad de los mismo. Por ejemplo, existen diferentes versiones de métodos `add` que se encuentran desde la posición 3 a 8 en la tabla 1. La herramienta seleccionará aquellos métodos que reciben menos parámetros o con menor complejidad de los mismos.

Una vez que alimentamos con toda la API de NCL a la herramienta, nos brindó el subconjunto de métodos, mínimo y suficiente, que se encuentra en la figura 1.1, que serán necesarios para brindarle asistencia a la técnica de *bounded model checking* explicada en la sección 4

3. Generación automática de objetos

En esta sección resumiremos la herramienta implementada en un trabajo previo [8] y que utilizaremos para obtener los métodos generadores de objetos a partir de una API.

Para realizar esto, utilizaremos Algoritmos Genéticos que son algoritmos de búsqueda no exhaustivo basado en la idea de hill climbing [9]. El espacio de búsqueda de este algoritmo esta compuesto de un conjunto muy grande de individuos (posibles soluciones del problema) comúnmente llamados *cromosomas*. Estos *cromosomas* son representados como vectores donde cada posición se la denomina *gen*. El objetivo del algoritmo es buscar una característica deseada en la población (conjunto de cromosomas). Esta búsqueda siempre es bajo la condición de que el algoritmo mantenga elitismo (guarde siempre al mejor individuo de la población). Para lograr esto se requiere una función de ajuste la cual asigna un valor a cada cromosoma posible. Para más detalles sobre estos algoritmos, recomendamos la lectura de [5].

En el caso de nuestro problema, los individuos son representados por los métodos de la API a analizar. Estos cromosomas lo representamos como vectores booleanos, que tendrán tamaño n que es el número de métodos de la API. Por ende, la posición i -th será verdadera si y solo si el cromosoma contiene el método i -th de la API. De esta manera el cromosoma que representa al subconjunto de métodos suficientes y minimales de esta API (ver figura 1.1), sera un vector que tendrá el valor 1 (valor *true*) en la posición 0, 7 y 25. La función de ajuste de nuestro problema computa la cantidad acotada de objetos que se puede construir usando la combinación de métodos que están presentes en el cromosoma a evaluar. Cuanto más objetos pueda construir un cromosoma, más alto será el número que retorne la función de ajuste. Para esto necesitamos un generador exhaustivo limitado para el conjunto de métodos. El límite k representa el número máximo de objetos que se pueden crear para cada combinación de rutinas y el número máximo de valores primitivos disponibles (por ejemplo, enteros de 0 a $k - 1$). Para este propósito, se desarrollo un prototipo que modifica la herramienta Randoop. Primero, se modifco Randoop para que funcione con un conjunto fijo de valores primitivos (enteros de 0 a $k - 1$). Luego, canonizamos los objetos generados por la ejecución de cada secuencia, y descartamos la secuencia si algún objeto tiene un índice igual o mayor que k . Para mas información acerca de la herramienta Randoop, recomendamos leer [7]. Para obtener buenos conjuntos de generadores de objetos, se realizó dos mejoras a la función de ajuste, por un lado, cuando hay dos conjuntos suficientes de generadores, siempre se elige el conjunto con menor cantidad de métodos y de esta manera evitamos incorporar métodos superfluos. Por otro lado, aquellos generadores de objetos con más parámetros, o con parámetros más complejos tienen un impacto negativo respecto de los generadores con parámetros más simples. Luego de esta explicación, la función de ajuste queda definida de la siguiente manera:

$$f(M) = \#cantidadDeObjetos(M) + \left(\frac{w_1 * \left(1 - \frac{\#M}{\#MT}\right) + w_2 * \left(1 - \frac{(\#PP(M) + w_3 * PR(M))}{(\#PP(MT) + w_3 * PR(MT))}\right)}{w_1 + w_2} \right)$$

Dado un cromosoma, M es el conjunto de métodos disponible en dicho individuo y MT es el conjunto de métodos que existen en la API. *cantidadDeObjetos* hace referencia a las cantidad de objetos que son generados por la herramienta Randoop personalizada para el enfoque planteado en el trabajo previo. La parte derecha de la sumatoria retorna un valor entre 0 y 1 que permite penalizar a los métodos que generan igual número de objetos de acuerdo a lo explicado anteriormente. En el dividendo de la fórmula general, el primer sumando penaliza los conjuntos que contengan un mayor número de método. Esto lo realiza restándole a 1 el resultado de la división entre el número de métodos en MT y el número de métodos en M . La constante $w1$ permite aumentar/disminuir el peso de este sumando con respecto al otro. En el segundo sumando se penaliza los conjuntos de métodos con parámetros más complejos. Similar a $w1$, la constante $w2$ sirve para darle más o menos peso a esta suma. $PP()$ es el número de parámetros primitivos en los métodos que contiene el subconjunto pasado como parámetro de esta función. También, cada parámetro de tipo referencia se le agrega una constante $w3$. $RP()$ es el número de parámetros de tipo referencia que hay en los métodos que contiene el subconjunto pasado como parámetro. Intuitivamente, el lado derecho del sumando computa el radio entre el número de parámetros en M (con el peso agregado de los parámetros de referencia) con el número de parámetros en MT (también ponderado). El resultado de todo esto, se resta a 1. Finalmente, dividimos esto entre $w1 + w2$ para obtener el número deseado en el intervalo $[0,1]$.

En la evaluación experimental realizada en el trabajo previo de esta herramienta, establecimos $w1 = 2$, $w2 = 1$, $w3 = 2$. Estos valores fueron lo suficientemente buenos para que la herramienta produjera conjuntos de constructores suficientes y mínimos en todos los casos de estudios de la herramienta.

4. Construcción de drivers eficientes usando *builders*

Como se discutió anteriormente, Java PathFinder (JPF) [10] es una herramienta de verificación automática de programas, que permite implementar diversos tipos análisis. En este trabajo la utilizaremos para realizar (bounded) model checking de software, es decir, para explorar todas las posibles ejecuciones de un programa con entradas de tamaño acotado. Las propiedades a verificar en JPF pueden especificarse usando la construcción `assert` de Java. `assert` evalúa una condición lógica y produce una falla en el programa en caso de que esta sea falsa en la ejecución corriente. En caso de que la condición sea verdadera, `assert` no tiene ningún efecto sobre la ejecución. Por ejemplo, el siguiente código especifica la propiedad de que para cualquier lista de entrada `t`, insertar al inicio un valor entero `v` (`addFirst(v)`, con `v` entre 0 y `b`, con `b` el límite máximo en la cantidad de enteros a analizar, ver más adelante) que no pertenece a la lista (ver explicación de `Verify.ignoreIf` abajo), y luego eliminar el primer elemento de la lista (el valor agregado recientemente, ejecutando `removeFirst`), resulta en una lista con una cantidad de elementos igual a la original (antes de ejecutar `addFirst` y `removeFirst`; notar que la cantidad de elementos previos se almacena en `oldSize` al inicio).

Algorithm 1 Propiedad a chequear

```

oldSize ← t.size()
value ← Verify.getInt(0, b)
Verify.ignoreIf(t.contains(v))
t.addFirst(value)
t.removeFirst()
assert oldSize = t.size() : "different size";

```

Este código utiliza dos directivas muy importantes de JPF:

- `Verify.ignoreIf(condicion)`: Evita explorar ejecuciones cuyo estado no cumplen con el predicado `condicion`. En el programa anterior, permite ignorar todas las ejecuciones en las que la lista contiene el valor `v` a insertar. Esto es similar a agregar una precondición a la propiedad a verificar.
- `int i = Verify.getInt(min,max)`: Esta construcción de JPF explora todas las posibles ejecuciones del programa que resultan de asignar a `i` cualquier valor entre `min` y `max`. Esto significa que JPF ejecutará el código luego de esta instrucción con `i=min`, con `i=min+1`, ..., `i=max`. Esto introduce no determinismo, ya que en principio no hay garantías del orden en que se asignarán valores a `i` (aunque JPF tiene opciones para elegir un orden de ejecución particular). En nuestro código previo, esto permite explorar todas las ejecuciones de nuestra propiedad con valores de `v` entre 0 y `b`.

Para poder realizar el análisis de esta propiedad, es necesario proveer a JPF los mecanismos necesarios para generar las listas de entrada para el parámetro `t`. Esto es necesario para cualquier tipo estructurado; notar que los valores enteros pueden generarse fácilmente con `Verify.getInt`. Nuestro objetivo es construir todas las listas posibles de tamaño máximo `b`, que almacenan elementos enteros entre 0 y `b` (verificación exhaustiva acotada). Notar que `b` es un parámetro del algoritmo, dado por el usuario. La generación exhaustiva acotada de estructuras para el parámetro `t` se implementa en el *driver* del algoritmo 3. Al inicio, el *driver* selecciona la cantidad de métodos a ejecutar, `nExec`, un número entre 0 y `b`. Como los métodos considerados agregan a lo sumo un elemento a la lista, ejecutar un máximo de `b` métodos resulta en listas de tamaño a lo sumo `b` (ejemplo, ejecutar `b` veces `addFirst`). Cada iteración del ciclo corresponde a la ejecución de un único método, seleccionado también de manera no determinística entre todos los disponibles. En el caso en que el usuario no conoce el conjunto de *builders* (y no desea realizar el difícil trabajo de seleccionarlos manualmente), la solución más segura para evitar descartar métodos importantes es utilizar todos los métodos disponibles en el módulo, como se muestra en el algoritmo 2. En el cuerpo del ciclo cada método tiene asignado un único entero, y se elige no determinísticamente un entero entre 0 y la cantidad de métodos para seleccionar el método a ejecutar la iteración corriente. Por ejemplo, si `methodN=3` se ejecuta el método `add`. Es fácil ver que la cantidad de ejecuciones posibles a explorar por JPF crece exponencialmente con la cantidad de métodos disponibles (por cada método que se ejecuta en una iteración, hay `m` posibles métodos para ejecutar en la iteración siguiente).

Si bien evitar este crecimiento exponencial no es posible en muchos casos, en este trabajo se propone utilizar sólo los builders detectados por el algoritmo genético explicado anteriormente para construir *drivers*, para acelerar la verificación de propiedades en JPF y mejorar su escalabilidad en casos que ocurren típicamente en la práctica (como el caso de `NodeCachingLinkedList` presentado aquí). Como se muestra en la figura 1.1, sólo 3 métodos conforman un conjunto suficiente y minimal para la construcción de *drivers* para `NodeCachingLinkedList`. Utilizando sólo esos métodos, obtenemos el *driver* del algoritmo 3, que genera exactamente los mismos objetos de `NodeCachingLinkedList` que el anterior (porque los builders son suficientes y minimales), y por lo tanto JPF explora las mismas ejecuciones de la propiedad con ambos *drivers*. Sin embargo, como se discute en la sección que sigue, eliminar métodos innecesarios del *driver* produce ganancias sustanciales en tiempo y escalabilidad en el análisis de nuestra propiedad de ejemplo usando JPF.

Algorithm 2 Driver API

```

t ← new NLC()
nExec ← Verify.getInt(0, bound)
for i = 0...nExec do
  methodN ← Verify.getInt(0, nMet -1);
  switch ( methodN )
    case 0:
      t.clear()
    case 1:
      t.removeFirst();
    case 2:
      t.addLast(Obj)
    case 3:
      t.add(t)
    ...
    case 28:
      t.getLast()
    case 29:
      t.get(Verify.getInt(0, bound))
    case 30:
      t.set(Verify.getInt(0, bound),Obj)
  end switch
end for

```

Algorithm 3 Driver con Herramienta

```

t ← new NLC()
nExec ← Verify.getInt(0, bound)
for i = 0...nExec do
  methodN ← Verify.getInt(0, nMet -1);
  switch ( methodN )
    case 0:
      t.addFirst(Obj)
    case 1:
      t.removeFirst()
  end switch
end for

```

5. Resultado Experimental

En esta sección nosotros realizaremos una evaluación de cómo se comporta la herramienta, implementada en el trabajo previo ya citado, para calcular los generadores de objetos en la estructura `Apache NodeCachingLinkedList`. Luego, realizaremos una comparación para demostrar como se puede mejorar (a partir de la obtención de métodos generadores) la eficiencia de una técnica de *bounded model checking*, más precisamente JPF. Todos los experimentos fueron corridos en máquinas con cuatro core Intel i7-6700 3.4GHz con 8GB de RAM, con el sistema operativo GNU/Linux. La primera parte del análisis consiste en chequear

la herramienta comentada en la sección 3. Pudimos examinar que los constructores identificados fueron los ya mencionados en la figura 1.1. La API usada como ejemplo contiene un total de 34 métodos, y el tiempo promedio de las 5 corridas fue de 1744 segundos. Inspeccionamos manualmente los resultados y descubrimos que el conjunto de generadores identificados automáticamente (*builders*) eran en todos los casos suficientes (todos los objetos factibles para la estructura pueden construirse usando los constructores) y mínimos (no contienen métodos superfluos).

La segunda parte de la evaluación se basa en la utilización de los métodos generadores obtenidos (*builders*), en el contexto de un análisis de programa, mas específicamente en la herramienta Java PathFinder. Estas rutinas generadoras pueden usarse, por ejemplo, para la creación de *drivers* útiles para guiar el análisis. Como ya se mostro anteriormente, el primer gran aporte a JPF es se encuentra en la escritura de los *drivers*. Queda claro que programar el *driver* a partir de todos los métodos de la API es una tárea compleja que requiere mas conocimiento del modulo bajo análisis como muestra el algoritmo 2. Para compensar esto, mostramos que el algoritmo que se utiliza como *driver* en caso de obtener previamente los *builders* es más sencillo y entendible para el usuario.

Luego, se realizo una comparación para determinar el tiempo (en segundos) que le lleva al verificador de software en comprobar una propiedad deseada, como la que se detalla en el algoritmo 1. En la tabla 2 se expone el tiempo que consume el verificador de software para diferentes *scopes* (tamaño maximo de las listas). Recordemos que en *bounded model checking* es necesario brindarle una cota, que cuanto más grande es, mas en profundidad podremos analizar las estructura y por ende, permite tener más posibilidad de detectar *bugs*, tal cual lo explicamos en secciones anteriores. Se observa en la tabla 2 que luego de un cota de tamaño 8, el *driver* sin la previa obtencion de los *builders* se queda sin tiempo (**TO**), con un presupuesto de tiempo de 1800 segundos, mientras que el *driver* creado a partir de la obtencion de los *builders* verifica la propiedad en menor tiempo.

Scope	Driver S/Builders	Driver C/Builders
3	9	1
5	250	3
8	TO	23
11	TO	185
13	TO	881

Tabla 2. Resultados de verificar la propiedad con JPF (en segundos)

6. Conclusiones

En este trabajo, utilizamos una herramienta que detecta automaticamente conjuntos de métodos generadores de objetos desde una la API para construir *drivers* que mejoren la eficiencia en técnicas de *bounded model checking*. La técnica propuesta fue analizada en un caso de estudio utilizando una implementación de listas con caché, y muestra resultados muy promisorios en cuanto a los ganancias en la eficiencia y escalabilidad obtenidos del análisis de una propiedad usando JPF.

En este trabajo los *drivers* se construyeron manualmente a partir de los *builders* que nos dió la herramienta presentada en [8]. Como un desafío a resolver es la idea de automatización de todo el proceso de la construcción de *drivers*. Esto quiere decir, lograr generar de manera automática los *drivers* a partir de la herramienta, la cual, de manera automática, genera los *builders* de un módulo. Lo que resta resolver, es realizar un trabajo y una inspección en la técnica de *bounded model checking* más precisamente en JPF para lograr unir ambas herramientas. Vale destacar, que dentro de los desafíos que se propone está el de realizar pruebas en nuevos casos de estudios.

Referencias

1. Abad, P., Aguirre, N., Bengolea, V.S., Ciolek, D., Frias, M.F., Galeotti, J.P., Maibaum, T., Moscato, M.M., Rosner, N., Vissani, I.: Improving test generation under rich contracts by tight bounds and incremental SAT solving. In: Sixth IEEE International Conference on Software Testing, Verification and Validation, ICST 2013, Luxembourg, Luxembourg, March 18–22, 2013. pp. 21–30 (2013)
2. Calcagno, C., Distefano, D., O’Hearn, P.W., Yang, H.: Compositional shape analysis by means of bi-abduction. *J. ACM* **58**(6), 26:1–26:66 (Dec 2011)
3. Clarke, E., Kroening, D., Lerda, F.: A tool for checking ANSI-C programs. In: Tools and Algorithms for the Construction and Analysis of Systems (TACAS 2004). Lecture Notes in Computer Science, vol. 2988, pp. 168–176. Springer (2004)
4. Fraser, G., Arcuri, A.: Evosuite: Automatic test suite generation for object-oriented software. In: Proceedings of the 19th ACM SIGSOFT Symposium and the 13th European Conference on Foundations of Software Engineering. pp. 416–419. ESEC/FSE ’11, ACM, New York, NY, USA (2011)
5. Goldberg, D.E.: Genetic Algorithms in Search, Optimization and Machine Learning. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 1st edn. (1989)
6. Huang, W., Milanova, A., Dietl, W., Ernst, M.D.: Reim & reiminfer: Checking and inference of reference immutability and method purity. In: Proceedings of the ACM International Conference on Object Oriented Programming Systems Languages and Applications. pp. 879–896. OOPSLA ’12, ACM, New York, NY, USA (2012)
7. Pacheco, C., Ernst, M.D.: Randoop: Feedback-directed random testing for java. In: Companion to the 22Nd ACM SIGPLAN Conference on Object-oriented Programming Systems and Applications Companion. pp. 815–816. OOPSLA ’07, ACM, New York, NY, USA (2007)
8. Ponzio, P., Bengolea, V.S., Politano, M., Aguirre, N., Frias, M.F.: Automatically identifying sufficient object builders from module apis. In: Fundamental Approaches to Software Engineering - 22nd International Conference, FASE 2019, Held as Part of the European Joint Conferences on Theory and Practice of Software, ETAPS 2019, Prague, Czech Republic, April 6–11, 2019, Proceedings (2019)
9. Russell, S., Norvig, P.: Artificial Intelligence: A Modern Approach. Prentice Hall Press, Upper Saddle River, NJ, USA, 3rd edn. (2009)
10. Visser, W., Mehltitz, P.: Model checking programs with java pathfinder. In: Proceedings of the 12th International Conference on Model Checking Software. pp. 27–27. SPIN’05, Springer-Verlag, Berlin, Heidelberg (2005)
11. Website of the Apache Collections library. <https://commons.apache.org/proper/commons-collections/>

Hacia un Sistema I Polimórfico

Cristian Sottile^{1,2,3}, Alejandro Díaz-Caro^{2,4}, and Pablo E. Martínez López²

¹ LIFIA, Facultad de Informática, Universidad Nacional de La Plata
La Plata, Buenos Aires, Argentina

² Departamento de Ciencia y Tecnología, Universidad Nacional de Quilmes
Bernal, Buenos Aires, Argentina

³ Comisión de Investigaciones Científicas de la Provincia de Buenos Aires (CICPBA)
La Plata, Buenos Aires, Argentina

⁴ Instituto de Ciencias de la Computación (CONICET-UBA)
Ciudad Autónoma de Buenos Aires, Argentina

Resumen Sistema I es un lambda cálculo simplemente tipado con pares, extendido con una teoría ecuacional obtenida a partir de los isomorfismos de tipos existentes entre los tipos simples con pares. En este trabajo en progreso proponemos una extensión de Sistema I hacia tipos polimórficos, añadiendo al sistema de tipos tanto el cuantificador universal como sus isomorfismos relacionados.

Keywords: Lambda cálculo; Teoría de tipos; Isomorfismos de tipos; Polimorfismo

1. Introducción

Dos tipos A y B se consideran isomorfos si existen dos funciones $f : A \Rightarrow B$ y $g : B \Rightarrow A$ tales que la composición $g \circ f : A \Rightarrow A$ es semánticamente equivalente a la identidad en A , y la composición $f \circ g : B \Rightarrow B$ es semánticamente equivalente a la identidad en B . Di Cosmo y sus colaboradores caracterizaron los isomorfismos existentes en diferentes sistemas: tipos simples, tipos simples extendidos con pares, polimorfismo, y otros (ver [2] para una referencia completa). Utilizando dicha caracterización, Díaz-Caro y Dowek [3] definieron Sistema I, un lambda cálculo con pares simplemente tipado donde los tipos isomorfos son considerados iguales. Así, si A y B son isomorfos, todo término de tipo A puede ser utilizado como un término de tipo B . Por ejemplo, el isomorfismo conocido como currificación dice que $(A \wedge B) \Rightarrow C$ es isomorfo a $A \Rightarrow B \Rightarrow C$, y por lo tanto una función esperando un par puede ser utilizada como una función esperando los argumentos por separado. Para esto Sistema I incluye a su vez una equivalencia entre términos: $t\langle r, s \rangle \rightleftharpoons trs$, ya que si t espera un par de elementos, también puede tomar uno a uno por separado.

Díaz-Caro y Martínez López [4] implementaron una versión preliminar de Sistema I extendido con un operador de recursión en Haskell, lo que produjo un lenguaje con características muy particulares; por ejemplo, utilizando el isomorfismo existente entre $A \Rightarrow (B \wedge C)$ y $(A \Rightarrow B) \wedge (A \Rightarrow C)$, se puede construir un

término que sea la proyección de una función que compute un par de elementos, para obtener mediante evaluación una de las funciones más simples que computan solo uno de los elementos del par, sin haber aplicado previamente la función y descartando así el código que computa el resultado que no nos interesa.

En este trabajo proponemos una extensión de Sistema I a tipos polimórficos, agregando los isomorfismos existentes en presencia del cuantificador universal, con el fin de enriquecer el lenguaje.

Este es un trabajo en progreso, y como tal, las propiedades formales del sistema se dejan para trabajo futuro.

El paper está organizado de la siguiente manera. En la Sección 2 presentamos Sistema I, tal y como fue definido por Díaz-Caro y Dowek [3]. En la Sección 3 presentamos nuestro aporte, que es la versión polimórfica de Sistema I. Finalmente en la Sección 4 damos algunas conclusiones y una descripción de los trabajos en progreso y a futuro.

2. Sistema I

2.1. Definiciones

Sistema I es una extensión al Lambda Cálculo Simplemente Tipado con Pares (al que llamaremos de forma abreviada Cálculo Simple). La sintaxis de tipos no se ve modificada, y Sistema I conserva los mismos tipos que el Cálculo Simple:

$$A ::= \tau \mid A \Rightarrow A \mid A \wedge A$$

Sin embargo, la extensión de Sistema I con respecto al Cálculo Simple consiste en agregar la siguiente regla de tipado:

$$[A \equiv B] \frac{t : A}{t : B} (\equiv)$$

válida para todo par de tipos A y B isomorfos. Este agregado, no menor, trae aparejado la modificación de la semántica operacional del cálculo, ya que, por ejemplo, una abstracción de tipo $A \Rightarrow (B \wedge C)$ podrá ser tipada con el tipo del par de funciones $(A \Rightarrow B) \wedge (A \Rightarrow C)$, y por lo tanto podremos proyectar una función a partir de una abstracción. Es decir, las abstracciones pueden convertirse en pares.

Los isomorfismos considerados son los cuatro existentes para el Cálculo Simple (cf. [2]), y se detallan en la Figura 1.

Los isomorfismos (1) y (2) definen la conmutatividad y asociatividad de la conjunción, el isomorfismo (3) es la distributividad de la implicación con respecto a la conjunción, y el isomorfismo (4) es la currificación.

La regla de tipado (\equiv) , para los isomorfismos considerados, induce ciertas equivalencias entre términos. La conmutatividad y asociatividad de la conjunción implican que el par de términos $\langle t, r \rangle$ y el par de términos $\langle r, t \rangle$ son indistinguibles, ya que ambos son tipados con $A \wedge B$ y con $B \wedge A$, por lo tanto, consideraremos que dichos términos son equivalentes, de la misma manera que

$$A \wedge B \equiv B \wedge A \quad (1)$$

$$A \wedge (B \wedge C) \equiv (A \wedge B) \wedge C \quad (2)$$

$$A \Rightarrow (B \wedge C) \equiv (A \Rightarrow B) \wedge (A \Rightarrow C) \quad (3)$$

$$(A \wedge B) \Rightarrow C \equiv A \Rightarrow B \Rightarrow C \quad (4)$$

Figura 1. Isomorfismos del Cálculo Simple

$\langle t, \langle r, s \rangle \rangle$ y $\langle \langle t, r \rangle, s \rangle$ son indistinguibles. Esta modificación implica que la proyección usual de pares definida con respecto a la posición (e.g. $\pi_i \langle t_1, t_2 \rangle \hookrightarrow t_i$) no está bien definida para nuestro sistema, y por consiguiente, Sistema I utiliza proyecciones con respecto al tipo:

$$\text{Si } t : A \text{ entonces } \pi_A \langle t, r \rangle \hookrightarrow t$$

Esta regla torna Sistema I en un sistema no determinista, ya que si t y r tienen tipo A , $\pi_A \langle t, r \rangle$ podrá reducir indistintamente a t o a r . Este no determinismo no presenta un problema mayor ya que, en presencia de la propiedad de preservación de tipos y si se piensa a Sistema I como un sistema de pruebas, implica que el sistema identifica las diferentes pruebas de proposiciones isomorfas (como una forma de *proof-irrelevance*). Por otra parte, si se considera a Sistema I como un lenguaje de programación, el determinismo se puede recuperar codificando los términos t y r del mismo tipo como $\lambda x^B.t$ y $\lambda x^C.r$, con $B \neq C$. Por lo tanto dicho no determinismo se considera una característica deseable y no un problema (ver discusión en [3]).

La sintaxis de términos entonces es la sintaxis del Cálculo Simple, excepto porque las proyecciones dependen de los tipos (lo que también implica que todos los términos son tipados, y por lo tanto utilizamos la notación *à la Church*, o sea con las variables explicitando su tipo):

$$t \quad := \quad x^A \mid \lambda x^A.t \mid tt \mid \langle t, t \rangle \mid \pi_A t$$

Se considera por convención que para todo tipo A existe un conjunto infinito de variables de tipo A denotado por \mathcal{V}_A tal que si $A \equiv B$ entonces $\mathcal{V}_A \equiv \mathcal{V}_B$, y si $A \neq B$ entonces $\mathcal{V}_A \cap \mathcal{V}_B = \emptyset$. Esto tiene como consecuencia que no hay variables diferentes con el mismo nombre (cf. [1, Página 26]).

Se definen de la forma usual las funciones FV_t (variables de término libres en términos), FTV_t (variables de tipo libres en términos), FTV_A (variables de tipo libres en tipos), y la substitución de términos en términos ($[x := r]t$).

El sistema de tipos de Sistema I es el sistema de tipos del Cálculo Simple con la modificación correspondiente a la proyección con respecto al tipo y la adición de la regla (\equiv), mencionada al principio de esta sección. Se detalla el sistema completo en la Figura 2. La presentación elegida para las reglas de tipado es sin un contexto específico (a diferencia de como es usual encontrar en los juicios de tipado más tradicionales, $\Gamma \vdash t : A$), ya que toda variable lleva su propio tipo y por lo tanto el contexto sería redundante.

$$\begin{array}{c}
\frac{[x \in \mathcal{V}_A]}{x : A} \text{ (ax)} \quad \frac{[A=B]}{t : A \quad t : B} \text{ (}\equiv\text{)} \\
\frac{t : B}{\lambda x^A . t : A \Rightarrow B} \text{ (}\Rightarrow_i\text{)} \quad \frac{t : A \Rightarrow B \quad r : A}{tr : B} \text{ (}\Rightarrow_e\text{)} \quad \frac{t : A \quad r : B}{\langle t, r \rangle : A \wedge B} \text{ (}\wedge_i\text{)} \quad \frac{t : A \wedge B}{\pi_A(t) : A} \text{ (}\wedge_e\text{)}
\end{array}$$

Figura 2. Reglas de tipado de Sistema I

De la misma manera que los isomorfismos (1) y (2) implican la conmutatividad y asociatividad de pares en los términos, así como un cambio en la proyección, el isomorfismo (3) implica que ciertos términos deben ser identificados: una abstracción que retorna un par de funciones se identifica con un par de abstracciones y un par aplicado a un término, distribuye dicha aplicación:

$$\begin{aligned}
\lambda x^A . \langle t, r \rangle &\rightleftharpoons \langle \lambda x^A . t, \lambda x^A . r \rangle \\
\langle t, r \rangle s &\rightleftharpoons \langle ts, rs \rangle
\end{aligned}$$

donde \rightleftharpoons es un símbolo simétrico.

Finalmente, el isomorfismo (4) induce la siguiente identificación de términos:

$$t \langle r, s \rangle \rightleftharpoons trs$$

sin embargo, esta identificación produce ambigüedad con la beta-reducción. Por ejemplo, si $t : A$ y $r : B$, el término $(\lambda x^{A \wedge B} . s) \langle t, r \rangle$ puede beta-reducir a $[x := \langle t, r \rangle]s$, pero a su vez este término será equivalente a $(\lambda x^{A \wedge B} . s)tr$, que beta-reduce a $([x := t]s)r$. Para evitar dicha ambigüedad, la beta-reducción deberá realizarse sólo en el caso en que el tipo esperado en la abstracción y el tipo del argumento coincidan:

$$\text{Si } r : A \text{ entonces } (\lambda x^A . t)r \hookrightarrow [x := r]t$$

Los isomorfismos (1) y (2) definen la conmutatividad y asociatividad de la conjunción, el isomorfismo (3) es la distributividad de la implicación con respecto a la conjunción, y el isomorfismo (4) es la currificación.

Resumiendo, la semántica operacional de Sistema I viene dada por la relación \hookrightarrow la cual se toma módulo la relación simétrica \rightleftharpoons . Es decir, se considera la relación $\rightarrow := \rightleftharpoons^* \circ \hookrightarrow \circ \rightleftharpoons^*$, donde \rightleftharpoons^* es la clausura transitiva de \rightleftharpoons . Como es usual, notaremos también con \rightarrow^* a la clausura reflexotransitiva de \rightarrow . Las reglas de ambas relaciones (\hookrightarrow y \rightleftharpoons) se detallan en la Figura 3.

2.2. Ejemplos

En esta sección mostramos algunos ejemplos para discutir la necesidad de las reglas presentadas.

El primer ejemplo muestra el uso de la equivalencia entre términos para producir aplicaciones no tradicionales, que no es posible construir en el Cálculo Simple. En particular, la función $apply = \lambda f^{A \Rightarrow B} . \lambda x^A . fx$ se puede aplicar a un par, e.g. $\langle g, t \rangle$ con $g : A \Rightarrow B$ y $t : A$, ya que la siguiente derivación de tipo es válida, debido al isomorfismo (4):

$$\begin{array}{l}
\langle r, s \rangle \rightleftharpoons \langle s, r \rangle \quad (\text{COMM}) \\
\langle r, \langle s, t \rangle \rangle \rightleftharpoons \langle \langle r, s \rangle, t \rangle \quad (\text{ASSO}) \\
\lambda x^A. \langle r, s \rangle \rightleftharpoons \langle \lambda x^A. r, \lambda x^A. s \rangle \quad (\text{DIST}_\lambda) \\
\langle r, s \rangle t \rightleftharpoons \langle rt, st \rangle \quad (\text{DIST}_{\text{app}}) \\
r \langle s, t \rangle \rightleftharpoons rst \quad (\text{CURRY}) \\
\\
\text{Si } s : A, (\lambda x^A. r) s \hookrightarrow [x := s] r \quad (\beta_\lambda) \\
\text{Si } r : A, \pi_A \langle r, s \rangle \hookrightarrow r \quad (\pi) \\
\\
\frac{t \rightleftharpoons r}{\lambda x^A. t \rightleftharpoons \lambda x^A. r} \quad \frac{t \rightleftharpoons r}{ts \rightleftharpoons rs} \quad \frac{t \rightleftharpoons r}{st \rightleftharpoons sr} \quad \frac{t \rightleftharpoons r}{\langle t, s \rangle \rightleftharpoons \langle r, s \rangle} \quad \frac{t \rightleftharpoons r}{\langle s, t \rangle \rightleftharpoons \langle s, r \rangle} \quad \frac{t \rightleftharpoons r}{\pi_A t \rightleftharpoons \pi_A r} \\
\frac{t \hookrightarrow r}{\lambda x^A. t \hookrightarrow \lambda x^A. r} \quad \frac{t \hookrightarrow r}{ts \hookrightarrow rs} \quad \frac{t \hookrightarrow r}{st \hookrightarrow sr} \quad \frac{t \hookrightarrow r}{\langle t, s \rangle \hookrightarrow \langle r, s \rangle} \quad \frac{t \hookrightarrow r}{\langle s, t \rangle \hookrightarrow \langle s, r \rangle} \quad \frac{t \hookrightarrow r}{\pi_A t \hookrightarrow \pi_A r}
\end{array}$$

Figura 3. Relaciones que definen la semántica operacional de Sistema I

$$\frac{\lambda f^{A \Rightarrow B}. \lambda x^A. fx : (A \Rightarrow B) \Rightarrow A \Rightarrow B \quad (\equiv) \quad \frac{g : A \Rightarrow B \quad t : A}{\langle g, t \rangle : (A \Rightarrow B) \wedge A} (\wedge_i)}{\lambda f^{A \Rightarrow B}. \lambda x^A. fx : ((A \Rightarrow B) \wedge A) \Rightarrow B \quad (\Rightarrow_e)} \quad \frac{}{(\lambda f^{A \Rightarrow B}. \lambda x^A. fx) \langle g, t \rangle : B}$$

La reducción ocurre de la siguiente manera:

$$(\lambda f^{A \Rightarrow B}. \lambda x^A. fx) \langle g, t \rangle \rightleftharpoons_{(\text{CURRY})} (\lambda f^{A \Rightarrow B}. \lambda x^A. fx) g t \hookrightarrow_{\beta}^2 g t$$

El segundo ejemplo muestra que la misma aplicación puede usarse de otras formas. Por ejemplo, $(\lambda f^{A \Rightarrow B}. \lambda x^A. fx) t g$ también está bien tipado, usando los isomorfismos (1) y (4), y reduce a $g t$, como se puede observar en la siguiente reducción.

$$\begin{aligned}
(\lambda f^{A \Rightarrow B}. \lambda x^A. fx) t g &\rightleftharpoons_{(\text{CURRY})} (\lambda f^{A \Rightarrow B}. \lambda x^A. fx) \langle t, g \rangle \\
&\rightleftharpoons_{(\text{COMM})} (\lambda f^{A \Rightarrow B}. \lambda x^A. fx) \langle g, t \rangle \rightarrow^* g t
\end{aligned}$$

También la función $\text{apply}' = \lambda x^{(A \Rightarrow B) \wedge A}. \pi_{A \Rightarrow B}(x) \pi_A(x)$ puede aplicarse a $g : A \Rightarrow B$ y $t : A$ como si estuviera currificada:

$$\begin{aligned}
(\lambda x^{(A \Rightarrow B) \wedge A}. \pi_{A \Rightarrow B}(x) \pi_A(x)) g t &\rightleftharpoons_{(\text{CURRY})} \\
(\lambda x^{(A \Rightarrow B) \wedge A}. \pi_{A \Rightarrow B} x \pi_A x) \langle g, t \rangle &\hookrightarrow_{\beta} \pi_{A \Rightarrow B} \langle g, t \rangle \pi_A \langle g, t \rangle \hookrightarrow_{\pi}^2 g t
\end{aligned}$$

Hemos mostrado diferentes maneras isomorfas de aplicar g a t mediante apply . De igual forma sucede con los pares y otras combinaciones.

Otro ejemplo que resulta de interés es el mencionado en la Sección 2.1: una función que devuelve pares se puede proyectar. Supongamos que tenemos el término proj_f definido como $\pi_{A \Rightarrow B}(\lambda x^A. \langle t, r \rangle)$, con $t : B$ y $r : C$; en primer lugar, es tipable, utilizando el isomorfismo (3):

$$\begin{aligned}
A \wedge B &\equiv B \wedge A & (1) \\
A \wedge (B \wedge C) &\equiv (A \wedge B) \wedge C & (2) \\
A \Rightarrow (B \wedge C) &\equiv (A \Rightarrow B) \wedge (A \Rightarrow C) & (3) \\
(A \wedge B) \Rightarrow C &\equiv A \Rightarrow B \Rightarrow C & (4) \\
\forall X. \forall Y. A &\equiv \forall Y. \forall X. A & (5) \\
\text{si } X \notin FTV_A(A) \quad \forall X. (A \Rightarrow B) &\equiv A \Rightarrow \forall X. B & (6) \\
\forall X. (A \wedge B) &\equiv \forall X. A \wedge \forall X. B & (7)
\end{aligned}$$

Figura 4. Isomorfismos de Sistema F con pares considerados

$$\frac{\lambda x^A. \langle t, r \rangle : A \Rightarrow (B \wedge C)}{\lambda x^A. \langle t, r \rangle : (A \Rightarrow B) \wedge (A \Rightarrow C)} (\equiv)$$

$$\frac{\lambda x^A. \langle t, r \rangle : (A \Rightarrow B) \wedge (A \Rightarrow C)}{\pi_{A \Rightarrow B}(\lambda x^A. \langle t, r \rangle) : A \Rightarrow B} (\wedge_e)$$

La reducción se produce como sigue:

$$proj_f = \pi_{A \Rightarrow B}(\lambda x^A. \langle t, r \rangle) \xrightarrow{(\text{DIST}_\lambda)} \pi_{A \Rightarrow B}(\lambda x^A. t, \lambda x^A. r) \xrightarrow{\hookrightarrow_\pi} \lambda x^A. t$$

Observar que la función es proyectada a pesar de no estar aplicada, devolviendo una función.

3. Sistema I Polimórfico

3.1. Definiciones

En esta sección describimos la extensión polimórfica a Sistema I. La sintaxis de tipos es extendida con el cuantificador universal, obteniendo los tipos de Sistema F [5] con pares:

$$A ::= X \mid A \Rightarrow A \mid A \wedge A \mid \forall X. A$$

Además consideramos también algunos de los isomorfismos que surgen como consecuencia de este nuevo símbolo (también caracterizados por Di Cosmo y sus colaboradores [2]). Los isomorfismos considerados se detallan en la Figura 4. Los primeros cuatro son los detallados en la sección precedente, y por lo tanto Sistema I Polimórfico incluye las mismas relaciones entre términos que las de Sistema I. El isomorfismo (5) nos indica que de alguna manera deberemos considerar la conmutatividad de los argumentos de una aplicación de tipos. Sin embargo, dicha conmutatividad haría que la aplicación del término a un tipo no sólo sea no determinista, sino que además no se podría probar la preservación de tipos. Por ejemplo, $(\lambda X. \lambda Y. \lambda x^X. x)[A][B]$ reduce a $\lambda x^A. x$, pero si se pueden conmutar los argumentos, también reduciría a $\lambda x^B. x$, para tipos A y B cualesquiera. Por consiguiente, agregamos una etiqueta a la aplicación de tipos para indicar a qué variable corresponde. Por ejemplo, el término anterior se escribirá

$$(\lambda X. \lambda Y. \lambda x^X. x)[A_X][B_Y]$$

$$\begin{array}{c}
\frac{[x \in \mathcal{V}_A]}{x : A} \text{ (ax)} \quad \frac{[A=B]}{t : A} \text{ (\equiv)} \\
\frac{t : B}{\lambda x^A . t : A \Rightarrow B} \text{ (\Rightarrow_i)} \quad \frac{t : A \Rightarrow B \quad r : A}{tr : B} \text{ (\Rightarrow_e)} \quad \frac{t : A \quad r : B}{\langle t, r \rangle : A \wedge B} \text{ (\wedge_i)} \quad \frac{t : A \wedge B}{\pi_A(r) : A} \text{ (\wedge_e)} \\
\frac{[X \notin FTV_i(t)]}{\Lambda X . t : \forall X . B} \text{ (\forall_i)} \quad \frac{t : \forall X . B}{t[A_X] : [X := A]B} \text{ (\forall_e)}
\end{array}$$

Figura 5. Reglas de tipado de Sistema I Polimórfico

lo que hace que la identificación de este término con $(\Lambda X . \Lambda Y . \lambda x^X . x)[B_Y][A_X]$ no modifique su semántica operacional.

Por lo antedicho, la sintaxis de términos de Sistema I Polimórfico es la siguiente:

$$t := x^A \mid \lambda x^A . t \mid tt \mid \langle t, t \rangle \mid \pi_A t \mid \Lambda X . t \mid t[A_X]$$

Las definiciones de variables libres y sustituciones se extienden de forma usual a las nuevas construcciones, con una excepción: en la sustitución de tipos en términos cerrados, la aplicación de tipos debe renombrar las variables por variables nuevas tomadas del conjunto correspondiente al tipo que se sustituye, ya que las variables se eligen de conjuntos diferentes dependiendo de su tipo.

Las reglas de tipado se establecen en la Figura 5, y son las reglas estándares de Sistema F con pares, a la cual hemos agregado la regla (\equiv) .

Las seis reglas que definen la relación simétrica \rightleftharpoons entre términos en Sistema I son heredadas directamente en Sistema I Polimórfico. Además, debemos agregar reglas a esta relación que reflejen los nuevos isomorfismos. La semántica operacional de Sistema I Polimórfico, que, al igual que en Sistema I, viene dada por la relación \hookrightarrow módulo la relación simétrica \rightleftharpoons , se detalla en la Figura 6.

3.2. Ejemplos

En esta sección introducimos ejemplos para mostrar la necesidad de las reglas presentadas. Los términos elegidos en los ejemplos no son tipables en Sistema F con pares.

La regla (P-SWAP) permite no prestar atención al orden de los argumentos de tipos, lo cual es una consecuencia del isomorfismo (5). Así, el término $(\Lambda X . \Lambda Y . \lambda x^X . \lambda y^Y . x)[B_Y][A_X]$ reduce a $\lambda x^A . \lambda y^B . x$, a pesar de que las instanciaciones de tipos están en diferente orden; la reducción procede así:

$$\begin{aligned}
(\Lambda X . \Lambda Y . \lambda x^X . \lambda y^Y . x)[B_Y][A_X] &\xrightarrow{(\text{P-SWAP})} (\Lambda X . \Lambda Y . \lambda x^X . \lambda y^Y . x)[A_X][B_Y] \\
&\hookrightarrow_{\beta_\lambda} (\Lambda Y . \lambda x^A . \lambda y^Y . x)[B_Y] \hookrightarrow_{\beta_\lambda} \lambda x^A . \lambda y^B . x
\end{aligned}$$

La regla $(\text{P-COMM}_{\forall_i \Rightarrow_i})$ permite no prestar atención al orden de los argumentos de términos y tipos, como en el siguiente ejemplo:

$$\begin{aligned}
(\Lambda X . \lambda x^A . \lambda f^{A \Rightarrow X} . fx)t &\xrightarrow{(\text{P-COMM}_{\forall_i \Rightarrow_i})} (\lambda x^A . (\Lambda X . \lambda f^{A \Rightarrow X} . fx))t \\
&\hookrightarrow_{\beta_\lambda} (\Lambda X . \lambda f^{A \Rightarrow X} . ft)
\end{aligned}$$

$$\begin{array}{l}
\langle r, s \rangle \rightleftharpoons \langle s, r \rangle \quad (\text{COMM}) \\
\langle r, \langle s, t \rangle \rangle \rightleftharpoons \langle \langle r, s \rangle, t \rangle \quad (\text{ASSO}) \\
\lambda x^A. \langle r, s \rangle \rightleftharpoons \langle \lambda x^A. r, \lambda x^A. s \rangle \quad (\text{DIST}_\lambda) \\
\langle r, s \rangle t \rightleftharpoons \langle rt, st \rangle \quad (\text{DIST}_{\text{app}}) \\
r \langle s, t \rangle \rightleftharpoons rst \quad (\text{CURRY}) \\
t[A_X][B_Y] \rightleftharpoons t[B_Y][A_X] \quad (\text{P-SWAP}) \\
\text{si } X \notin \text{FTV}_A(A) \quad \Lambda X. \lambda x^A. r \rightleftharpoons \lambda x^A. \Lambda X. r \quad (\text{P-COMM}_{\forall_i \Rightarrow_i}) \\
\text{si } X \notin \text{FTV}_A(A) \quad (\lambda x^A. t)[B_X] \rightleftharpoons \lambda x^A. t[B_X] \quad (\text{P-COMM}_{\forall_e \Rightarrow_i}) \\
\Lambda X. \langle r, s \rangle \rightleftharpoons \langle \Lambda X. r, \Lambda X. s \rangle \quad (\text{P-DIST}_{\forall_i \wedge_i}) \\
\langle r, s \rangle [A_X] \rightleftharpoons \langle r[A_X], s[A_X] \rangle \quad (\text{P-DIST}_{\forall_e \wedge_i}) \\
\pi_{\forall X. A}(\Lambda X. r) \rightleftharpoons \Lambda X. \pi_A r \quad (\text{P-DIST}_{\forall_i \wedge_e}) \\
\text{si } t : \forall X. (B \wedge C) \quad (\pi_{\forall X. Bt})[A_X] \rightleftharpoons \pi_{[X:=A]B}(t[A_X]) \quad (\text{P-DIST}_{\wedge_e \forall_e}) \\
\\
\text{Si } s : A, (\lambda x^A. r)s \hookrightarrow [x := s]r \quad (\beta_\lambda) \\
\text{Si } r : A, \pi_A(\langle r, s \rangle) \hookrightarrow r \quad (\pi) \\
(\Lambda X. r)[A_X] \hookrightarrow [X := A]r \quad (\beta_A) \\
\\
\frac{t \rightleftharpoons r}{\lambda x^A. t \rightleftharpoons \lambda x^A. r} \quad \frac{t \rightleftharpoons r}{ts \rightleftharpoons rs} \quad \frac{t \rightleftharpoons r}{st \rightleftharpoons sr} \quad \frac{t \rightleftharpoons r}{\langle t, s \rangle \rightleftharpoons \langle r, s \rangle} \quad \frac{t \rightleftharpoons r}{\langle s, t \rangle \rightleftharpoons \langle s, r \rangle} \quad \frac{t \rightleftharpoons r}{\pi_A t \rightleftharpoons \pi_A r} \\
\frac{t \rightleftharpoons r}{\Lambda X. t \rightleftharpoons \Lambda X. r} \quad \frac{t \rightleftharpoons r}{t[A_X] \rightleftharpoons r[A_X]} \\
\frac{t \hookrightarrow r}{\lambda x^A. t \hookrightarrow \lambda x^A. r} \quad \frac{t \hookrightarrow r}{ts \hookrightarrow rs} \quad \frac{t \hookrightarrow r}{st \hookrightarrow sr} \quad \frac{t \hookrightarrow r}{\langle t, s \rangle \hookrightarrow \langle r, s \rangle} \quad \frac{t \hookrightarrow r}{\langle s, t \rangle \hookrightarrow \langle s, r \rangle} \quad \frac{t \hookrightarrow r}{\pi_A t \hookrightarrow \pi_A r} \\
\frac{t \hookrightarrow r}{\Lambda X. t \hookrightarrow \Lambda X. r} \quad \frac{t \hookrightarrow r}{t[A_X] \hookrightarrow r[A_X]}
\end{array}$$

Figura 6. Relaciones que definen la semántica operacional de Sistema I Polimórfico

La regla $(\text{P-COMM}_{\forall_e \Rightarrow_i})$ es consecuencia del isomorfismo (6). Por ejemplo, el término $(\lambda x^{\forall X. (X \Rightarrow X)}. x)[A_X](\Lambda X. \lambda x^X. x)$, está bien tipado. Notamos con \mathbb{X} al tipo $\forall X. (X \Rightarrow X)$:

$$\frac{\frac{\lambda x^{\mathbb{X}}. x : \mathbb{X} \Rightarrow \mathbb{X}}{\lambda x^{\mathbb{X}}. x : \forall Y. \mathbb{X} \Rightarrow (Y \Rightarrow Y)} (\equiv)}{(\lambda x^{\mathbb{X}}. x)[A_X] : \mathbb{X} \Rightarrow (A \Rightarrow A)} (\forall_e) \quad \frac{\Lambda X. \lambda x^X. x : \mathbb{X}}{((\lambda x^{\mathbb{X}}. x)[A_X])(\Lambda X. \lambda x^X. x) : A \Rightarrow A} (\Rightarrow_e)$$

La reducción ocurre como sigue:

$$\begin{aligned}
& (\lambda x^{\forall X. (X \Rightarrow X)}. x)[A_X](\Lambda X. \lambda y^X. y) \rightleftharpoons_{(\text{P-COMM}_{\forall_e \Rightarrow_i})} \\
& (\lambda x^{\forall X. (X \Rightarrow X)}. x)[A_X](\Lambda X. \lambda y^X. y) \hookrightarrow_{\beta_\lambda} (\Lambda X. \lambda y^X. y)[A_X] \hookrightarrow_{\beta_A} \lambda y^A. y
\end{aligned}$$

La reglas $(\text{P-DIST}_{\forall_i \wedge_i})$ y $(\text{P-DIST}_{\forall_i \wedge_e})$ son consecuencia del isomorfismo (7). Por ejemplo, el término $\pi_{\forall X.(X \Rightarrow X)}(\Lambda X.\langle \lambda x^X .x, t \rangle)$ está bien tipado:

$$\frac{\Lambda X.\langle \lambda x^X .x, t \rangle : \forall X.((X \Rightarrow X) \wedge A)}{\Lambda X.\langle \lambda x^X .x, t \rangle : (\forall X.(X \Rightarrow X)) \wedge (\forall X.A)} \quad (\equiv)$$

$$\frac{\Lambda X.\langle \lambda x^X .x, t \rangle : (\forall X.(X \Rightarrow X)) \wedge (\forall X.A)}{\pi_{\forall X.(X \Rightarrow X)}(\Lambda X.\langle \lambda x^X .x, t \rangle) : \forall X.(X \Rightarrow X)} \quad (\wedge_e)$$

Las siguientes son dos reducciones alternativas de dicho término:

$$\pi_{\forall X.(X \Rightarrow X)}(\Lambda X.\langle \lambda x^X .x, t \rangle) \xrightarrow{(\text{P-DIST}_{\forall_i \wedge_i})} \pi_{\forall X.(X \Rightarrow X)}\langle \Lambda X.\lambda x^X .x, \Lambda X.t \rangle \hookrightarrow_{\pi} \Lambda X.\lambda x^X .x$$

$$\pi_{\forall X.(X \Rightarrow X)}(\Lambda X.\langle \lambda x^X .x, t \rangle) \xrightarrow{(\text{P-DIST}_{\forall_i \wedge_e})} \Lambda X.\pi_{X \Rightarrow X}\langle \lambda x^X .x, t \rangle \hookrightarrow_{\pi} \Lambda X.\lambda x^X .x$$

Otra consecuencia del isomorfismo (7) es la regla $(\text{P-DIST}_{\forall_e \wedge_i})$, para la cual podemos considerar el siguiente ejemplo:

$$\langle \Lambda X.\lambda x^X .\lambda y^A .t, \Lambda X.\lambda x^X .\lambda z^B .r \rangle [C_X]$$

el cual está bien tipado. Por simplicidad, asumimos $t : D$ y $r : E$ con $X \notin \text{FTV}_A(D) \cup \text{FTV}_A(E)$:

$$\frac{\langle \Lambda X.\lambda x^X .\lambda y^A .t, \Lambda X.\lambda x^X .\lambda z^B .r \rangle : \forall X.(X \Rightarrow A \Rightarrow D) \wedge \forall X.(X \Rightarrow B \Rightarrow E)}{\langle \Lambda X.\lambda x^X .\lambda y^A .t, \Lambda X.\lambda x^X .\lambda z^B .r \rangle : \forall X.((X \Rightarrow A \Rightarrow D) \wedge (X \Rightarrow B \Rightarrow E))} \quad (\equiv)$$

$$\frac{\langle \Lambda X.\lambda x^X .\lambda y^A .t, \Lambda X.\lambda x^X .\lambda z^B .r \rangle [C_X] : (C \Rightarrow A \Rightarrow D) \wedge (C \Rightarrow B \Rightarrow E)}{\langle \Lambda X.\lambda x^X .\lambda y^A .t, \Lambda X.\lambda x^X .\lambda z^B .r \rangle [C_X]} \quad (\forall_e)$$

La reducción ocurre como sigue:

$$\langle \Lambda X.\lambda x^X .\lambda y^A .t, \Lambda X.\lambda x^X .\lambda z^B .r \rangle [C_X] \xrightarrow{(\text{P-DIST}_{\forall_e \wedge_i})} \langle (\lambda x^X .\lambda y^A .t)[C_X], (\lambda x^X .\lambda z^B .r)[C_X] \rangle \hookrightarrow_{\beta_A} \langle \lambda x^C .\lambda y^A .t, \lambda x^C .\lambda z^B .r \rangle$$

Finalmente, la regla $(\text{P-DIST}_{\wedge_e \forall_e})$, también consecuencia del mismo isomorfismo, nos permite considerar el siguiente ejemplo:

$$(\pi_{\forall X.(X \Rightarrow X)}(\Lambda X.\langle \lambda x^X .x, r \rangle))[A_X]$$

el cual también está bien tipado. Asumiendo $r : C$, tenemos la siguiente derivación:

$$\frac{\Lambda X.\langle \lambda x^X .x, r \rangle : \forall X.((X \Rightarrow X) \wedge C)}{\Lambda X.\langle \lambda x^X .x, r \rangle : (\forall X.(X \Rightarrow X)) \wedge (\forall X.C)} \quad (\equiv)$$

$$\frac{\Lambda X.\langle \lambda x^X .x, r \rangle : (\forall X.(X \Rightarrow X)) \wedge (\forall X.C)}{\pi_{\forall X.(X \Rightarrow X)}(\Lambda X.\langle \lambda x^X .x, r \rangle) : \forall X.(X \Rightarrow X)} \quad (\wedge_e)$$

$$\frac{\pi_{\forall X.(X \Rightarrow X)}(\Lambda X.\langle \lambda x^X .x, r \rangle) : \forall X.(X \Rightarrow X)}{(\pi_{\forall X.(X \Rightarrow X)}(\Lambda X.\langle \lambda x^X .x, r \rangle))[A_X] : A \Rightarrow A} \quad (\forall_e)$$

La reducción ocurre como sigue:

$$(\pi_{\forall X.(X \Rightarrow X)}(\Lambda X.\langle \lambda x^X .x, r \rangle))[A_X] \xrightarrow{(\text{P-DIST}_{\wedge_e \forall_e})} \pi_{A \Rightarrow A}(\Lambda X.\langle \lambda x^X .x, r \rangle)[A_X] \hookrightarrow_{\beta_A} \pi_{A \Rightarrow A}\langle \lambda x^A .x, [X := A]r \rangle \hookrightarrow_{\pi} \lambda x^A .x$$

4. Conclusión

En este paper hemos presentado una extensión a Sistema I a tipos polimórficos, extendiendo el número de isomorfismos a considerar y, en consecuencia, extendiendo la equivalencia entre términos del lenguaje.

Este es un trabajo en progreso, y como tal, aún estamos desarrollando las pruebas de corrección (preservación de tipos y normalización fuerte). Además, como trabajo a futuro planeamos extender la implementación presentada por Díaz-Caro y Martínez López [4] con las construcciones introducidas aquí.

Referencias

1. Barendregt, H.: The lambda calculus: its syntax and semantics, Studies in logic and the foundations of mathematics, vol. 103. Nord-Holland (1981)
2. Di Cosmo, R.: Isomorphisms of types: from λ -calculus to information retrieval and language design. Progress in Theoretical Computer Science, Birkhauser (1995)
3. Díaz-Caro, A., Dowek, G.: Proof normalisation in a logic identifying isomorphic propositions. In: Proceedings of FSCD 2019. LIPIcs, vol. 131, pp. 14:1–14:23. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik (2019)
4. Díaz-Caro, A., Martínez López, P.E.: Isomorphisms considered as equalities: Projecting functions and enhancing partial application through an implementation of λ^+ . In: Proceedings of IFL'15. pp. 9:1–9:11. ACM (2015)
5. Girard, J.Y.: Une extension de l'interpretation de Göedel à l'analyse, et son application à l'élimination des coupures dans l'analyse et la théorie des types. In: Proceedings of the second Scandinavian logic symposium. pp. 63–92 (1971)

Implementación Básica de Typestates en Rust

Marcelo Arroyo, Simón Gutiérrez Brida, Pablo Ponzio

Dpto de Computación, FCEFQyN, Universidad Nacional de Río Cuarto

Resumen Generalmente la API de un módulo describe las operaciones disponibles, aunque el orden lícito de aplicación de las mismas queda implícito o documentado externamente debido a que los lenguajes de programación generalmente no proveen mecanismos de especificación del *protocolo* de uso.

Typestates permite especificar *estados* de objetos de un determinado tipo. Cada estado habilita ciertas operaciones y prohíbe otras, permitiendo especificar el *protocolo* de uso de una API determinada.

En este trabajo se presenta un patrón de implementación de *typestates* en el lenguaje de programación **Rust** y se analiza su sistema de tipos y mecanismos que permiten la verificación del *typestate* en tiempo de compilación, mostrando que cumple con las propiedades requeridas por las propuestas de verificación modular descritas en la bibliografía especializada.

1. Introducción

Cuando se desarrolla un componente de software, principalmente en el paradigma imperativo u orientado a objetos, se debe definir su interface (API) pero también es importante tener en cuenta el estado del objeto que representa al componente.

A modo de ejemplo, a una variable de tipo `File` se pueden aplicar las operaciones `read`, `write` y `close` luego de un `open` exitoso, o en un iterador (como en Java), la operación `next()` debería ser precedida por `hasNext()`.

En ingeniería de software, generalmente se utilizan máquinas de estados para describir estas clases de objetos y a nivel de diseño se utilizan varios formalismos y notaciones para su descripción y análisis.

Sin embargo, casi la totalidad de los lenguajes de programación carecen de mecanismos directos para describir el estado y sus transiciones. La programación basada en *typestates* [4] permite definir objetos no sólo en términos de clases (o tipos) sino también en los cambios de estado de manera explícita, describiendo los cambios de estado en base a la aplicación de las operaciones o métodos. Esto permite definir el *protocolo* que se debe cumplir en cuanto al uso de la componente, imponiendo la aplicación de las operaciones en los estados correspondientes. Sería deseable que esto pudiese ser verificado estáticamente.

Si bien muchos componentes de software se diseñan de esa manera, en la implementación se deben incluir verificaciones en tiempo de ejecución en las

operaciones para garantizar el uso correcto y en el código el *protocolo* queda implícito.

La idea de *typestates* [1] extiende la idea de *tipo* con el de *estado interno* y su incorporación en un lenguaje de programación permitiría la verificación en tiempo de compilación, evitando tener que considerar los estados en ejecución y el uso de mecanismos como devolución de códigos de error o disparo de excepciones.

Si bien la gran mayoría de los lenguajes de programación de amplia adopción no incluyen soporte para *typestates*, algunos lenguajes más recientes, en particular **Rust** [2], incluyen mecanismos en su sistema de tipos que permiten implementarlo adecuadamente y permiten su verificación durante la compilación.

La figura 1 muestra un simple ejemplo de una API sobre archivos.

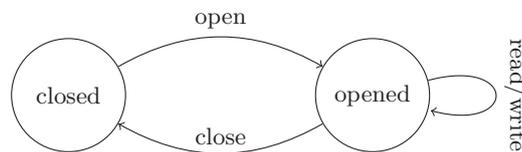


Figura 1. Sistema de transición de estados de una api File

En la próxima sección se describen las principales características de **Rust**, principalmente su sistema de tipos. Luego se presenta una propuesta de un patrón de implementación de *typestates*. En la sección 4 se analiza el soporte del sistema de tipos y mecanismos de **Rust** para implementar las ideas propuestas en [4] y [5]. También se consideran herramientas de verificación estática en otros lenguajes como Java y C++. En esta sección se encuentra la principal contribución de este trabajo. Finalmente se describen las conclusiones y trabajo futuro.

2. Características de Rust

Rust [2] es un lenguaje de programación reciente (Julio 2010), originalmente desarrollado por Graydon Hoare en *Mozilla Research*¹.

Entre sus principales características se pueden destacar:

- Sintácticamente similar a C++, aunque con construcciones más concisas.

¹ El navegador Mozilla Firefox, inicialmente escrito en C++, ya contiene varios subsistemas reescritos en **Rust**.

- Multiparadigma: Si bien se puede considerar un lenguaje imperativo, tiene mecanismos de soporte para programación orientada a objetos, genericidad y concurrencia.
- Adecuado para la *programación de sistemas* (software de base, embebido, etc) y aplicaciones que requieran alto rendimiento, como por ejemplo un navegador web moderno.
- Comparte la filosofía de C++ de brindar *abstracciones de costo cero*.
- Gestión manual de la memoria pero garantizando el desarrollo de programas *seguros* en cuanto a que previene tener *memory leaks* y *dangling references*.
- Es un lenguaje fuertemente tipado, con verificación estática de control de aliasing y tiempo de vida de variables.

2.1. Tipos de datos

Los tipos primitivos de Rust incluyen

- *Básicos*: lógicos (`bool`), caracteres Unicode (`char`), numéricos (enteros `i8`, `i16`, `i32`, `i64` y sus versiones sin signo `u8`, ... y reales en punto flotante `f32`, `f64`).
- *Estructurados*: registros (`structs`), tuplas, arreglos de tamaño fijo (de tipo `[T; N]`, con T como tipo base y N como tamaño).
- *Uniones disjuntas* (`enums`): Las enumeraciones pueden usarse como patrones en sentencias `match` para realizar *pattern-matching*.

```

1 struct Point { x: i32, y: i32 } // struct
2 struct Color {i32, i32}        // struct tuple
3 enum RGB {Red, Green, Blue}   // disjoint union
4 let mut a = [1,2,3];           // mutable array
5 let first_two = &a[0..2];      // immutable slice
6                               //(first 2 elements)
7 let pair = (i32, i32);         // tuple literal
8 let my_ref = &pair;           // immutable reference

```

Las variables y referencias son inmutables por omisión, excepto que se use el modificador `mut`.

La biblioteca estándar define muchos tipos útiles como strings (`str`), vectores (arreglos de longitud variable), y otros.

2.2. Polimorfismo

Rust provee *polimorfismo paramétrico acotado*, similar a la genericidad de Java y polimorfismo *ad hoc* (*type classes* o *traits*).

Es posible definir tipos y funciones con parámetros de tipos, como `vec<T>` y `fn max<T>(v: vec[T]) -> T { ...}`.

La manera de emular la herencia se logra mediante el uso de *traits*, las cuales podríamos decir que son similares a las *interfaces* al estilo de Java, aunque técnicamente son más del estilo de los *conceptos* de C++20 [10] o las *type classes* de Haskell. En las comunidades de usuarios de C++, Rust, Scala y otros, los *traits* representan *requisitos* (u operaciones) que determinados tipos deberán implementar.

Los *traits* son extensibles, permitiendo definir jerarquías y cualquier tipo puede implementar múltiples *traits*. Este mecanismo es menos restricto que la *herencia* y facilita la *extensibilidad y adaptabilidad de componentes*.

Es posible implementar un *trait* para cualquier tipo de dato. La biblioteca estándar define algunos *traits* de utilidad y algunos de ellos está integrados en el compilador, como por ejemplo, los *traits Copy*, el cual provee la operación `copy()` para implementar *copy semantics* y *Drop*, que provee la operación `drop()` que indica la liberación de recursos adueñados y la destrucción del objeto invocante. Estos *traits* son conocidos por el compilador y tenidos en cuenta por el *borrow checker*.

Esta forma de estructurar tipos de datos y operaciones, permite extensibilidad multidimensional sin tener que definir una jerarquía de clases con las restricciones que ello implica. Estos mecanismos están presentes en otros lenguajes de programación modernos como Scala.

Para más detalles sobre **Rust**, se recomienda el libro oficial disponible en la página web: <https://doc.rust-lang.org/book/>.

2.3. Gestión de memoria

Rust al contrario de otros lenguajes como Go y Java, entre otros, no utiliza *recolección automática de basura*². En su lugar se basa en el concepto (ampliamente usado en C++) de *RAII (Resource Acquisition is Initialization)*, es decir que los recursos de un objeto son adquiridos en su construcción y liberados en su destrucción siguiendo la idea de *ownership* en la cual el objeto es responsable de la liberación de los recursos que él creó.

Rust ofrece referencias mutables (`mut&`) e inmutables (`&`) con verificación estática (en tiempo de compilación) de su uso seguro:

Una referencia no puede tener un tiempo de vida mayor que el valor referenciado.

Rust almacena los valores típicamente en el *stack*, aunque permite almacenar valores en el *heap* mediante el uso de constructores como `Box<T>` (el equivalente a `new` en otros lenguajes de programación, como Java o C++).

Además ofrece opcionalmente en su biblioteca estándar *smart pointers*, como `Rc<T>` (contador de referencias, similar a `std::shared_ptr<T>` de C++0x).

² Esto va en consonancia con el objetivo de ofrecer el máximo rendimiento posible, al estilo de C.

A pesar de que no dispone de un recolector de basura, el compilador fuerza a que la utilización de referencias sea de manera segura, es decir que si un programa compila, éste no contiene errores como *lagunas de memoria*, acceso a *referencias nulas*, o *condiciones de carrera* (*race conditions*) en presencia de concurrencia usando memoria compartida (*threads*).

Esta seguridad en el manejo de la memoria se basa en las siguientes restricciones o invariantes que un programa debe cumplir:

1. *Ownership*:

- Cada valor tiene una única variable como *dueño*.
- Cuando la variable se destruye (finaliza su alcance), el valor asociado se destruye.
- Es posible tener múltiples referencias inmutables.

La asignación (y el pasaje de parámetros) sigue *move semantics*, es decir que la variable original cede su *ownership* al destino, lo que invalida el uso futuro de la primera. Este sistema es una implementación de *affine types*.

2. *Borrowing*: Mecanismo estático que lleva la pista de *ownership* y tiempo de vida de valores y referencias.

Además de verificar las condiciones de *ownership* debe contemplar la transferencia temporal de variables (*owners*) a otras variables o referencias, como es el caso de pasaje de parámetros por referencia y en alcances (scopes) interiores.

El siguiente ejemplo muestra un uso indebido de variables:

```

1 let mut x = 5;
2 let y = &mut x;
3 *y += 1;
4 println!("{}", x);

```

El compilador detecta un uso de *x* en la línea 4 (ha cedido su valor a *y*).

En cambio el siguiente código es legal.

```

1 let mut x = 5;
2 {
3     let y = &mut x;
4     *y += 1;
5 }
6 println!("{}", x);

```

ya que dentro del bloque (líneas 2-5) se hace referencia al valor adueñado por *y* (únicamente) y a la salida del bloque, *x* recupera su condición de (único) *dueño*.

Notar que en este ejemplo el compilador puede determinar que el tiempo de vida de *y* es menor al de *x*.

Este mismo escenario es válido para el pasaje de referencias como parámetros a funciones.

3. Implementación de `typestates`

El objetivo de `typestates` es garantizar, en tiempo de compilación, el cumplimiento del *protocolo* de uso asociado a una API. La idea más simple de implementación es asociar a cada estado un tipo que provea las operaciones aplicables en ese estado, un enfoque comúnmente encontrado en implementaciones de lenguajes de dominio específico *DSL*. Cada operación retorna un objeto del tipo correspondiente a las operaciones disponibles, de acuerdo al protocolo de uso de la API al que pertenece el objeto³. En la Figura 3 se muestra un ejemplo de la API `StFile` (typestate File).

Es importante notar que para las operaciones que cambian el estado, como `open` o `close`, el objeto (`self`) se pasa por valor, lo que tiene como consecuencia que el objeto se *consume* (*borrow*), es decir que no pueda usarse en operaciones subsiguientes. Recordar que Rust usa *move semantics* por omisión. Por eso al compilar el código del siguiente ejemplo:

```

1 fn main() {
2     let f = file ();
3     let mut f = f.open (); // we define a "new" f here
4     let mut g = f;
5     f.read ();
6     let f = f.close ();
7 }
```

Obtenemos el siguiente error de compilación:

```

error[E0382]: borrow of moved value: 'f'
--> typestate.rs:51:2
|
3 |   let mut f = f.open();
|     ----- move occurs because 'f' has type 'StFile<FileOpened>',
|           which does not implement the 'Copy' trait
4 |   let mut g = f;
|                 - value moved here
5 |   f.read();
|   ^ value borrowed here after move
```

Este ejemplo muestra que el *borrow checker* lleva el control de *ownership*. Si eliminamos del ejemplo de uso anterior las líneas 3 y 4,

```

1 fn main() {
2     let f = file ();
3     f.read ();
4     f.close ();
5 }
```

obtenemos el siguiente error:

³ El uso de tipos parametrizados permite simplificar la implementación.

```

1 // States (phantom types)
2 struct FileClosed;
3 struct FileOpened;
4
5 struct File; // actual file i/o impl.
6
7 struct StFile<State> {
8     // real implementation of File i/o
9     _impl: File,
10    // zero size data, not represented in runtime
11    _state: State
12 }
13
14 // Implementation for closed state.
15 // User only can call open()
16 impl StFile<FileClosed> {
17     // open() consumes self and return a opened file
18     fn open(self) -> StFile<FileOpened> {
19         File {
20             _impl: self._impl,
21             _state: FileOpened
22         }
23     }
24 }
25
26 // Implementation for open state.
27 // User can use read(), write() and close()
28 impl StFile<FileOpened> {
29     // This operations lead object in the same state
30     fn read(&mut self) {}
31     fn write(&mut self) {}
32
33     // consume self and return a new closed state object
34     fn close(self) -> StFile<FileClosed> {
35         File {
36             _impl: self._impl,
37             _state: FileClosed
38         }
39     }
40 }
41
42 // simple factory for file in initial state (closed)
43 fn file() -> StFile<FileClosed> {
44     File {
45         _impl: File,
46         _state: FileClosed
47     }
48 }

```

Figura 2. File con tpestates.

```

error[E0599]: no method named 'read' found for type 'StFile<FileClosed>'
      in the current scope
--> tpestate.rs:3:4
3 |     f.read();
  |     ^^^^^

```

porque hemos violado el protocolo al no haber invocado previamente a `open` y se aplica la operación `read()` sobre un objeto en estado `Closed`.

El siguiente listado muestra un uso legítimo del protocolo y la compilación es exitosa.

```

fn main() {
    let f = file();
    let mut f = f.open();
    f.read();
    f.close();
}

```

Como se puede apreciar, el único inconveniente que esto genera es que ante cada operación que produce un cambio de estado se deberá usar una nueva variable, ya que esta operación retornará un valor de un nuevo tipo. Esto no le quita simplicidad al uso de estos tipos ya que `Rust` permite el rehúso de identificadores y además omitir el tipo (inferencia automática).

4. Verificación estática

En esta sección se analizan y justifican porqué el sistema de tipos y el *borrow checker* de `Rust` permiten realizar la verificación de *tpestates* cuando se implementan en un estilo como el que se muestra en la sección anterior.

En [5] y [4] se describe una propuesta de análisis estático modular de *tpestates*.

Cualquier algoritmo de verificación de *tpestates* deberá llevar la pista de cada estado de un valor en un punto de programa determinado, teniendo en cuenta las transiciones de estados.

Con lo anterior no alcanza si el lenguaje permite *aliasing*, por ejemplo, mediante el uso de *referencias*. En cuyo caso, un verificador estático deberá verificar que un valor puede ser accedido desde varias referencias lo que hace la verificación mucho más compleja.

En los artículos referenciados arriba, Bierhoff, Aldrich y sus colegas han propuesto lenguajes con construcciones o anotaciones que permiten al sistema de tipos tener control y así poder *razonar* en presencia de *aliasing*.

Particularmente, en [5], se propone un algoritmo de verificación de *tpestates* que requiere que las referencias tengan las siguientes características:

1. *Lógica lineal*: Cada variable puede ser asignada sólo una vez.

2. Permisos de acceso

- *full*: Permite acceso de lectura/escritura al valor referenciado. En este caso, sólo deberá haber una única referencia al valor (en cada scope).
- *pure*: Acceso de sólo lectura. En este caso, puede haber múltiples referencias *puras*.

En **Rust**, usando variables inmutables y gracias a *move semantics* se logra implementar lógica lineal, mientras que el *borrow checker* garantiza que en presencia de aliasing, existe una única referencia mutable en uso en un bloque (*scope*) dado, o bien pueden existir múltiples referencias inmutables.

Mediante estos mecanismos, junto con el patrón de implementación propuesto, se logra que la verificación estática realizada por el compilador prácticamente imite el algoritmo propuesto en [5].

Es importante notar que en otros lenguajes de programación, como por ejemplo C++, aunque soporta *move semantics*, la verificación de *lógica lineal* sólo puede ocurrir en tiempo de ejecución (cuando se intente acceder a un valor movido anteriormente se producirá una excepción), ya que a diferencia de **Rust**, no se realiza ningún chequeo en tiempo de compilación.

En otros lenguajes la verificación de *typestate* podría hacerse mediante el uso de herramientas externas, como por ejemplo con el *clang static analyzer* [7,9,3], que contiene soporte para el análisis de *dataflow*, como se propone en [6].

5. Conclusiones y trabajo futuro

El uso de *typestates* es cada vez más importante para el desarrollo de software confiable. Existen pocos lenguajes de programación con soporte para su verificación estática. Algunos de los lenguajes que lo soportan son experimentales y no han sido adoptados ampliamente en producción.

Si bien en las versiones iniciales de **Rust** se incluía soporte para especificar *typestates*, esta característica fue eliminada en su versión 1.0. Sin embargo, en este trabajo se muestra una posible estrategia de implementación sin costo extra en tiempo de ejecución, es decir que no se requiere más memoria ni chequeos dinámicos, aprovechando el sistema de tipos y el verificador (*borrow checker*) para conseguir análisis estático de *typestates* en **Rust**.

Muchos componentes de la biblioteca estándar de **Rust** se están desarrollando siguiendo el concepto de *typestates*.

Como trabajo futuro se está desarrollando un *checker* para el *Clang static analyzer* [3] ya que provee un framework de ejecución simbólica, el cual permitirá al desarrollador especificar el sistema de transición de estados representando el *typestate* y analizar su uso. El checker en desarrollo usa otro checker existente (el *move checker*). El principal desafío es la *emulación* del *borrow checker* de **Rust**.

Referencias

1. Robert E. Strom and Shaula Yemini. *Typestates: A Programming Language Concept for Enhancing Software Reliability*. IEEE Transactions on Software Engineering, vol SE-12, n° 1, 1986.
2. Steve Klabnik and Carol Nichols. *The Rust Programming Language (Ed. 2018)*. <https://doc.rust-lang.org/book/>
3. Clang static analyzer. <http://clang-analyzer.llvm.org/>
4. Jonathan Aldrich, Joshua Sunshine, Darpan Saini, Zachary Sparks. *Typestate-Oriented Programming*. Onward 09. ACM 78-1-60558-768-4/09/10. 2009/2010.
5. Kevin Bierhoff, Jonathan Aldrich. *Modular Typestate Checking of Aliased Objects*. OOPSLA'07, October 21–25, 2007, Montréal, Québec, Canada. ACM 978-1-59593-786-5/07/0010. 2007.
6. Thomas Reps, Susan Harwitz, Mooly Sagiv. *Precise Interprocedural Dataflow Analysis via Graph Reachability*. POPL '95 Proceedings of the 22nd ACM SIGPLAN-SIGACT symposium on Principles of programming languages. Pages 49-61. ISBN:0-89791-692-1.
7. Zhongxing Xu, Ted Kremenek, Jian Zhang. *A Memory Model for Static Analysis of C Programs*. International Symposium on Leveraging Applications of Formal Methods, 2010. Lecture Notes In Computer Science. Vol. 6415.
8. J. Aldrich, et al. *Permission Based Programming Languages*. ICSE'11, May 21–28, 2011. ACM 978-1-4503-0445-0/11/05. 2011.
9. Hari Hampapuram, Yue Yang, and Manuvir Das. *Symbolic Path Simulation in Path-Sensitive Dataflow Analysis*. PASTE '05 Proceedings of the 6th ACM SIGPLAN-SIGSOFT workshop on Program analysis for software tools and engineering. Pages 52-58. ACM SIGSOFT Software Engineering Notes. Volume 31 Issue 1, January 2006. Pages 52-58.
10. Bjarne Stroustrup. *Concepts: The Future of Generic Programming*. Technical Report P0557r1. Morgan Stanley and Columbia University. 2017.

Inspecciones para Mejorar la Calidad de Modelos en Lenguaje Natural

Graciela D. S. Hadad^{1,2}, Alberto Sebastián¹

¹ Facultad de Tecnología Informática, Universidad de Belgrano, Argentina

² Escuela de Informática, Universidad Nacional del Oeste, Argentina
{graciela.hadad, alberto.sebastian}@comunidad.ub.edu.ar

Resumen. Las inspecciones han probado ser de gran eficacia en la detección de defectos en código fuente, por lo que esta técnica se ha adaptado para verificar modelos elaborados en el proceso de especificar requisitos. La literatura ha propuesto algunas variantes de inspección en el modo de detectar defectos en estos modelos, sin embargo, los estudios empíricos realizados no alcanzan para establecer qué variante puede ser más apropiada en determinadas situaciones, en función de la calidad exigida y tiempos disponibles. Dado el uso frecuente de modelos escritos en lenguaje natural en el proceso de requisitos, se ha llevado a cabo un experimento para comparar la eficiencia y eficacia de tres variantes de inspección aplicadas a 5 modelos léxico. Los resultados obtenidos permiten establecer una aproximación inicial en cuanto a que la variante basada en la lectura constructiva detecta un número superior de defectos, principalmente omisiones y ambigüedades, aunque insumiendo más tiempo.

Palabras Clave: Verificación, Inspecciones, Ingeniería de Requisitos, Modelos en Lenguaje Natural.

1 Introducción

Se han diseñado muy variados métodos para sistematizar el desarrollo de software y mejorar su nivel de calidad. Sin embargo, muchos proyectos de software han fracasado debido principalmente a una pobre especificación de requisitos. Una respuesta a este problema está dada en el marco de la Ingeniería de Requisitos (IR) mediante la verificación de sus modelos usando técnicas efectivas como las inspecciones [1-3], pues estos modelos son la base para construir los siguientes artefactos en el proceso de desarrollo de software.

En la IR han surgido diversas modalidades para detectar defectos mediante una inspección [1] y se han realizado estudios sobre la efectividad de la técnica [2-6], sin embargo, quedan aún cuestiones no resueltas sobre cual variante puede ser más conveniente según el caso. Es por ello que se presenta un estudio comparativo de tres variantes de inspección aplicadas sobre un modelo denominado Léxico Extendido del Lenguaje (LEL) [7]. Este modelo define la terminología utilizada en el contexto de aplicación bajo estudio, y se lo construye al iniciar un proceso de IR, por lo que su

calidad puede impactar en modelos derivados de él, escritos en lenguaje natural (LN), tales como escenarios, casos de uso e historias de usuario [7,8]. El propósito del estudio ha sido identificar la eficacia y eficiencia de las tres variantes, permitiendo colaborar en la toma de decisiones sobre la variante de inspección apropiada para la calidad esperada y los plazos previstos.

En la sección 2 se resume la técnica de inspección y sus variantes en la IR. La sección 3 presenta el modelo léxico y las variantes de inspección aplicadas al mismo. En la sección 4 se describe el experimento realizado y los datos obtenidos de cada variante de inspección. En la sección 5 se analizan los resultados, y finalmente en la sección 6 se exponen conclusiones y trabajos futuros.

2 Inspecciones en la Ingeniería de Requisitos

Las inspecciones fueron inicialmente diseñadas para detectar defectos en código fuente [9], y dada su alta eficacia [10,11] fueron adoptadas para verificar modelos de diseño y de requisitos. La técnica involucra la participación de inspectores, autores del material, un moderador y un escriba, siguiendo un proceso de 6 etapas: 1) Planificación de la inspección; 2) Visión General, donde los autores describen a los inspectores el material a inspeccionar; 3) Preparación o lectura del material por parte de los inspectores; 4) Reunión de Inspección, donde los inspectores revisan el material detectando defectos y los autores los confirman o rechazan, asistidos por el moderador mientras el escriba registra los hechos; 5) Corrección de los defectos a cargo de los autores; y 6) Seguimiento del estado del material.

En la IR se han propuesto diferentes variantes para identificar defectos en una inspección, centrando la detección en la etapa de Preparación, y ellas son [9]:

Lectura ad-hoc: el inspector lee el material e identifica defectos dependiendo exclusivamente de su experiencia, en un modo desestructurado.

Lectura basada en lista de comprobación: el inspector usa una lista de control que le sirve de ayuda-memoria para identificar defectos.

Lectura basada en procedimientos: el inspector sigue procedimientos detallados para capturar defectos, en un modo altamente estructurado.

Lectura constructiva: el inspector produce un artefacto intermedio a partir del material y sobre dicho artefacto detecta los defectos.

En el trabajo presentado en [6], se evaluó la influencia del nivel de experiencia del inspector, resultando que profesionales lograron mayor eficacia que estudiantes al detectar defectos en casos de uso, bajo la lectura ad-hoc. Resultados similares se habían presentado sobre especificaciones de requisitos [3]. En otro experimento [2], se evaluó la variante con procedimientos, observándose que la importancia de la definición del procedimiento decrecía a mayor experiencia del inspector. En [12] se recomienda usar variantes de lectura más sistematizadas para no depender de la experiencia del inspector. Por otro lado, en el trabajo expuesto en [5] se comprobó, usando inspectores estudiantes, que se detectaba en promedio más defectos usando la lectura basada en procedimientos que la lectura con lista de control, aunque insumía más tiempo. Son casi nulos los trabajos que analizan la efectividad de la lectura constructiva, siendo pocos los que comparan en un mismo experimento las variantes

existentes; esto permitiría tener mejor conocimiento de ellas al requerirse su uso.

3 Inspecciones del Modelo Léxico

Los modelos en LN, aunque son fáciles de construir y colaboran con la elicitación, validación y negociación, presentan habitualmente omisiones y ambigüedades [4,13]. En un estudio empírico [13] se observó que el 50% de los casos de uso contenía ambigüedades. En [14] se estimaron omisiones superiores al 57% en 9 conjuntos de escenarios. Valores similares de omisiones se estimaron sobre el modelo LEL [15].

3.1 El Modelo Léxico Extendido del Lenguaje

El modelo LEL [7] tiene una estructura que aporta más información que un simple glosario, pues cada término o símbolo se describe con una *Noción* (la denotación del término) y un *Impacto* (su connotación). El modelo permite describir sinónimos, homónimos y jerarquías de términos. Se le aplican dos principios: el de circularidad y el de vocabulario mínimo, por los que la definición de cada símbolo se realiza procurando utilizar al máximo el propio vocabulario del léxico.

Los símbolos del LEL se clasifican en: *Sujetos* (entidades activas), *Objetos* (entidades pasivas), *Verbos* (actividades) o *Estados* (condiciones de sujetos, objetos o verbos). Según esta clasificación, sus nociones e impactos tienen una semántica diferente. La Fig. 1 muestra dos símbolos del LEL, cuyos términos subrayados (hipervínculos) identifican otros símbolos del LEL.

Símbolo: PRESTADOR	Tipo: Sujeto	Símbolo: EXPEDIENTE	Tipo: Objeto
Noción: <ul style="list-style-type: none"> Es el médico de una prestadora para controlar la facturación proveniente del Instituto de Salud. Impacto: <ul style="list-style-type: none"> Audita expedientes de los pacientes para autorizar el pago o rechazar el pago de una prestación. 		Noción: <ul style="list-style-type: none"> Es el conjunto de información de un paciente que genera el Instituto. Contiene la historia clínica, facturas, estudios e imágenes DICOM. Impacto: <ul style="list-style-type: none"> Se envía al prestador para auditar. 	

Fig. 1. Ejemplos de símbolos del LEL

3.2 Variantes de Inspección para el Modelo Léxico

En las variantes bajo estudio, se sigue el proceso de inspección propuesto por Fagan [9] con algunas adaptaciones para modelos en LN [4], como el LEL. No se realiza la etapa Visión General, dado que estos modelos deberían comprenderse por sí mismos sin requerir una descripción de los autores, con lo cual se evita la parcialidad de los autores, logrando mayor independencia entre inspectores y autores. La etapa Preparación se focaliza en detectar defectos, en lugar de solo una lectura del material. Esto lleva a que, en la Reunión de Inspección, los inspectores presenten los defectos hallados previamente y los autores los acepten o rechacen.

La categorización de defectos, aplicable a modelos en LN, es: *Discrepancias* (inconsistencias), *Errores* (elementos incorrectos), *Omisiones* (elementos faltantes) y

Ambigüedades (elementos entendidos de formas diferentes o no entendibles). Cada defecto es calificado por su grado de severidad en: *Alto, Medio* o *Bajo*.

Se utilizaron las siguientes variantes para inspeccionar el modelo LEL: i) Lectura con lista de comprobación: *Variante usando CheckList*; ii) Lectura con procedimientos: *Variante basada en Formularios* [16]; y iii) Lectura constructiva: *Variante basada en la construcción de Mapas Conceptuales* [17].

En las pautas de cada variante, se especifica el tipo de defecto a capturar y su nivel de severidad. En la Preparación, el inspector aplica las pautas especificadas y genera un reporte con el detalle de cada defecto encontrado con su tipo y severidad, midiendo el tiempo insumido. En la Reunión de Inspección, los defectos capturados son confirmados o rechazados, midiéndose el tiempo de la reunión y el número de defectos aceptados por tipo y severidad.

Variante usando CheckList (VCH). El inspector usa una lista con 44 ítems de control sobre el LEL durante la Preparación. Cada ítem tiene predefinido el tipo de defecto y su nivel de severidad (ver Fig. 2).

Ítem de Control	Tiempo hh:mm	Cantidad Defectos	Tipo de Defecto	Severi- dad
20. Verificar que toda mención a un símbolo en las descripciones de otros símbolos esté identificada (referencia no identificada)	00:15	4	Omisión	Media
24. Verificar que los símbolos estén descriptos con oraciones atómicas, concretas y directas (Un solo verbo en cada oración de la noción y del impacto)	00:08	3	Ambigüedad	Media
29. Verificar que todo símbolo Sujeto represente una entidad activa: rol, persona, área u otra que realice alguna acción	00:15	1	Error	Alta
36. Verificar que todo símbolo Objeto describa en su impacto las acciones que se le aplican o que se realizan con él	00:17	2	Omisión	Alta
43. Verificar que se mantenga la coherencia en las descripciones de un símbolo general respecto a sus símbolos especializados (jerarquía de símbolos)	00:20	0	Discrepancia	Media

Fig. 2. Ejemplo de una lista parcial de ítems de control

Variante basada en Formularios (VFR). El inspector completa 12 formularios siguiendo la guía de cada formulario. Las guías indican el objetivo del formulario, cómo llenarlo y cómo analizar los datos volcados para identificar defectos. Cada guía estipula el tipo y severidad de los defectos que permite detectar (ver Fig. 3).

FORMULARIO V: VERIFICACIÓN DE SÍMBOLOS TIPO OBJETO							
NOCIÓN:		definición del objeto: qué representa, sus características / otros objetos con los que se relaciona					
IMPACTO:		acciones que pueden ser aplicadas al objeto o realizadas con él					
Nº Símbolo	Tipo Erróneo	NOCIÓN			IMPACTO		
		SOBRANTE	¿Dónde va?	FALTANTE	SOBRANTE	¿Dónde va?	FALTANTE
4	NO	NO		NO	NO		NO
7	NO	NO		NO	"El prestador..., verifica la FACTURA EMITIDA y aprueba el pago."	No es del símbolo	NO
5	NO	NO		NO	"Contiene el detalle de cada FACTURA EMITIDA."	Va en Noción	NO
Tiempo insumido (hh:mm): 0:33							
GUÍA FORMULARIO V: VERIFICACIÓN DE SÍMBOLOS TIPO OBJETO							
Objetivo: detectar si el tipo Objeto asignado al símbolo es correcto y si su noción e impacto coincide con el contenido especificado para un símbolo del tipo Objeto.							
Pasos:							
<ul style="list-style-type: none"> ◆ Registrar en la columna SÍMBOLO el número de símbolo a analizar. ◆ Registrar en la columna TIPO ERRÓNEO si el símbolo está bien o mal clasificado teniendo en cuenta la definición del tipo Objeto. En caso de mal clasificado, registrarlo en el Formulario XIII. ◆ Determinar si existe alguna parte de la noción del símbolo que sobre o falte con respecto a la definición del tipo Objeto. Si se encontrara alguno, mencionar en la columna NOCIÓN en columna SOBRANTE o FALTANTE la porción de texto correspondiente. ◆ 							
Análisis:							
En el caso de encontrar un Tipo Erróneo (Error / Alta) puede indicar un defecto en la clasificación del símbolo o que la definición (noción e impacto) no se corresponde con el tipo asignado. Cuando se detectan faltantes, es un indicativo de símbolos incompletos según la definición del tipo (Omisión / Alta). Cuando se detectan sobrantes que corresponden al otro componente del mismo símbolo, se trata de ...							

Fig. 3. Ejemplo de un formulario y su procedimiento

Variante construyendo Mapas Conceptuales (VMC). El inspector construye un mapa conceptual por cada símbolo del LEL, para luego analizarlos a través de sus conceptos y relaciones, siguiendo 13 pasos para detectar distintos defectos. Cada paso indica su objetivo, cómo identificar defectos en los mapas, el tipo de defecto y su severidad, y brinda recomendaciones para la corrección del LEL (ver Fig. 4).

PASO 1: Detectar sub-grafos inconexos en un mapa conceptual		
Acción de Detección	Tipo de Defecto	Severidad
Buscar en el mapa conceptual sub-grafos sin relación con el Concepto Raíz Símbolo.	Omisión	Baja
Defecto en el LEL	Ejemplo:	
Detectar frases gramaticales omitidas en la noción o impacto que conectan con el símbolo bajo estudio.		
Corrección en el Mapa		
1. Establecer una relación entre el Concepto Raíz Símbolo y el concepto desconectado, ya sea un Concepto Secundario Símbolo o un Concepto no Símbolo. 2. De corresponder, reescribir las relaciones derivadas del Concepto Secundario Símbolo o del Concepto no Símbolo.		
Tratamiento sugerido en el LEL		
Reescribir la oración en la noción o impacto del símbolo bajo estudio en función del mapa conceptual corregido.		

Fig. 4. Ejemplo de un paso de análisis y un mapa conceptual con el defecto a buscar

4 Diseño del Experimento

Se realizó un estudio empírico de las tres variantes de inspección antes descritas, aplicadas sobre el modelo léxico, con el fin de poder establecer qué variantes pueden ser más eficaces para capturar determinados defectos y qué esfuerzo requieren.

En este estudio se utilizaron 5 glosarios distintos, los que habían sido construidos en contextos de sistemas de información en organizaciones argentinas, aunque desarrollados como parte de un curso de grado. Cada glosario fue inspeccionado usando cada una de las 3 variantes de inspección. A cada inspector se le asignó un glosario y una variante de inspección a aplicar. Los inspectores eran estudiantes de grado, independientes de los autores del LEL.

Tanto en la Preparación como en la Reunión de cada inspección, el inspector tomó datos detallados de tiempos, cantidad de defectos, tipos de defectos y severidad, por cada ítem de control, formulario o paso según la variante de inspección que se estaba utilizando. En cada Reunión, participaron el inspector, quien cumplía también el rol de escriba, los autores del LEL y el moderador, quien era un docente del curso. En la Tabla 1 se exponen los datos de cada inspección referidos a defectos detectados en la Preparación, confirmados en la Reunión, el grado de confirmación, defectos confirmados por tipo de defecto y severidad, y el tiempo total insumido.

Respecto a las amenazas a la validez interna del experimento, los inspectores seleccionados eran estudiantes de grado del curso Ingeniería de Requisitos de distintas universidades y ciclos lectivos. Todos los inspectores tenían conocimiento y práctica previo sobre las características y el proceso de construcción del modelo LEL. Los LELs utilizados se crearon con anterioridad al experimento, por otros estudiantes, siguiendo el mismo proceso, independiente del experimento. Los inspectores tuvieron una capacitación de 2 horas sobre la técnica de inspección y la variante a aplicar, y recibieron la documentación necesaria para aplicar la técnica y registrar los datos

necesarios. En las pautas de las 3 variantes, estaba estipulado el tipo y nivel de severidad para cada defecto a capturar, evitando la interpretación del inspector y, por ende, el desvío en las métricas a utilizar para el estudio comparativo. Las 3 variantes habían sido probadas previamente en otros LELs y calibradas para calificar de la misma manera cada defecto.

Tabla 1. Defectos capturados y tiempo por inspección

LEL #	Variante de Inspección	Cantidad Símbolos	Tiempo HH:MM	Defectos en Preparación	Defectos Confirmados	Confirmados / Preparación	TIPO DE DEFECTO				SEVERIDAD		
							Discrep.	Error	Omisión	Ambig.	Alta	Media	Baja
1	Checklist	22	09:29	48	22	45,8%	0	7	5	10	0	14	8
1	Formularios	22	10:39	27	17	63,0%	0	3	14	0	1	14	2
1	Mapas	22	06:55	68	50	73,5%	0	0	25	25	14	36	0
2	Checklist	27	04:01	50	49	98,0%	0	14	22	13	18	27	4
2	Formularios	27	07:12	57	52	91,2%	0	4	37	11	1	38	13
2	Mapas	27	09:37	91	70	76,9%	0	2	41	27	1	61	8
3	Checklist	43	03:36	127	94	74,0%	2	27	51	14	9	33	52
3	Formularios	43	06:14	115	92	80,0%	4	13	73	2	23	28	41
3	Mapas	43	15:45	90	81	90,0%	0	0	59	22	12	37	32
4	Checklist	27	03:40	66	48	72,7%	0	5	13	30	33	10	5
4	Formularios	27	10:35	24	25	104,2%	1	2	17	5	9	2	14
4	Mapas	27	05:47	57	62	108,8%	0	0	35	27	21	14	27
5	Checklist	30	05:39	44	31	70,5%	0	2	22	7	2	27	2
5	Formularios	30	05:14	45	42	93,3%	0	1	24	17	6	12	24
5	Mapas	30	10:56	54	48	88,9%	0	0	25	23	5	41	2

Respecto a las amenazas a la validez externa, los sujetos del experimento son solo representativos de inspectores sin experiencia en inspecciones y con escasa experiencia en la IR. Esto implica que los resultados obtenidos no son representativos de las inspecciones en la industria, pero pueden ser válidas si solo se dispone de inspectores novatos. El material inspeccionado eran LELs elaborados, a su vez, por estudiantes sin experiencia en su construcción, por lo que los resultados pueden no ser representativos de la cantidad de defectos reales que pueden aparecer en este tipo de modelo construido por profesionales con experiencia en IR. En cuanto al tamaño de los LELs (22 a 43 símbolos) corresponden a proyectos de software pequeños a medianos, típicos de pymes. El proceso de inspección utilizado fue propuesto y evaluado en la literatura [2,4,16]. El número de inspecciones analizadas (15) indica una muestra pequeña para poder generalizar algunos resultados. En base a estas consideraciones, se asume que este experimento podría ser replicado, obteniendo nuevos resultados que amplíen la muestra presentada en este trabajo, permitiendo alcanzar resultados más precisos o ajustados a diferentes realidades de la industria.

5 Análisis de Resultados

En un análisis descriptivo de los resultados generales obtenidos de las inspecciones, se observa que la VMC aunque requiere más tiempo por símbolo (19 min) es la que detecta en promedio más defectos por símbolo (2,13), mientras que por el contrario la VFR, que lleva un esfuerzo en tiempo un poco menor (17 min) que la VMC, detecta a su vez un número mucho menor de defectos por símbolo (1,43) (ver Fig. 5-A y B). Esto hace evidente que en promedio el tiempo para detectar un defecto en la VFR es 1,7 veces superior que en las otras variantes (ver Fig. 5-C), aunque entre esas 2 variantes la VMC detecta más defectos por símbolo que la VCH (ver Fig. 5-A). En

promedio, la VMC insume 1,6 veces más tiempo por símbolo que la VCH y apenas 1,1 más que la VFR (ver Fig. 5-B), lo que debería tenerse en cuenta al establecer tiempos a dedicar a una inspección.

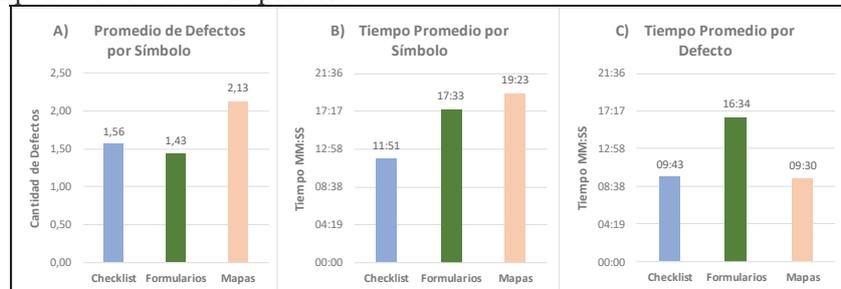


Fig. 5. Valores promedio de defectos y tiempos por variante de inspección

En particular, en 3 inspecciones la VMC llevó más tiempo de inspección que las otras 2 variantes, mientras que la VCH insumió menos tiempo en 3 inspecciones (ver Fig. 6-A), e insumió también menos tiempo por defecto en las mismas 3 inspecciones (ver Fig. 6-B). Sin embargo, debe notarse que tanto la VCH como la VFR presentan una dispersión excesivamente alta en la tasa promedio de tiempo requerido por símbolo (coeficientes de variación: 0,98 y 0,87 respectivamente), y también alta dispersión en la tasa promedio de tiempo por defecto (coeficientes de variación: 0,69 y 0,49 respectivamente). Esto se visualiza claramente en la Fig. 6, donde se muestran dichas tasas para cada inspección. Estas dispersiones, en cierta medida, debilitan las observaciones previas. Cabe mencionar que la VMC, por el contrario, tuvo para ambos tipos de tasas coeficientes de variación bajos; en la Fig. 6 se observan tasas de tiempo relativamente similares en 4 de las 5 inspecciones de VMC. Esto permitiría realizar proyecciones de tiempo según el tamaño del LEL.

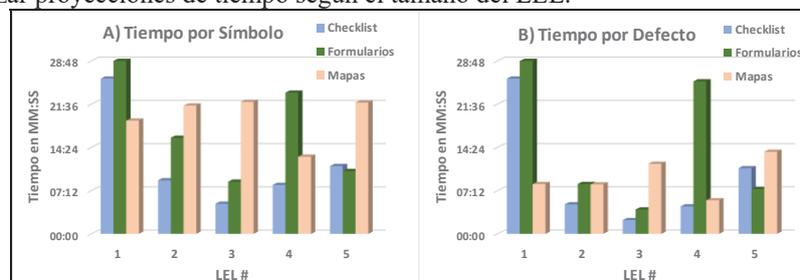


Fig. 6. Tasas de tiempos por LEL inspeccionado

En la Fig. 7-A se exponen los defectos confirmados en la Reunión de Inspección, donde se observa que en 4 de los 5 LELs la VMC tuvo las mayores tasas de detección por símbolo, llegando a duplicar dicha tasa en 2 LELs. Todas las inspecciones de la VMC superaron la tasa de detección del 1.60 defectos por símbolo, mientras que la VFR por el contrario tuvo 3 inspecciones por debajo de esa tasa y la VCH tuvo 2 inspecciones por debajo.

Por otro lado, las 3 variantes presentan una buena tasa promedio de defectos confirmados respecto a defectos capturados en la Preparación (con bajos coeficientes de variación): en la VFR y la VMC los autores confirmaron en promedio más del 86%

de los defectos, y en la VCH confirmaron en promedio el 72%. Analizando cada inspección, en 4 LELs el porcentaje de defectos confirmados respecto a capturados en a Preparación fue superior para VFR y VMC (ver Fig.7-B). Esto indica que VCH podría ser proclive a levantar defectos que no son tales, pudiendo considerarse que las otras 2 variantes son más precisas en guiar hacia la detección de defectos reales.

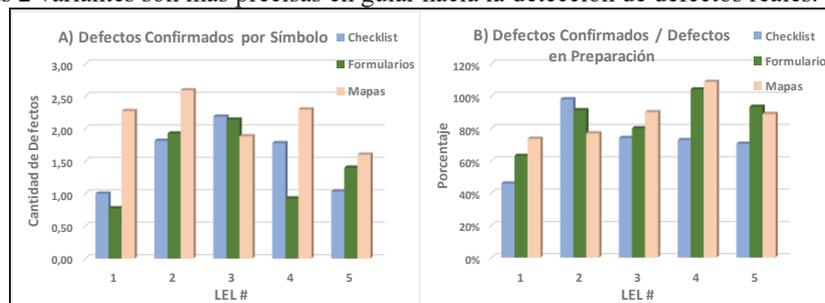


Fig. 7. Tasas de defectos por LEL inspeccionado

Otra métrica utilizada es la tasa de defectos detectados por símbolo según su nivel de severidad. Se observa una gran variación en el número de defectos por severidad entre los LELs y respecto de las variantes (ver Fig. 8), principalmente para defectos de alta y baja severidad. Para severidad media (ver Fig. 8-B), la VMC superó a las otras variantes en los 5 LELs, duplicando la tasa de detección en 2 de ellos y detectando 1,5 veces más defectos en otros 2 LELs.

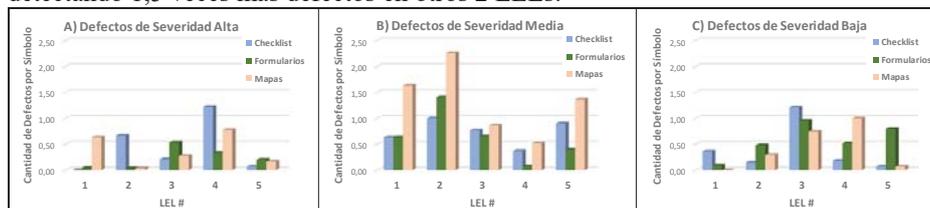


Fig. 8. Defectos confirmados por símbolo según nivel de severidad

Las 3 variantes están diseñadas para detectar omisiones y ambigüedades principalmente, en menor medida errores y solo algunos tipos de discrepancias, debido a las características del LEL y a qué defectos pueden capturarse en una verificación. Es por ello, que en los 5 LELs, las tasas de discrepancias y de errores detectados son muy bajas. Solo 2 inspecciones de la VFR y 1 inspección de la VCH detectaron Discrepancias (tasas ≤ 0.09 defectos por símbolo). La VCH fue la variante que logró tasas más altas de detección de Errores en los 5 LELs, duplicando (4 LELs) y hasta triplicando (1 LEL) el número de errores respecto a la VFR, siendo que la VMC solo detectó errores en un LEL y con una tasa muy baja (ver Fig. 9-A). La VMC detectó más omisiones que las otras variantes en 4 LELs, capturando en promedio 1,23 omisiones por símbolo. La VCH y la VFR tuvieron una variación significativa en sus tasas de detección respecto a los 5 LELs, siendo la VCH la que detectó la menor cantidad de omisiones en los 5 LELs (ver Fig. 9-B). Respecto a ambigüedades, al igual que en omisiones, la VMC detectó más cantidad de ambigüedades que las otras variantes en 4 LELs, capturando en promedio 0,88 por símbolo, duplicando la tasa en 3 LELs respecto a la VCH, y siendo la VFR la que detectó menos ambigüedades en los 5 LELs (ver Fig. 9-C). También en este tipo de

defecto la VCH y la VFR tuvieron una variación significativa en su tasa de detección. En resumen, se observa mejor posicionada a la VMC para la detección de omisiones y ambigüedades, prestando menos atención en capturar discrepancias y errores. Las otras variantes están más preparadas para identificar estos dos tipos de defectos, en desmedro de omisiones y ambigüedades.

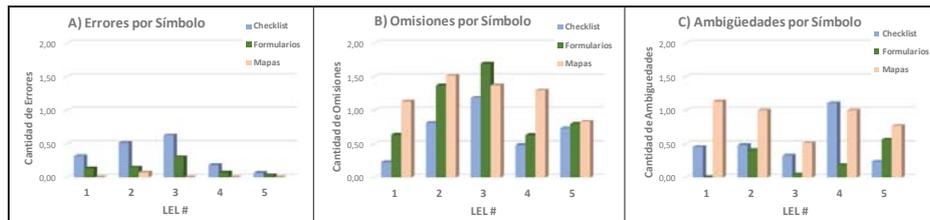


Fig. 9. Tipos de defectos confirmados por símbolo

6 Conclusiones

Es conocido que las inspecciones de software detectan mayor cantidad de defectos que las pruebas, aunque insumen más tiempo [10,11], también se ha probado su eficacia en modelos de la IR [1,3,5,6], proponiéndose distintas maneras de captura defectos. Es por ello que se ha llevado a cabo un experimento para establecer la efectividad de 3 variantes de inspección utilizando 5 léxicos.

En base al análisis de los resultados se observa que la variante basada en lectura constructiva (creando mapas conceptuales) logra tasas de detección de defectos muy superiores a las otras dos variantes (lectura con checklist y lectura basada en procedimientos), pero insumiendo más tiempo, mientras que la lectura con checklist lleva en promedio menos tiempo, pero captura menos defectos. Esto corrobora los resultados presentados en [2,5] sobre VCH y VFR. Por otro lado, debe notarse que en el caso de la lectura con checklist y la basada en procedimientos, los tiempos de inspección tuvieron alta dispersión respecto de los 5 LELs en este estudio empírico acotado, dado lo cual resulta difícil poder estimar tiempos a insumir en una inspección de este modelo para cualquiera de estas 2 variantes. Las 3 variantes tuvieron buenas tasas de confirmación de defectos en la reunión de inspección, siendo la variante con checklist la que en 4 LELs tuvo más rechazos que las otras 2, lo cual estaría indicando que brinda menos apoyo para identificar defectos. Respecto a la severidad de defectos, se observa una alta disparidad entre los 5 LELs y las 3 variantes utilizadas; solo se puede señalar que la variante constructiva tuvo tasas muy superiores de detección de defectos de mediana severidad frente a las otras. Se observa que la variante con checklist estaría más preparada para detectar errores, mientras que la variante constructiva estaría más enfocada en detectar omisiones y ambigüedades frente a las otras variantes. Estas observaciones están basadas en evidencia empírica proveniente de una muestra pequeña de 5 inspecciones por variante, realizadas por inspectores novatos sobre los mismos 5 LELs, creados también por novatos, por lo que estos resultados solo pueden dar indicios de cómo podrían capturarse ciertos defectos de una manera más eficaz, teniendo inspectores inexpertos y considerando los tiempos disponibles en un dado proyecto de software.

Para precisar los resultados obtenidos, se prevé replicar este experimento sobre LELs de mayor tamaño, y extender el experimento al modelo de escenarios o casos de uso con las variantes estudiadas en este trabajo. Aunque existen estudios sobre el impacto de la experiencia del inspector en la efectividad de algunas variantes [2,3,6], casi no se ha estudiado para la lectura constructiva, lo cual es un camino a seguir.

Referencias

1. Regnell, B., Runeson, P., Thelin, T.: Are the perspectives really different? Further experimentation on scenario-based reading of requirements. *Empirical Software Engineering*, 5(4), 331-356 (2000).
2. Porter, A.A., Votta, L.G.: Comparing Detection Methods for Software Requirements Inspections: A Replication Using Professional Subjects. *Empirical Software Engineering*, 3(4), 355-380 (1998).
3. Albayrak, Ö., Carver, J.C.: Investigation of individual factors impacting the effectiveness of requirements inspections: a replicated experiment. *Empirical Software Engineering*, 19(1), 241-266 (2014).
4. Leite, J.C.S.P., Doorn, J.H., Hadad, G.D.S., Kaplan, G.N.: Scenario Inspections, *Requirements Engineering Journal*, 10(1), 1-21. Springer-Verlag, Londres (2005).
5. Paech, B., Denger, C., Kerkow, D., Von Knethen, A.: Requirements Engineering for Technical Products: Integrating Specification, Validation and Change Management. En: Maté Silva (eds) *Requirements Engineering for Sociotechnical Systems*, IGI-Global (2005).
6. Armijos, A.F., Monsalve, C., Ullón, R.H., Maya, R.D., Romero, J.A.: Calidad en la inspección de requerimientos de software: una propuesta de inspección. En: XIX Ibero-American Conference on Software Engineering, Quito, Ecuador (2016).
7. Leite, J.C.S.P., Doorn, J.H., Kaplan, G.N., Hadad, G.D.S., Ridaio, M.N.: Defining System Context using Scenarios. En: J.C.S.P. Leite, J.H. Doorn (eds.) *Perspectives on Software Requirements*, pp. 169-199. Kluwer Academic Publishers, Springer US (2004).
8. Antonelli, L., Rossi, G., Leite, J.C.S.P., Oliveros, A.: Deriving requirements specifications from the application domain language captured by Language Extended Lexicon. En: 15th Workshop on Requirements Engineering, Buenos Aires (2012).
9. Fagan, M.E.: Design and Code Inspections to reduce Errors in Program Development, *IBM Systems Journal*, 15(3), 182-211 (1976).
10. Laitenberger, O., Debaud, J.M.: An Encompassing Life-Cycle Centric Survey of Software Inspection. *Journal of Systems and Software*, 50(1), 5-31 (2000).
11. Jones, C.: Measuring defect potentials and defect removal efficiency. *CrossTalk, The Journal of Defense Software Engineering*, 21(6), 11-13 (2008).
12. Ciolkowski C., Laitenberger, O., Biffel, S.: Software reviews: The state of the practice, *IEEE software*, 20(6), 46-51 (2003).
13. Ben Achour, C., Rolland, C., Maiden, N.A.M., Souveyet, C.: Guiding Use Case Authoring: Results of an Empirical Study. En: *International Symposium on Requirements Engineering*, Limerick, Irlanda, IEEE Computer Society Press, pp. 36-43 (1999).
14. Ridaio, M., Doorn, J.H.: Estimación de Completitud en Modelos de Requisitos Basados en Lenguaje Natural. En: 9th Workshop on Requirements Engineering, pp.151-158 (2006).
15. Hadad, G.D., Litvak, C., Doorn, J.: Problemas y Soluciones en la Completitud de Modelos en Lenguaje Natural. En: II Congreso Argentino de Ingeniería, Tucumán, T366 (2014).
16. Kaplan, G., Hadad, G., Doorn, J., Leite, J.: Inspección del Léxico Extendido del Lenguaje. En: 3rd Workshop on Requirements Engineering, Río de Janeiro, Brazil, pp.70-91 (2000).
17. Sebastián, A., Hadad, G.: Mejoras a un Modelo Léxico mediante Mapas Conceptuales. En: XXI Congreso Argentino de Ciencia de la Computación, Buenos Aires, pp.526-535 (2015).

Integración de funcionalidades BPM en Portales Web

Anahí Soledad Rodríguez¹, Patricia Bazan¹, José Martínez Garro², Juan Pablo Pérez Bergues², Francisco Javier Díaz¹.

¹ LINTI – ²Facultad de Informática – UNLP,
arodriguez@linti.unlp.edu.ar, pbaz@info.unlp.edu.ar,
josemartinezzgarro@yahoo.com.ar, jpablop@linti.unlp.edu.ar, jdiaz@info.unlp.edu.ar.

Abstract. Los Portales Web han ido evolucionando y han aportado herramientas de gestión para desarrollar portales de internet que ofrezcan al usuario un acceso unificado a servicios y recursos. BPM (*Business Process Management*), por su parte, es hoy un paradigma maduro que cuenta con herramientas estables para su aplicación en las organizaciones. Una aplicación muy habitual de BPM es proveer una solución para orquestar y componer servicios y que, además, debe integrarse otras soluciones de la organización. Este trabajo propone la definición de un marco de trabajo y un modelo arquitectónico para integrar una solución BPM en un Portal Web. A su vez, se adopta el concepto de *portlet* como elemento de construcción de los Portales Web y su vinculación funcional con los gestores de procesos de negocio.

Keywords: Portal web, Portal BPM, web Service, Portlet, BPM.

1 Introducción

La evolución de los sitios web ha ido creciendo acorde al del avance de la tecnología. Los sitios y Portales Web no solo son utilizados para brindar información, sino que, a medida que fueron evolucionando, comenzaron a presentar funcionalidades más complejas, siendo utilizados también como dispositivos de publicidad para cualquier tipo de empresa y organización. A fines de los '80 Tim Berners-Lee escribió la primera propuesta del proyecto *World Wide Web* (WWW). El propósito del proyecto era favorecer el intercambio de información científica entre universidades e institutos del mundo. A principios de los 90, junto a Robert Cailliau se formalizó el proyecto "*WorldWideWeb*", en el cual varios documentos de hipertexto podían ser visitados por aplicaciones llamadas navegadores web. Luego Tim Berners-Lee logró hacer funcionar el primer servidor y navegador web en el *CERN* en una computadora *NeXT*. Dicho navegador web permitía la edición de la página web desde su interfaz, y presentaba muchas características que hoy en día tienen los navegadores web. [1]. Luego a mediados de los '90 aparecieron los navegadores web *Mosaic*, *Netscape*, *Opera* e *Internet Explorer* lo que hizo que la utilización de Internet sea más masiva. Los sitios web comenzaron a ser más atractivos con la utilización de videos y animaciones logradas con la aparición de la plataforma de multimedia Adobe Flash [2], con las nuevas versiones del lenguaje HTML, la utilización de *javascript* y *applet* de java. A partir del año 2000 la interfaz de las páginas web comenzó a complejizarse con elementos gráficos y aplicaciones que brindan servicios. Con el aumento de

ancho de banda en los servicios de internet, se sumó la posibilidad de realizar *streaming* de video y audio en tiempo real. El uso de dispositivos móviles hizo crecer en diseño y navegabilidad a los sitios web ambos soportados por lenguajes de programación como *HTML5*, *javascript*, *json* y *nodejs* [3] [41]. Esta evolución tecnológica de las aplicaciones Web ha alcanzado a las organizaciones que, actualmente cuentan con presencia en la Web no solo con portales de información, sino con aplicaciones que acercan los servicios de las organizaciones a su público, representado por clientes tanto externos como internos. Por su parte, BPM ha cobrado gran importancia los últimos años en las organizaciones, dejando de ser solo un producto para convertirse en un paradigma de servicios y soluciones. La adopción de BPM en las organizaciones se encuentra en un grado de madurez avanzado, y brinda beneficios que se ven reflejados en las mejoras en la productividad. Una aplicación muy habitual de BPM es la de orquestar y componer servicios y su integración con las soluciones de la organización. Este trabajo propone la definición de un marco de trabajo y un modelo arquitectónico para integrar una solución BPM en un Portal Web. A su vez, se adopta el concepto de *portlet* como elemento para construcción de los Portales Web integrados a los gestores de procesos de negocio.

El presente trabajo se organiza de la siguiente manera en la Sección 2 se presentan una revisión de la bibliografía de trabajos relacionados con la presente investigación. En la Sección 3 se introducen conceptos, definiciones y características de Portales Web y *portlet*, aportando las bases conceptuales y metodológicas que sustentan la propuesta de este trabajo. En la sección 4 se propone un marco de trabajo y un modelo de arquitectura para lograr con eficiencia y eficacia la integración de una solución BPM a un Portal Web. Finalmente se arriban algunas conclusiones.

2 Trabajos Relacionados

La revisión bibliográfica dejó en evidencia que la investigación en el tema lleva varios años y permitió obtener definiciones y un marco conceptual relacionados con los temas principales del presente trabajo. La búsqueda de trabajos relacionados se realizó en *Scopus*, *IEEE Xplore Digital Library*, *Google* y *Google Scholar*, con las palabras claves “Portal Web”, “portales verticales y horizontales”, “herramientas para la construcción de portales”, “portlet”, “Portal BPM”. Los trabajos analizados se agrupan entre los que aportan cuestiones conceptuales y los que aportan enfoques metodológicos que validan nuestra propuesta.

En [4] [5] [6] [7] se abordan los conceptos básicos, características relacionadas con Portales Web y Sitios Web, aportando las bases conceptuales y definiciones para el desarrollo de este trabajo. En [8] se presenta una clasificación de Portales Web en función del público al que van dirigidos, y de los contenidos que pretenden cubrir, como ser: portales especializados, corporativos y generales, en comparativa a los presentado en [9] [10] en el cual se definen como verticales, horizontales y diagonales, estos temas son tratados en la Sección 3.

En [11] queda de manifiesto la necesidad de brindar funcionalidades de distintas aplicaciones a través de servicios web en una arquitectura orientada a servicios (SOA) y valida la necesidad de la combinación de tecnologías BPM con servicios web. Al

igual que en [12], se plantean los requerimientos básicos para definir componentes de la interfaz para presentar funcionalidades BPM en SOA, presentando una posible implementación con *portlet*. Por último, un trabajo con gran relevancia es [37] el cual presenta la posibilidad de tener un modelo de datos unificado y un acceso genérico a la funcionalidad del motor BPMS. Este trabajo ha sido un importante aporte para nuestra propuesta metodológica aportando un modelo y acceso genérico a las funcionalidades de un motor de BPMS para ser brindadas en un portal.

3 Marco Conceptual - Definiciones

Los Portales Web son sitios web que proveen un único punto de acceso a información y funcionalidades, que pueden estar dirigidos a un usuario autenticado del portal o a un público en general. Su objetivo es que el usuario disponga de información personalizada y funcionalidades provistas por el mismo en un solo lugar [8]. En cambio, los sitios web clásicos están dirigidos al público en general brindando información y gestión de contenidos [13] [14]. Las siguientes categorías definen características que ayudan a diferenciar los Portales Web de los sitios Web que son comparadas en la Tabla 1 [15] [16]:

- **Audiencia:** se refiere al conjunto de usuarios a los cuales está dirigida la información expuesta en el Portal o sitio Web. Pueden clasificarse a su vez en horizontales – dirigidos a un cierto grupo de interés de índole general-, verticales – orientados a usuarios autenticados- y diagonales – combinación de anteriores como las redes sociales [8] [9] [10].
- **Objetivo:** se refiere al propósito general y específicos para cumplir las metas propuestas en el sitio Web o portal.
- **Comunicación con el usuario:** se refiere a la interacción entre el sitio Web o portal y el usuario, en cómo se desarrolla la interacción.
- **Seguridad:** se refiere al control de acceso y permisos de los usuarios en el portal Web o portal, puede ser de libre acceso o con control de autenticación. Puede ser público o de acceso privado.
- **Contenido:** se refiere a la visibilidad y tipo de información o contenido ofrecido por el sistema (sitio Web o portal).
- **Actualización de información:** se refiere a la frecuencia de actualización de la información del sistema.

3.1 Definición y Características de *portlet*

Los *portlets* son componentes activos de los Portales Web, se los asemeja a las ventanas de los sistemas operativos por las funcionalidades que presentan, dado que cada *portlet* dentro de la página del portal cuenta con su propio título y conjunto de botones para poder maximizar, minimizar entre otras preferencias visuales propias de una ventana de una interfaz [17]. Los Portales Web también pueden ofrecer funcionalidades a través de APIs con comunicación con Servicios Web. Esta tecnología no alcanza para la interacción compleja de flujos que se extienden a lo

largo de varias páginas web, ya que devuelven datos sin procesar. Otra desventaja es que la operatoria de cómo se solicitan los parámetros o cómo se devuelven, más el hecho de saber cómo procesarlos queda en la responsabilidad del cliente [18] [19]. La utilización del estándar JAVA 3.0 (JSR-362) permite la interoperabilidad de *portlet* en diferentes Sitios Web, logrando reutilizar estos componentes [20]. Otra propuesta para facilitar funcionalidades en los portales surge a través de *servlets*, que son componentes en lenguaje JAVA que procesan las peticiones de un navegador web y retornan una respuesta [21].

Tabla 1 - Comparación de características

Característica	Sitio Web	Portal Web
Audiencia	Usuario con un determinado rol o perfil	Público en general
Objetivo	Plataforma para integrar funcionalidades, sistemas e información	Gestor de contenidos de páginas Web
Comunicación con usuario	Bidireccional	Unidireccional
Seguridad	Control de acceso a las funcionalidades según perfil, garantizada por la plataforma	Las restricciones de acceso que hubiere debe ser programadas ad-hoc.
Contenido	Colección de páginas web relacionadas	Conjunto de recursos y servicios
Actualización de Información	En general la actualización es frecuente	En general la actualización es escasa o poco frecuente

Las similitudes *servlets* y *portlets* se organizan en torno a [22] [17]:

- Lenguaje de desarrollo: se desarrollan en JAVA.
- Entorno de despliegue: utilizan un descriptor de despliegue basado en XML.
- Entorno de ejecución: cuentan con un contenedor especializado que gestiona el ciclo de vida.
- Mecanismo de interacción: utilizan el paradigma petición/respuesta sincrónico.

Por otra parte, las diferencias entre *portlets* y *servlets* [23] son:

- El ciclo de vida del *portlet* es: *init - render - process action - resource action - process event - destroy*, en cambio los *servlets* es: *init - serve - destroy*.
- Los *portlets* generan un fragmento de un documento mientras que los *servlets* generan un documento completo.
- Los *portlets* no se acceden directamente a través de la URL.
- Los *servlets* no se comunican entre sí.

La creación de un portal con la utilización de *portlets* permite construir los portales de manera modular abstrayendo los aspectos gráficos de los estrictamente funcionales, como por ejemplo la estructura de los contenidos respecto de la seguridad y autenticación de usuarios [24].

3.2 Herramientas para la construcción de portales

Las herramientas para la creación de Portales Web permiten crear y gestionar fácilmente los contenidos, la creación y organización de la interfaz, administrar usuarios, entre otras funcionalidades brindadas por los mismos.

Existe un número considerable de herramientas adaptadas a las necesidades, requerimientos y perfiles de distintos negocios, empresas u organizaciones, que pueden ser utilizadas para crear Portales Web.

A su vez, resulta interesante analizarlas a la luz de un conjunto de categorías que las definen y encuadran. Las cuales son: el tipo de licencia, la disponibilidad de código, la utilización de *portlets* y la última actualización.

- **Tipo de licencias:** las licencias son un contrato entre el autor del programa y el usuario, siguiendo una serie de términos y cláusulas que el usuario debe cumplir para su uso.
- **Disponibilidad del código:** dependiendo de la licencia se puede determinar si el código está disponible para su redistribución/modificación. Esto puede ser útil para posibles actualizaciones o modificaciones que se necesiten para la utilización dentro de una organización para implementar una funcionalidad específica.
- **Utilización de portlets:** los *portlets* son componentes Web que se generan de manera dinámica, ante la petición de un usuario [26].
- **Última actualización:** contar con la fecha de la última actualización del software es importante para saber si se cuenta con actualizaciones y soporte del software por parte de la comunidad o de la empresa creadora.

Los proveedores de herramientas para la creación de portales están siguiendo estándares y prácticas comunes favoreciendo la flexibilidad, y que el desarrollo de los portales sea más fácil y sencillo.

En la actualidad existen diferentes herramientas para la creación de Portales Web, la elección de la misma dependerá de las funcionalidades y necesidades del portal que se necesite utilizar en la organización. Algunas de las herramientas más populares [25], mejor posicionadas y analizadas en el reporte “*Magic Quadrant for Horizontal Portals*” [27], son comparadas en la Tabla 2 según las características definidas anteriormente.

Tabla 2 - Comparación de Herramientas

Aplicación/ herramienta	Licencia	Disponibilidad del código	Utiliza portlet	Última actualización
OpenPortal [28]	Licencia Pública General de GNU	Si	Si	20/02/2001
UPortal [29]	Licencia Apache, Versión 2.0	Si	Si	23/04/2019
Adobe Muse [30]	Privada	No	No	01/03/2018

Drupal [31]	Licencia Pública General de GNU, versión 2 o posterior	Si	No	01/05/2019
Liferay Portal [32]	Liferay Portal CE License [5] (Licencia Pública General de GNU, versión 2)	Si	Si	24/04/2019
Jetspeed [33]	Apache License Version 2.	Si	Si	09/05/2016
GetIn [34]	LGPL	Si	Si	21/10/2013
IBM WebSphere Portal [35]	Propietaria	No	Si	25/09/2018

4 Enfoques metodológicos para integrar funcionalidades BPM en portales WEB

Las funcionalidades propias de la gestión de procesos de negocio incorporadas a los Portales Web pueden aumentar la productividad de toda organización, así como proporcionar otras ventajas propias de la gestión de procesos de negocio. Para considerar la integración de BPM con un Portal Web se deben tener en cuenta las siguientes consideraciones [36]:

- **Integración del portal web con las funcionales del BPMS.** La integración del portal web con el motor de BPMS se puede realizar a través de *portlets* los cuales se comunicarán y extraerán información a través de la API proporcionada por el BPMS. Al realizar dicha integración a través de *portlets* se tiene la ventaja de contar con toda la funcionalidad provista por el entorno de ejecución (*portlet container*), encargado del ciclo de vida de las instancias del *portlet*. Otra manera de integrar al portal web con el BPMS, es a través de por ejemplo un *iFrame* dentro de las páginas web que pueden consumir funcionalidades y datos del BPMS a través de la API, pero perdiendo las funcionalidades ya provistas por el entorno de ejecución del contenedor de *portlet* como son la seguridad, personalización, etc. En [37] se propone la integración entre el portal web y el motor de BPMS a través de la API REST / SOAP disponible en el BPMS. Los inconvenientes encontrados en el trabajo presentado radican en la rapidez de incorporación de cambios en la comunicación con el BPMS, por lo que se propone una capa intermedia la cual generalice la comunicación con diferentes BPMS.
- **Contenido: Audiencia y Función.** Esta característica la podemos ver cómo los actores y funcionalidades del BPMS, con los cuales se tiene como evidencia que

dependiendo del actor autenticado en el portal se puede determinar el tipo de tareas y funcionalidades que tiene este tiene asociadas. Por lo cual el contenido y la funcionalidad que se presenten en el portal dependen pura y exclusivamente del tipo de usuario al cual estén dirigidos.

- **Modo de acceso: Externo vs. Interno.** Un portal con modo de acceso externo está dirigido a público en general sin la necesidad de estar autenticado dentro del portal. En contraposición de un portal interno el cual está dirigido por ejemplo a los empleados de una empresa o también a personas pertenecientes a una institución. En el caso de un portal con funcionalidades del gestor de procesos del tipo externo se debería brindar información a usuarios generales, y podría tener alguna funcionalidad brindada a través de un servicio web (ofrecer servicios a usuarios sin acceso) orquestado por los procesos de negocio de la organización. Para el caso de los actores/usuarios internos se accedería a las funcionalidades para el usuario que esté autenticado a través de las cuales tenga acceso a funciones netamente del proceso, como ser la ejecución de tareas, alertas de tareas pendientes, y funcionalidades provistas por un social BPM con esto se permite que los distintos actores involucrados en el proceso de negocio trabajen colaborativamente en el mismo a través por ejemplo, de comentarios, notificaciones, tareas etiquetadas, presentadas en [38] o chats inmersos en la ejecución del proceso de negocio.

- **Autenticación Gerencial vs Autenticación operativa.** En esta característica podemos tener o bien una puerta de entrada con información dirigida al actor (usuario) autenticado, por lo cual el actor entrante podría ejecutar tareas del proceso y poder visualizar funcionalidades de negocio de otros servicios brindados en el portal. Para la otra característica de Panel de control el usuario autenticado podría tener acceso a los resultados de la ejecución de los procesos, de tareas, diferente información de indicadores establecidos por la gerencia y que podría tener un impacto en el negocio.

Las características descritas permiten construir Portales Web funcionalmente más completos y que logran sumar en sus funcionalidades, las provistas por un gestor de BPM. Por otro lado, las capacidades de autenticación y modo de acceso, aumentan la posible audiencia de la gestión por procesos, usualmente destinada a un público interno.

4.1 Modelo Arquitectónico

La arquitectura propuesta integra componentes de manera desacoplada y comunicadas a través de un elemento genérico como es el API Gateway. Como se muestra en la Figura 1, la capa de *frontend* resuelve la interfaz de usuario, la conforman los portales web que incluyen los *portlets* para el acceso a los servicios del *backend*. La introducción de un componente de API Gateway en la capa *middleware* permite, además de orquestar los servicios, una independencia, ya sea de protocolo o parámetros, entre los servicios brindados en la capa de *backend* y la utilización de los mismos por parte de los *portlets*. Una API Gateway permite también brindar autenticación y autorización independiente de servicios, *caching* en casos necesarios, entre otros beneficios [39]. Por otro lado se encuentran los componentes del *backend*:

servidor de *Portlet*, el contenedor de *Portlet*, contenedor de servidor, motor de BPMS, sistemas de ECM (Enterprise Content Management) y gestores de documentos, los cuales consumen los datos ofrecidos por el sistema de información para ofrecerlos mediante servicios a través del API Gateway [40].

El motor de BPM en el *backend* constituye un elemento de orquestación de servicios junto con la gestión del flujo de actividades que se refleja en la consola de usuario conteniendo la lista de actividades a realizar según su rol.

Los sistemas de ECM y los gestores de documentos fueron incluidos en la arquitectura dado que son componentes útiles para las funcionalidades propuestas en una organización.

El sistema de información en el *backend* contiene los datos que son consumidos a nivel gerencial y de toma de decisiones desde el portal y que también es accedido a través de *portlets*.

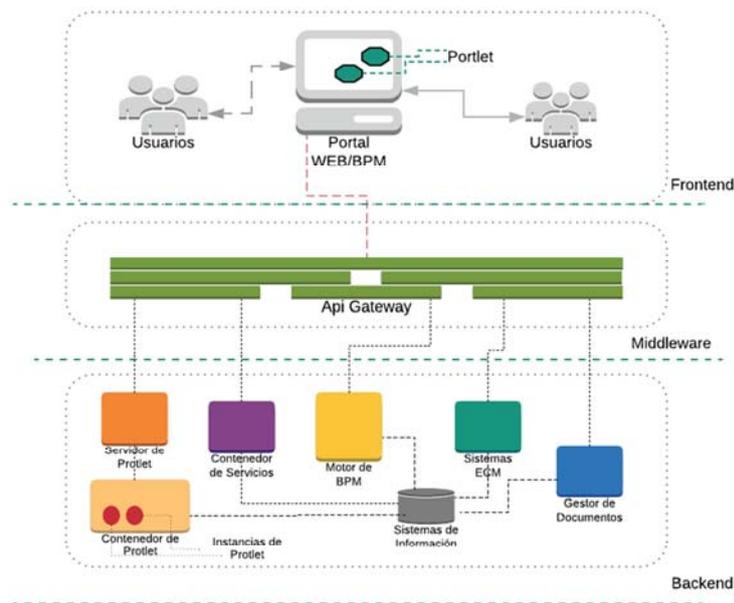


Figura 1 – Modelo Arquitectónico para Integrar BPM con Portales Web

5 Conclusiones

La adopción del paradigma BPM brinda varios beneficios en las organizaciones. La integración este paradigma en Portales Web puede ofrecer a los usuarios un acceso unificado a aplicaciones y recursos.

En este trabajo se presenta el marco conceptual de Portales Web, para luego analizar y evaluar sus características en las herramientas existentes. Este análisis pretende definir criterios comparativos que puedan ser tenidos en cuenta a la hora de elegir soluciones para el despliegue de Portales Web. Asimismo, se define un marco de trabajo y un modelo arquitectónico para la integración de Portales WEB y soluciones BPM. Esta arquitectura se sustenta en el uso del concepto de API Gateway, como un componente que organiza y comunica distintas funcionalidades de las aplicaciones, construidas con enfoque de orientación a servicios orquestados por proceso de negocio.

Referencias

1. CERN, (Consultado Mayo 2019) <https://home.cern/science/computing/birth-web/short-history-web>
2. Adobe Flash, (Consultado Mayo 2019) <https://get.adobe.com/es/flashplayer/about/>
3. La evolución de la Web, (Consultado Mayo 2019), <http://www.evolutionoftheweb.com/>
4. Usability testing on government agencies web portal: A study on Ministry of Education Malaysia (MOE) web portal” Education Malaysia.pdf.
5. Avila, G., Sánchez, A., de Zayas, L. M. R., Jiménez, P., & Basnuevo, M. (2010). El Portal Holguín una ventana al mundo por la red de redes. Ciencias Holguín, 10(4).
6. Burgess, S., Bingley, S., & Tatnall, A. (2005). Matching the revenue model and content of horizontal portals. In Second International Conference on Innovations in Information technology, UAE University, Dubai, UAE
7. Sieber, S., & Sabatier, J. V. (2003). Market bundling strategies in the horizontal portal industry. International Journal of Electronic Commerce, 7(4), 37-54.
8. García-Gómez, J. C. (2001). Portales de internet: concepto, tipología básica y desarrollo. El profesional de la información, 10(7-8), 4-13.
9. Burgess, S., Bingley, S., & Tatnall, A. (2005). Matching the revenue model and content of horizontal portals. In Second International Conference on Innovations in Information technology, UAE University, Dubai, UAE.
10. Sieber, S., & Sabatier, J. V. (2003). Market bundling strategies in the horizontal portal industry. International Journal of Electronic Commerce, 7(4), 37-54
11. Van Der Aalst, W. M., Leymann, F., & Reisig, W. (2007). The role of business processes in service oriented architectures. International Journal of Business Process Integration and Management, 2(2), 75.
12. Hohwiller, J., & Schlegel, D. (2011, June). Integration of UI services into SOA based BPM applications. In International Conference on Business Information Systems (pp. 53-64). Springer, Berlin, Heidelberg.
13. Liferay, (Consultado Mayo 2019), <https://www.liferay.com/es/blog?regionCategoryId=en-us&blogCategoryId=digital-strategy&title=portal-vs-website-when-to-use-each>
14. Avila, G., Sánchez, A., de Zayas, L. M. R., Jiménez, P., & Basnuevo, M. (2010). El Portal Holguín una ventana al mundo por la red de redes. Ciencias Holguín, 10(4)
15. TechDifferences, (Consultado Mayo 2019), <https://techdifferences.com/difference-between-website-and-portal.html>
16. Usability testing on government agencies web portal: A study on Ministry of Education Malaysia (MOE) web portal” Education Malaysia.pdf
17. Sarin, A. (2012). Portlets in action. Manning.
18. Trujillo, S., Batory, D., & Diaz, O. (2007, May). Feature oriented model driven development: A case study for portlets. In 29th International Conference on Software Engineering (ICSE'07) (pp. 44-53). IEEE.

19. Diaz, O., & Rodriguez, J. (2004). Portlets as web components: An introduction. *Journal of Universal Computer Science*, 10(4), 454-472.
20. <http://recursositc.educacion.es/observatorio/web/ca/software/servidores/1087-liferay-portal?showall=1>
21. García de Jalón, J., Ignacio Rodríguez, J., & Imaz, A. (2012). Aprende Servlets de Java como si estuviera en primero.
22. Liferay, (Consultado Mayo 2019), <http://liferaytechnologies.blogspot.com/p/basic-difference-between-servlet-and.html>
23. Servlets Vs Portlets. (Consultado Mayo 2019), <https://fanatech.wordpress.com/servlets-vs-portlets/>
24. Liferay, (Consultado Mayo 2019), https://dev.liferay.com/es/develop/tutorials/-/knowledge_base/7-0/portlets
25. Byte, (Consultado Junio 2019), <https://www.revistabyte.es/comparativa-byte-ti/herramientas-para-la-construccion-de-portales/>
26. Gozalo Macías, S. (2010). Diseño y desarrollo de un portlet para la gestión de cursos en Liferay integrado con el sistema de gestión de workflows JBPM (Bachelor's thesis).
27. Gartner Magic Quadrant (Consultado Mayo 2019) <https://www.gartner.com/en/research/methodologies/magic-quadrants-research>
28. OpenPortal, (Consultado Junio 2019), <http://openportal.sourceforge.net/>
29. UPortal, (Consultado Junio 2019), <https://www.apereo.org/projects/uportal>
30. Adobe Muse, (Consultado Junio 2019), www.adobe.com/es
31. Drupal 7, (Consultado Junio 2019), <https://www.drupal.org>
32. Liferay Portal, (Consultado Junio 2019), <https://www.liferay.com/es/home>
33. Jetspeed, (Consultado Junio 2019), <http://portals.apache.org/jetspeed-2/>
34. GetIn, (Consultado Junio 2019), <http://gatein.jboss.org/>
35. IBM WebSphere Portal, (Consultado Junio 2019) https://www.ibm.com/us-en/marketplace/ibm-digital-experience?mhq=portal&mhsr=ibmsearch_p
36. Liferay, (Consultado Mayo 2019), <https://www.liferay.com/es/blog/es-es/transformacion-digital/tipos-de-portales-tres-principales-areas-a-considerar>
37. Delgado, A., Clegari, D., & Arrigoni, A. (2016). Towards a generic BPMS user portal definition for the execution of business processes. *Electronic Notes in Theoretical Computer Science*, 329, 39-59.
38. Bazán, P., Garro, J. M., Giandini, R., & Diaz, J. (2015, October). Mejora de la monitorización y ejecución de procesos de negocio con integración y socialización. In 2015 Latin American Computing Conference (CLEI) (pp. 1-12). IEEE.
39. Song, M., Zhang, C., & Haihong, E. (2018, November). An Auto Scaling System for API Gateway Based on Kubernetes. In 2018 IEEE 9th International Conference on Software Engineering and Service Science (ICSESS) (pp. 109-112). IEEE.
40. WhatIs?, (Consultado Junio 2019), <https://whatistechtarget.com/definicion/API-gateway-application-programming-interface-gateway>
41. Luján-Mora, S. (2002). Programación de aplicaciones web: historia, principios básicos y clientes web. Editorial Club Universitario.

Plantilla de parametrización de Consultas de Mashup

Graciela Vidal y Sandra Casas

GISP - Instituto de Tecnología Aplicada
Universidad Nacional de la Patagonia Austral
Campus universitario Piloto Lero Rivera s/nro. Río Gallegos Santa Cruz
vidalgracielaunpa@gmail.com, scasas@unpa.edu.ar

Resumen El acceso a la información en la Web así como el formato de los contenidos ha cambiado. Por esta razón el proceso de desarrollo también se ha modificado. Esto ha dado lugar al surgimiento de nuevas prácticas como Mashup, que utiliza la web como plataforma de desarrollo. Las APIs cumplen un rol fundamental en este tipo de desarrollo. A su vez el cúmulo de recursos disponibles dificulta la selección adecuada de los mismos. El trabajo presenta una plantilla preliminar para seleccionar y generar código para la ejecución de consultas simples y complejas de videos utilizando las APIs de Datos de Youtube y Vimeo.

Palabras clave: APIs, mashup, Youtube, Vídeo

1 Motivaciones

La forma de acceder a la información y el formato en el cual está disponible a través de la web ha cambiado. En consecuencia las aplicaciones web y su proceso de desarrollo también se han modificado. Las aplicaciones actuales son el resultado de la integración de múltiples recursos disponibles en la web, tales como APIs, contenidos y componentes, más precisamente consiste en extraer información de diferentes fuentes, combinarla y presentarla en otras formas [1]. Esta joven práctica de desarrollo se denomina Mashup, la misma utiliza como plataforma de desarrollo a la Web y además combina técnicas ya establecidas como incrustación, agregación e integración para generar nuevas aplicaciones [2]. Esta nueva forma de desarrollo permite implementar nuevas aplicaciones y servicios web completos [3], además posibilita la incorporación de contenidos actualizados en sitios web ya desarrollados.

El video es un recurso audiovisual con amplia aceptación, si bien son populares desde hace tiempo en la industria de la música, resultan un medio muy utilizado para difundir información de todo tipo en la actualidad. Las plataformas web de videos tales como Youtube y Vimeo entre otras, representan un claro ejemplo de la evolución y crecimiento de este tipo de contenidos.

Ante estos cambios, los sitios basados en videos no sólo evolucionan en cuanto al volumen de información y la cantidad de usuarios, sino que también disponen de nuevas herramientas para desarrolladores. Esto promueve las tendencias actuales como mashup y facilita la distribución y el acceso a videos desde otras aplicaciones.

Las APIs (Applications Programming Interfaces) cumplen un rol fundamental en el proceso de desarrollo mashup. El surgimiento de estas prácticas también ha provocado el crecimiento de sitios web como ProgrammableWeb [4], Google developers [5] y Yahoo Developer Network [6], cuyo principal propósito es ofrecer colecciones de APIs pertenecientes a diferentes dominios como Fotos, Videos, Social, eCommerce, Música, Tools, entre otros. Las APIs presentan sus recursos, especificaciones, referencias y ejemplos en cada sitio, los mismos están destinados a desarrolladores experimentados. Sin embargo, el volumen de recursos disponibles en cada sitio es elevado por lo tanto resulta complejo seleccionar los adecuados.

El trabajo, presenta una plantilla preliminar que permite generar automáticamente el código para ejecutar consultas simples y complejas sobre videos, basadas en diferentes parámetros (categoría, términos y fecha), para ser integrado a diferentes aplicaciones. Esta herramienta contribuye a la selección de recursos mashup desde una API, más precisamente en la generación de consultas.

Este trabajo está organizado de la siguiente manera en la Sección 2 se exponen conceptos básicos sobre Mashup y se mencionan trabajos relacionados al desarrollo en esta práctica, en la Sección 3 el trabajo muestra una exploración de las APIs de Datos de los sitios Youtube y Vimeo, en la Sección 4 se presenta una Plantilla que permite parametrizar consultas y resultados para mashup, en la Sección 5 se exponen las conclusiones.

2 Mashup

Mashup se define como una práctica de desarrollo joven que utiliza la Web como plataforma y técnicas como incrustación, agregación e integración para la manipulación de recursos y datos con el fin de obtener aplicaciones complejas o incorporar funcionalidad en otras.

Un mashup (o Web mashup) es una aplicación que integra dos o más componentes mashup en cualquier capa de la aplicación, por ejemplo, en la capa de datos, lógica de aplicación y presentación, además pueden o no estar en comunicación cada entre ellos [7].

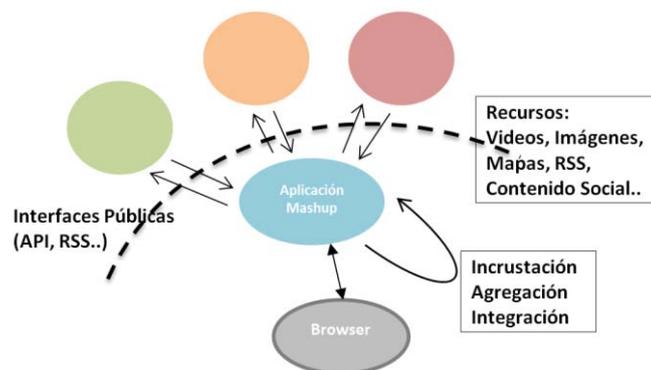


Figura 1. Arquitectura de Mashup

En el proceso de mashup se identifican los siguientes roles [8][9]:

Desarrollador de componente: es quien selecciona recursos (arquitecturas, bases de datos, estilos, formatos, etc.) luego desarrolla un componente mashup, describe su especificación y lo publica en la web.

Creador mashup (composer): es quien descubre, selecciona los componentes mashup y lleva a cabo la “mezcla”, esto puede realizarse utilizando herramientas mashup o manualmente.

Usuario mashup: es quien configura y utiliza la aplicación final.

Mashup es considerado un paradigma impulsado por el usuario, los desarrolladores mashup desean crear soluciones que satisfagan sus necesidades situacionales.

El proceso de desarrollo mashup puede llevarse a cabo de dos formas: manual o asistido. En el primer escenario no existen APIs, el desarrollador debe analizar los componentes existentes, seleccionar el adecuado e implementar la aplicación. El problema que presenta esta forma es que las interfaces no son estables y que está orientado sólo a programadores expertos. El segundo escenario propone herramientas y plataformas, las cuales simplifican el proceso de desarrollo a usuarios menos experimentados.

Se han llevado a cabo estudios [10] para asistir en la selección de componentes, tal vez la tarea más compleja de este proceso, los cuales proponen un framework basado en la calidad de los componentes. El mismo consiste en la definición de un conjunto pasos para analizar y seleccionar componentes candidatos.

En [11] se aborda el desarrollo Mashup como una metodología simple basada en componentes con la cual usuarios finales pueden crear sus propias aplicaciones convirtiéndolos en desarrolladores e innovadores.

3 Exploración y análisis de APIs de Datos de sitios web de videos

Youtube posee una amplia sección destinada a los desarrolladores, la misma se organiza de la siguiente manera: Documentación, Widgets y herramientas, API de análisis, API de datos, Recursos para Android y Recursos para iOS. Estas herramientas permiten agregar funcionalidad de Youtube a todo tipo de aplicaciones, las operaciones son aplicables a diferentes recursos: Activities, Channels, Playlist, GuideCategories, Search, Videos, entre otros. La API se encuentra organizada por recursos y algunas de las funciones disponibles para cada recurso son list, insert, delete y update.

Los requisitos para utilizar los métodos disponibles en esta API son:

Obtener una clave API o un token OAuth 2.0, esto depende del tipo de solicitud que se envíe y del alcance de los datos a los cuales se pretende acceder (público o privado). Tanto para obtener una clave API como para un token es necesario acceder con una cuenta Gmail y registrar la aplicación a desarrollar, luego se deben seleccionar y habilitar las APIs adecuadas para ser invocadas en la aplicación.

La API de Datos de Youtube (v3) se encuentra organizada de la siguiente manera:

Inicio: breve descripción de los recursos y lenguajes de programación para los cuales está compatible la API de datos. Además de presentar Blogs de desarrolladores, repositorios y asistencia ante problemas en la implementación.

Guías: esta opción permite acceder a : 1) pasos requeridos para comenzar a utilizar la API, 2) los recursos y operaciones disponibles, 3) cuotas la API de datos utiliza una cuota para garantizar que los programadores utilicen el servicio según lo previsto, sin afectar la calidad del mismo o que limiten el acceso de las demás personas, 4) bibliotecas, 5) solicitudes autorizadas, 6) guías y tutoriales, 7) guía de implementación y migración

Referencias: en esta opción se describen en detalle cada uno de los recursos y los métodos disponibles para cada uno. Algunos de los recursos de la API de datos son: Activities, ChannelBanners, Channels, GuideCategories, PlaylistItems, Playlist, Search, Suscriptions, entre otros. Cada uno de los métodos se identifica por nombre, Solicitud HTTP y descripción.

Muestras: se presentan fragmentos de código implementados en diferentes lenguajes de programación, tales como Go, Java, Javascript, .NET, PHP, Python y Ruby.

Asistencia: permite realizar preguntas relacionadas al uso de la API de datos de Youtube en todas sus versiones.

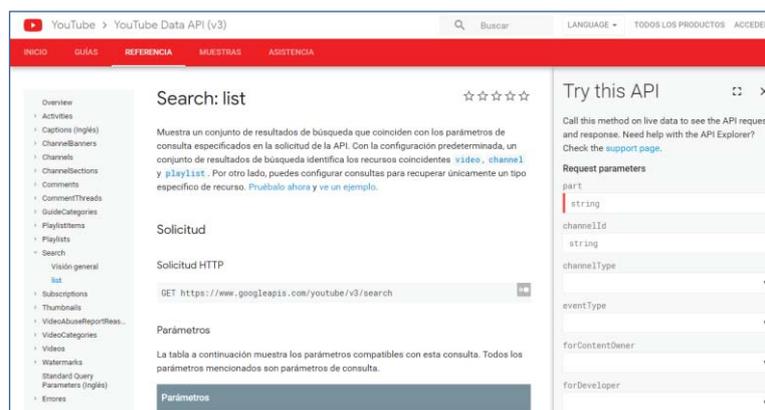


Figura 2. Distribución y especificación de recursos de la API de Datos de Youtube (v3)

La API de Vimeo fue construida para cumplir con los principios y prácticas REST (Representación de Estado de Transferencia), es un estilo arquitectónico de diseño para interactuar con recursos en línea, en este caso videos, utilizando métodos HTTP y estándar como GET, POST y PATCH.

El primer paso para utilizar la API de Vimeo consiste en registrar la aplicación, esta puede referirse a una aplicación móvil completa, una página web dinámica o un script de pocas líneas.

La API de Vimeo está disponible para ser integrada en diferentes lenguajes de programación como PHP, Python, Java y otros. Para realizar llamadas a la API se requiere un token de acceso, el cual consiste en una cadena de caracteres que cumple funciones de identificar a la aplicación y determinar un rango de acciones disponibles o ámbitos a los cuales puede acceder la aplicación.

El sitio web de la API de Vimeo, presenta la información de diferentes formas, tanto para desarrolladores nuevos o avanzados, incluye tutoriales, bibliotecas para

diferentes lenguajes de programación y referencias a la API, en la cual se describen todas las solicitudes y la estructura de los resultados.

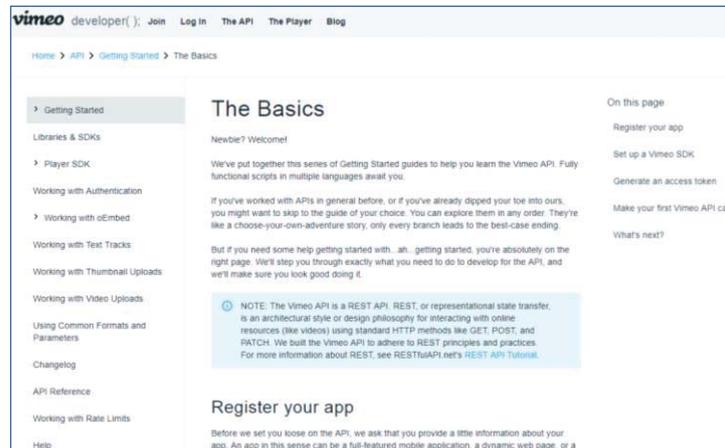


Figura 3. Recursos destinados a desarrolladores en el sitio Vimeo.

En la Tabla 1 se resumen diversos aspectos analizados sobre cada una de las API, los mismos están relacionados a las herramientas que proponen cada una para implementar consultas basadas en diferentes criterios.

Tabla 1. Análisis de API de Datos de YouTube y Vimeo.

	API de Datos de Youtube	API de Datos de Vimeo
Método de autenticación	<code>gapi.client.setApiKey(..)</code>	Authorization: bearer {token} ó <code>var vimeoToken = "xxxxxxx";</code>
Método para cargar la API	<code>gapi.client.load(..)</code>	No requiere
Método o Solicitud para consultas	Métodos <code>gapi.client.youtube.search.list(..)</code> <code>gapi.client.youtube.videos.list(..)</code>	Solicitud endpoint <code>https://api.vimeo.com/{endpoint}? camp=holo;..Access token=vimeoToken;</code>
Formato de los resultados	JSON Estructura ítems	JSON Estructura data
Permite parametrizar los resultados?	Si <code>part: snippet, player,..</code>	Si <code>fields=name, description, embed,..</code>

4 Plantilla de parametrización de consultas con APIs de Datos

El trabajo propone una plantilla preliminar en la cual es posible generar fragmentos de código parametrizado de consultas, utilizando las API de Youtube y Vimeo. El lenguaje utilizado para ambas APIs es Javascript, en el proceso de consulta se distinguen tres instancias:

Autenticación: el desarrollador debe registrarse y generar las claves de acceso para realizar las llamadas.

Consulta: se define una consulta parametrizada a partir de la API y del criterio seleccionado, además es posible parametrizar los resultados.

Resultados: los resultados son presentados un archivo JSON en el cual se describe los recursos que responden a la consulta.

La Figura 4 muestra la plantilla de configuración de consultas parametrizadas, el usuario debe seleccionar la API, el tipo de consulta y el criterio en el cual se basará la consulta.

Figura 4. Plantilla diseñada para generar código para consultas.

En la Tabla 2 se detallan los tipos de consulta que genera la plantilla definida, las mismas están basadas en tres parámetros: categoría, término y fecha.

Tabla 2. Tipos de consultas

		Categoría	Término	Fecha
Consulta	Simple	X		
			X	
				X
	Compleja	X	X	
		X	X	
			X	X

Consultas Simples

El desarrollador selecciona una API, luego el tipo de consulta que desea obtener, a partir del criterio seleccionado se habilita el campo correspondiente, en el caso de categoría, se despliega un listado con todas las categorías de cada sitio. Si la consulta estará basada en un término se habilita un campo para ingresar el mismo, en el caso de fecha es posible seleccionar la misma desde un date. En la Figura 5 se muestra el código generado utilizando la API de Datos de Youtube, basada en fecha.

Figura 5. Código de consulta de API YouTube.

El código generado podrá ser integrado a la aplicación que lo requiera, en este trabajo se utilizó el lenguaje Javascript, la ejecución del mismo por medio de un objeto request, obtiene como resultado un archivo JSON. El mismo consiste en una estructura array denominada ítems, cada ítems, a su vez puede contener otra estructura de ítems, lo que se define como propiedades anidadas. El desarrollador puede especificar en la consulta, los ítems y propiedades anidadas que incluirá la estructura resultante. Si no se especifican ítems en la consulta se incluirán todas las propiedades en el resultado. En la Figura 6 se presenta la estructura utilizada por la API de Datos de Youtube, en la misma se observan los siguientes ítems: kind y etag cuyos valores son únicos, por otro lado los ítems id y snippet contienen otros ítems.

```

"items": [
  {
    "kind": "youtube#searchResult",
    "etag": "\"Bdx4f4ps3xCOOo1W291nTLkRZ_c/-_HH2WqkK2AteMCoG7Lghwtqx2I\"",
    "id": {
      "kind": "youtube#video",
      "videoId": "7hPMmzKs62w"
    },
    "snippet": {
      "publishedAt": "2015-06-18T07:00:01.000Z",
      "channelId": "UCmQmc-Y4N9btVMUS1BfnyDw",
      "title": "Madonna - Bitch I&#39;m Madonna ft. Nicki Minaj",
      "description": "You're watching the official music video for \"Bitch I'm Madonna\"",
      "thumbnails": {
        "default": {
          "url": "https://i.ytimg.com/vi/7hPMmzKs62w/default.jpg",
          "width": 120,
          "height": 90
        }
      }
    }
  }
]

```

Figura 6. Estructura de resultados con API de Datos de Youtube

El desarrollador que utilice la plantilla propuesta deberá previamente, obtener una clave API en el sitio correspondiente. En el caso de Youtube deberá invocar el método `setApiKey(..)` para establecer su clave y el método `load(..)` para incorporar la API a la aplicación. El ejemplo 1 muestra el código generado para ejecutar una consulta de videos por categoría, invocando el método `gapi.client.youtube.videos.list(..)`, la lista de parámetros incluye: `part`, en el cual se indica qué subestructuras son requeridas en los resultados, en este caso `snippet` y `player`, `maxResults` permite definir la cantidad de recursos que incluirá cada página, `chart`: incluye los videos más populares y sus categorías, `order`: ofrece diferentes criterios para ordenar los resultados y por último el campo `videoCategoryId` en el cual se indica la categoría seleccionada para la consulta.

```
gapi.client.youtube.videos.list({part: "snippet,player", maxResults: 5,
chart:"mostPopular" order: "viewCount",videoCategoryId: mySearch });
```

Ejemplo 1. Consulta por categoría en Youtube.

En ejemplo 2, se presenta un fragmento de código para una consulta de videos por categoría en el sitio Vimeo, en este ejemplo la categoría es *animación* y se trata de un parámetro seleccionado en la plantilla, además se detalla la lista de campos que incluirá la estructura del resultado, en este caso se requiere: name, description, embed, created_time, modified_time. El desarrollador deberá proporcionar su token de acceso para ejecutar la consulta.

```
https://api.vimeo.com/categories/animation/videos?name,description,embed,
created_time,modified_time&access_token={API_KEY}
```

Ejemplo 2. Consulta por categoría en Vimeo.

Los resultados obtenidos utilizando la API de Vimeo son presentados en una estructura similar a la que propone la API de Youtube, excepto en la denominación de algunos campos. En Figura 7 se muestra la estructura se denomina data, de igual forma existen campos de la estructura que contienen otros campos, como por ejemplo embed, que a su vez contiene otras estructuras que incluyen más campos, como badges y live.

```
"data": [
{
  "uri": "/videos/89797442",
  "name": "RUDENS X MADONA X ŠKODEK",
  "description": "I spent 1 week in my hometown called Madona together ..",
  "type": "video",
  "link": "https://vimeo.com/user10987738/rudensxmadonaxskodek",
  "duration": 110,
  "width": 1280,
  "language": null,
  "height": 720,
  "embed": {
    "html": "<iframe src=\"https://player.vimeo.com/video/89797442?title=...\"",
    "badges": {
      "hdr": false,
      "live": {
        "streaming": false,
        "archived": false
      },
      "staff_pick": {
        "normal": false,
        "best_of_the_month": false,
        "best_of_the_year": false,
        "premiere": false
      },
      "vod": false,
      "weekend_challenge": false
    }
  },
  "created_time": "2014-03-22T18:47:33+00:00",
  "modified_time": "2018-02-06T00:17:21+00:00",
  "release_time": "2014-03-22T18:47:33+00:00",
```

Figura 7. Estructura de resultados con API de Datos de Vimeo

Consultas complejas

La flexibilidad de ambas APIs en cuanto a los parámetros admitidos en la consulta permitió establecer solicitudes basadas en varios criterios. En el ejemplo 3 se define una consulta utilizando la API de Youtube basada en término q y categoría (`videoCategoryId`), además se especifica el tipo de recurso *video*, se indica un criterio de ordenamiento y se restringe a los publicados desde el año 2015. Como parte del resultado se requiere el ítem `snippet`.

```
gapi.client.youtube.search.list({part: "snippet",type: "video",q:
encodeURIComponent($("#search").val()).replace(/%20/g, "+" ),maxResults: 5,order:
"viewCount",publishedAfter: "2015-01-01T00:00:00Z", videoCategoryId:"1" });
```

Ejemplo 3. Consulta compleja basada en término y categoría

El análisis sobre la taxonomía de los resultados en ambas APIs, permitió identificar un conjunto de propiedades equivalentes, es decir que se denominan de diferente forma pero representan la misma información. Esto permitió parametrizar la información que será incluida en la respuesta.

5 Conclusiones

El trabajo propone una plantilla preliminar que permite generar código para integrar y ejecutar consultas simples y complejas utilizando las API de Datos de Youtube y Vimeo. Se realizó un análisis de cada API, enfocado en dos instancias, en primer lugar en el proceso que proponen para definir las consultas, en el cual se identificaron tres pasos autenticación, consulta y resultados. La API Youtube utiliza métodos, mientras que Vimeo utiliza endpoint para realizar solicitudes de consulta.

Luego se realizó una exploración de la taxonomía definida por cada APIs para describir los recursos y presentar los resultados. Se observó una gran similitud en la estructura utilizada para describir los recursos, se identificaron propiedades equivalentes de manera que las respuestas a las consultas presenten la misma información, para cualquiera de las dos APIs.

La plantilla permite generar, a partir de una simple selección de API, criterios y tipo de consulta (simple o compleja), fragmentos de código parametrizados, para ejecutar consultas en lenguaje Javascript.

El enfoque de plantillas puede ser aplicado a consultas de otros dominios. El trabajo futuro está dirigido a desarrollar generadores de código enfocados en otras operaciones disponibles en las APIs mencionadas, posibilitar que el desarrollador seleccione desde la plantilla qué campos desea incluir en los resultados, e incluso incorporar nuevas APIs, para luego ser utilizado en proceso de desarrollo mashup.

Referencias

1. Makki S. K. & Sangtani J. Data Mashups & Their Applications in Enterprises. Internet and Web Applications and Services, ICIW '08. Third International Conference Athens: IEEE, pp 445-450. (2008)
2. Matera M., & Picozzi M. Next Generation Web Apps. Towards Transformative UX. Politecnico de Milano Dpto. Electronica e Informática (2015)
3. Yu J., Benatallah B., Casari F. & Daniel F. Understanding Mashup Development. Service Mashup. IEEE Computer Society. ISSN: 1089-7801.(2008)
4. ProgramableWeb <https://www.programmableweb.com/> (2019)
5. Google Developer <https://developers.google.com/> (2019)
6. Yahoo Developer Network <https://developer.yahoo.com/api/> (2019)
7. Daniel, F., & Matera M.. Mashups Concepts, Models and Architectures. Milano, Italy: ISBN: 978-3-642-55049-2. Springer. (2014).
8. T. Janner et al., Patterns for Enterprise Mashups in B2B Collaborations to Foster Lightweight Composition and End User Development. Proc. IEEE Int'l Conf. Web Services (ICWS 09) pp. 976–983 (2009)
9. Gamble T.& Gamble R. Monoliths to Mashups: Increasing Opportunistic Assets. IEEE Software. pp 9-24 (2008)
10. Saeed A.A Quality-based Framework for Leveraging the Process of Mashup Component Selection. ISSN: 1651-4769. Department of Applied Information Technology. University of Gothenburg Sweden (2009)
11. Daniel F., Matera M.& Weiss M. Next in Mashup Development: User-Created Apps on the Web. IEEE Computer Society. ISSN: 1520-9202 (2011)

Propuesta de Framework para el Diseño de Interacción en las Redes Sociales de Video

Héctor Reinaga y Sandra Casas

GISP - Instituto de Tecnología Aplicada
Universidad Nacional de la Patagonia Austral
Campus universitario Piloto Lero Rivera s/nro. Río Gallegos Santa Cruz
hreinaga@uarg.unpa.edu.ar, scasas@unpa.edu.ar

Abstract. Los portales o redes sociales de video como YouTube, Vimeo y Dailymotion se encuentran entre los sitios más populares en Internet. Los usuarios de estos sitios forman una red social, que proporcionan un medio poderoso para compartir, organizar y encontrar contenido y contactos. La información producida a partir de este dinamismo, puede ser usada en otras plataformas y aplicaciones. Paralelamente, existen metodologías, modelos y patrones aplicables al desarrollo y diseño web, sin embargo no se han especificado herramientas de este estilo para el desarrollo de sitios propiamente de videos. Este trabajo propone un framework de diseño de interacción para las redes sociales de video, con el objeto de diseñar e implementar portales y redes sociales de video en general, que se ha desarrollado a partir de la exploración de los sitios de videos de mayor popularidad en la actualidad, y del análisis de patrones de diseño de interacción reconocidos.

Palabras claves: videos, portales de videos, redes sociales, patrones de diseño.

1 Introducción

El video es el formato de información preferido entre los internautas, siendo YouTube uno de los tres espacios de Internet más populares en la actualidad, junto a Google y Facebook. Estos espacios constituyen exponentes de la web 3.0, que se caracteriza por la producción de contenidos, además del intercambio de información entre usuarios [1]. Las características y propiedades, los contenidos, las funciones, la colaboración, y la interacción, enriquecen la comunicación y generan nuevos esquemas de interoperación. La información producida a partir de este dinamismo, puede ser usada en otras plataformas y aplicaciones, y además resultará vital para la evolución de las mismas.

Asimismo, existen metodologías, modelos y patrones aplicables al desarrollo y diseño web, sin embargo no se han especificado herramientas de este estilo para el desarrollo de sitios propiamente de videos.

Por estas razones, proyectando nuevos escenarios de uso, se propone especificar las redes sociales basadas en videos (RSV) a través de un framework para el diseño de interacción. El uso de un framework para cualquier tipo de desarrollo, reduce el

tiempo de elaboración e implementación, y ayuda a hacer un trabajo mantenible y escalable, según las características del mismo [2].

A partir de la exploración y estudio de los principales portales de video de la actualidad, como YouTube, Vimeo y Dailymotion; se identificaron aquellos elementos de interacción que se producen, utilizando un enfoque de patrones de autores reconocidos. Del estudio realizado sobre un total de 309 patrones, se identificaron la aplicación de 75 patrones en estas redes.

Esta información derivó en la definición de un Framework de Diseño de Interacción para Redes Sociales de Video, con el objeto de diseñar e implementar portales de video y redes sociales de video en general.

El presente trabajo se organiza de la siguiente manera, en la Sección 2 se exponen los Materiales y Métodos, en la Sección 3 se presentan los Resultados, en la Sección 4 se expone el Framework para el Diseño de la Interacción de RSV, en la Sección 5 los Trabajos relacionados, y en la Sección 6 las Conclusiones.

2 Materiales y Métodos

El trabajo se basó en una exploración sobre los sitios de videos con mayor popularidad en la actualidad. A continuación, se detallan los pasos de estudio:

Análisis de estructura y funcionalidades. Con el fin de analizar la estructura y funcionalidades de los portales, se utilizaron wireframes [3]. Los wireframes son gráficos que representan cómo estarán organizados los elementos en las páginas web sin elementos gráficos, de tal manera que implementan aspectos generales del diseño sin entrar en detalles.

Se generaron prototipos rápidos de la interfaz gráfica correspondiente a la pantalla denominada “Resultado de la búsqueda” de cada uno de los portales de videos analizados. Para ello se utilizó una herramienta web free denominada Webfree.cc [4]. De acuerdo a los wireframes obtenidos, se analizaron detalladamente su interfaz gráfica para encontrar espacios o funcionalidades en común entre estos portales.

Análisis de patrones de diseño de interacción aplicados. Un patrón de interacción o patrón de diseño de interfaz de usuario [5][6], es una solución basada en el diseño de un componente software que resuelve un problema de interacción de un usuario final con una interfaz de usuario [7].

En base a los catálogos de patrones de diseño de interacción de varios autores, como los 131 patrones propuestos por Van-Wellie [8], los 54 patrones de Toxboe [9], y los 124 patrones de Crumlish-Malone [10]; se construyó un inventario de patrones de interacción de acuerdo a las funcionalidades encontradas.

Se analizaron detalladamente cada catálogo de patrones con el fin de evaluar si los patrones pueden ser aplicables a las funcionalidades definidas.

Definición de propuesta de framework. En base a las funcionalidades definidas, y del inventario de patrones creado, permitieron centralizar en un repositorio las propuestas de patrones de interacción teniendo en cuenta las funcionalidades determinadas.

3 Resultados

La reproducción de la organización de espacios y elementos obtenidos mediante wireframes (Fig. 1), resultantes de la pantalla “Resultado de la búsqueda” de las RSV analizadas, se lograron encontrar cinco espacios comunes.



Fig. 1: Wireframes de los portales YouTube, Vimeo, y Dailymotion.

Las funcionalidades en común entre los portales fueron:

- Búsqueda:** permite al usuario introducir una consulta y obtener un conjunto de resultados. Para ello, el usuario ingresa una/s palabras/s que desea buscar en una caja de texto.
- Menú propio:** menú específico que presenta cada portal de video en lo que refiere a la cuenta de usuario, y otras funciones. Se accede al menú ingresando con una cuenta de usuario.
- Video:** espacio o zona donde se reproduce el video resultante de una operación de búsqueda por parte del usuario, incluye las funciones propias de reproducción, tales como título del video, autor, nombre del canal, cantidad de reproducciones, botones de reproducción, “Me gusta”, “Compartir”, “Denunciar”, etc.
- Comentarios:** espacio donde los usuarios pueden ingresar sus comentarios del video buscado.
- Resultado búsqueda/Videos relacionados:** en este espacio se encuentran los videos resultantes de un proceso de búsqueda, y que se relacionan a la/s palabra/s buscada/s.

De acuerdo a los wireframes obtenidos, se generó la Tabla 1, en la cual se representa con una "X" aquellas funcionalidades que los portales analizados tienen en común, dejando en blanco las restantes. Se tuvieron en cuenta solo aquellas funcionalidades que aparecen en al menos 2 portales.

Tabla 1. Funcionalidades en común de los portales de video.

	YOUTUBE	VIMEO	DAILYMOTION
BUSQUEDA	X	X	X
PUBLICIDAD	X		
MENU PROPIO	X	X	X
VIDEO	X	X	X
COMENTARIOS (WIKI)	X	X	X
RESULTADOS BUSQUEDA /VIDEOS RELACIONADOS	X	X	

Se construyó un inventario de patrones en base al catálogo de los autores vistos [8][9][10], y de acuerdo a las funcionalidades definidas. Se agruparon los patrones de acuerdo a su aplicación en cada funcionalidad. Muchos de ellos pueden ser aplicados directamente (por ejemplo, los patrones de Van-Wellie para buscar o compartir). Otras veces, distintas fuentes explican el mismo patrón dándole diferentes nombres y matices.

Del análisis realizado al catálogo de patrones de Van Wellie, solo 23 patrones fueron aplicados; para el caso de Toxboe, solo 32 patrones fueron tomados en cuenta; y finalmente del catálogo de Erin Malone, solamente 20 patrones formaron parte del inventario de patrones generado; haciendo un total de 75 patrones aplicables.

En Tabla 2 se detallan por catálogo de autores, los patrones aplicados a cada funcionalidad. Para una mejor descripción, se usaron abreviaturas por autor (T=Toxboe, W=Willie, M=Malone), y se enumeraron de manera ascendente.

Tabla 2. Catálogo de patrones aplicables a las funcionalidades.

Funcionalidad	RESULTADO		
	TOXBOE	WILLIE	MALONE
Búsqueda	T1) Input Prompt T2) Rule Builder T3) Autocomplete	W1) Navigation Tree W2) Autocomplete W3) Search Box W4) Table Filter	M1) Filtering M2) Realtime Search
Menú propio	T4) Notifications T5) Fat Footer T6) Home link T7) Horizontal Dropdown Menu T8) Categorization T9) FAQ T10) Account Registration T11) Paywal	W5) Customizable Window W6) Headerless Menu W7) FAQ W8) Site Map W9) Footer Sitemap W10) Login W11) Registration W12) Form	M3) Sign Up or Registration M4) Sign Out M5) Terms of Service M6) Groups M7) Private Conversation
Video	T12) Vote To Promote T13) Flagging & Reporting T14) Morphing Controls T15) Favorites T16) Follow T17) Auto-sharing T18) Reaction T19) Invite Friends	W13) Scrolling Menu W14) Slideshow W15) Action Button	M8) Saving M9) Add / Subscribe M10) Direct Sharing M11) Public Sharing M12) Thumbs Up/Down Style M13) Ratings M14) Following

Comentarios	T20) Morphing Controls T21) Flagging & Reporting T22) Progressive Disclosure T23) Feedback loops T24) Reaction	W16) Comment Box	M15) Blogs- Consume M16) Blogs - Ownership M17) Lifecycle M18) Comments M19) Public Conversations M20) Group Conversation
Videos relacionados / Resultado búsqueda	T25) Carousel T26) Article List T27) Pagination T28) Continuous Scrolling T29) Thumbnail T30) Pull to refresh T31) Table Filter T32) Search Filters	W17) Accordion W18) Home Link W19) Paging W20) Minesweeping W21) Search Results W22) Carrousel W23) Thumbnail	

En la Tabla 3 se detallan por cada RSV estudiada, los patrones de diseño de interacción que fueron aplicados, dividido por catálogo de autores, y en base a las cinco funcionalidades definidas.

Tabla 3. Patrones aplicados a los portales de video, dividido por funcionalidades definidas.

Funcionalidad	YOUTUBE			VIMEO			DAILYMOTION		
	Toxboe	Willie	Malone	Toxboe	Willie	Malone	Toxboe	Willie	Malone
Búsqueda	T1 T2 T3	W1 W2 W3 W4	M1 M2	T1 T3	W1 W2 W3 W4	M1 M2	T1	W1 W3	M2
Menú propio	T4 T6 T7 T8 T9 T10	W5 W6 W7 W10 W11 W12	M3 M4 M5 M6 M7	T4 T5 T6 T7 T8 T9 T10	W5 W6 W7 W8 W9 W10 W11 W12	M3 M4 M5 M6 M7	T4 T5 T6 T8 T9 T10 T11	W7 W10 W11 W12	M3 M4 M5 M6
Video	T12 T13 T14 T15 T16 T17 T18 T19	W15	M8 M9 M10 M11 M12 M13 M14	T12 T13 T14 T15 T16 T17 T18 T19	W13 W14 W15	M8 M9 M10 M11 M12 M13 M14	T12 T13 T14 T15 T16 - T18 -	W15	M9 M10 M12 M13 M14
Comentarios	T20 T21 T22 T23 T24	W16	M15 M16 M17 M18 M19 M20	T20 T21 T23	W16	M15 M16 M17 M18 M19 M20	T20 T21 T22 T23 T24	-	-
Videos relacionados/ Resultado búsqueda	T25 T26 T28 T29 T30 T31 T32	W17 W18 W20 W21 W22 W23	-	T25 T26 T27 T29 T30 T31 T32	W17 W18 W19 W21 W22 W23	-	T25 T26 T28 T29 T30	W17 W18 W21 W22 W23	-

De este proceso, y de acuerdo a los catálogos analizados, se obtuvo un inventario de patrones de interacción de los tres portales o RSV más populares.

Con este inventario se centralizan en un repositorio, las propuestas de patrones de interacción teniendo en cuenta las funcionalidades determinadas.

4 Framework para el Diseño de la Interacción de Redes Sociales de Videos

El estudio realizado permitió identificar cinco funcionalidades del entorno de diseño de interacción de los portales vistos, que forman parte del framework propuesto: Búsqueda, Menú propio, Video, Comentarios, y Resultados de búsqueda/Videos relacionados.

Además de la especificación para cada una de las funcionalidades identificadas, se ha determinado ciertas características encontradas, denominada elementos de interacción, conforme si el patrón se encuentra en todas o al menos dos de estos portales. Un elemento de interacción se considera obligatorio, si se encuentra presente en todos los sitios explorados, mientras que los optativos aparecen con menor frecuencia que las anteriores.

A continuación, se describen las funcionalidades con los elementos de interacción (patrones) que la componen:

- a) Búsqueda:
 - i. Elementos de interacción obligatorios: Input Prompt (T1), Navigation Tree (W1), Search Box (W3), Realtime Search (M2).
 - ii. Elementos de interacción opcionales: Autocomplete (T3), Table Filter (W4), Filtering (M1).
- b) Menú propio:
 - i. Elementos de interacción obligatorios: Notifications (T4), Home link (T6), Categorization (T8), FAQ (T9), Account Registration (T10), FAQ (W7), Login (W10), Registration (W11), Form (W12), Sign Up or Registration (M3), Sign Out (M4), Terms of Service (M5), Groups (M6).
 - ii. Elementos de interacción opcionales: Fat Footer (T5), Horizontal Dropdown Menu (T7), Customizable Window (W5), Headerless Menu (W6), Private Conversation (M7).
- c) Video:
 - i. Elementos de interacción obligatorios: Vote To Promote (T12), Flagging & Reporting (T13), Morphing Controls (T14), Favorites (T15), Follow (T16), Reaction (T18), Action Button (W15), Add / Subscribe (M9), Direct Sharing (M10), Thumbs Up/Down Style (M12), Ratings (M13), Following (M14).
 - ii. Elementos de interacción opcionales: Auto-sharing (T17), Invite Friends (T19), Saving (M8), Public Sharing (M11).
- d) Comentarios:
 - i. Elementos de interacción obligatorios: Morphing Controls (T20), Flagging & Reporting (T21), Feedback loops (T23).
 - ii. Elementos de interacción opcionales: Progressive Disclosure (T22), Reaction (T24), Comment Box (W16), Blogs-Consume (M15), Blogs-Ownership

(M16), Lifecycle (M17), Comments (M18), Public Conversations (M19), Group Conversation (M20).

e) Resultados de Búsqueda /Videos relacionados:

- i. Elementos de interacción obligatorios: Carousel (T25), Article List (T26), Thumbnail (T29), Pull to refresh (T30), Accordion (W17), Home Link (W18), Search Results (W21), Carrousel (W22), Thumbnail (W23).
- ii. Elementos de interacción opcionales: Continuous Scrolling (T28), Table Filter (T31), Search Filters (T32).

Se pudo determinar que las funcionalidades en las cuales se observan más cantidad de patrones son: Menú Propio (21 patrones de diferentes catálogos), Video (18 patrones), Resultados de búsqueda/Videos relacionados (15 patrones), y Comentarios (12 patrones).

En la Fig. 2 se detalla el framework propuesto que define un enfoque de los elementos o características que describen las funcionalidades disponibles en las RSV, generando un marco para diseñar e implementar portales de video en general.

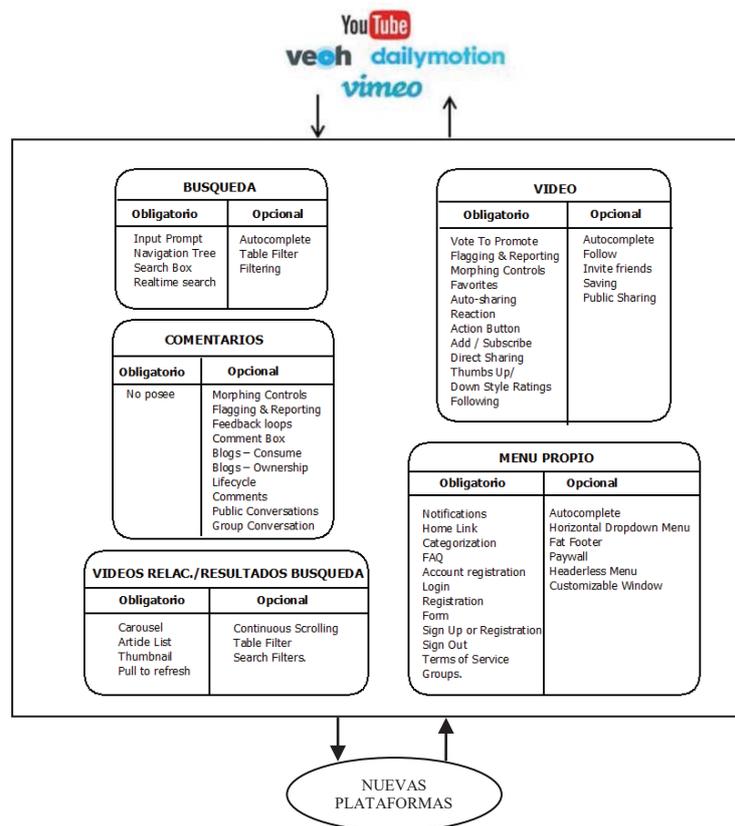


Fig. 2: Framework para el diseño de la interacción en Redes Sociales de Videos.

4.1 Aplicación del Framework para el Diseño de la Interacción de RSV

El framework propuesto se compone básicamente de las cinco funcionalidades en común encontradas de los portales analizados, y de los elementos de interacción (obligatoria y opcional) que poseen cada una de las funciones.

Para la aplicación del framework propuesto, los portales o RSV tendrán las funcionalidades y elementos de interacción definidos, sobre el cual se podrán añadir otras funciones propias de cada portal.

En la Fig. 3 se presenta una versión prototipo del framework, con los elementos de interacción obligatoria (rojo) y opcional (verde), y las funcionalidades definidas.



Fig. 3: Prototipo del framework propuesto para el diseño de la interacción en RSV.

En Anexo 2 se describen las referencias a las abreviaturas descritas en la Fig. 3, que representan los elementos de interacción (patrones).

5 Trabajos relacionados

Existen frameworks para el desarrollo web con diferentes propósitos: de descripción de recursos [11], de especificación de componentes [12], y de especificación de aspectos reusables [13], entre otros.

Relacionado al tema propuesto, no se han hallado trabajos específicos sobre algún framework de diseño de interacción para las RSV, pero sí para el diseño de la interacción en sitios web en general [14][15], lo que se complementa con variadas guías y lineamientos. En [14] se propone un framework de clasificación de videos que considera la red social del usuario, y lo aplica a la búsqueda de videos como en YouTube para realizar búsquedas más aproximadas de acuerdo a su interés y preferencias. En otro caso, [15] presenta una caracterización de propuestas para el desarrollo de la interfaz de usuario de sistemas interactivos basados en distribución de contenido de video, en contextos de entretenimiento (Netflix y Youtube), o en contextos de aprendizaje (Udacity).

6 Conclusión

Este trabajo propone, de manera preliminar, un framework de diseño de interacción para las redes sociales de video, que se ha desarrollado a partir de la exploración de los sitios de videos de mayor popularidad en la actualidad, como son YouTube, Dailymotion, y Vimeo; y del análisis de patrones de diseño de interacción reconocidos.

La experiencia permitió identificar, a través del desarrollo de un wireframe, las funcionalidades en común de estos portales: Búsqueda, Menú propio, Video, Comentarios, y Resultados búsqueda/Videos relacionados; y se construyó un inventario de patrones que son aplicados a cada RSV vista. Asimismo, permitió determinar los elementos de interacción obligatorios y opcionales, dependiendo si los patrones aparecen con mayor o menor frecuencia en los portales.

Del estudio realizado se determinó que sobre un total de 309 patrones, se aplicaron 75 en estas redes; siendo 41 patrones los elementos de interacción obligatorios, y los opcionales de 24 patrones.

Esta información derivó en la definición de un framework, con el objeto de diseñar e implementar portales de video y redes sociales de video en general.

Como trabajo futuro, se orienta a continuar ampliando el estudio a otras RSV como también en sus versiones para celulares, y analizar el catálogo de patrones de otros autores. Asimismo, evaluar la implementación de una herramienta prototipo que permita generar o gestionar portales de video.

Referencias

1. Melo Herrera, B.: Web 2.0, Web 3.0 y Web 4.0. Universidad Icesi. Colombia, <http://www.icesi.edu.co/blogs/egatic/2009/05/21/web-20-web-30-y-web-40/> (2009)
2. Larman, C.: UML y Patrones. Una introducción al análisis y diseño orientado a objetos y al proceso unificado. Pearson Educación, Madrid (2003)
3. Carreras, O.: Wireframes, <http://olgacarreras.blogspot.com.es/2007/02/wireframes.html> (2012)
4. Sitio oficial wireframe.cc, <http://wireframe.cc>
5. Borchers. J.: A Pattern Approach to Interaction Design. Department of Computer Science. Darmstadt University of Technology. Alexanderstr. 6, 64283 Darmstadt, Germany. (2001)
6. Alexander, C., Ishikawa, S., Silverstein, M., Jacobson, M., Fiksdahl-King, I. & Angel, S.: "A Pattern Language". Oxford University Press, New York. (1977)
7. Palacios, A., García, R., Oliva, M., Granollers, T.: Exploración de patrones de interacción para su uso en la web semántica. El Profesional de la Información. 24. 749. 10.3145/epi.2015.nov.06. (2015)
8. Van-Welie, M.: A pattern library for interaction design. Welie.com, <http://www.welie.com> (2007)
9. Toxboe, A.: User interface desing pattern. UI patterns. <http://ui-patterns.com> (2011)
10. Crumlish, C.; Malone, E.: Designing social interfaces. O'Reilly; Yahoo Press. http://www.designingsocialinterfaces.com/patterns/Main_Page (2009)
11. Potts, A., Snow E., Mitchell N., Lloyd A. World Wide Web Consortium Publishes Public Draft of Resource Description Framework (RDF). Key Industry Players Collaborate to Develop Interoperable Metadata for the Web. (1997)

12. Ackermann, J., Brinkop, F., Conrad, S., Fettke, P., Frick, A., Glistau, E., Jaekel, H., Kotlar, O., Loos, P., Mrech, H., Ortner, E., Overhage, S., Raape, U., Sahm, S., Schmietendorf, A., Teschke, T. & Turowski, K.: Standardized Specification of Business Components. German Society of Informatics. (2002)
13. Vidal, G., Casas, S.: Development and Evaluation of a Software Repository Reusable Aspects. IEEE Conference Publications DOI: 10.1109/SCCC.2016.7836045 - 35th. International Conference of the Chilean Computer Science Society (SCCC). Valparaíso Chile. (2016)
14. Liang G., Hung-Hsuan, Ch., Jung Hyun, K., Xiaolong, Z., Lee Giles: SNDocRank: A social network-based video search ranking framework. 367-376. 10.1145/1743384.1743443. (2010)
15. Ruiz, A., Arciniegas J.L., Giraldo, W.: Caracterización de marcos de desarrollo de la interfaz de usuario para sistemas interactivos basados en distribución de contenido de video. Revista Chilena de Ingeniería. Vol.26 N°2 Arica Jun. (2018) <http://dx.doi.org/10.4067/S0718-33052018000200339>.

Anexo

Seguidamente se detallan las abreviaturas utilizadas para describir los elementos de interacción del framework propuesto. La abreviatura se compone de una letra y un número; para el caso de la letra, es la inicial del nombre del autor del catálogo, seguido de un número correlativo.

TOXBOE	WILLIE	MALONE
T1: Input Prompt	W1: Navigation Tree	M1: Filtering
T2: Rule Builder	W2: Autocomplete	M2: Realtime Search
T3: Autocomplete	W3: Search Box	M3: Sign Up or Registration
T4: Notifications	W4: Table Filter	M4: Sign Out
T5: Fat Footer	W5: Customizable Window	M5: Terms of Service
T6: Home link	W6: Headerless Menu	M6: Groups
T7: Horizontal Dropdown Menu	W7: FAQ	M7: Private Conversation
T8: Categorization	W8: Site Map	M8: Saving
T9: FAQ	W9: Footer Sitemap	M9: Add / Subscribe
T10: Account Registration	W10: Login	M10: Direct Sharing
T11: Paywal	W11: Registration	M11: Public Sharing
T12: Vote To Promote	W12: Form	M12: Thumbs Up/Down Style
T13: Flagging & Reporting	W13: Scrolling Menu	M13: Ratings
T14: Morphing Controls	W14: Slideshow	M14: Following
T15: Favorites	W15: Action Button	M15: Blogs- Consume
T16: Follow	W16: Comment Box	M16: Blogs - Ownership
T17: Auto-sharing	W17: Accordion	M17: Lifecycle
T18: Reaction	W18: Home Link	M18: Comments
T19: Invite Friends	W19: Paging	M19: Public Conversations
T20: Morphing Controls	W20: Minesweeping	M20: Group Conversation
T21: Flagging & Reporting	W21: Search Results	
T22: Progressive Disclosure	W22: Carrousel	
T23: Feedback loops	W23: Thumbnail	
T24: Reaction		
T25: Carousel		
T26: Article List		
T27: Pagination		
T28: Continuous Scrolling		
T29: Thumbnail		
T30: Pull to refresh		
T31: Table Filter		
T32: Search Filters		

PWA para unificar el desarrollo Desktop, Web y Mobile

Verónica Aguirre¹ , Agustín Ortu¹ , Lisandro Delía¹ , Pablo Thomas¹ ,
Leonardo Corbalán¹ , Germán Cáseres¹ , Patricia Pesado¹ 

¹ Instituto de Investigación en Informática LIDI (III-LIDI). Facultad de Informática –
Universidad Nacional de La Plata, La Plata, Argentina.
Centro Asociado a la Comisión de Investigaciones Científicas de la Provincia de Buenos
Aires (CIC)
{vaguirre, ldelia, pthomas, corbalan, gcaseres, ppesado}@lidi.info.unlp.edu.ar
ortu.agustin@gmail.com

Abstract. El crecimiento continuo de la potencia de cómputo de los dispositivos móviles y su relativo bajo costo, son algunas de las razones de su gran expansión. Actualmente, por ejemplo, en la Argentina 9 de cada 10 personas poseen un smartphone. Esta masividad, local y mundial, ha sido acompañada por la evolución de la industria del software para estos dispositivos con el liderazgo de dos sistemas operativos: Android e iOS. Esta segmentación ha generado la necesidad de disponer aplicaciones móviles para ambas plataformas y consecuentemente han surgido diferentes enfoques de desarrollo en respuesta a esta necesidad. Este trabajo presenta el análisis de un enfoque de desarrollo de aplicaciones móviles surgido recientemente, denominado Aplicaciones Web Progresivas o PWA, por sus siglas en inglés, como una alternativa novedosa a los enfoques existentes.

Palabras claves: PWA, Aplicaciones Web Progresivas, Aplicaciones Web Móviles, Desarrollo Multiplataforma

1 Introducción

Las aplicaciones móviles están presentes en múltiples dominios, se caracterizan por brindar acceso a la información y estar disponibles en todo momento desde cualquier dispositivo móvil con acceso a internet. La mayoría de las actividades que se realizaban desde una computadora, tales como revisar el correo electrónico o visitar una red social, en la actualidad son realizadas desde dispositivos móviles [1].

Actualmente, para el desarrollo de aplicaciones de software se debe contemplar los diversos dispositivos existentes (computadoras de escritorio, notebooks, smartphones, tablets, wearables, etc.) y los sistemas operativos disponibles (Android, iOS, Windows, entre otros). El desarrollo específico para cada plataforma puede requerir un trabajo costoso, por tal razón los proveedores de software buscan soluciones alternativas más económicas con similar calidad en el producto final [2]. Existen dos tipos de aplicaciones móviles: nativas y multiplataforma.

Las aplicaciones nativas se desarrollan utilizando tecnologías específicas, dando lugar a un nuevo proyecto de desarrollo para cada plataforma que se desee cubrir.

Por el contrario, existen diferentes enfoques de desarrollo que parten de un único proyecto base y permiten cubrir múltiples plataformas [3][4]: aplicaciones web móviles, aplicaciones híbridas, aplicaciones interpretadas y aplicaciones generadas por compilación cruzada; aceptados por la academia y la industria desde hace varios años [1][3][4][5]. Estos enfoques han sido analizados por los autores de este trabajo en [1][2][6][7][8], no obstante el escenario está en constante evolución.

En forma reciente ha surgido una nueva alternativa para el desarrollo de aplicaciones web móviles denominada Aplicaciones Web Progresivas (PWA por sus siglas en inglés). Esta nueva alternativa se introdujo por primera vez en la conferencia de desarrolladores de Google I/O en mayo de 2016 en San Francisco, California, Estados Unidos [9].

En este trabajo se presenta PWA, sus principales características, un ejemplo de uso, y un minucioso análisis comparativo con las aplicaciones web móviles.

Este artículo está estructurado de la siguiente manera: la sección 2 resume el desarrollo de aplicaciones web móviles, mientras que la sección 3 introduce el enfoque PWA. La sección 4 presenta un minucioso análisis comparativo entre las aplicaciones web móviles y las PWA. Finalmente se presentan conclusiones y se propone trabajo futuro.

2 Aplicaciones web móviles

Las aplicaciones web móviles se desarrollan utilizando tecnología web estándar como HTML, CSS y Javascript. Por tal motivo, el desarrollo, la distribución y las pruebas son relativamente sencillas, siendo una opción a considerar para tener presencia en el escenario de los dispositivos móviles. Estas aplicaciones no necesitan la aprobación de algún fabricante para su publicación y son independientes de la plataforma: sólo es necesario un navegador y acceso a internet [2]. El gran desafío que presenta el desarrollo de este tipo de aplicaciones es lograr una experiencia de usuario satisfactoria considerando que existen una amplia diversidad de dispositivos (computadoras, smartphones de diferentes tamaños, tablets, consolas de videojuegos, televisores inteligentes, wearables, entre otros) en los cuales la aplicación va a ser utilizada [10].

El concepto de aplicación web móvil “adaptable” (conocido como “web responsive”), el cual consiste en que el sitio puede adecuarse a los diferentes tamaños de pantalla de los dispositivos, ha surgido como un mecanismo para evitar construir la misma aplicación para cada tipo de dispositivo. Actualmente existen diversos frameworks de desarrollo web que facilitan este trabajo, tales como Bootstrap [11] Foundation [12], Bulma [13], entre otros.

De acuerdo a lo presentado en [2] y [3] las aplicaciones web móviles pueden ralentizar el tiempo de respuesta dado que requieren interacción cliente-servidor. Asimismo, resultan menos atractivas que una aplicación instalada en el dispositivo, ya que se utiliza el navegador web como intermediario. Por otra parte, no es posible utilizar todas las características del dispositivo (sensores, estado de la batería, almacenamiento, entre otros) ni hacer que la aplicación se ejecute en segundo plano [2].

3 PWA

Las aplicaciones denominadas PWA son aplicaciones web móviles que aprovechan las nuevas posibilidades y APIs que brindan las nuevas tecnologías Web, cómo los Service Worker (script que se ejecuta en segundo plano y permite la implementación de funcionalidades que no requieren de una página web ni interacción del usuario) [14] y Web App Manifest (archivo JSON que permite especificar metadatos de la aplicación tales como nombre, color e icono que la distingue) [15] [16] [17], entre otras. Esto permite que una aplicación web pueda incorporar algunas de las características históricamente exclusivas de las aplicaciones nativas, tales como funcionamiento offline, la recepción de notificaciones push y disponer de un ícono de acceso en el lanzador de aplicaciones.

PWA emplea un conjunto de tecnologías que permite a una aplicación web superar algunas de las limitaciones subyacentes al enfoque web móvil, y brindan al usuario la sensación de estar utilizando una aplicación nativa [18].

Si bien este enfoque está orientado a los dispositivos móviles, permite realizar instalaciones en computadoras de escritorio, por lo que constituye una posibilidad para unificar el desarrollo de aplicaciones, independientemente del tipo de dispositivo [19]. Esta característica pondera a las PWA por sobre el resto de los enfoques para el desarrollo de aplicaciones móviles.

Como contrapartida, debido a que las PWA se instalan por intermedio de un navegador web, su disponibilidad está limitada al soporte del mismo. Por ejemplo Safari, el navegador web de Apple, no es compatible con todas las APIs necesarias para la ejecución de las PWAs [20].

Al margen de las tecnologías necesarias para su implementación, Google propone una serie de requerimientos base que una aplicación web debe cumplir para ser considerada una PWA: (a) funcionar offline (sin conectividad se debe recuperar la información del último acceso), (b) responder cualquier solicitud en menos de 5 segundos, (c) brindar experiencia de uso similar a una aplicación nativa, (d) utilizar diseño adaptable y (e) utilizar el protocolo HTTPS [14][21][22]. En [23] se indica una lista completa de características deseables para una PWA bien diseñada e implementada.

Como prueba de concepto de esta nueva tecnología, se utilizó el sitio web de la Facultad de Informática de la UNLP [24]. Éste se encuentra desarrollado con el sistema de gestión de contenido Wordpress [25] y a través de la instalación del plugin Super PWA [26], se lo convirtió en una PWA. La configuración de este plugin requirió establecer un nombre e ícono para identificar la aplicación en el lanzador de aplicaciones del dispositivo; se definió la URL de inicio, la página a mostrar en caso de no tener conectividad y la orientación de inicio por defecto.

En la figura 1a se presenta el sitio accedido desde un navegador. En la parte inferior de la pantalla se muestra un banner para instalar la PWA en la pantalla principal del dispositivo. Una vez instalada, se genera un ícono de acceso en el lanzador de aplicaciones. La figura 1b reproduce la interfaz de la PWA de la Facultad de Informática de la UNLP.



Figura 1a. Aplicación web adaptable tradicional



Figura 1b. Aplicación PWA

4 Aplicaciones Web Móviles vs. PWA

Para las organizaciones implica un esfuerzo significativo desarrollar y mantener aplicaciones móviles específicas para cada sistema operativo. Por tal motivo resulta importante evaluar los enfoques de desarrollo multiplataforma y decidir cuál se adapta mejor a las necesidades del proyecto. La elección del enfoque a utilizar afecta el ciclo de vida de la aplicación. El cambio de enfoque en un proyecto avanzado tiene un alto costo de reingeniería.

En [2] los autores del presente trabajo realizaron un minucioso análisis comparativo de las características que deben ser evaluadas para determinar cuál enfoque de desarrollo se adecua a las necesidades de un proyecto, sin considerar PWA.

En esta sección se profundiza el análisis de aplicaciones web móviles utilizando la taxonomía de características y escala de valores posibles propuesta en [2]. El resultado

consiste en tres tablas comparativas entre aplicaciones web móviles tradicionales y PWA.

La Tabla 1 presenta características no funcionales a considerar en el desarrollo de aplicaciones móviles; la Tabla 2 describe aspectos técnicos de interés para los desarrolladores; y la Tabla 3 sintetiza las características relacionadas con la gestión del proyecto de software.

Tabla 1. Análisis comparativo de características no funcionales.

Característica \ Tecnología	Aplicaciones Web	PWA
Experiencia de usuario	Muy bajo	Alto
Interfaz de usuario	Web	Web
Performance	Muy bajo	Medio
Modo de instalación	No requiere instalación. Se accede a través de un navegador web	Requiere instalación a través de un navegador web
Consumo de batería	Bajo	Bajo
Uso del disco/tamaño de la aplicación	Muy bajo	Bajo
Renderización de la imagen	Muy alto	Muy alto
Tiempo de inicio	Bajo	Muy bajo

Tabla 2. Análisis comparativo de características de desarrollo.

Tecnología Característica	Aplicaciones Web	PWA
Entorno de desarrollo integrado (IDE)	Múltiples opciones. No hay un IDE oficial.	Múltiples opciones. No hay un IDE oficial.
Lenguaje de programación	HTML, CSS, JavaScript más otro lenguaje del lado del servidor	HTML, CSS, JavaScript más otro lenguaje del lado del servidor
Open source/Licencia y costo	Libre	Libre
Diseño de interfaz de usuario	HTML, CSS, JavaScript.	HTML, CSS, JavaScript.
Curva de aprendizaje	Muy bajo	Bajo
Acceso a características específicas del dispositivo.	Muy bajo	Medio

Tabla 3. Análisis comparativo de características de la gestión de proyectos de software

Característica \ Tecnología	Aplicaciones Web	PWA
Plataformas destino	Todas	Todas (iOS presenta algunas limitaciones)
Velocidad y costo de desarrollo	Muy bajo	Bajo
Mantenimiento	Muy bajo	Muy bajo
Grado de madurez	Muy alta	Medio
Viabilidad a largo plazo	Muy alta	Alto
Categoría de la aplicación	Social, Turística o Institucional	Social, Turística o Institucional
Uso offline	No	Si
Reusabilidad de código	Total	Total
Distribución	A través de un navegador web	A través de un navegador web.
Acceso	A través de un navegador web	Inicio desde el lanzador de aplicaciones.
Usuarios Potenciales	Ilimitado	Ilimitado

Se puede apreciar que los valores en muchas características son exactamente iguales. No obstante existen diferencias que merecen ser destacadas. Las PWA ofrecen ventajas en los siguientes aspectos: (a) el acceso se realiza directamente desde el lanzador de aplicaciones sin la necesidad de recurrir al navegador, (b) la experiencia de usuario se ve mejorada debido a que se ejecuta en pantalla completa, ocultando los elementos del

navegador, (c) la performance es superior dado que se encuentran instaladas en el dispositivo y utilizan contenido almacenado en caché, (d) permiten el acceso a características específicas del dispositivo como por ejemplo la posibilidad de recibir notificaciones push y (e) funcionan sin conectividad.

Por otra parte, las aplicaciones web móviles tradicionales resultan una mejor alternativa en los siguientes aspectos: (a) son más viables a largo plazo dado que no están condicionadas por la plataforma, (b) la curva de aprendizaje y el esfuerzo de desarrollo es menor dado que no requieren contemplar los requisitos presentados en la sección 3 para las PWA y (c) el grado de madurez es superior dada la aparición reciente de las PWA.

5 Conclusiones y Trabajo Futuro

En este trabajo se analiza PWA, un nuevo enfoque de desarrollo de aplicaciones multiplataforma que permite a las aplicaciones web tradicionales incorporar características que son propias del enfoque móvil nativo.

Como prueba de concepto de este nuevo enfoque, se utilizó el sitio web de la Facultad de Informática de la UNLP.

Dado que una PWA se construye a partir de una aplicación web móvil, resulta de interés realizar un análisis comparativo entre ellas. A partir de este análisis, se pueden señalar aspectos de las PWA que las convierten en una mejor opción:

- a) se instalan en el lanzador de aplicaciones
- b) se ejecutan desde el lanzador de aplicaciones
- c) funcionan sin conectividad
- d) mejoran la experiencia de usuario
- e) permiten incorporar características de aplicaciones nativas, como por ejemplo la recepción de notificaciones push

Por otra parte, debido a su reciente aparición, presentan las siguientes desventajas respecto de las aplicaciones web tradicionales:

- a) menor grado de madurez
- b) mayor curva de aprendizaje y esfuerzo de desarrollo
- c) presentan incompatibilidades en algunas plataformas, por ejemplo iOS [20]

No obstante este enfoque está orientado a los dispositivos móviles, es de destacar que permite realizar instalaciones en computadoras de escritorio, por lo que constituye una posibilidad para unificar el desarrollo de aplicaciones, independientemente del tipo de dispositivo y sistema operativo.

Por todo lo expuesto, se concluye que las PWA tienen el potencial suficiente para establecerse como una alternativa sólida en el desarrollo de software.

A efectos de profundizar el estudio del presente trabajo, se plantea extender el análisis comparativo de PWA con los restantes enfoques de desarrollo considerados en [2].

Como futura línea de investigación, se propone explorar el concepto de Trusted Web Activity (TWA) [27] y su relación con PWA.

Bibliografía

1. Delia, L., Galdamez, N., Thomas, P., Corbalan, L., & Pesado, P. (2015, May). Multi-platform mobile application development analysis. In *Research Challenges in Information Science (RCIS)*, 2015 IEEE 9th International Conference on (pp. 181-186). IEEE.
2. Lisandro Delia, Pablo Thomas, et. al., *Development Approaches for Mobile Applications: Comparative Analysis of Features*. SAI 2018: Intelligent Computing. Proceedings of the 2018 Computing Conference, Volume 2. Springer, ISBN: 978-3-030-01176-5, Noviembre 2018.
3. Spyros Xanthopoulos, Stelios Xinogalos, *A Comparative Analysis of Cross-platform Development Approaches for Mobile Applications*, BCI 2013, Greece
4. A Survey and Taxonomy of Core Concepts and Research Challenges in Cross-Platform Mobile Development. November 2018 *ACM Computing Surveys* 51(5):1-34 DOI: 10.1145/3241739 Andreas Bjørn-Hansen Andreas Bjørn-Hansen Tor-Morten Grønli Tor-Morten Grønli Gheorghita Ghinea Gheorghita Ghinea
5. Dalmasso I., Datta S.K., Bonnet C. Nikaein N., *Survey, comparison and evaluation of cross platform mobile application development tools*, *Wireless Communications and Mobile Computing Conference (IWCMC)*, 2013 9th International.
6. L. Delía, N. Galdamez, L. Corbalan, P. Pesado and P. Thomas, "Approaches to mobile application development: Comparative performance analysis," 2017 Computing Conference, London, 2017.
7. Corbalán L., Thomas Pablo, Lisandro Delía, et al., *A Study of Non-functional Requirements in Apps for Mobile Devices*, *Communications in Computer and Information Science*, vol 1050. Springer, Cham, Springer, ISBN 978-3-030-27712-3, Julio 2019
8. Corbalan L.; Fernandez Sosa J.; Cuitiño A.; Delia L.; Caseres G.; Thomas P.; Pesado P., *Development Frameworks for Mobile Devices: A Comparative Study about Energy Consumption (ICSE)*, *MobileSoft 2018 5th IEEE/ACM International Conference on Mobile Software Engineering and Systems on*, Gothenburg Sweden, 2018.
9. <https://events.google.com/io2016/>
10. <https://developers.google.com/web/fundamentals/design-and-ux/responsive/>
11. <https://getbootstrap.com/>
12. <https://foundation.zurb.com/>
13. <https://bulma.io/>
14. Matt Gaunt. *Introducción a los service workers*
15. <https://developers.google.com/web/fundamentals/web-app-manifest/>
16. <https://www.w3.org/TR/appmanifest/>
17. <https://developer.mozilla.org/en-US/docs/Web/Manifest>
18. *Progressive Web Apps: an alternative to the native mobile Apps (portugués)*. 2018 13th Iberian Conference on Information Systems and Technologies (CISTI)
19. <https://developers.google.com/web/progressive-web-apps/desktop>
20. *Progressive Web Apps for the Unified Development of Mobile Applications*
21. <https://developers.google.com/web/fundamentals/codelabs/your-first-pwapp/>
22. <https://developers.google.com/web/progressive-web-apps>
23. <https://developers.google.com/web/progressive-web-apps/checklist>
24. <https://info.unlp.edu.ar>
25. <https://es.wordpress.com/>
26. <https://superpwa.com/>
27. <https://developers.google.com/web/updates/2019/02/using-twa>

Rapid Releases and Testing Problems at the industry: A survey

Maximiliano A. Mascheroni^{1,2}, Emanuel Irrazábal², Juan A. Carruthers²,
Juan A. Pinto²

¹ Facultad de Informática. Universidad Nacional de La Plata.
La Plata, Buenos Aires, Argentina.

² Facultad de Ciencias Exactas y Naturales y Agrimensura.
Universidad Nacional del Nordeste. Corrientes, Argentina.

{agustin.mascheroni, emanuelirrazabal}@gmail.com
{juan.carruthers, juan_al_pinto}@hotmail.com

Abstract. Rapid releases and continuous software development are established practices in modern agile projects. The advantages of them are widely known across the software development community, but there are some studies which mention that there are still challenges to face. According to them, there are different open issues which are affecting the implementation of an adequate testing process. With the aim of validating if these problems are present in real projects, in this paper we present the results of a survey whose goal was to validate whether the industry is experiencing similar issues and their causes. The findings demonstrate that both the industry and academic side are aligned, and that there is still a need for processes and tools regarding the testing process in continuous development.

Keywords: Continuous testing, continuous software development, continuous delivery, continuous deployment, survey.

1 Introduction

Currently, there are many companies which develop software that are moving from traditional release cycles to rapid releases. Practitioners of continuous deployment (CDP) or continuous delivery (CD) concepts, claim that deploying software to production continuously offers various benefits to companies and their end users [1–4]. However, according to some authors, there are still some challenges to face [5–11, 12, 13]. One of these claims is the relation between rapid release models and software quality [14]. Examples of software quality issues are: not enough time for testing [15], slow bug fixing [16], small scope for the testing stage [17].

With the goal of identifying any type of problems related to software quality in continuous development environments, several studies have been carried out by many authors. It has been found that the most reported issues are: *time-consuming testing*; *flaky tests*; *ambiguous test results*; *GUI automated testing*; *dynamic Web UI*

automated testing; data testing; big data testing; mobile automated testing, non-functional automated testing; automated testing of applications composed by cloud services; testing as a service; and webservice automated testing [11, 18, 19].

Also, in [18] it has been analyzed the relationship between these problems. It has been found that the root problems are time-consuming testing, flaky tests and lack of frameworks for testing certain type of applications. Fig 1 shows this relationship.

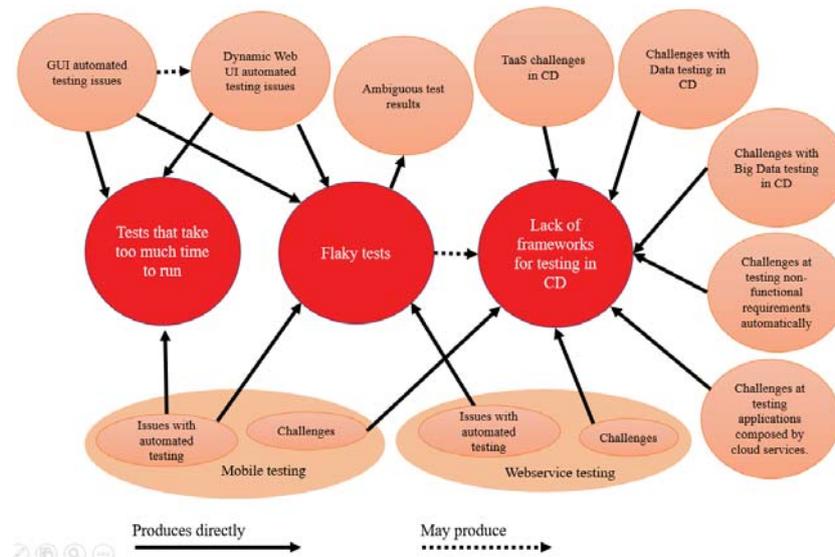


Fig. 1. Relationship between CD testing problems [18].

As it was aforementioned, the findings of the studies show that the more important testing-related issues in rapid releases are time-consuming testing, flaky tests and lack of frameworks. However, those results were taken only from academic sources which are not directly related to the industry. Thus, this research aims to validate whether the industry is facing the same problems, they have never experienced them, or they have already got rid of them.

Apart from this introductory section, in Section 2 it is described the survey and its components. The findings of the survey are presented in Section 3, which are discussed later in Section 4. Finally, conclusions and future work are highlighted in Section 5.

2 The survey

This research employed a survey as the main data gathering method, following Kitchenham's principles of survey research [20–25]. In this section, it is described the procedure followed to design and conduct the study.

2.1 Research goals

The first step was setting the goals which were derived from previous studies. In the context of rapid releases at the industry, the goals of this research are:

- Validate if the industry is having the same testing problems reported by the academic sources.
- Detect new challenges and open issues at implementing the testing process.

2.2 Survey design and development

The survey is a cross sectional survey [26], because it is necessary to get a snapshot about the testing process in the present. The survey instrument was the questionnaire, and after the design of it, a pilot process was performed in order to test the validity and readability of the questions. As part of it, project managers and team mates supplied feedback for improving the survey. Based on that feedback, and after making the modifications to the survey, the process was repeated with other project managers.

The target population was companies that have projects and teams which work with rapid releases. However, as it is impossible to send a questionnaire to all of the continuous software development projects around the world, based on similar surveys that have a sample size in a range between 100 and 250 respondents [27–32], the sample size for this research was 255 projects. The method for sharing the questionnaire was the use of social networks. It was provided a briefly introduction where respondents were able to see the goals of the study, the value of their participation and how they were going to be benefited from it. According to Kitchenham [23], respondents will be more motivated to supply complete and precise responses if they can see that the results are likely to be useful to them.

There were four sections in the survey, where the first section was the introductory part with the purpose of the study, concepts and clarifications. The second section was on metadata (project information) such as type of software being developed, the number of members in the team, their roles and timebox duration. The third section was on the use of continuous development practices, which results will be used for other studies. The fourth section was on problems, challenges and solutions in the testing process using open questions. The reason for open questions is because they allow respondents to describe all the problems they have and the solutions or workarounds they implemented. The survey can be found in the following link: <https://forms.gle/8ULNMgVxzone3WvE6>.

3 Results

In this section, the findings of the survey are presented. During a period of four months, a total of 287 responses were obtained. However, after rejecting results with incomplete questionnaires and with answers that were not consistent each other, a total of 255 projects was gotten.

The projects were grouped based on the size of the team and the type of application being developed (see Fig. 2). It is very important to highlight that most of the teams are working in more than one platform. For the team size, it was used a classification proposed by Yang et al. [33], based on O'Connor and Yang metrics [34], as follows:

- Small team: < 16 members.
- Medium team: 16 – 45 members.
- Large team: > 45 members.

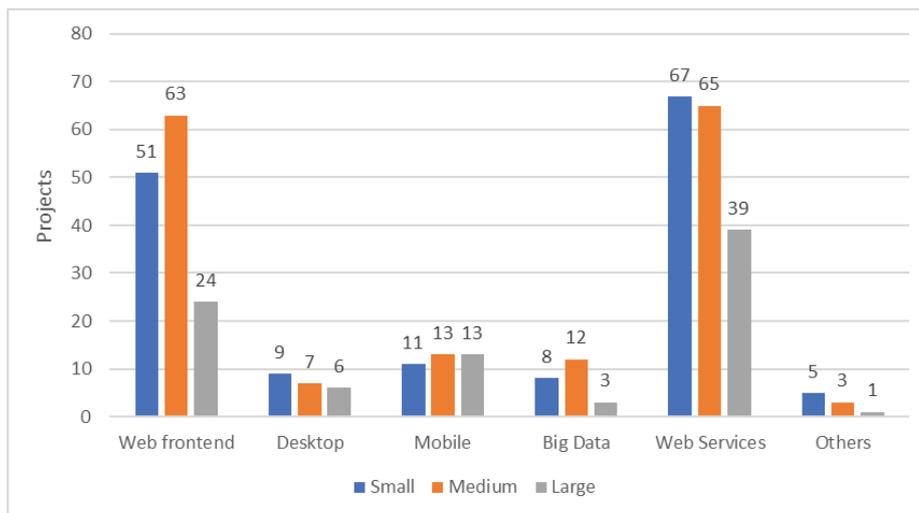


Fig. 2. Number of projects grouped by team size and platform being developed.

3.1 Testing Problems at the industry

As it was described in Section 1, the literature reports that there are some problems related to the testing in continuous development environments. The purpose of this survey was to validate whether the industry is facing the same issues or not. The results obtained from the survey can be seen in Table 1 and Fig. 3.

As it can be seen in Table 1, most of the outlined problems match with the ones reported by the literature. There were also new problems detected:

1. Lack of procedures, patterns and good practices for automated testing in continuous development.
2. Unstable environments.

Flaky test was the problem with greater occurrence (224 times). According to the companies, there are several reasons for having flaky tests such as missing or incorrect test data, GUI components that cannot be found, timeouts due to network issues or unstable environments, dynamic GUI elements which take long to load and inconsistency test code. Regarding GUI automated testing, the most reported issue

was maintaining the test code at changes on the UI element's locators. Furthermore, GUI automated tests are tests that take much longer time to run than any other test. This issue was reported by 179 companies. In the same way, running automated scripts on a page which has dynamic components is a challenge for 47 projects, because they have mentioned that tests fail when those elements are not shown yet. Time-consuming testing affects also mobile (17) and web service testing (58).

Table 1. Testing problems found by the survey.

Problem	N° of projects
Flaky tests	224
Time-consuming testing	179
GUI automated testing	124
Lack of procedures, patterns and good practices for automated testing in rapid releases environments	112
Ambiguous test results	65
Web service testing	58
Dynamic Web UI automated testing	47
Unstable environments	36
Data testing	26
Non-functional automated testing	23
Big Data testing	18
Mobile testing	17
Automated testing of applications composed by cloud services	2

A group of 26 people have reported that testing data attributes automatically is also a challenge. Similarly, there are some projects implementing big data tools and frameworks and most of them have stated that there is a lack of information, documentation or guides on the web about big data testing. These projects mentioned the need for verification mechanisms at ETL stages and data analysis. Other companies have outlined issues with automated testing in mobile applications, where they highlight the lack of robust frameworks, processes and models, especially for running the tests continuously. The same problems are faced by teams which work on web services or that develop applications with cloud services.

For testing non-functional requirements, 17 projects were not able to integrate performance test scripts on a CI pipeline. Another 6 teams reported issues with lack of documentation and tools about security testing on CI, CDP or CD. Also, according to 65 people, understanding the reports is an issue because lack of details about failures.

Finally, 112 projects stated in different ways that there is a need for procedures, patterns and good practices for automated testing in rapid releases environments. For example, some teams mentioned that there is lack of good practices for getting a good coverage of automated tests for different layers in the application. Other projects have stated that there is a lack of standards about test data management strategies and test automation framework architecting. Some respondents mentioned that while there are thousands of tutorials, trainings, blogs and courses about traditional test automation, there is lack of them about continuous test automation (continuous testing).

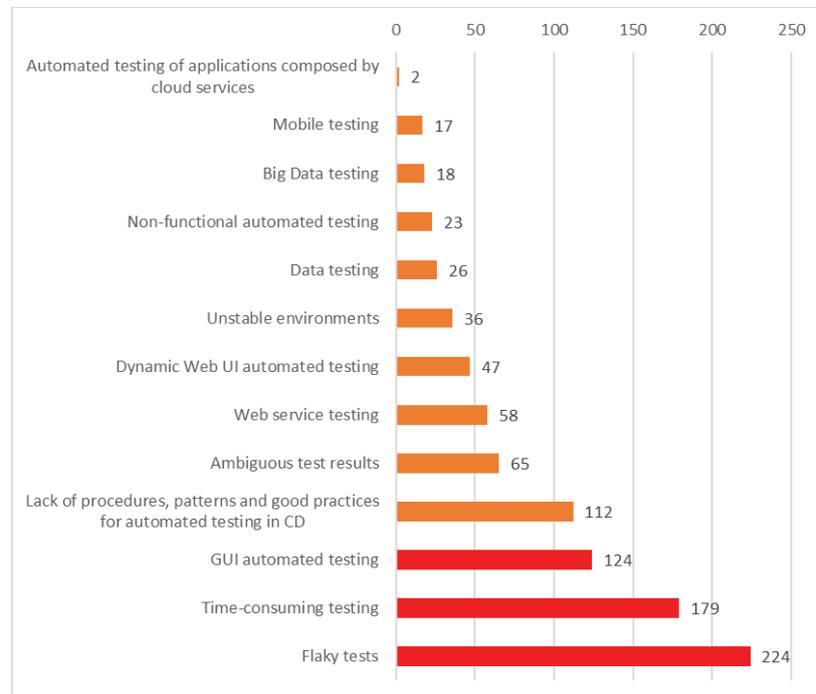


Fig. 3. Reported problems.

4 Discussion

In the last section the results found in the survey have been presented. From there, a deeper analysis can be performed, and different conclusions can be taken. In this section, the findings of the survey will be analyzed by discussing the reported testing problems.

The most reported problems were flaky automated tests, tests that take long to run, automated tests for GUI applications and lack of frameworks, tools, procedures and documentation for automated testing in rapid releases. The metadata obtained from the second part of the survey will be used to analyze each testing problem. Thus, the analysis is used later for determining whether a problem reported by the literature is also valid for the industry or not.

Table 2 presents the number of testing problems by the type of platform being developed using the following annotation:

- P1: Flaky tests
- P2: Time-consuming testing
- P3: GUI automated testing
- P4: Lack of procedures, patterns and good practices for automated testing in rapid releases environments

- P5: Ambiguous test results
- P6: Web service testing
- P7: Dynamic Web UI automated testing
- P8: Unstable environments
- P9: Data testing
- P10: Non-functional automated testing
- P11: Big Data testing
- P12: Mobile testing
- P13: Automated testing of applications composed by cloud services

Table 2. Number of testing problems by type of platform being developed.

Platform	P1	P2	P3	P4	P5	P6	P7	P8	P9	P10	P11	P12	P13
Web frontend	134	126	92	46	34	0	47	18	12	12	0	0	1
Desktop	17	19	17	16	7	0	0	4	2	2	0	0	0
Mobile	19	29	14	12	5	0	0	6	1	2	0	17	0
Big Data	1	1	0	15	0	0	0	1	1	1	18	0	0
Web Services	48	3	0	20	19	58	0	7	8	4	0	0	0
Other	5	1	1	3	0	0	0	0	2	2	0	0	1
Total	224	179	124	112	65	58	47	36	26	23	18	17	2

Flaky tests (P1), time-consuming testing (P2) and lack of frameworks (P4) were reported in all the different type of projects. Data testing (P9) and non-functional automated testing (P10) were also outlined by all of them, but they just represent a low percentage of the total: P9 is 10% (26 over 255) and P10 is 9% (23 over 255), where 255 is the total of projects. Using the same formula, P1 is 87%, P2 is 70% and P4 is 43%.

GUI automated testing (P3) was reported by all the teams except the ones that work on big data and web service-only applications. Thus, the total of projects that will be considered are 198 (as the sum of all the platforms except big data and web service projects). After applying the mentioned formula, P3 is 62% (124 over 198).

Using the same criteria: web service testing (P6) is 34% (58 over 171), dynamic Web UI automated testing (P7) is 34% (47 over 138), big data testing (P11) is 78% (18 over 23), mobile testing (P12) is 45% (17 over 37), and automated testing of applications composed by cloud services is 1% (2 over 147).

As regards of unstable environments (P8) and ambiguous test results (P5), not all the teams have reported this problem. However, on the one hand, a report is an artifact of any type of test execution (no matter the platform). On the other hand, any type of test runs on a test environment which may be unstable or not. Thus, those problems are not tied to a certain type of technology. Therefore, P5 is 25% (65 over 255) and P8 is 14% (36 over 255).

Finally, Table 3 shows the severity of each problem reported by the literature at the industry, using the following criteria: 0% is not an issue; 0,1% to 19,9% is a very low severity issue; 20% to 39,9% is a low severity issue; 40% to 59,9% is a medium severity issue; 60% to 79,9% is a high severity issue; and finally 80% to 100% is a critical issue.

Table 3. Relationship between the testing problems in the literature and the industry.

Problem	Reported by the literature	Severity at the industry
Flaky tests	Yes	Critical (87%)
Time-consuming testing	Yes	High (70%)
GUI automated testing	Yes	High (62%)
Lack of procedures, patterns and good practices for automated testing in rapid releases environments	No	Medium (43%)
Ambiguous test results	Yes	Low (25%)
Web service testing	Yes	Low (34%)
Dynamic Web UI automated testing	Yes	Low (34%)
Testing as a service (TaaS)	Yes	Not an issue (0%)
Unstable environments	No	Very Low (14%)
Data testing	Yes	Very Low (10%)
Non-functional automated testing	Yes	Very Low (9%)
Big Data testing	Yes	High (78%)
Mobile testing	Yes	Medium (45%)
Automated testing of applications composed by cloud services	Yes	Very Low (1%)

5 Conclusions and future works

In this paper it has been presented the findings of a survey whose main goal was to validate whether the industry is experiencing the same testing problems that academic sources have reported.

Most of the literature's testing problems match with the ones reported by the organizations which have participated. The only issue that was not reported by these companies was Testing as a Service (TaaS). On the other hand, two new issues have been outlined by the respondents: unstable environments, and lack of frameworks such as procedures, good practices, models, and patterns for continuous automated testing in rapid releases.

The results also have demonstrated that most of the issues that can be found in the literature are present in the industry too. The most critical problem found at the industry is having flaky tests. Also, time-consuming testing, GUI automated testing and big data testing are high severity issues.

The findings of this survey might contribute to new research lines on software quality. It is therefore that in future works deeper research will be carrying on, focusing on the most critical testing problems in rapid releases.

Acknowledgment

This paper was developed in the context of a PhD tesis on computer science and it has been supported by the projects "Metodologías y herramientas emergentes para contribuir con la calidad del software" (PI 17F018 SCyT UNNE) and "Análisis e

Implementación de tecnologías emergentes en sistemas computacionales de aplicación regional” (PI 17F017 SCyT UNNE).

We also thank to the 255 respondents of the survey for their contribution to this study.

References

1. Humble, J., Farley, D.: Continuous Delivery: Reliable Software Releases through Build, Test, and Deployment Automation. Pearson Professional, Upper Saddle River, NJ (2010).
2. Duvall, P., Matyas, S., Glover, A.: Continuous integration: Improving software quality and reducing risk. Pearson Professional, Upper Saddle River, NJ (2007).
3. Fowler, M.: Continuous Integration, <https://martinfowler.com/articles/continuousIntegration.html>.
4. Fowler, M.: Continuous Delivery, <https://martinfowler.com/bliki/ContinuousDelivery.html>.
5. Chen, L.: Continuous Delivery at Scale: Challenges and Opportunities. In: 2018 IEEE/ACM 4th International Workshop on Rapid Continuous Software Engineering (RCoSE). pp. 42–42 (2018).
6. Chen, L.: Continuous Delivery: Huge Benefits, but Challenges Too. *IEEE Software*. 32, 50–54 (2015). <https://doi.org/10.1109/MS.2015.27>.
7. Chen, L.: Continuous Delivery: Overcoming adoption challenges. *Journal of Systems and Software*. 128, 72–86 (2017). <https://doi.org/10.1016/j.jss.2017.02.013>.
8. Neely, S., Stolt, S.: Continuous Delivery? Easy! Just change everything (well, maybe it is not that easy). In: 2013 Agile Conference. pp. 121–128 (2013). <https://doi.org/10.1109/AGILE.2013.17>.
9. Fitzgerald, B., Stol, K.-J.: Continuous software engineering and beyond: Trends and challenges. In: Proceedings of the 1st International Workshop on Rapid Continuous Software Engineering. pp. 1–9. ACM, New York, NY, USA (2014). <https://doi.org/10.1145/2593812.2593813>.
10. Shahin, M., Babar, M.A., Zhu, L.: Continuous Integration, Delivery and Deployment: A Systematic Review on Approaches, Tools, Challenges and Practices. *IEEE Access*. 5, 3909–3943 (2017). <https://doi.org/10.1109/ACCESS.2017.2685629>.
11. Laukkanen, E., Itkonen, J., Lassenius, C.: Problems, causes and solutions when adopting continuous delivery — A systematic literature review. *Information and Software Technology*. 82, 55–79 (2017). <https://doi.org/10.1016/j.infsof.2016.10.001>.
12. Shankland, S.: Rapid-release Firefox meets corporate backlash, <https://www.cnet.com/news/rapid-release-firefox-meets-corporate-backlash/>.
13. Kaply, M.: Why Do Companies Stay on Old Technology?, <https://mike.kaply.com/2011/06/23/why-do-companies-stay-on-old-technology/>.
14. Khomh, F., Dhaliwal, T., Zou, Y., Adams, B.: Do faster releases improve software quality? An empirical case study of Mozilla Firefox. In: 2012 9th IEEE Working Conference on Mining Software Repositories (MSR). pp. 179–188 (2012). <https://doi.org/10.1109/MSR.2012.6224279>.
15. Porter, A., Yilmaz, C., Memon, A.M., Krishna, A.S., Schmidt, D.C., Gokhale, A.: Techniques and processes for improving the quality and performance of open-source software. *Software Process: Improvement and Practice*. 11, 163–176 (2006). <https://doi.org/10.1002/spip.260>.
16. Baysal, O., Davis, I., Godfrey, M.W.: A Tale of Two Browsers. In: Proceedings of the 8th Working Conference on Mining Software Repositories. pp. 238–241. ACM, New York, NY, USA (2011). <https://doi.org/10.1145/1985441.1985481>.

17. Mäntylä, M.V., Khomh, F., Adams, B., Engström, E., Petersen, K.: On Rapid Releases and Software Testing. In: 2013 IEEE International Conference on Software Maintenance. pp. 20–29 (2013). <https://doi.org/10.1109/ICSM.2013.13>.
18. Mascheroni, M.A., Irazábal, E.: Problemas que afectan a la calidad de software en entrega continua y pruebas continuas. Presented at the XXIV Congreso Argentino de Ciencias de la Computación (La Plata, 2018). (2018).
19. Mascheroni, M.A., Irazábal, E.: Continuous Testing and Solutions for Testing Problems in Continuous Delivery: A Systematic Literature Review. *Computación y Sistemas*. 22, (2018). <https://doi.org/10.13053/cys-22-3-2794>.
20. Pfleeger, S.L., Kitchenham, B.A.: Principles of Survey Research: Part 1: Turning Lemons into Lemonade. *SIGSOFT Softw. Eng. Notes*. 26, 16–18 (2001). <https://doi.org/10.1145/505532.505535>.
21. Kitchenham, B.A., Pfleeger, S.L.: Principles of Survey Research Part 2: Designing a Survey. *SIGSOFT Softw. Eng. Notes*. 27, 18–20 (2002). <https://doi.org/10.1145/566493.566495>.
22. Kitchenham, B.A., Pfleeger, S.L.: Principles of Survey Research: Part 3: Constructing a Survey Instrument. *SIGSOFT Softw. Eng. Notes*. 27, 20–24 (2002). <https://doi.org/10.1145/511152.511155>.
23. Kitchenham, B., Pfleeger, S.L.: Principles of Survey Research Part 4: Questionnaire Evaluation. *SIGSOFT Softw. Eng. Notes*. 27, 20–23 (2002). <https://doi.org/10.1145/638574.638580>.
24. Kitchenham, B., Pfleeger, S.L.: Principles of Survey Research: Part 5: Populations and Samples. *SIGSOFT Softw. Eng. Notes*. 27, 17–20 (2002). <https://doi.org/10.1145/571681.571686>.
25. Kitchenham, B., Pfleeger, S.L.: Principles of Survey Research Part 6: Data Analysis. *SIGSOFT Softw. Eng. Notes*. 28, 24–27 (2003). <https://doi.org/10.1145/638750.638758>.
26. Kitchenham, B.A., Pfleeger, S.L.: Personal Opinion Surveys. In: Shull, F., Singer, J., and Sjøberg, D.I.K. (eds.) *Guide to Advanced Empirical Software Engineering*. pp. 63–92. Springer London, London (2008). https://doi.org/10.1007/978-1-84800-044-5_3.
27. Koschke, R.: Software Visualization in Software Maintenance, Reverse Engineering, and Re-engineering: A Research Survey. *Journal of Software Maintenance*. 15, 87–109 (2003). <https://doi.org/10.1002/smr.270>.
28. Chow, T., Cao, D.-B.: A survey study of critical success factors in agile software projects. *Journal of Systems and Software*. 81, 961–971 (2008). <https://doi.org/10.1016/j.jss.2007.08.020>.
29. Lethbridge, T.C.: A Survey of the Relevance of Computer Science and Software Engineering Education. In: *Proceedings of the 11th Conference on Software Engineering Education and Training*. pp. 0056–. IEEE Computer Society, Washington, DC, USA (1998).
30. R. VIJAYASARATHY, L., TURK, D.: Agile software development: A survey of early adopters. *Journal of Information Technology Management*. 19, (2008).
31. Garousi, V., Zhi, J.: A survey of software testing practices in Canada. *Journal of Systems and Software*. 86, 1354–1376 (2013). <https://doi.org/10.1016/j.jss.2012.12.051>.
32. Mao, K., Capra, L., Harman, M., Jia, Y.: A survey of the use of crowdsourcing in software engineering. *Journal of Systems and Software*. 126, 57–84 (2017). <https://doi.org/10.1016/j.jss.2016.09.015>.
33. Yang, L.-R., Huang, C.-F., Wu, K.-S.: The association among project manager's leadership style, teamwork and project success. *International Journal of Project Management*. 29, 258–267 (2011). <https://doi.org/10.1016/j.ijproman.2010.03.006>.
34. T. O'Connor, J., Yang, L.-R.: Project Performance versus Use of Technologies at Project and Phase Levels. *Journal of Construction Engineering and Management-asce - J CONSTR ENG MANAGE-ASCE*. 130, (2004). [https://doi.org/10.1061/\(ASCE\)0733-9364\(2004\)130:3\(322\)](https://doi.org/10.1061/(ASCE)0733-9364(2004)130:3(322)).

Reparación de programas aplicando templates: Generando workarounds permanentes a través de SAT

Marcelo Uva¹, Pablo Ponzio^{1,2} y Nazareno Aguirre^{1,2}

¹ Universidad Nacional de Río Cuarto (UNRC)

² Consejo Nacional de Investigaciones Científicas y Técnicas (CONICET)
{uva,pponzio,naguirre}@dc.exa.unrc.edu.ar

Resumen La aplicación de procesos modernos y sistemáticos en el desarrollo de sistemas informáticos con estrictos estándares de calidad no ha podido evitar la presencia de defectos en el software. A pesar de extensas test-suites, un alto porcentaje de sistemas son implantados con fallas. El desarrollo de técnicas de reparación de programas ha cobrado gran importancia en esta última década. El concepto de *workaround* ha sido utilizado por diversos enfoques para la reparación de programas. Un *workaround* aprovecha la redundancia implícita en un sistema con el propósito de evitar fallas detectadas en runtime. A partir de una técnica de cómputo automático de *state-bound workarounds* (workarounds dependientes de estado) que utiliza SAT-Solving, se propone un procedimiento de búsqueda de *workaround templates* independientes de estado, es decir, *state-free workarounds* que permitan mejorar los tiempos de reparación de programas.

Keywords: Reparación de programas, Workarounds, SAT-Solving

1. Introducción

La aplicación de procesos modernos y sistemáticos en el desarrollo de sistemas informáticos cumpliendo con estrictos estándares de calidad no ha podido evitar la aparición de defectos en el software. A pesar de ejecutar extensas test-suites, un alto porcentaje de los sistemas es implantado con fallas. Muchas de las denominadas fallas de campo, requieren de un período de tiempo largo para ser solucionadas efectivamente por los desarrolladores luego de ser reportadas. Esto se debe, entre otras causas, a la dificultad en reproducir aquellos ambientes en los cuales las fallas ocurren. La complejidad del software, la constante adaptación y extensión de los sistemas, sumada a la presión por disminuir los tiempos de producción y comercialización, define un contexto no muy favorable a la producción de software confiable. Estas circunstancias, combinadas con las demandas de disponibilidad del software hacen que las técnicas, que dotan al software de la capacidad de tolerar y reparar fallas hayan cobrado gran importancia en la actualidad. La noción de *workaround* es utilizada en este sentido y aprovecha

la redundancia intrínseca contenida dentro de un módulo, buscando ejecuciones alternativas a fin de reparar errores introducidos por los desarrolladores o bien provocadas por factores ambientales. En los trabajos [1,2,3] se ha utilizado el concepto de *automatic workaround* como mecanismo de recuperación de fallas sobre dominios específicos como por ejemplo, aplicaciones web. El cómputo de *workarounds* en estos enfoques es realizado sobre modelos abstractos (máquinas de estados) generadas manualmente a partir de un análisis acabado del comportamiento del sistema. En esta misma línea de investigación, en [4,5] se propone un enfoque similar, utilizando en este caso secuencias alternativas equivalentes definidas por los usuarios. Los autores del presente trabajo han propuesto técnicas que utilizan tecnologías basadas en SAT-Solving para el cómputo automático de *workarounds*[6]. A diferencia de los enfoques previos mencionados, estas técnicas son aplicadas sobre abstracciones propias de programas orientados a objetos, generadas a partir de especificaciones formales JML (Java Modeling Language)[7] incluídas en el código fuente de programas Java a modo de anotaciones. Estas técnicas utilizan el estado corriente donde se ha originado la falla como parte activa del cómputo de los *workarounds*, estableciendo así una especialización del concepto de *workaround* en *state-bound workarounds* y *state-free workaround*. En el primer caso, las soluciones alternativas utilizan el estado corriente del sistema para proporcionar una ejecución de reparación y en el segundo caso las soluciones son independientes del estado. Idealmente se desearía contar siempre con *state-free workarounds* para utilizarlos ante la ocurrencia de un error, lamentablemente, este tipo de *workarounds* son más costosos de encontrar y en muchas ocasiones, no existen. Un concepto derivado de estas técnicas es el de *workaround templates*. Estos definen una clase especial de *workarounds* que se mueven entre los dos tipos anteriores. Un *workaround templates* es una solución alternativa generalizable a partir de una serie de *state-bound workarounds* obtenidos. Un *workaround templates* puede ser categorizado en muchos casos como un *state-free workaround* haciendo intervenir minimamente a un SAT-Solver para concretizar los parámetros generalizados.

En este trabajo se propone un mecanismo de cómputo de *workaround templates* independientes de estado, esto permitirá su utilización como *state-free workaround* aplicando SAT-Solving. El resto del presente trabajo está organizado de la siguiente manera: en la sección 2 una revisión de los trabajos vinculados a la propuesta. En la sección 3 se presenta la técnica de reparación de la cual se deriva el procedimiento de cómputo de *workaround templates* independientes de estado. El mecanismo de detección de *workaround templates* y su posterior evaluación como *state-free workaround* es presentado en la sección 4. Luego, en la sección 5 se presentan algunos resultados obtenidos. Finalmente, las conclusiones y posibles trabajos futuros se presentan en la sección 6.

2. Preliminares

El concepto de *workaround* ha sido utilizado en técnicas de reparación de programas a través de diversos enfoques. Se presentan a continuación algunos

de los trabajos más relevantes. También se describen brevemente los lenguajes de modelado Alloy[8] y DynAlloy[9,10] junto con sus correspondientes herramientas de análisis automático: Alloy Analyzer y DynAlloy Analyzer utilizadas en el presente trabajo.

2.1. Automatic workarounds

El concepto de *workaround* fue definido inicialmente en el contexto de sistemas *self-healing* [1]. Un *workaround* aprovecha la redundancia implícita dentro de un sistema de software con el propósito de salvar las fallas que puedan producirse en él. Un *workaround* para una rutina m en un estado s_i define un procedimiento P compuesto por una secuencia de otras acciones pertenecientes al mismo módulo que permiten alcanzar el estado esperado s_f . Con el objetivo de automatizar el cómputo de *workarounds* en [1,3] se presentó un procedimiento que requiere de un modelo (máquina abstracta de estados) con el comportamiento de todo el sistema para poder encontrar estas alternativas de ejecución. El mismo grupo de investigación propuso también en [4] una arquitectura de reparación en donde inicialmente se debe proveer un conjunto de *workarounds* mediante reglas de re-escritura de términos. A partir de las mismas, el mecanismo de reparación tendrá la posibilidad de encontrar nuevas reparaciones combinando las ya conocidas.

2.2. Alloy

Alloy[8] es un lenguaje de especificación de primer orden basado en el uso de relaciones y orientado al modelado de propiedades estructurales de sistemas. Alloy utiliza operadores relacionales, definidos en lógica relacional (formalismo subyacente en el lenguaje) tales como el operador de composición, clausura transitiva y reflexo-transitiva con la finalidad de reducir la complejidad de la expresión de algunas propiedades estructurales comunes en lógica de primer orden. Alloy posee una sintaxis simple y clara, fácil de comprender. Alloy Analyzer es una herramienta de verificación que implementa un análisis automático de las especificaciones Alloy, basado en análisis de satisfactibilidad booleana (SAT). El problema de satisfactibilidad de una fórmula en lógica proposicional consiste en encontrar una asignación de valores de verdad para las variables, de tal forma que la misma se evalúe como verdadera.

En Alloy los tipos de datos están definidos mediante *Signaturas*. Alloy permite definir restricciones, predicados y aserciones. Mediante la palabra reservada *fact*, es posible establecer propiedades que se asumen válidas en todos los modelos. Por otro lado también es posible definir predicados y aserciones. En el primer caso para verificar la existencia de instancias de modelos que cumplan con una especificación, y en el segundo caso para verificar que una propiedad se cumple para todo modelo generado dentro de los scopes definidos para cada signatura. En la figura 1(a) se presenta, a modo de ejemplo, una especificación Alloy para listas simplemente encadenadas.

2.3. DynAlloy

Alloy es un lenguaje conveniente, simple y expresivo para construir modelos de software *estáticos*. Para el caso del modelado de sistemas *dinámicos* que capturan la ejecución de sistemas mediante, cambios de estado, Alloy no posee una forma sencilla para representar este comportamiento. DynAlloy [9,10] es una extensión de Alloy que incorpora constructores que permiten capturar fácilmente la noción de cambio de estado. La sintaxis y la semántica de DynAlloy está basada sobre lógica dinámica. DynAlloy extiende Alloy con *acciones básicas*, *programas*, y *asepciones de corrección parcial*. Las acciones básicas son definidas a través de pre y pos condiciones. Por ejemplo, una acción que elimina todos los elementos de una lista simplemente encadenada podría definirse como `removeAll` en la figura 1 (b). Esta acción atómica actualiza el `head` y el `size` de la lista utilizando el operador relacional (`++`) de sobre-escritura. Por un lado la acción `removeAll` posee los campos de `List`, `head` y `size` con parámetros explícitos, en lugar de utilizar `this` se utiliza `thisz` (ya que la primera es una palabra reservada en Alloy). Los programas DynAlloy se construyen utilizando acciones de asignación (`:=`), `skip`, tests (acción skip con guardas, `[expr]?`) y acciones atómicas como casos base, combinadas mediante la aplicación de composición secuencial (`;`), elección no determinística (`+`) e iteración (`*`).

Alloy y DynAlloy son lo suficientemente expresivos como para modelar abstracciones de programas Java [11] y especificaciones JML, y se han utilizado como lenguajes intermedios para diversos análisis, incluida la verificación acotada y la generación de casos de tests de programas Java con anotaciones JML [12].

3. Cómputo de *state-bound workarounds*

En [6] se propuso una técnica de cómputo automática de *state-bound workarounds*. Este procedimiento de búsqueda utiliza el estado corriente del programa como información para encontrar una solución alternativa ante una eventual falla. Esta técnica es aplicable a programas Java anotados con especificaciones JML (Java Modeling Language) [7]. El procedimiento de búsqueda de *state-bound workarounds* se realiza a través de un programa DynAlloy generado a partir de las acciones atómicas DynAlloy mapeadas de las especificaciones JML de los métodos del módulo. Dada una clase C , y m_1, m_2, \dots, m_k métodos públicos en C , donde cada método m_i posee su contrato JML, pre_{m_i} y $post_{m_i}$, las acciones DynAlloy a_1, a_2, \dots, a_k , correspondientes a cada uno de los métodos m_1, m_2, \dots, m_k son utilizadas para generar el programa de búsqueda DynAlloy. Cada acción atómica a_i es definida como:

$$\text{act } a_i \{ \text{pre } \{ pre_{m_i}^A \} \text{ post } \{ post_{m_i}^A \} \}.$$

El programa de reparación modela todas las posibles composiciones secuenciales de las acciones a_1, a_2, \dots, a_k . El operador `*`, es interpretado como un bucle de cero o más iteraciones [10], por lo que al momento de ejecutar DynAlloy Analyzer será necesario fijar este valor de cota superior de iteraciones o *loops*

<pre> one sig Null { } sig Node { elem: Int, next: Node+Null } sig List { head: Node+Null, size: Int } fact acyclicLists { all l: List all n: Node n in l.head.*next => not (n in n.^next) } pred getFirst[l: List, result': Int] { l.head != Null and result' = l.head.elem } assert getFirstEqGetLast { all l: List all n1, n2: Int l.size = 1 and getFirst[l, n1] and getLast[l, n2] => n1 = n2 } run getFirst for 5 but 1 List, 5 Int check getFirstEqGetLast for 5 but 1 List, 5 Int </pre>	<pre> act removeAll[thiz: List, head: List -> one (Node+Null), size: List -> one Int] { pre { } post { head' = head ++ (thiz -> Null) and size' = size ++ (thiz -> 0) } } program choose[l: List, result: Int] { local [chosen: Boolean, curr: Node+Null] chosen := false; curr := l.head; ([curr!=Null]?; (result:=curr.elem; chosen:=true)+(skip)); curr:=curr.next)*; [chosen = true]? } assertCorrectness chooseIsCorrect[l: List, result: Int] { pre { l.size>0 and repOK[l] } program = choose[l, result] post { some e: l.head.*next.elem e=result' } } run choose for 5 but 1 List, 5 Int, 5 lurs check chooseIsCorrect for 5 but 1 List, 5 Int, 5 lurs </pre>
--	--

(a)

(b)

Figura 1. Especificaciones Alloy and DynAlloy para listas enlazadas.

unrolls. Este valor determinará el alcance o *scope* máximo de composiciones de acciones, es decir, la cantidad de acciones que podrá contener el *workaround*. Bajo el supuesto de que un método m_i de la clase produce una falla en tiempo de ejecución en un estado concreto s_i (modelado con un predicado Alloy s_i^A) se contruye la siguiente aserción de corrección parcial:

$$\{ s_i^A \} (a_1 + a_2 + \dots + a_{i-1} + a_{i+1} + \dots + a_k) * \{ \neg post_{m_i}^A \}$$

La aserción será ser verificada automáticamente utilizando DynAlloy Analyzer. Un contra-ejemplo encontrado que no satisfaga la aserción anterior estará compuesta por una secuencia de estados s_{A_0}, \dots, s_{A_j} tal que:

- s_{A_0} es un estado s_i^A ;
- existe una secuencia $a_{p(1)}; a_{p(2)}; \dots; a_{p(j)}$ de operaciones tales que $\langle s_{A_i}, s_{A_{i+1}} \rangle$ están relacionados por la relación de transición $a_{p(i)}$; y
- s_{A_j} es un estado s_f^A que *no* satisface $\neg post_{m_i}^A$, es decir, que satisface $post_{m_i}^A$.

Dado que s_i^A y $post_{m_i}^A$ son predicados Alloy que caracterizan los estados s_i y la pos condición del método m_i , respectivamente, cada contra-ejemplo será en efecto un *workaround* formado por una secuencia de acciones, vinculadas a los métodos de la clase C , que producirán una transición de estados del programa desde s_i con el propósito de arribar a un estado que satisfaga $post_{m_i}$. En el caso de que DynAlloy Analyzer no encuentre un contra-ejemplo a la aserción anterior, dentro del *scope* previsto, no será garantía de que no existe un *workaround* en scopes superiores. El procedimiento de búsqueda se realiza de manera incremental, para poder determinar la alternativa de menor tamaño (en cantidad e acciones involucradas). A continuación, y a modo de ejemplo, se muestran algunos *state-bound workarounds* encontrados para listas simplemente encadenadas. En este caso se supone que un programa Java, al ejecutar una línea *Integer v = lista.set(0,4)*, la cual debería almacenar en la variable v el valor contenido en la posición 0 de la lista y luego almacenar en esa posición 4, produce un incumplimiento en su contrato. En el cuadro siguiente se muestran algunos de los *state-bound workarounds* encontrados en función del número de iteraciones(*).

Invocación a reparar: Integer v = lista.set(0,4)	
Estado inicial: {lista = [10,5,20]}	
Estado final: {v=10 , lista= [4,5,20]}	
Nro. de Iteraciones (*)	state-bound workarounds
1	<i>No existen workarounds de longitud 1.</i>
2	<i>Integer v= lista.remove(0) ; lista.addFirst(4);</i>
	<i>Integer v= lista.remove(10) ; lista.addFirst(4);</i>
	<i>lista.addFirst(4) ; Integer v = remove(1);</i>
	...
3	...
4	...
5	<i>lista.clear() ; lista.addFirst(20); lista.addFirst(5) ;</i> <i>lista.addFirst(4) ; Integer v= lista.get(0)</i>

Cuadro 1. State-bound workaronds para un escenario de falla sobre listas

4. Búsqueda de *workaround templates* y *state-free workarounds*

La reparación de fallas utilizando *state-bound workaronds* posee un condimento adicional de complejidad vinculado a los parámetros de invocación. Luego de analizar los *state-bound workaronds* obtenidos en la evaluación experimental de la técnica presentada en la sección 3, se pudieron identificar *workarounds templates*. Éstos establecen soluciones generalizables fijando la secuencia de acciones de invocación y dejando “libres” los parámetros a ser completados por un

SAT-Solver. Se propone a continuación un procedimiento que permite identificar a un *workaround templates* como un *state-bound workarounds*, es decir que pueda ser utilizable para cualquier estado. Esto también se realiza aplicando técnicas de SAT-Solving. El procedimiento está dividido en las siguientes etapas:

- Etapa 1: Se aplica el algoritmo de cómputo de *state-bound workarounds* para el método m obteniendo como resultado un *state-bound workaround wkt*.
- Etapa 2: Se evalúa si *wkt* puede ser generalizable a un *workaround template*. Para ello se fija la secuencia de las acciones contenidas en *wkt* y se “liberan” los parámetros de los métodos involucrados. El SAT-Solver (Alloy Analyzer), buscará los valores apropiados para alcanzar la pos condición del método m .
- Etapa 3: Para verificar si *wkt* es un *workaround template* válido, se generan una serie de escenarios aleatorios en donde se asume una falla en m y se intenta repararlo utilizando el template derivado de *wkt*.
- Etapa 4: Como se mencionó anteriormente, un *workaround templates* es un *state-bound workaround* cuya “confianza” es brindada por la prueba del template bajo escenarios generados aleatoriamente. A pesar de que este mecanismo brinda confianza acerca de las posibilidades de reparación del *workaround templates*, no asegura una reparación efectiva para todos los casos. Es decir, no es un *state-free workaround*. Para promover a este *workaround templates* a la categoría de *state-free workaround* se utiliza Alloy Analyzer, en este caso. El SAT-Solver intentará generar un estado (dentro de los alcanzables) que satisfaga la precondición del método asociado con el template y que al ejecutar el mismo (dando la posibilidad de que el SAT-Solver defina los parámetros actuales) no satisfaga la pos condición del método. Es decir, buscará generar un estado inicial para el cual ejecutando el template no se repare la falla. En caso de encontrarse el contra-ejemplo el *workaround templates* no podrá ser promovido a *state-free workaround* dentro de los scopes definidos. En caso de que no se encuentre un contra-ejemplo, el *workaround templates* se incluirá en una lista de *state-free workaround* los cuales serán invocados en un primer momento a la hora de reparar una falla.

El cómputo de *workaround templates* permite encontrar soluciones alternativas que mejoran el rendimiento de la técnica de búsqueda de *state-bound workaround*. Este cómputo es incorporado a la técnica presentada en [6]. Ante un escenario de falla, el algoritmo de búsqueda de reparaciones buscará inicialmente en la lista de los *state-free workaround* y *workaround templates* para intentar reparar el problema. En la sección 5 podemos ver algunos resultados obtenidos al utilizar el procedimiento de búsqueda y la mejora en los tiempos de reparación utilizando *workaround templates* y *state-free workaround*.

5. Evaluación de la propuesta

En el cuadro 2, se muestran algunos de los *workaround templates* identificados para métodos de la clase `java.util.TreeSet`, el valor *True* en la columna *Condición de Aplicabilidad* indica que el *template* es un *state-free workaround*. En el cuadro 3 se presenta la aceleración obtenida por la técnica al incorporar *workaround templates* independientes de estado.

Cuadro 2. Ejemplos de *state-bound workarounds* y *workaround templates* encontrados para `java.util.TreeSet`

Método	State-Bound Workaround	Workaround Template	Condición de Aplicabilidad
<code>add(e)</code>	<code>{1,7,10}.add(10)=={1,7,10}.contains(-1)</code>	<code>s.add(X):=s.contains(Y)</code>	X belongs to s
<code>ceiling(e)</code>	<code>{1,7,10}.ceiling(10)=={1,7,10}.floor(10)</code> <code>{1,7,10}.ceiling(8)=={1,7,10}.floor(11)</code> <code>{1,7,10}.ceiling(20)=={1,7,10}.floor(-1)</code>	<code>s.ceiling(X):=s.floor(Y)</code>	True
<code>contains(e)</code>	<code>{2,4}.contains(5)=={2,4}.isEmpty()</code> <code>{2,4}.contains(4)=={2,4}.add(5);{2,4,5}.remove(5)</code>	<code>s.contains(X):=s.isEmpty()</code> <code>s.contains(X)==s.add(Y);s.remove(Z)</code>	s not empty X in s
<code>first()</code>	<code>{2,3,4}.first()=={2,3,4}.floor(2)</code>	<code>s.first():=s.floor(X)</code>	True
<code>floor(e)</code>	<code>{1,7,3,10}.floor(10)=={1,7,3,10}.ceiling(10)</code> <code>{1,7,3,10}.floor(8)=={1,7,3,10}.ceiling(5)</code> <code>{1,7,3,10}.floor(-1)=={1,7,3,10}.ceiling(100)</code>	<code>s.floor(X):=s.ceiling(Y)</code>	True
<code>higher(e)</code>	<code>{2,3,5}.higher(4)=={2,3,5}.floor(5)</code> <code>{2,3,5}.higher(-10)=={2,3,5}.floor(-1)</code>	<code>s.higher(X)==s.floor(Y)</code>	True
<code>last()</code>	<code>{2,3,4}.last()=={2,3,4}.floor(4)</code>	<code>s.last():=s.floor(X)</code>	True
<code>lower(e)</code>	<code>{-1,3,4}.lower(2) == {-1,3,4}.floor(-1)</code> <code>{1,3,24}.lower(-1) == {1,3,24}.floor(-2)</code>	<code>s.lower(X)==s.floor(Y)</code>	True
<code>pollFirst()</code>	<code>{2,4,7}.pollFirst()=={2,4,7}.first();{2,4,7}.remove(2)</code>	<code>s.pollFirst():=s.first();s.remove(X)</code>	True

6. Conclusiones y trabajos futuros

El concepto de *workaround* ha sido utilizado en técnicas de reparación de programas a través de diversos enfoques. En trabajos anteriores hemos propuesto una técnica de cómputo automático de *state-bound workarounds*. Esta técnica

Cuadro 3. Aceleración lograda usando wokarounds templates para `Java.util.TreeSet`

Treesets: 136 estructuras analizadas.; tam. mín.: 11, tam. máx.: 22, Tam. Promedio:13.16			
Método a reparar	Tiempo Promedio de rep.sin templates	Tiempo de rep. con templates	Aceleración
add	0:00:42	0:00: 03	14
ceiling	0:00:43	0:00: 13	3.31
clear	0:00:44	0:00: 02	22
contains	0:00:44	0:00: 02	22
first	0:00:28	0:00: 06	4.67
floor	0:00:43	0:00: 12	3.6
higher	0:00:27	0:00: 13	2.1
is_empty	0:00:22	0:00: 02	11
last	0:00:30	0:00: 06	5
lower	0:00:22	0:00: 07	3.14
poll_first	0:00:44	0:00: 05	8.8
remove	0:00:15	0:00:07	8.5

es aplicable a programas Java anotados con especificaciones JML y utiliza SAT-Solving como mecanismo de búsqueda. La reparación de fallas utilizando *state-bound workarond* posee un condimento adicional de complejidad vinculado a los parámetros de invocación. Se observó que los *state-bound workaronds* pueden ser generalizables liberando los parámetros de invocación. A esos workarounds genéricos los denominamos *workaroud templates*. En este trabajo se propuso un mecanismo de cómputo automático de *workaroud templates* independientes de estado (*state-free workaronds*). Los resultados obtenidos en la evaluación experimental han sido muy positivos. La utilización de *state-free workaroud* permite mejorar notablemente el rendimiento de la técnica de búsqueda de *workarounds*. En el cuadro3 observamos una aceleración promedio de reparación muy significativa, las mismas van de 3.31 a 22 veces más rápida utilizando *workaroud templates* comparado con el enfoque tradicional. Por otro lado, la detección de *workaroud templates* independientes de estado (*state-free workaronds*) módulo intervención del SAT-Solver para fijar los parámetros de invocación, brinda la posibilidad de computar previamente workarounds permanentes, independientemente de los tiempos requeridos. Este proceso de SAT se lleva a cabo en una instancia previa, y pueden ser almacenados en un repositorio para su utilización. Como trabajos futuros, se preve continuar con optimizaciones aplicables a la técnicas de análisis de SAT utilizadas. Se espera incrementar los límites de exploración alcanzados hasta el momento.

Referencias

1. A. Carzaniga, A. Gorla and M. Pezzè, *Self-healing by means of automatic workarounds*, in Proceedings of 2008 ICSE Workshop on Software Engineering for Adaptive and Self-Managing Systems SEAMS 2008, Leipzig, Germany, May 12-13, ACM, 2008.
2. A. Carzaniga, A. Gorla, N. Perino and M. Pezzè, *Automatic Workarounds for Web Applications*, in Proceedings of the 18th ACM SIGSOFT International Symposium on Foundations of Software Engineering FSE 2010, Santa Fe (NM), USA, ACM, 2010.

3. A. Carzaniga, A. Gorla, N. Perino and M. Pezzè, *RAW: runtime automatic workarounds*, in Proceedings of the 32nd ACM/IEEE International Conference on Software Engineering ICSE 2010, New York (NY), USA, ACM, 2010.
4. A. Carzaniga, A. Gorla, A. Mattavelli, N. Perino and M. Pezzè, *Automatic recovery from runtime failures*, in Proceedings of the 35th International Conference on Software Engineering ICSE 2013, San Francisco (CA), USA, IEEE/ACM, 2013.
5. A. Carzaniga, A. Gorla, N. Perino and M. Pezzè, *Automatic Workarounds: Exploiting the Intrinsic Redundancy of Web Applications*, ACM Trans. Softw. Eng. Methodol. 24(3), ACM, 2015.
6. Marcelo Uva, Pablo Ponzio, Germán Regis, Nazareno Aguirre, Marcelo F. Frias: *Automated Workarounds from Java Program Specifications Based on SAT Solving*. FASE 2017: 356-373.
7. P. Chalin, J. R. Kiniry, G. T. Leavens and E. Poll, *Beyond Assertions: Advanced Specification and Verification with JML and ESC/Java2*, in Proceedings of 4th International Symposium on Formal Methods for Components and Objects FMCO 2005, LNCS 4111, Springer, 2005.
8. D. Jackson, *Software Abstractions: Logic, Language, and Analysis*, The MIT Press, 2006.
9. J. Galeotti and M. Frias, *DynAlloy as a Formal Method for the Analysis of Java Programs*, in Proceedings of Software Engineering Techniques SET 2006: Design for Quality, Warsaw, Poland, IFIP 227, Springer, 2006.
10. M. Frias, J. Galeotti, C. López Pombo and N. Aguirre, *DynAlloy: Upgrading Alloy with Actions*, in Proceedings of International Conference on Software Engineering ICSE 2005, St. Louis, Missouri, USA, ACM, 2005.
11. B. Liskov and J. Guttag, *Program Development in Java: Abstraction, Specification and Object-Oriented Design*, Addison-Wesley, 2000.
12. J.P. Galeotti, N. Rosner, C. López Pombo and M. Frias, *Analysis of Invariants for Efficient Bounded Verification*, in Proceedings of the Nineteenth International Symposium on Software Testing and Analysis, ISSTA 2010, Trento, Italy, July 12-16, ACM, 2010.

Reuso de Funcionalidades en el Subdominio de Patrimonio Cultural

Sofia Caballero¹, Agustina Buccella¹, and Alejandra Cechich¹

GIISCO Research Group

Departamento de Ingeniería de Sistemas - Facultad de Informática

Universidad Nacional del Comahue

Neuquen, Argentina

{sofia.caballero,agustina.buccella,alejandra.cechich}@fi.uncoma.edu.ar

Resumen La búsqueda de métodos, procesos, y/o técnicas que maximicen el reuso son constantes dentro del área de la ingeniería de software. En particular las líneas de productos de software (LPS) tienen como objetivo analizar aspectos comunes para que puedan ser reutilizados dentro de un dominio. En este trabajo proponemos un proceso de desarrollo de una LPS basado en la reutilización de artefactos de software existentes, enfocándonos en el método para el reuso de funcionalidades.

Keywords: Líneas de Productos de Software, Reuso de Dominios, Patrimonio Cultural

1. Introducción

Las tendencias actuales dentro del área de la ingeniería de software se basan en la generación de mejoras que permitan desarrollos de software más rápidos, con mejor calidad y menor costo. Esas tres ventajas principales en general se obtienen mediante la aplicación de técnicas y/o métodos que maximicen el reuso de artefactos de software comenzando desde las fases tempranas de cualquier desarrollo. En particular las líneas de productos de software (LPS) se enfocan en encontrar similitudes dentro de un dominio de aplicación de manera de reusarlas cada vez que un producto es derivado dentro del mismo. Así, un aspecto principal para una LPS es definir primero el dominio donde será aplicada.

En este sentido hemos trabajado primeramente dentro del dominio geográfico buscando similitudes entre los productos posibles a desarrollar y lo hemos especializado en subdominios (de ecología marina [5] y paleontología [6]) para minimizar la variabilidad inherente. A su vez hemos definido una metodología de desarrollo [4,5], la cual se basa en la construcción de artefactos de software como recursos necesarios que pueden ser reutilizados.

En este contexto, nuestro trabajo está relacionado con aquellas propuestas en la literatura que proponen reuso de artefactos de software en alguna de las etapas de desarrollo. Por ejemplo en [13] se realiza una revisión sobre técnicas basadas en grafos, en razonamiento de casos, en ontologías etc. Entre aquellas basadas en grafos, las cuales se asemejan en cierto sentido a lo propuesto en este

artículo, podemos citar a [12] donde se aplica un algoritmo de correspondencia gráfica para encontrar similitud entre los diagramas de secuencia de UML y en [14], donde se calcula la similitud estructural de diagramas de clases utilizando una técnica de correspondencia de gráficos inexacta. También en [11] se describe un modelo de estimación de esfuerzo basado en la reutilización de casos de uso, y en [1] los autores se centran en la correspondencia de diagramas UML desde múltiples vistas.

En este trabajo se presenta un método que usa algunas de las técnicas presentes en literatura previa para aplicarlas al desarrollo de LPSs dentro del dominio geográfico. El proceso en general tiene como objetivo crear, reutilizar y extender artefactos de software a medida que se desarrollan LPSs en nuevos subdominios.

El artículo se organiza de la siguiente manera. En la Sección 2 se presentan los trabajos previos en donde se describen los artefactos de software y dominios trabajados y se describe el proceso basado en reuso de dominios. En la Sección 3 se presenta el método para el reuso de funcionalidades el cual extiende y aplica dichos artefactos con un ejemplo de uso. Finalmente se describen las conclusiones y trabajos futuros.

2. Contexto y Trabajos Previos

En trabajos previos [4,5,6] hemos propuesto una metodología de desarrollo de LPS que se basa en la creación y extensión de artefactos de software a partir de una taxonomía de dominios (Figura 1).

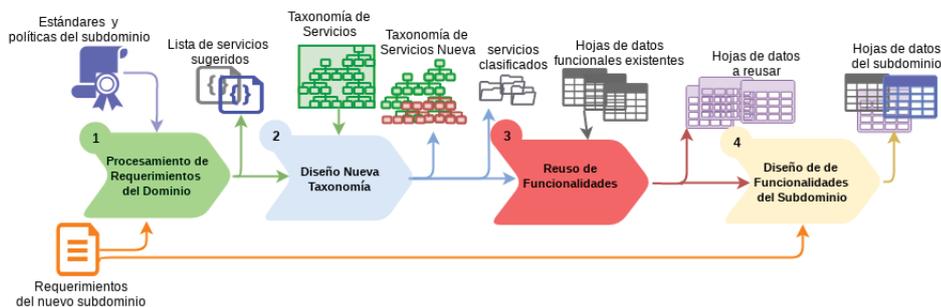


Figura 1: Primeros pasos del proceso de desarrollo basado en reuso de dominios

Durante el (1) *Procesamiento de Requerimientos del Dominio* se deben elicitar y definir todos los requerimientos que posee el subdominio a analizar. Se recibe como entradas los *requerimientos del subdominio*, los cuales provienen de los usuarios expertos y los *estándares* que existen en cuanto a servicios y funcionalidades comunes. Luego, la salida de este paso es la *lista de servicios requeridos* en el subdominio.

En el (2) *Diseño Nueva Taxonomía* se ingresan la *taxonomía de servicios* y la lista de servicios requeridos obtenidos del paso anterior. Así, se debe ingresar

a la taxonomía existente los servicios nuevos definidos, es decir, se deben ubicar en el lugar correcto de la jerarquía de dominios. Para esto hemos creado una herramienta llamada *Mapeo de Servicios* [2,3] que asiste al ingeniero de software en determinar si el servicio ya pertenece a otro subdominio dentro de la taxonomía, o si debe ser agregado, junto con lugares optativos donde ingresarlo.

En particular, hemos trabajado en el análisis del dominio geográfico y en los subdominios de ecología marina [5] y paleontología [6]. A partir de los estándares geográficos definidos por el OGC (Open Geospatial Consortium)¹ y el comité ISO TC211², hemos construido los servicios base para clasificar aquellos servicios específicos necesitados en cada subdominio. Dentro de estos estándares hemos utilizado el estándar ISO 19119³ para definir la clasificación inicial de servicios para luego dentro de cada subdominio utilizar los estándares específicos como la norma ISO 21127⁴ o el modelo de referencia CIDOC-CRM⁵ para el subdominio paleontológico.

Luego, para modelar las funcionalidades de las LPSs, con sus partes comunes y variables, utilizamos una estructura denominada *hojas de datos funcionales*. Las mismas se componen de cinco ítems: (1) una identificación, (2) una descripción textual de la función principal, (3) el dominio (o subdominio) en el que se incluye o ha sido creada, (4) una lista de servicios (de la taxonomía) utilizada para representar la funcionalidad, y (5) un conjunto de modelos de variabilidad que pueden ser cualquier tipo de artefacto gráfico (uno o más), siempre y cuando muestren las interacciones entre los servicios listados. En general el modelo de variabilidad que hemos utilizado se basa en una notación gráfica similar a los diagramas de colaboración (de UML) y la notación de variabilidad definida en OVM (Orthogonal Variability Model) [10].

Durante el (3) *Reuso de Funcionalidades*, analizamos las funcionalidades previamente diseñadas en las LPSs para los subdominios existentes con respecto a las funcionalidades requeridas para el nuevo subdominio. En el trabajo presentado en [6] hemos desarrollado un análisis de reusabilidad basado en la función de distancia entre grafos presentada en [8,7]. En este artículo, extenderemos dicho análisis de forma que permita encontrar servicios y funcionalidades existentes en subdominios previos, para el caso de la construcción de uno nuevo dentro de una misma jerarquía de dominios.

Finalmente, el (4) *Diseño de funcionalidades del Subdominio* se refiere al diseño de las funcionalidades nuevas no existentes en las LPSs previas. Dichas funcionalidades deben ser diseñadas de acuerdo a la metodología de LPS presentada en [4,6,9].

¹ <http://www.opengeospatial.org/>

² <http://www.isotc211.org/>

³ Services International Standard 19119, ISO/IEC, 2005.

⁴ ISO 21127:2014. Information and documentation. A reference ontology for the interchange of cultural heritage information

⁵ CIDOC Conceptual Reference Model Version 6.0 – <http://www.cidoc-crm.org>

3. Método para el Reuso de Funcionalidades

Para mostrar cómo trabaja el método (parte del tercer paso en la Figura 1) y los casos creados para el análisis de reuso, vamos a aplicarlo a un nuevo subdominio – *patrimonio cultural paleontológico*. Este subdominio gestiona aspectos específicos sobre cómo registrar las piezas paleontológicas encontradas, cómo solicitar permisos de exploración y excavación, intervalos de tiempo en los que deben realizarse los procedimientos administrativos y legales y sanciones en caso de incumplimiento. Al mismo tiempo, tiene varios aspectos en común con los subdominios previamente desarrollados (biología marina y paleontología) e incluso con el geográfico.

El método de reuso toma como entrada la nueva taxonomía, la cual es la salida del segundo paso del proceso presentado previamente (Figura 1) y realiza una serie de comparaciones que permiten obtener un *costo de reuso*.

En la Figura 2 podemos observar una parte de la nueva taxonomía conteniendo servicios pertenecientes a los diferentes subdominios. Los servicios en color negro son derivados de la norma ISO 19119⁶. Luego, los de color rojo son del subdominio de ecología marina, los azules del subdominio paleontológico, y los de patrimonio cultural se muestran en verde. Estos últimos son justamente los adicionados en este paso 2.

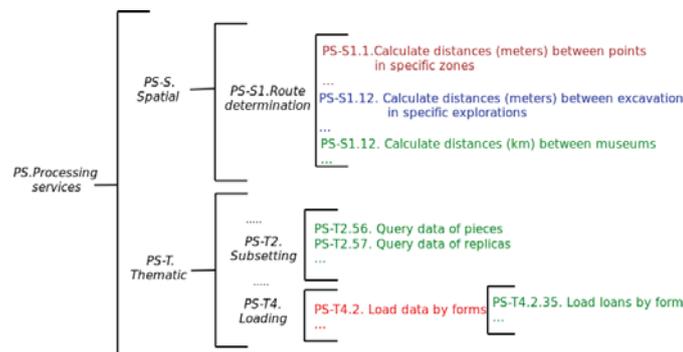


Figura 2: Parte de los servicios de procesamiento de la taxonomía incluyendo los de patrimonio cultural paleontológico

Luego, con esta nueva taxonomía, aplicamos el método de reuso de funcionalidades que recibe como entrada una hoja de datos funcional del subdominio nuevo y analiza específicamente los ítems de la *lista de servicios* utilizados y el *modelo de variabilidad*. Por una cuestión de simplicidad, llamaremos a ambos ítems *datasheet* (que es el nombre en inglés utilizado en trabajos previos).

⁶ ISO 19119:2016. Geographic information-Services <https://www.iso.org/standard/59221.html>

3.1. El método paso a paso

En el siguiente algoritmo podemos ver los pasos del método:

```

1  Por cada datasheet del subdominio nuevo:
2    Traducir el datasheet a un grafo (Dn)
3    Por cada datasheet existente en otros subdominios:
4      Traducir el datasheet existente a un grafo (De)
5      Por cada nodo de Dn (Dn.nodo):
6        Por cada nodo de De (De.nodo):
7          Aplicar clasificación de servicios y de variabilidad (Eq 2)
8          Mapear De.nodo <=> Dn.nodo con mejor costo de sustitucion
9          Calcular costos para eliminacion
10     Mostrar las 3 De's con mejores costos

```

Traducción del datasheet a un grafo (Dn). La instrucción 2 se realiza en base al modelo de variabilidad. Como describimos en la sección anterior, nuestro modelo de variabilidad es similar a un diagrama de colaboraciones de UML y la notación OVM para anotar la variabilidad. En la Figura 3a) podemos observar parcialmente el modelo de variabilidad creado dentro de la hoja de datos funcional de *Agregar una nueva pieza* del subdominio de patrimonio cultural. Como vemos, el diagrama modela el hecho de que una pieza paleontológica puede ser cargada en el sistema de un museo ya sea porque es recibida de un préstamo, una donación, un intercambio, o como resultado de una excavación. Tanto el préstamo como la donación son servicios que siempre van a estar considerados (obligatorios), en cambio las excavaciones o intercambios son optativos ya que los museos no necesariamente poseen áreas de exploración o permiten intercambio de piezas. A su vez, en la Figura 3b) mostramos el grafo generado correspondiente, el cual posee la misma información pero traducida a dicha estructura.

Traducción de las datasheet existentes a un grafo (De). Luego, a partir de la instrucción 3 el algoritmo inicia el cálculo del costo de transformar el grafo traducido del subdominio nuevo Dn a los grafos traducidos de cada datasheet existente (instrucción 4).

Cálculo de costos. Para el cálculo usamos la función general definida en la Ecuación 1:

$$\text{costo}_{total}(Dn, De) = \frac{\min}{e_1, \dots, e_k \in P(Dn, De)} \sum_{i=1}^k c(e_i) \quad (1)$$

donde $P(Dn, De)$ se refiere al conjunto de pasos de edición requeridos para transformar Dn en De , y $c(e) \geq 0$ es el costo de cada operación de edición e .

Para calcular este costo separamos las operaciones de edición entre *sustitución* y *eliminación*. La inserción de nodos no es considerada ya que no es relevante considerar el costo de agregar un nuevo nodo para llegar a construir un De . Buscamos reusar lo posible para construir Dn , por lo que debemos buscar lo que podemos reusar para lograrlo.

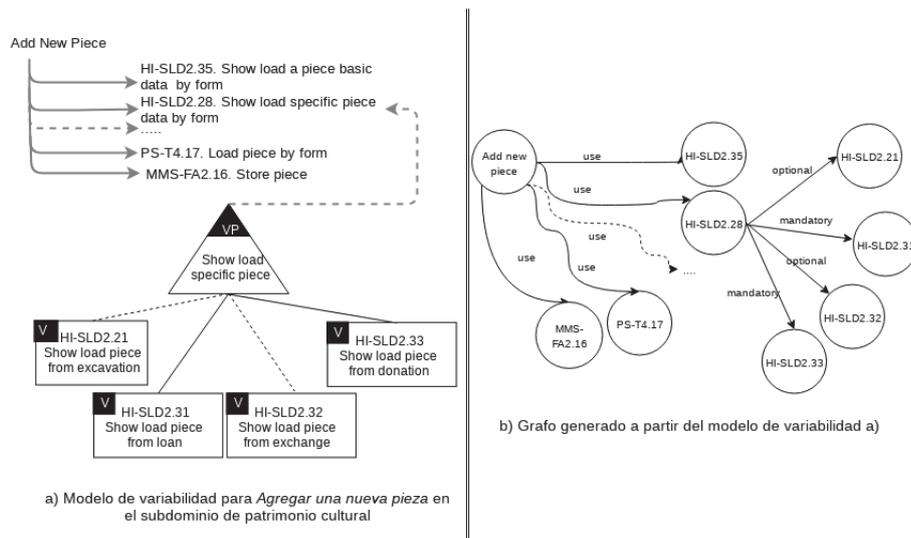


Figura 3: Modelo de variabilidad y grafo para la funcionalidad *Agregar una nueva pieza*

Cálculo del costo de sustitución. Para calcular el costo de sustitución (instrucciones 5 a 8) buscamos nodos que sean similares y les asignamos el costo de sustituir uno por el otro. Así, tomamos un nodo del grafo nuevo (*Dn.nodo*) y cada nodo de cada grafo existente (*De.nodo*) por vez y los comparamos uno a uno. La comparación nodo por nodo nos permite mapear cuál se corresponde con cual y el costo asociado. Este costo de sustitución se asigna según la siguiente ecuación (Ecuación 2):

$$costo_{sust}(n1, n2) = \frac{cl_{ser}(n1, n2) + cl_{var}(n1, n2)}{2} \tag{2}$$

donde *n1* es *Dn.nodo* y *n2* es *De.nodo*. La ecuación devuelve el promedio del costo de analizar el resultado de la *clasificación de servicios* (*cl_{ser}*) de los nodos analizados según la nueva taxonomía de servicios y la *clasificación de variabilidad* (*cl_{var}*) según como están interactuando los nodos dentro de los grafos. Cada una de estas funciones devuelve un resultado de 0 a 1.

Para la *clasificación de servicios* entre un nodo *Dn.nodo* y un nodo *De.nodo* definimos los siguientes casos:

- *Dn.nodo* es un servicio completamente reusable ya que es exactamente igual a *De.nodo*. Esto significa que ya existía en la taxonomía de servicios dentro de algún subdominio previo, de esta manera su reuso es completo y se agrega el mapeo $Dn.nodo \leq SCR \Rightarrow De.nodo$ provisorio y costo 0.
- *Dn.nodo* es un servicio reusable por especialización ya que aparece como especialización de *De.nodo*. Así, cierto reuso podría ser aprovechado, y se agrega el mapeo $Dn.nodo \leq SRE \Rightarrow De.nodo$ provisorio y costo 0,5.

- En cualquier otro caso el costo es 1 ya que se asume que ningún reuso puede ser aplicado.

Ambos casos son representativos ya que consideran hipotéticamente el esfuerzo de implementar un servicio. En el primer caso, el costo 0 significa que el servicio ya existe y por lo tanto no requiere una reimplementación o adaptación. En cambio, en el segundo caso, el costo de 0,5 asume un esfuerzo de reimplementar un servicio que ya está similarmente implementado por otro servicio que lo contiene. Este caso sucede por ejemplo si debemos comparar el servicio PS-T4.2 (del subdominio de ecología marina) con el servicio PS-T2.35 (de patrimonio cultural) de la Figura 2.

Para la *clasificación de variabilidad* (cl_{var}) utilizamos las dependencias de variabilidad que involucran a cada nodo. Las dependencias de variabilidad analizadas son: obligatorio, opcional, alternativo, variante y uso. Por ejemplo, en la Figura 3 el nodo HI-SLD2.21 es opcional, el HI-SLD2.33 es obligatorio y el PS-T4.17 es de uso. Así a cada nodo se clasifica según estas dependencias y luego se los compara y asigna un costo según se muestra en la Tabla 1.

<i>Dn.nodo.variabilidad</i>	<i>De.nodo.variabilidad</i>	Costo
uso	obligatorio	0,2
uso	opcional	0,7
	variante	
	alternativo	
obligatorio	opcional	0,7
	variante	
	alternativo	
opcional	variante	0,3
	alternativo	
opcional	uso	0,2
	mandatorio	

Cuadro 1: Costo según dependencias de variabilidad

Nuevamente estos costos son hipotéticos asumiendo: (1) un costo bajo cuando las dependencias son de uso y obligatorias en *De.nodo.variabilidad* ya que van a estar implementadas en las LPS previas; y (2) un costo mayor cuando las dependencias son obligatoria o uso en *Dn.nodo.variabilidad* y variante, alternativa u opcional en las *De.nodo.variabilidad*, ya que asumimos que posiblemente este servicio puede no estar implementado ya que ningún producto ha sido instanciado con el mismo. Luego, y como podemos observar en la instrucción 7, de todas las comparaciones de un nodo de *Dn.nodo* aplicadas a cada nodo de *De.nodo* nos quedamos con aquella que haya devuelto el menor costo y generamos un mapeo final de $Dn.nodo \Leftrightarrow De.nodo$.

Para comprender mejor las posibilidades, en la Tabla 2 mostramos algunos ejemplos de comparaciones de nodos para el caso de querer sustituir el *Dn.nodo* con *De.nodo*. Por ejemplo, la primera fila muestra el caso de un nodo con una dependencia de uso. Este nodo se mapea con otro nodo con la misma dependencia y al mismo tiempo ambos nodos representan el mismo servicio en la taxonomía.

De esta forma el costo de sustituir es nulo. En cambio en la segunda fila, el nodo A se encuentra como un servicio especializado del nodo B en la taxonomía, por lo que el costo es de 0,5. Luego como ambos nodos participan por igual en la dependencia de variabilidad (el costo es 0) generando un costo final de 0,25.

<i>Dn.nodo</i>	<i>De.nodo</i>	<i>cla_ser</i>	<i>cla_var</i>	<i>costo_sust</i>
use → (Dn.nodoA)	use → (De.nodoB)	SCR	uso = uso	$\frac{0+0}{2} = 0$
use → (Dn.nodoA)	use → (De.nodoB)	SRE	uso = uso	$\frac{0,5+0}{2} = 0,25$
use → (Dn.nodoA)	optional → (De.nodoB)	SCR	uso = opcional	$\frac{0+0,7}{2} = 0,35$

Cuadro 2: Casos de aplicación de la Ecuación 2

Cálculo de los costos de eliminación. Luego faltaría determinar los costos de eliminación para aquellos nodos que no fueron mapeados (instrucción 9). Los nodos no mapeados son aquellos que devuelven 1 en la función $cla_{ser}(Dn.nodo, De.nodo)$ ya que significa que el *Dn.nodo* está ubicado como un servicio nuevo en la taxonomía sin ser parte de ningún subdominio. En este caso, el costo de eliminar es de 0,2 ya que asumimos que se requiere el esfuerzo de analizar lo que se debe borrar o ignorar del servicio ya implementado existente.

Obtención de mejores costos. Por último se calcula la función final de costo (Ecuación 1) y se listan los *De* que hayan resultado en los primeros mejores costos obtenidos (instrucción 10).

Aplicando el método en nuestro ejemplo. Como un ejemplo completo de comparación de dos grafos, en la Figura 4b) mostramos parcialmente el grafo correspondiente al subdominio paleontológico para *Agregar una nueva pieza*. En ese subdominio, el agregado de una pieza es sólo considerado cuando se realiza una excavación [6], por lo que podemos observar que en el subdominio de patrimonio este caso es considerado como una de 4 opciones (Figura 4a). El resultado parcial de la comparación de haber aplicado el método completo lo podemos ver en la Tabla 3 en donde se mapean 4 nodos (primeras 4 filas) en forma completa (cada uno con diferentes dependencias de variabilidad) y se deben eliminar otros 4 nodos (segundas 4 filas) que no pertenecen al grafo de paleontología. Con todos estos resultados parciales, la Ecuación 1 retorna un $costo_{total}$ de 0,9.

Aunque este valor final no tenga asociada una función de costo que mida algún atributo como esfuerzo, adaptabilidad, etc., para nosotros es comparativo en cuanto al resultado que genera cada comparación del grafo nuevo con cada

uno de los existentes. Es decir, el resultado de 0,9 será útil si consideramos el resultado de haberlo comparado con los demás datasheets existentes. En base a estos resultados el método puede mostrar en orden descendente (estilo ranking) los datasheets con mejores costos totales de forma que el ingeniero de software pueda determinar qué reuso efectivo le dará a la nueva funcionalidad.

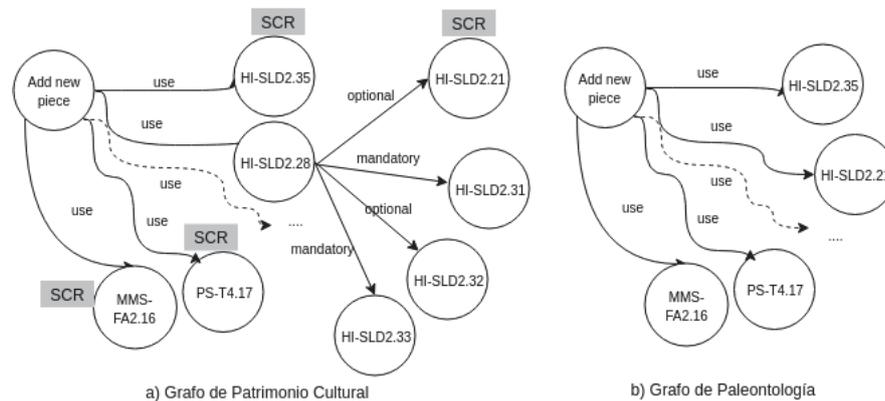


Figura 4: Grafos de los subdominios paleontológico y patrimonio cultural para la funcionalidad *Agregar una nueva pieza*

<i>Dn.nodo</i>	<i>De.nodo</i>	<i>cla_{scr}</i>	<i>cla_{var}</i>	<i>costo_{sust}</i>	<i>costo_{ins/elim}</i>
HI-SLD2.35	HI-SLD2.35	SCR	<i>uso = uso</i>	0	-
HI-SLD2.21	HI-SLD2.21	SCR	<i>opcional = uso</i>	0,1	-
PS-T4.17	PS-T4.17	SCR	<i>uso = uso</i>	0	-
MMS-FA2.16	MMS-FA2.16	SCR	<i>uso = uso</i>	0	-
HI-SLD2.28	-	-	-	-	0,2
HI-SLD2.31	-	-	-	-	0,2
HI-SLD2.32	-	-	-	-	0,2
HI-SLD2.33	-	-	-	-	0,2

Cuadro 3: Resultados parciales de comparar los grafos de la Figura 4a) y b)

4. Conclusiones y Trabajo Futuro

En este trabajo presentamos un proceso de desarrollo de LPS basado en el reuso de artefactos previos creados dentro de un jerarquía de dominios geográficos. Este proceso utiliza estos artefactos para extenderlos durante la creación de una nueva LPS en el subdominio de patrimonio cultural paleontológico. En particular hemos presentado nuestro método de reuso de funcionalidades el cual permite encontrar funcionalidades existentes similares a las previamente diseñadas

para maximizar el reuso efectivo de la nueva LPS. Como este es un trabajo preliminar, debemos todavía analizar el resto de los casos de similitud que pueden surgir incluyendo similitudes por servicios cruzados (servicios que no están en el datasheet que se analiza pero sí en la taxonomía) y luego validar los resultados. También estamos desarrollando una herramienta de soporte que permita agilizar los cálculos e iniciar evaluaciones en casos de estudio.

Referencias

1. Adamu, A., Zainon, W.M.N.W.: Multiview similarity assessment technique of uml diagrams. *Procedia Computer Science* 124, 311 – 318 (2017), 4th Information Systems International Conference (ISICO)
2. Arias, M., Buccella, A., Cechich, A.: Búsqueda de funcionalidades basada en expansión de consultas para lps. In: *Proceedings of the CACIC'16: XXII Congreso Argentino de Ciencias de la Computación*. San Luis, Argentina (2016)
3. Arias, M., DeRenzis, A., Buccella, A., Flores, A., Cechich, A.: Classification-based mining of reusable components on software product lines. *IEEE Latin America Transactions* 14(2), 870–876 (Feb 2016)
4. Buccella, A., Cechich, A., Arias, M., Pol'la, M., Doldan, S., Morsan, E.: Towards systematic software reuse of gis: Insights from a case study. *Computers & Geosciences* 54(0), 9 – 20 (2013)
5. Buccella, A., Cechich, A., Pol'la, M., Arias, M., Doldan, S., Morsan, E.: Marine ecology service reuse through taxonomy-oriented SPL development. *Computers & Geosciences* 73(0), 108 – 121 (2014)
6. Buccella, A., Cechich, A., Porfiri, J., Diniz Dos Santos, D.: Taxonomy-oriented domain analysis of gis: A case study for paleontological software systems. *ISPRS International Journal of Geo-Information* 8(6) (2019), <https://www.mdpi.com/2220-9964/8/6/270>
7. Bunke, H., Shearer, K.: A graph distance metric based on the maximal common subgraph. *Pattern Recognition Letters* 19(3), 255 – 259 (1998)
8. Dijkman, R., Dumas, M., van Dongen, B., Käärik, R., Mendling, J.: Similarity of business process models: Metrics and evaluation. *Inf. Syst.* 36(2), 498–516 (Apr 2011), <http://dx.doi.org/10.1016/j.is.2010.09.006>
9. Pesce, F., Caballero, S., Buccella, A., Cechich, A.: Reusing a geographic software product line platform: A case study in the paleontological sub-domain. In: De Giusti, A.E. (ed.) *Computer Science – CACIC 2017*. pp. 145–154. Springer International Publishing, Cham (2018)
10. Pohl, K., Böckle, G., Linden, F.J.v.d.: *Software Product Line Engineering: Foundations, Principles and Techniques*. Springer-Verlag New York, Inc., Secaucus, NJ, USA (2005)
11. Rak, K., Car, , Lovrek, I.: Effort estimation model for software development projects based on use case reuse. *Journal of Software: Evolution and Process* 31(2), 2119 (2019)
12. Robinson, W.N., Woo, H.G.: Finding reusable uml sequence diagrams automatically. *IEEE Software* 21(5), 60–67 (Sep 2004)
13. Salami, H., A. Ahmed, M.: Uml artifacts reuse: State of the art. *International Journal of Soft Computing and Software Engineering* 3 (02 2014)
14. Salami, H.O., Ahmed, M.A.: A framework for class diagram retrieval using genetic algorithm. In: *SEKE* (2012)

Risk determination for the implantation process of software systems

Felipe Ortiz¹, Marisa Panizzi^{1,2} Rodolfo Bertone³,

¹ Master in Information Systems Engineering, Graduate School. Universidad Tecnológica Nacional. Regional Buenos Aires. Castro Barros 91. (C1178AAA). CABA. Argentina.

² Information Systems Engineering Department. Universidad Tecnológica Nacional. Facultad Regional Buenos Aires. Medrano 951 (C1179AAQ), C.A.B.A, Argentina.

³ Institute for Informatics Research- III-LIDI. School of Informatics. Universidad Nacional de La Plata (UNLP) 50 y 120 – La Plata, Argentina, ortizfd@gmail.com, marisapanizzi@outlook.com, pbertone@lidi.unlp.edu.ar

Abstract. From the analysis of software companies in Argentina, weaknesses in risk management have been observed. This impacts in quality management because risk planning is a requirement specified by all standards. As part of a general study about the implantation of software systems, the aim of this work is to analyze the risks associated to such process. This proposal envisages the activities and tasks of the ISO/IEC 12207 standard transition process. For the assessment of the proposed risks, the ISO/IEC 31010 standard is adopted. Furthermore, associated procedures are suggested to either avoid or mitigate risks. The work was tested in a real environment to determine its viability. The case study consisted of the risk analysis of the implantation of the management system module of a multinational company's advertising agency. This revealed flaws in the management of the analyzed risks and provided feedback for the study.

Keywords: software process, implantation process of software systems, risk management, case study, improvement.

1 Introduction

The implantation of software systems is the phase of the development life cycle in which the software product is transferred to the client. The implantation process contains practices that tend to pose problems such as the lack of (external) components, incomplete downloads and erroneous implantations [1]. Problems that may occur in the implantation phase are transferred and eventually resolved in the maintenance phase. Some companies usually take months and even years to complete the implantation of a software system in its entirety. This is why an efficient software implantation will considerably save resources in terms of costs and effort. [2].

Software projects are high-risk activities that generate variable performance results [3]. A risk is the probability that a loss will occur. In a software development project, the loss could occur through the decrease in software product quality, the increase in development costs, the delay in completion or a failure, among other losses [4]. A great number of projects lack formal approaches for risk management. The identification of such approaches often depends informally on the skills and the level of expertise of the software administrators [5].

In the context of this research, a conceptual definition of software systems implantation is analyzed due to the varied existing terminology. It is called “Implantation phase and user’s acceptance” by the Métrica v3 [6] methodology, “Transition” by the ISO/IEC 12207 standard [7], “Deployment Phase” by the DSDM method (Dynamic Systems Development Method) [8], and “Deployment Flow” by the RUP (Rational Unified Process) [9].

The aim of this work is to enhance the implantation process of software systems through the comprehensive management of a set of risks (to avoid, to mitigate and/or to transfer).

Section 2 describes the related works, section 3 presents a preliminary proposal of a set of risks for the process under study focused especially on the activities and tasks carried out in such process. Section 4 exhibits a case study aimed at validating the proposal in a real context and finally, section 5 presents the conclusions and future work.

2 Related works

A systematic mapping study (SMS) was performed in accordance with the process proposed by Barbara Ann Kitchenham in [10], available in [11]. As a result of the SMS, it was observed that the most used methodologies, methods and standards dealing with risks management are CMMI [12], PMIBOK [13] and Software Risk Evaluation [14].

For this work, an adequate tridimensional vision of the implantation process is considered [15]: “Process/Product/Person”. The so called “*Process*” dimension includes phases or stages, activities and tasks that compose the process of implantation. The “*Product*” dimension envisages characteristics such as size, complexity, design characteristics, performance and quality level. The “*Person*” dimension includes informatics professionals and system’s users.

Based on the SMS results and in order to find out how CMMI [12], PMIBOK [13] and Software Risk Evaluation [14] approach the implantation phase, a comparative analysis was performed through the DESMET method [16] based on the “Process/Product/Person” characteristics. In addition, MAGERIT [17] was added to the study because it is one of the pioneering methodologies in risk management. The comparative analysis is shown in Table 1.

Table 1. Evaluation of methodologies, methods and standards.

Implantation Phase	Methodologies, methods and standards considered			
	CMMI-DEV	PMIBOK	S.R.E.	MAGERIT
“Process” Dimension	YES	YES	YES	YES
“Product” Dimension	NO	YES	YES	YES
“Person” Dimension	NO	NO	NO	NO

The results shown in Table 1 led to the following conclusions:

- All the methodologies, methods and standards (SOFTWARE RISK EVALUATION, CMMI, PMIBOK and MAGERIT) show compliance in the “*Process*” dimension.
- Except CMMI, SOFTWARE RISK EVALUATION, PMIBOK and MAGERIT show compliance in the “*Product*” dimension.
- Finally, the “*Person*” dimension does not show compliance in any of the methodologies, methods and standards analyzed.

3 Risk Proposal

This work focuses on the “*Process*” dimension. This proposal envisages the activities and tasks of the transition process under the ISO/IEC 12207 standard [7] as it is an internationally recognized standard. The risk classification used is the one proposed in [5] with adaptations to this work and to the evolution of software engineering in recent decades. For this work, the adaptation implied that the risks considered comprise the entire life cycle of the software and the research proposal focuses on the implantation process. For risk assessment (see Table 2, VH = Very High, H = High, M = Medium, L = Low, VL = Very Low), the ISO/IEC 31010 standard proposal [18] is adopted as one of the main risk management references for the software industry internationally.

Table 2. Risk Assessment Scale according to the ISO/IEC 31010 standard [18].

RISK	PROBABILITY					
		VL	L	M	H	VH
IMPACT	VH	H	VH	VH	VH	VH
	H	M	H	H	VH	VH
	M	L	M	M	H	H
	L	VL	L	L	M	M
	VL	VL	VL	VL	L	L

$$\text{Risk} = [\text{Probability} * \text{Impact}]$$

The activities considered in the "transition" process of the ISO/IEC 12207 standard are: A1. Implantation preparation, 2. Implantation execution and 3. Management of the results of the implantation. In Table 3, two tasks are presented for each of the activities that are related to a proposed risk.

Table 3. Risks for each activity

Activity	Tasks	Risks
A1 – Implantation preparation	T1 – Identify restrictions of the technology	R1 Little investment in technology
	T2 – Obtain access to the enabled environments, systems or services	R2 Friction between software management and top executives.
A2 – Implantation execution	T3 – Adapt software elements to restrictions	R3 Cost overruns
	T4 – Register requirement compliance	R4 Low user satisfaction
A3 – Implantation results management	T5 – Register results	R5 Lack of specialization
	T6 – Maintain traceability of elements	R6 Inadequate tools and methods

4 Case study

In this section, a preliminary case study is described, following the guidelines proposed in [19]. The main objective is to examine the feasibility of applying the risks set (Table 3) for the process of implanting software systems in a real environment. According to Robson's classification [20], it falls under the scope of exploratory studies.

To achieve the objective, the following research questions (RQ) are defined:

RQ1: Were the risks adequately managed during the software system implantation process activities?

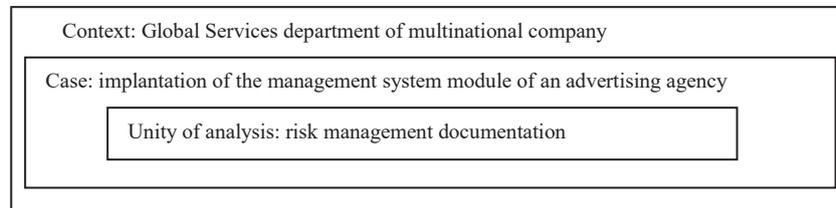
The aim of this question is to obtain information on the risks that were identified in the process execution and the treatment given by the company in order to compare them with the proposal made.

RQ2: How can the implantation process of software systems be improved in this company?

This question attempts to determine the way in which the consultant can enhance its implantation process. For this purpose, it is proposed that a set of risks be identified along with their procedures in order to avoid, mitigate and/or transfer them.

For the case study, data were collected in the Global Services department of a multinational company. This company has been in the Argentine Republic for 20 years and offers professional consulting services. The study focuses on the department mentioned as it carries out the software development projects.

The study is a holistic unique case (Fig. 1) and follows Yin's classification [21].

Fig. 1. Classification of case studies based on Yin`s definition [21].

For the collection of data on the risks of the implantation executed, a third grade technique was used combined with an independent method according to the classification proposed in [22]. To collect information on risks, a data collection template with a codification scheme was defined according to the template approach mentioned in [21]. The template codification scheme consists of a set of 3 groups, each of them corresponding to the 3 activities of the transition process established in the ISO/IEC 12207 standard [7] (1. Implantation preparation, 2. Implantation execution and 3. Implantation results management). Table 6 presents the extract of the coding scheme for data collection. The results presented correspond to the case study of the management system module implantation of an advertising agency. This system was developed to suit the client.

Table 7 shows the traceability of the documents reviewed for each activity of the transition process under the ISO/IEC 12207 standard [7].

Table 6. Extract of the data collection codification scheme [5]

Group	Category	Description
A1 – Implantation preparation	R1: Little investment in technology	Poor investment in technology is endemic in the industry and it affects companies of all sizes.
	R2: Friction between software management and top executives.	Friction between executives is endemic in software projects and it occurs approximately in 30% or 50% of large companies.
	R3: Cost overruns	Cost overruns are endemic in the software field.
A2 – Implantation execution	R4: Low user satisfaction	Low user satisfaction is found in between approximately 25% and 30% of business applications.
A3 – Implantation results management	R5: Lack of specialization	Lack of specialization is endemic in the software industry.
	R6: Inadequate tools and methods.	Lacking the tools or support for them, or eventually failing to define a clear and documented work methodology known by all the personnel.

The results obtained contribute to validate de preliminary proposal of the whole set of risks, which is part of the research process undertaken in the Master`s thesis of the first author of this study.

Table 7. Traceability of reviewed documents.

Documents/ Activities	A1	A2	A3
Work plan	R2		
General Documentation	R1		R6
User requirement		R4	
Change control document		R3	
Project standards	R2		R5
Requirements for the installation environment	R1		
Installation test procedure	R2		R6
Acceptance test procedure		R4	
Smoke test instructions			R6
Data entry instructions			R5
Acceptance test instructions		R4	
Installation script	R1		
Progress report	R2	R3	
Meeting memo	R2		
Smoke test results	R2		
Acceptance test results		R4	
Installation completion report	R2	R4	R5
Training registry		R4	R5

Table 8 presents the assessment of the risks detected as a result of the document analysis.

Table 8: Risk assessment for the case study.

Activity	Risk	Assessment
A1	R1	[Probability (M) * Impact (H)] H
	R2	[Probability (H) * Impact (H)] VH
A2	R3	[Probability (L) * Impact (H)] H
	R4	[Probability (VH) * Impact (H)] VH
A3	R5	[Probability (H) * Impact (M)] H
	R6	[Probability (M) * Impact (L)] M

The results related to the research questions defined for the case study are presented below:

RQ1: Were the risks adequately managed during the software system implantation process activities?

Based on the documentation analyzed, flaws in the management of risks proposed for the implantation process activities were identified:

- Activity 1 – Implantation preparation: The general documentation of the project and the work plan contain generic information, giving rise to misunderstandings and subjectivities. This is one of the reasons that caused very important delays (shown in the meeting memos) and frictions between different sectors within the organization, because the project standards were not met. In addition, the productive environment did not comply with the necessary

requirements by the estimated date set in the work plan, as shown in the progress reports available.

- Activity 2 – Implantation execution: Ambiguous user requirements along with technical problems, multiple complaints and a very low satisfaction level of final users generated cost overruns. An analysis of the progress reports, change control documents and meeting memos revealed that the client considered cancelling the project on several occasions.
- Activity 3 – Implantation results management. Lack of specialization of the different areas involved and of key users hindered an adequate knowledge transfer and project closure. This is observed in the low attendance to training workshops and in incomplete project completion reports.

RQ2: How can the implantation process of software systems be improved in this Company?

Adequate risk management makes it possible to either avoid risks or have procedures available to mitigate them. Table 9 presents the procedures to avoid, mitigate or transfer each of the risks under study based on the proposal by [5]:

Table 9. Procedures associated to each risk.

Risk (R)	Procedure (P)
R1 – Little investment in technology	P1 – Accurate software measurements are the best preventive method for this type of risk. The methodology is based on adequately managing costs, deadlines and other quantitative and qualitative factors associated to the Organization's technology projects.
R2 – Friction between software management and top executives.	P2 – Once friction is generated between top executives and software management, it is not easy to continue the project properly. Some control approaches introduce radical changes, such as externalization of software management and reduction of the number of applications to be implanted.
R3 – Cost overruns	P3 – As the project moves forward, it becomes increasingly difficult to control the associated costs. Cost overruns can stem from many reasons. The best form of mitigation is a detailed follow-up of the project plan. Any excess in time or resources used may cause cost overruns. Specifically, staff working overtime may be a factor causing the risk.
R4 – Low user satisfaction	P4 – User satisfaction is a complex and multifaceted issue. Some of the seemingly effective preventive measures include usability laboratories and having a user experience specialist. In addition, annual or bi-annual user satisfaction surveys are the basic control mechanism to ensure it.
R5 – Lack of specialization	P5 – A preventive method is to create an inventory of the employees' abilities in the Company and to establish specialization criteria and training study plans based on the project.
R6 – Inadequate tools and methods.	P6 – The most effective approach for preventing inadequate software engineering tools is to conduct surveys and generate metrics of the kinds of tools used by the software industry.

The lessons learned from this case are listed below:

- **Method selection:** A preliminary validation of a set of risks was required for the “process” dimension of the software system implantation process and their assessment in a real context. The results obtained enabled the analysis of its evolution and refinement; therefore, the method used is considered to have yielded the expected results.
- **Collected data.** Although the software system implantation process documentation was reviewed in order to analyze how the risks were managed, the case may have been strengthened if the data collected had been complemented with another source or with quantitative data.
- **Codification selected.** The codification scheme selected for the design of the data collection and analysis template was adequate and it allowed the systematic registration of risk information.
- **Results report.** Although the case includes two research questions, the work performed is considered to provide an adequate degree of detail for the purposes of understanding the phenomenon under study.

5 Conclusions and future work

A set of risks related to the software system implantation process was presented, with a focus on the “Process” dimension, since the phases or stages, activities and tasks involved in it are particularly interesting. This proposal considered the activities and tasks of the transition process under the ISO/IEC 12207 standard. The proposal of the ISO 31010 standard was adopted in order to evaluate the risks proposed.

A case study was conducted to determine the viability in a real environment. It consisted of the risk analysis of an implantation of the management system module of an advertisement agency which is part of a multinational company that offers professional counselling services. The document analysis revealed risk management flaws, including lack of specialized staff for the project, conflicting interests among the areas involved and failure to comply with the requirements of the installation environment.

The lessons learned from the case showed that the research method used was appropriate for validating the proposal. A set of recommendations were given to the company as a result of the case in order to enhance its software system implantation process in future projects.

The following future lines of work are identified: (a) to further define risks for the “process” dimension, (b) to define the set of risks for the “product” and “person” dimensions, (c) to validate the proposals in different case studies.

References

1. Jansen, S., Brinkkemper, S. Definition and validation of the key process of release, delivery and deployment for product software vendors: Turning the ugly duckling into a

- swan IEEE International Conference on Software Maintenance, ICSM, art. no. 4021334, pp. 166-175 (2006).
2. Subramanian, N. The software deployment process and automation. *CrossTalk*, 30 (2), pp. 28-34 (2017).
 3. Charette, Robert. Why Software Fails. *Spectrum*, IEEE. 42. 42 - 49. 10.1109/MSPEC.2005.1502528, (2005).
 4. Dhlamini, John & Nhamu, Isaai & Kaihepa, Admire. Intelligent risk management tools for software development. 33-40. 10.1145/1562741.1562745, (2009)
 5. Jones C., Assessment and control of software risk, Yourdon Press, (1994).
 6. PAe, Métrica versión.3. Portal de Administración Electrónica. Gobierno de España (2001).
 7. IEEE ISO/IEC/IEEE 12207:2017(E). Systems and software engineering — Software life cycle processes (2017).
 8. Agile Business Consortium. DSDM. <https://www.agilebusiness.org>. Accessed February 2017.
 9. IBM. Rational Software. Péraire C., Edwards M, Fernandes A., Mancin E. y Carroll K. The IBM Rational Unified Process for Systems (2007).
 10. Kitchenham B., Dyba T., Jorgensen M. Evidence-Based Software Engineering. (págs. 273-281). Washington, DC, USA: International Conference on Software Engineering. (2004).
 11. Felipe Ortiz, Mauricio Davila, Marisa Panizzi y Rodolfo Bertone. State of the art determination of risk management in the implantation process of computing systems. Congreso Internacional sobre Avances en Nuevas Tendencias y Tecnologías (ICAETT 2019). Ecuador, Guayaquil Ecuador, 29 al 31 de mayo (2019).
 12. CMMI Institute, «Capability Maturity Model Integration,» 2010. [En línea]. Available: <https://cmmiinstitute.com/>
 13. Project Management Institute, [En línea]. Available: <https://www.pmi.org/pmbok-guide-standards> (2013)
 14. Software Engineering Institute, «Software Risk Evaluation Method,» (1999). https://resources.sei.cmu.edu/asset_files/TechnicalReport/1999_005_001_16799.pdf
 15. Marisa Panizzi, Mauricio Davila, Agustin Hodes, Pablo Vázquez, Felipe Ortiz, Rodolfo Bertone, Alejandro Hossian. (2019). Aportaciones al proceso de implantación de sistemas informáticos. XXI Workshop de Investigadores en Ciencias de la Computación (WICC 2019). San Juan, 25 y 26 de abril. ISBN: 978-987-3619-27-4.
 16. Kitchenham B., Linkman S., Law D.T. DESMET: A method for evaluating software engineering methods and tools. Keele University (1996)
 17. Portal de administración electrónica, «MAGERIT v.3 : Metodología de Análisis y Gestión de Riesgos de los Sistemas de Información,» 2012. [En línea]. Available: https://administracionelectronica.gob.es/pae_Home/pae_Documentacion/pae_Metodolog/pae_Magerit.html.
 18. International Organization for Standardization, «ISO/IEC 31010:2009,» 2009. [En línea]. Available: <https://www.iso.org/standard/51073.html>
 19. Runeson P, Höst M, Rainer A, Regnell B. Case study research in software engineering: guidelines and examples. Wiley Publishing, Hoboken (2012).
 20. C. Robson. Real world research 2nd edition. Blackwell.(2002)
 21. Yin, R.K.: Case study research: design and methods. 5th Edition. Sage Publications. (2014).
 22. Lethbridge TC, Sim SE, Singer J Studying software engineers: data collection techniques for software field studies. *Empir Softw Eng* 10(3):311–341 (2005).

Un Análisis preliminar sobre reparación de modelos *Alloy* utilizando *Sketching*

César Cornejo^{1,2}, Germán Regis¹, and Nazareno Aguirre^{1,2}

¹ Departamento de Computación, FCEFQyN,
Universidad Nacional de Río Cuarto,

{ccornejo, gregis, naguirre}@dc.exa.unrc.edu.ar

² Consejo Nacional de Investigaciones Científicas y Técnicas (CONICET)

Resumen El tamaño y complejidad de los sistemas de software modernos muestran de manera taxativa la necesidad e importancia de contemplar las etapas tempranas en el desarrollo de software. En particular, una de estas etapas que permite tener una comprensión más abstracta y general del sistema como un todo, es la etapa de *modelado*. Si bien existe una variada gama de lenguajes para tal fin, dos características que consideramos importantes para su elección son su *output* como entrada para las etapas siguientes y su versatilidad en el análisis. En este sentido, los lenguajes con algún grado de formalismo subyacente prevalecen al permitir construir herramientas automáticas o semi-automáticas para su procesamiento.

Al igual que en las siguientes etapas del desarrollo, el *modelado* no se encuentra exento de errores como producto de una actividad humana. Para abordar este problema, diferentes técnicas y herramientas fueron propuestas. En este trabajo proponemos combinar dos técnicas conocidas con el objetivo de reparar posibles errores en modelos especificados en *Alloy*. Utilizando el *testing* como herramienta para localizar errores, nuestra técnica emplea el concepto de *Sketching* para descubrir y proponer una posible reparación de los mismos.

1. Introducción.

Debido al tamaño y complejidad de los sistemas de software modernos, la utilización de *modelos* en etapas tempranas del desarrollo es una técnica común, principalmente porque nos permite comprender y razonar desde un punto de vista más abstracto, alejados de los detalles de la implementación.

Está comprobado que esta comprensión abstracta del sistema contribuye a disminuir posibles errores en etapas futuras del desarrollo. Aún así, la etapa de modelado no deja de ser una actividad realizada por humanos y en consecuencia, no está exenta de errores. En este sentido, contar con la posibilidad de razonar formalmente, con sustento matemático, sobre estos modelos es una característica importante a la hora de escoger lenguajes para realizar esta tarea. Por esta razón, y cada vez con mayor frecuencia, es común encontrar algún grado de formalismo y sistematización en etapas tempranas de desarrollo como por ejemplo, captura de requisitos [12,1,2] además de modelado.

En general, el grado de popularidad de los lenguajes, en cualquier etapa del desarrollo de software, está fuertemente influenciado por las herramientas y recursos asociados a él. Si bien esta observación es evidente en los lenguajes de programación, durante la implementación, no es menos importante en las etapas tempranas. Contar con técnicas y herramientas automáticas o semi-automáticas posibilita entre otras cosas, analizar y verificar de manera sistemática ciertas propiedades deseables de nuestros modelos y actuar en consecuencia.

En este sentido, numerosas técnicas de análisis y verificación han sido propuestas para la etapa de modelado, como por ejemplo, la noción de *modelchecking* basado en representación explícita de estados y utilizando lógicas modales para la especificación de propiedades [7] o técnicas basadas en *sat-solving* que exploran de manera exhaustiva los posibles comportamientos de modelos acotados [6,10]. Más aún algunas técnicas con gran difusión y uso en etapas avanzadas, como el *testing*, se han extrapolado a las etapas iniciales para lograr una mayor confianza los modelos. Un ejemplo de ello es AUnit [11], una herramienta que permite generar y analizar casos de prueba unitario (test de unidad) para modelos Alloy.

Este trabajo se sitúa en el escenario donde tenemos un modelo Alloy como producto inicial para nuestro sistema y luego de analizarlos mediante pruebas unitarias encontramos errores, es decir, ciertos casos de prueba *no* son satisfechos por nuestro modelo. En esta situación, aprovechando la formalidad subyacente y utilizando mecanismos automáticos, proponemos una técnica que permita reparar automáticamente el modelo defectuoso en cuestión. Para lograr tal fin, utilizando la información de los casos de prueba no superados y la técnica de *sketching* para generar automáticamente predicados que, al sustituir a los originales, permitan alcanzar el caso de prueba fallidos. Si bien, las técnicas basadas en *testing* no garantizan una confianza absoluta [3,15], creemos que aportan confianza en nuestro trabajo y son plausibles de aplicar en la práctica gracias a su escalabilidad.

El trabajo procede de la siguiente manera: inicialmente repasaremos los conceptos subyacentes, mostrando de manera sintética el lenguaje elegido y el funcionamiento de las herramientas asociadas. Luego en la sección 3, explicaremos nuestra técnica, enfocándonos en *sketching* como técnica de reparación. Finalmente en las secciones 4 y 5 mencionaremos el rumbo a seguir y expondremos las conclusiones.

2. Preliminares

En esta sección realizaremos una breve revisión de los lenguajes y herramientas subyacentes a nuestro trabajo. Como lo mencionamos anteriormente, el contexto de nuestro trabajo se sitúa en la etapa de modelado y particularmente estamos interesados en lenguajes de modelado con fundamentos formales.

Alloy [6] [5] es el lenguaje elegido para este trabajo, dado que es un lenguaje de especificación formal orientado a modelos. Está definido en términos de una

semántica relacional, su sintaxis incluye constructores con notación orientada a objetos y soporta análisis automatizado acotado. Para ello provee su herramienta asociada Alloy Analyzer.

Un modelo Alloy es un conjunto de restricciones que describe implícitamente un conjunto de estructuras, estos modelos se definen esencialmente en términos de dominios de datos y operaciones entre estos dominios.

Para definir el dominio de datos utilizamos *signaturas*, para las operaciones sobre estos dominios el lenguaje nos provee *predicados* y *funciones*. Además nos permite definir restricciones sobre el modelo con fórmulas que se imponen como válidas, es decir *axiomas*. Una vez especificado nuestro modelo, mediante la utilización del Alloy Analyzer podemos buscar instancias que lo satisfagan o bien, mediante la definición de *asepciones*, podemos explorar exhaustivamente todas las posibles instancias (acotadas) del modelo en busca de contraejemplos que las contradigan.

La Fig. 1 presenta un ejemplo en el que definimos *listas simplemente encadenadas*, nuestro dominio en este modelo son las signaturas *List* y *Node*, para completar la especificación necesitamos agregar la propiedad de aciclicidad (*Acyclic*).

```

1 one sig List {
2   header: lone Node
3 }
4
5 sig Node {
6   link: lone Node
7 }
8
9 pred Acyclic() {
10  all n: Node | n in List.header.*link => n !in n.^link
11 }

```

Figura 1. Ejemplo de modelo Alloy

Una vez que tenemos nuestro modelo especificado, una pregunta que surge es si expresa exactamente lo que esperamos. Una forma de ganar confianza aplicando técnicas más simples que la especificación de propiedades, es realizar *testing*. Si bien no da garantías absolutas, poder llevar a cabo esta tarea sólo requiere de conocer instancias válidas (concretas) que nuestro modelo debería cumplir.

AUnit [11] es una herramienta que extiende al Alloy Analyzer que nos permite escribir y ejecutar casos de prueba (tests) sobre nuestra especificación Alloy. Esta extensión nos brinda un entorno de prueba formal y presenta una gramática adicional para escribir los casos de prueba. Continuando con el ejemplo de listas simplemente encadenadas, la Figura 2 muestra un ejemplo de uso de AUnit.

Además, la herramienta admite la generación automática de casos de prueba que puede aplicarse sobre todo nuestro modelo, o bien parcialmente mediante la selección de predicados particulares. La herramienta soporta una interfaz fácil

de utilizar y con reportes avanzados como por ejemplo métricas de cobertura de código.

```

val Test5{
  some disj List0 : List | some disj Node0, Node1 : Node | {
    List = List0
    Node = Node0 + Node1
    header = List0->Node0
    link = Node0->Node0
    @cmd:{ !Acyclic[] } }
}
@Test Test5: run Test5 for 3 expect 1

```

Figura 2. Ejemplo de AUnit test

En ciertas ocasiones, la etapa de modelado, puede ser una sucesión de refinamientos sobre un modelo parcial inicial. Para abordar estos casos, no sólo es imprescindible contar con un lenguaje que nos de soporte, sino también contar con técnicas que ayuden a transitar los ciclos de refinamiento hasta llegar al modelo deseado. Ejemplos de ello son los Sistemas Transición Modales (MTS) [4] que permiten capturar diferentes nociones de obligación en los modelos y luego, de manera automática, sintetizar controladores que garanticen el cumplimiento de los objetivos. En el caso de Alloy, una técnica que podemos aplicar para “completar” modelos parciales es *sketching*.

ASketch [14] es una herramienta que dado un modelo Alloy con *holes* (huecos marcados sintacticamente) nos permite generar predicados de manera que el modelo satisfaga un conjunto de casos de prueba asociado a él. Estos *holes* o comodines pueden ser operadores binarios, de comparación, lógicos, cuantificadores, operadores unarios o incluso expresiones de navegación sobre el dominio del modelo como se detalla en la Tabla 1. Estos operadores se anotan dentro de los predicados para debilitar (especificar parcialmente) restricciones o propiedades del modelo.

Como ejemplo, en la Figura 3 se anota el predicado *Acyclic* para indicar la cuantificación sobre elementos de la signatura *Node*. Tal que, los elementos de node que cumplen con una comparación con otra expresión de navegación, necesariamente cumplen otra comparación con una expresión de navegación que puede o no ser la misma.

Una vez anotado el modelo, **ASketch** genera los predicados candidatos, aquellos que cumplen con los *holes* indicados sintacticamente, pero que superen los casos de prueba asociado. El principal objetivo de esta herramienta es ayudar al usuario a especificar un modelo Alloy de manera correcta en base a las instancias que describen los casos de prueba asociado.

```

pred Acyclic() {
  -- holes:
  \Q,q\ n: Node | n \CO,co\ \E,e\ => n \CO,co\ \E,e\
}

```

Figura 3. Predicado *Acyclic* con *holes*

Hole	Notación	Candidatos
Operador binario	\BO\	&, +, -
Operador de comparación	\CO\	=, in, !=, !in
Operador lógico	\LO\	, &&, <=>, =>
Cuantificador	\Q\	all, no, some, lone, one
Operador Unario	\UO\	no, some, lone, one
Operador de expresión unaria	\UOE\	~, *, ^
Operador de formula unaria	\UOF\	!, ε
Expresión	\E\	cualquier expresión

Tabla 1. Operadores para escribir *holes* de ASketch.

3. Hacia la reparación de modelos Alloy con ASketch

Para evaluar *sketching* como técnica de reparación, partimos de un modelo correcto y generamos automáticamente un conjunto de casos de prueba para el mismo mediante AUnit. Luego utilizaremos este conjunto para definir si las reparaciones candidatas verdaderamente lo son.

En pos de generar un escenario para evaluar nuestra propuesta, incorporamos al modelo una serie de errores. Cabe destacar que comprobamos que dichos errores sean detectados por los casos de prueba que generamos anteriormente.

Aprovechando el análisis que realiza AUnit sobre nuestro modelo erróneo, y en particular el reporte de cobertura de código, podemos distinguir cuáles son los predicados que infringen a los casos de prueba no superados. Una vez identificados, procedemos a anotar dichos predicados con los operadores (*holes*) que mencionamos anteriormente en la Tabla 1, es decir, debilitamos la especificación de dicho predicado, supuestamente erróneo, tornándolo parcial.

Ahora sí, tomando esta especificación parcial comenzamos el proceso de generación mediante ASketch, una vez obtenidos todos los predicados candidatos, los incorporamos (de a uno por vez en caso de haber más de uno) en el modelo erróneo sustituyendo al predicado parcial y contrastamos con todo el conjunto de casos de prueba. Cabe recordar que pueden existir más de un predicado potencialmente erróneo. Si logramos reemplazar todos los predicados erróneos, entonces hemos encontrado una *reparación* de nuestro modelo que pasa todos los casos de prueba asociados a él.

3.1. Casos de estudio

Para llevar adelante la evaluación que planteamos, tomamos de la literatura[14,8] distintos casos de estudio con su conjunto de casos de prueba que utilizaremos evaluar nuestra propuesta.

Single Linked List El modelo correcto para este caso es el utilizado como ejemplo en la sección anterior. En la Figura 5 mostramos el predicado *acyclic* con *holes* y tres candidatos generados por ASketch.

```

pred Acyclic() {
  -- holes:
  \Q,q\ n: Node | n \CO,co\ \E,e\ => n \CO,co\ \E,e\
  -- candidato 1:
  all n: Node | n in n.^link => n !in List.header.^link
  -- candidato 2:
  all n: Node | n in n.^link => n !in List.header.^link
  -- candidato 3:
  all n: Node | n in List.header.^link => n !in n.^link
}

```

Figura 4. Predicado Alloy con holes

Luego de probar los tres casos candidatos en AUnit, observamos que todos los casos pasan correctamente todos los casos de prueba que utilizamos para evaluar como reparación.

Grade Para este caso también generamos los casos de prueba automáticamente para el modelo. Además se modifica el predicado *PolicyAllowsGrading* insertando errores para lograr un modelo erróneo. La Figura 6 muestra el predicado con *holes* en donde AUnit identifica posibles errores y, finalmente, las soluciones generadas que propone ASketch.

En este caso de estudio, los tres predicados candidatos *no logran* pasar los casos de prueba, es decir, no logramos una reparación.

Árboles Coloreados Este ejemplo aborda la especificación de la conocida estructura de datos de árboles coloreados. La Figura 7 exhibe un modelo correcto para el mismo. De manera similar a los casos anteriores, se genera el conjunto de casos de prueba que oficiará de oráculo. Para conseguir un escenario más complejo, para este modelo vamos a considerar más de un predicado erróneo mediante su manipulación. Luego de ejecutar ASketch, evaluamos los candidatos consiguiendo las tres reparaciones.

4. Conclusiones

Alloy es un lenguaje de especificación formal muy utilizado para el modelado de sistemas, su herramienta Alloy Analyzer facilita un análisis completamente

```

1 abstract sig Person {}
2
3 sig Student extends Person {}
4
5 sig Professor extends Person {}
6
7 sig Class {
8   assistant_for: set Student,
9   instructor_of: one Professor
10 }
11
12 sig Assignment {
13   associated_with: one Class,
14   assigned_to: some Student
15 }
16
17 pred PolicyAllowsGrading(s: Person, a: Assignment) {
18   s in a.associated_with.assistant_for || s in a.associated_with.instructor_of
19   s !in a.assigned_to
20 }

```

Figura 5. Predicado Alloy Grades

```

pred PolicyAllowsGrading(s: Person, a: Assignment) {
  -- Holes:
  \E,e\ \C0,co\ \E,e\ \L0,lo\ \E,e\ \C0,co\ \E,e\
  \E,e\ \C0,co\ \E,e\
  -- Candidato 1:
  Class.assistant_for !in Class.assistant_for &&
  a.associated_with.assistant_for = a.associated_with.instructor_of
  Class.assistant_for in s
  -- Candidato 2:
  s != Class.instructor_of =>
  a.associated_with.assistant_for = a.associated_with.instructor_of
  a.assigned_to !in a.associated_with.assistant_for
  -- Candidato 3:
  a.associated_with.assistant_for !in a.associated_with.instructor_of =>
  a.assigned_to != s
  a.associated_with.assistant_for != Class.assistant_for
}

```

Figura 6. Predicado PoliciAllowsGrading *Hole* y candidatos de reparación

automatizado para explorar los modelos que genera. Las extensiones que estudiamos son de mucha utilidad en ambos casos y nutren al análisis que se realiza en este tipo de lenguaje haciéndolo más completo y más simple para el usuario, tanto para evaluar nuestros modelos mediante técnicas como el *testing*, como así también para sintetizar predicados de especificaciones parciales, guiado por casos de prueba.

En este sentido de analizar y explorar de manera automática un modelo, analizar la reparación de los mismos de manera automática se acerca a completar las herramientas que facilitan el uso de este tipo de lenguajes y ayudan al usuario en su especificación.

Luego de la evaluación que realizamos y los resultados obtenidos, podemos concluir que la reparación de modelos mediante la técnica de *sketching* tiene

```

1 abstract sig Color {}
2 one sig Red extends Color {}
3 one sig Blue extends Color {}
4
5 sig Node {
6   neighbors: set Node,
7   color: one Color
8 }
9
10 pred undirected {
11   neighbors = ~neighbors
12   no iden & neighbors
13 }
14
15 pred graphIsConnected {
16   all n1: Node | all n2: Node-n1 |
17     n1 in n2.^neighbors
18 }
19
20 pred treeAcyclic {
21   all n1, n2: Node |
22     n1 in n2.neighbors => n1 !in n2.^(neighbors-(n2->n1))
23 }
24
25 run {} for 3 Node

```

Figura 7. Especificación Alloy para Árboles Coloreados

muchos puntos de interés para trabajar, es decir, tiene muchas potenciales optimizaciones a realizar. Cabe notar que esta técnica, guiada por casos de prueba no está exenta del problema abordado en [15]. Es decir, consideramos la reparación tomando como oráculo un conjunto de casos de prueba, lo cual, las garantías de correctitud están directamente determinadas por la calidad de dicho conjunto. En este sentido, al observar los resultados obtenidos, y en particular para el caso de estudio *Grade*, *ASketch* logra sintetizar predicados candidatos como reparación, pero al ser evaluados por la suite de test de aceptación los mismos no se comportan como un modelo correcto.

5. Trabajo futuro.

Tal como observamos en esta evaluación y en [14], la calidad de generación de predicados que *ASketch* nos provee esta ligada fuertemente a los conjuntos de casos de prueba y los fragmentos que utilizamos. Sobre los conjuntos de casos de prueba, el uso de herramientas automáticas de generación de test es una buena opción. *AUnit*nos permite generar test para los modelos de manera *bounded exhaustive* [11] ó realizando *mutation testing* [13]. Además de utilizar estos test generados de manera automática y reemplazar los propuestos por el usuario, también se puede mejorar la generación de test utilizando técnicas de *field exhaustive* [9]. Mediante esta técnica también pueden generarse de manera automática los fragmentos que utilizamos para realizar *sketching*. Ambos casos nos permitirían tener mejores recursos para la generación de predicados de *ASketch*, y así mejorar la calidad de los predicados generados.

```

pred undirected {
  -- Holes:
  \E,e1\ \C0,co\ \E,e1\
  \U0,uo\ \E,e1\
  -- Candidato 1:
  neighbors = ~neighbors
  no iden & neighbors
  -- Candidato 2:
  ~neighbors, =, neighbors
  no iden & neighbors
  -- Candidato 3:
  neighbors in ~neighbors
  no iden & neighbors
}

pred graphIsConnected {
  all n1: Node | all n2: Node-n1 |
  -- Holes:
  \E,e2\ \C0,co\ \E,e2\
  -- Candidato 1:
  n2.(neighbors) != n2.^(neighbors-n2->n1)
  -- Candidato 2:
  n2.^(neighbors-n2->n1) != n2.(neighbors)
  -- Candidato 3:
  n2.^(neighbors-n2->n1) != n2.(neighbors)
}

pred treeAcyclic {
  all n1, n2: Node |
  -- Holes:
  \E,e2\ \C0,co\ \E,e2\ \L0,lo\ \E,e2\ \C0,co\ \E,e2\
  -- Candidato 1:
  n2.^(neighbors-n2->n1) in n2.(neighbors) <=>
  n2.^(neighbors-n2->n1) in n2.(neighbors-n2->n1)
  -- Candidato 2:
  n1 !in n2.(neighbors) <=> n2.(neighbors) in n2.^(neighbors-n2->n1)
  -- Candidato 3:
  n2.(neighbors-n2->n1) = n2.^(neighbors-n2->n1) <=>
  n2.^(neighbors-n2->n1) in n2.(neighbors)
}

```

Figura 8. Predicados CTree con *Holes* y candidatos de reparación para cada uno

Otro punto a trabajar es evaluar fuertemente la generación de candidatos como reparaciones que sólo cumplen con los test brindados pero no solucionan el problema general. El oráculo (test) es muy débil para asegurar las propuestas como reparación [15]. Es por esto que creemos necesaria una evaluación de la técnica que presentamos para evaluar la presencia de falsos positivos en las reparaciones que *ASketch* nos genera.

Referencias

1. Lilian Burdy, Yoonsik Cheon, David R. Cok, Michael D. Ernst, Joseph R. Kiniry, Gary T. Leavens, K. Rustan M. Leino, and Erik Poll. An overview of JML tools and applications. *STTT*, 7(3):212–232, 2005.
2. Renzo Degiovanni, Dalal Alrajeh, Nazareno Aguirre, and Sebastián Uchitel. Automated goal operationalisation based on interpolation and SAT solving. In *36th ICSE '14, Hyderabad, India - May 31 - June 07, 2014*, pages 129–139, 2014.
3. Edsger W. Dijkstra. The humble programmer. 1972.
4. Nicolás D'Ippolito, Dario Fischbein, Marsha Chechik, and Sebastián Uchitel. MT-SA: the modal transition system analyser. In *23rd IEEE/ACM (ASE 2008), 15-19 September 2008, L'Aquila, Italy*, pages 475–476, 2008.
5. Daniel Jackson. Alloy: a lightweight object modelling notation. *ACM Trans. Softw. Eng. Methodol.*, 11(2):256–290, 2002.
6. Daniel Jackson. *Software Abstractions: Logic, Language, and Analysis*. The MIT Press, 2006.
7. Jeff Magee and Jeff Kramer. *Concurrency - state models and Java programs (2. ed.)*. Wiley, 2006.
8. Tim Nelson, Natasha Danas, Daniel J. Dougherty, and Shriram Krishnamurthi. The power of "why" and "why not": enriching scenario exploration with provenance. In *Proceedings of the 2017 11th, ESEC/FSE 2017, Paderborn, Germany, September 4-8, 2017*, pages 106–116, 2017.
9. Pablo Ponzio, Nazareno Aguirre, Marcelo F. Frias, and Willem Visser. Field-exhaustive testing. In *Proceedings of the 24th ACM SIGSOFT, FSE 2016, Seattle, WA, USA, November 13-18, 2016*, pages 908–919, 2016.
10. Germán Regis, César Cornejo, Simón Gutiérrez Brida, Mariano Politano, Fernando Raverta, Pablo Ponzio, Nazareno Aguirre, Juan Pablo Galeotti, and Marcelo F. Frias. Dyalloy analyzer: a tool for the specification and analysis of alloy models with dynamic behaviour. In *Proceedings of the 2017 11th ESEC/FSE 2017, Paderborn, Germany, September 4-8, 2017*, pages 969–973, 2017.
11. Allison Sullivan, Kaiyuan Wang, and Sarfraz Khurshid. Aunit: A test automation tool for alloy. In *11th IEEE International Conference on Software Testing, Verification and Validation, ICST 2018, Västerås, Sweden, April 9-13, 2018*, pages 398–403, 2018.
12. Axel van Lamsweerde. *Requirements Engineering - From System Goals to UML Models to Software Specifications*. Wiley, 2009.
13. Kaiyuan Wang, Allison Sullivan, and Sarfraz Khurshid. Mualloy: a mutation testing framework for alloy. In *Proceedings of the 40th ICSE 2018, Gothenburg, Sweden, May 27 - June 03, 2018*, pages 29–32, 2018.
14. Kaiyuan Wang, Allison Sullivan, Darko Marinov, and Sarfraz Khurshid. Asketch: a sketching framework for alloy. In *Proceedings of the 2018 ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering, ESEC/SIGSOFT FSE 2018, Lake Buena Vista, FL, USA, November 04-09, 2018*, pages 916–919, 2018.
15. Luciano Zemín, Simón Gutiérrez Brida, Ariel Godio, César Cornejo, Renzo Degiovanni, Germán Regis, Nazareno Aguirre, and Marcelo F. Frias. An analysis of the suitability of test-based patch acceptance criteria. In *10th IEEE/ACM International Workshop on Search-Based Software Testing, SBST@ICSE 2017, Buenos Aires, Argentina, May 22-23, 2017*, pages 14–20, 2017.

Una propuesta para clasificación de roles de usuarios en foros de discusión técnicos

Nadina Martínez Carod, Gabriela Aranda, Valeria Zoratto, Christian Murray

Grupo GIISCo, Facultad de Informática, Universidad Nacional del Comahue
Buenos Aires 1400 (8300) Neuquén, Argentina
{nadina.martinez|gabriela.aranda|vzoratto}@fi.uncoma.edu.ar,
cristianmurray@hotmail.com

Resumen Este trabajo se enfoca en la detección de roles de usuarios participantes en foros de discusión técnicos, con el propósito de incorporar dicha clasificación como parte de una herramienta de recuperación de información de foros de discusión. El objetivo de dicha herramienta es, a partir de una consulta específica, determinar un ranking de hilos de discusión relevantes, seleccionados de varios foros de discusión. El ranking resultante estará ordenado según diferentes características de calidad, dentro de las cuales serán considerados los roles de los usuarios participantes. En este trabajo se definen las características de los distintos roles, la forma de clasificar los mismos y se presenta una validación preliminar del modelo propuesto en un caso de estudio.

1. Introducción

Los foros de discusión técnicos facilitan la creación de espacios de debate científico y permiten compartir conocimiento específico en ambientes informales, lo que los transforma en bitácoras actualizadas sobre temas específicos, que muy a menudo son utilizados para la recuperación de información (RI). Los usuarios que participan en dichos foros comparten conocimientos y experiencias, siendo éstas de gran utilidad cuando un problema surge repetidamente, por lo que el desafío fundamental radica entonces, en detectar los hilos de discusión que contienen las soluciones más adecuadas para un problema particular, analizando los debates que suceden en uno o varios foros de discusión a la vez. Con ese objetivo en mente, en [1] se definió un conjunto inicial de características de calidad para evaluar las soluciones encontradas en diferentes foros de discusión, seguido del diseño de un modelo para obtener un ranking de soluciones posibles, considerando diferentes ópticas. Por un lado se propuso un conjunto preliminar de métricas de calidad [2], el cual fue probado en un grupo de hilos de discusión reales [3]. Por otro lado, se planteó un modelo de clasificación basado en el análisis del texto de los hilos bajo estudio [4]. Posteriormente, se extendió dicho modelo mediante la incorporación de sinónimos según la estructura semántica de las oraciones [5], para lo cual se utilizó la base de datos léxica WordNet [6] de manera conjunta con la herramienta de procesamiento de lenguaje natural Stanford POS Tagger [7].

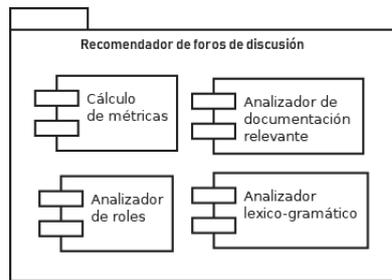


Figura 1. Esquema del recomendador de foros

Para extender la propuesta inicial de un recomendador de foros se ha agregado una nueva componente que utiliza el rol de los usuarios que participan compartiendo opiniones y experiencias en el foro, bajo la premisa que el rol del usuario está en estrecha relación con el conocimiento del mismo. Para entender la funcionalidad de la herramienta en construcción, en la Figura 1 se muestra el esquema del recomendador de foros de discusión, con las componentes mencionadas. A continuación se describirá más en detalle el agregado del componente *Analizador de roles*.

El resto del artículo está organizado de la siguiente manera: en la Sección 2 se introduce una herramienta que mantiene y gestiona la información contenida en foros de discusión para establecer el ranking de la información existente en dichos foros. Esto se realiza incorporando la funcionalidad de clasificar los participantes de los foros técnicos, para luego poder asociarles un grado o nivel de conocimiento en el tema. En la Sección 3 se valida la estrategia propuesta mediante la utilización de un caso de estudio. Finalmente, se analizan los resultados obtenidos comparándolos con una estimación de la experiencia de los participantes mediante una encuesta personal, se presentan las conclusiones y líneas de trabajo futuro.

2. Recomendador de foros

En [1] se presentó el modelo conceptual de la información contenida en foros de discusión desde el punto de vista del usuario externo, identificándose las entidades más importantes y sus atributos. Siguiendo este modelo, se plantea la arquitectura de la herramienta *Recomendador de foros* [8] cuyo fin es recomendar una lista de hilos de discusión técnicos, ordenada de acuerdo a determinada prioridad. Esta herramienta está diseñada en tres capas, como se muestra en la Figura 2.

En la capa de Datos se destaca el *Repositorio de hilos*, donde se mantienen los archivos recuperados en formato HTML, el *Repositorio XML*, preservado con fines estadísticos y de control. Luego se agrega el repositorio de *Frases de roles* que son formaciones semánticas específicas para una taxonomía de roles.

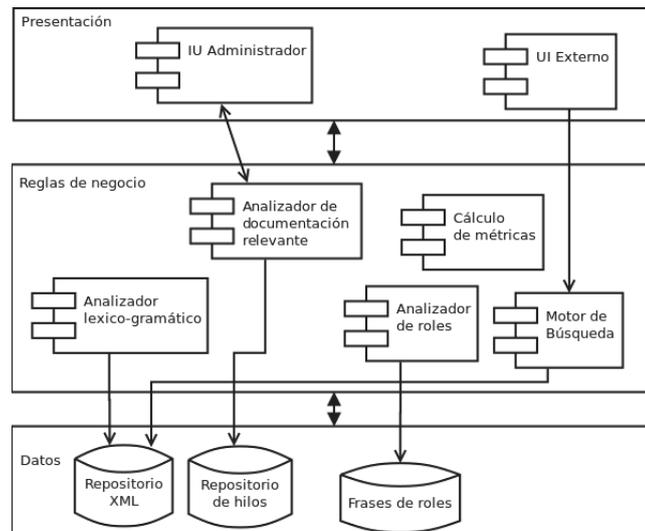


Figura 2. Arquitectura del recomendador de foros propuesto

La capa de Negocio de la aplicación incluye el componente *Motor de búsqueda*, encargado de la búsqueda y descarga de hilos de discusión disponibles en la web; mientras que el proceso de generación de documentos XML asociado se completa agregando etiquetas de algunos atributos puntuales y actualizando el repositorio de archivos XML. El componente *Analizador de documentación relevante* analiza la sintáxis de los fragmentos del hilo de discusión y los clasifica en base a la documentación Oracle Java [4]. Luego, el componente *Analizador léxico-gramático* mejora dicha clasificación mediante la incorporación de sinónimos utilizando WordNet de manera conjunta con la aplicación Stanford POS Tagger [5]. El componente *Motor de búsqueda* interactúa con la capa de presentación, solicitando al usuario que ingrese una cadena de búsqueda que represente su problema y, como resultado, devuelve una lista rankeada de soluciones candidatas, requiriendo los servicios del resto de los componentes que conforman las *Reglas de negocio*. El componente que se agrega en esta capa es el *Analizador de roles*, el cual será explicado a continuación en detalle.

La capa de Presentación incluye las interfaces de *Usuario Externo*, que provee una interfaz Web para que el recomendador de foros sea utilizado por usuarios interesados en encontrar soluciones a un problema técnico particular, y la de *Usuario Administrador*, para usuarios que realicen tareas de mantenimiento, carga y actualización de la base de datos, así como definición de nuevas métricas.

3. Analizador de Roles

El rol de un usuario participante está estrechamente relacionado con el conocimiento y expertitud del mismo; por consiguiente el componente *Analizador*

de roles tiene como función determinar los roles de los participantes de un hilo de discusión a partir de características y comentarios que contienen sus posts dentro de ese hilo. Para ello, partiendo de los hilos recuperados, se buscan frases o construcciones semánticas mantenidas en un repositorio de *Frases de roles*, identificándolas y clasificándolas, determinando así un rol para cada uno de los participantes de un hilo.

Se pueden diferenciar 5 roles distintos, los cuales se identifican como *Líder*, *Jefe*, *Moderador*, *Novato* y *Conflictivo*, cada uno de ellos puede ser identificado con un tipo de frases determinadas. Por ejemplo, los participantes con rol *Líder* son los usuarios con mayor conocimiento en el tema de discusión, emiten respuestas a las consultas, ayudan activamente y son reconocidos por sus aportes. Generalmente, luego de dar sus sugerencias, buscan verificar que sus respuestas hayan sido útiles. Se reconocen por comenzar sus posts con frases del tipo "...lo primero que..." , "...prueba con este código..." , "...debes usar..." , entre otras; y terminan sus post con frases del tipo "...espero que te sirva..." , "...ya nos cuentas..." .

Las personas con categoría *Jefe* tienen bastante conocimiento en el tema de discusión pero sólo dan directivas de solución. En general dan pautas, emiten links, sugieren búsquedas pero no resuelven los problemas. Los *Moderadores* conocen los temas y construyen lazos de comunicación entre usuarios, también supervisan; los usuarios *Novatos* tienen poco conocimiento sobre el tema de discusión; los usuarios *Conflictivos*, a diferencia de los otros roles, tienen comportamientos negativos en la conversación, emitiendo mensajes provocadores e irrelevantes en los temas que se discuten. Los usuarios *Novatos*, a su vez, tienen dos subroles bien determinados, el *Novato Implicado*, que si bien tiene poco conocimiento, a veces ayuda expresando la experiencia que tuvo él con un problema determinado y el *Novato Incipiente* tiene muy poco conocimiento y además no aporta ni contesta preguntas de otros usuarios. Los usuarios *Conflictivos* se categorizan a su vez en *Publicitario* (emite propagandas o información sin sentido), *Impostor* (amenaza o intenta engaños informáticos) y *Agresivo*, que utiliza lenguaje agresivo, grita (escribe en mayúsculas) o utiliza lenguaje indebido.

La Figura 3 esquematiza el proceso propuesto para la identificación de roles junto a la propuesta de evaluación de los resultados obtenidos. El proceso de *detección de frases de roles*, analiza el texto de los hilos e identifica las oraciones a partir del repositorio de *Frases por roles*, almacenado como parte de la taxonomía de roles. Mediante el proceso de *identificación de roles* se determina el rol final de cada usuario participante, los cuales están asociados a un nivel de conocimiento.

Por otro lado, para validar la metodología, para cada hilo se realiza un proceso de *clasificación manual* de los usuarios participantes, en base a una encuesta de opinión a un grupo de evaluadores humanos.

El objetivo del proceso de *comparación de resultados* es contrastar los resultados obtenidos mediante ambos procedimientos y proceder a su análisis en la etapa de *análisis de resultados*.

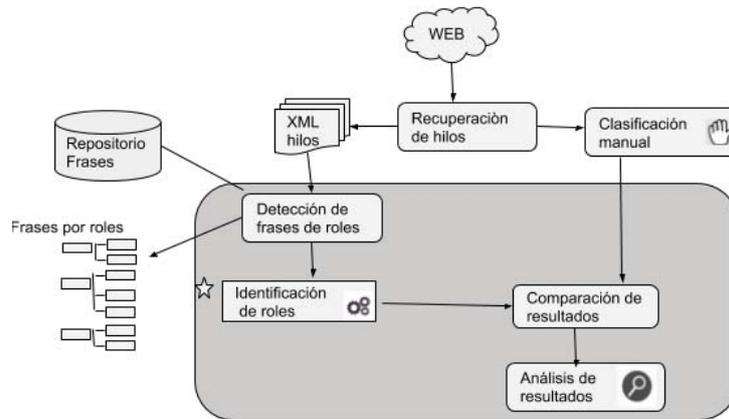


Figura 3. Esquema del proceso de identificación de roles en hilos de discusión

Detección de roles

Para poder detectar el rol que cumple un usuario dentro de un hilo de discusión se analiza el texto escrito por ellos en cada post utilizando como soporte el repositorio de frases definido anteriormente. Para ello, primero se definen las diferentes componentes que constituyen cada rol, como se puede apreciar en la Tabla 1. En dicha Tabla se observa que cada rol tiene un *id* asociado, un nivel de expertitud y un valor único (pR_i) que se encuentra dentro del rango de valores determinado para ese rol (*Rango según Rol R_i*). Por ejemplo, en la primera fila, el *Rol Líder* – que corresponde al rol de mayor conocimiento –, tiene el nivel 1 de *Expertitud*, y como valor pR_i : 99 que, en este caso, es el mayor del rango [90-99]. Esto se cumple en todos los casos, salvo para los subroles del rol *Conflictivo*, en los cuales se elige el menor valor del rango (puntajes 40, 30 y 20).

Cuando se analiza el texto del post de un usuario, se buscan frases que mapeen con las frases del repositorio. Luego de detectar dichas frases se puede saber a que rol pertenece un usuario en ese post, entonces se le asigna un *id rol* para el usuario en ese post y un valor pR_i , y así se hace con cada post perteneciente a ese usuario específico. Luego dicho valor pR_i es utilizado para poder detectar el rol global que cumple el usuario en el hilo de discusión.

La estrategia para detectar los roles de los diferentes participantes se basa en una fórmula que utiliza los valores obtenidos en cada post del participante, generando así un valor final por usuario. Este valor final se va a encontrar entre alguno de los rangos de roles establecidos anteriormente (Tabla 1) y va a determinar el rol que cumple un usuario dentro de un hilo de discusión. El valor mencionado se calcula a partir de la siguiente fórmula:

$$p(u) = \frac{\sum_{i=1}^n (c_i * pR_i)}{\sum_{i=1}^n c_i} \quad (1)$$

donde u es el Usuario que participa en un hilo de discusión de un foro; $p(u)$ es el puntaje de un usuario u ; i es un escalar identificador del rol (entre 1 y 8);

Tabla 1. Asignación de puntos para roles de usuarios

Rol	Id rol R_i	Expertitud	Valor asignado (pR_i)	Rango según $RolR_i$	
Líder	R_1	1	$pR_1=99$	90-99	
Jefe	R_2	2	$pR_2=89$	80-89	
Moderador	R_3	3	$pR_3=79$	70-79	
Novato	Implicado	R_4	4.1	$pR_4=69$	60-69
	Incipiente	R_5	4.2	$pR_5=59$	50-59
Conflictivo	Publicitario	R_6	5.1	$pR_6=40$	40-49
	Impostor	R_7	5.2	$pR_7=30$	30-39
	Agresivo	R_8	5.3	$pR_8=20$	20-29

c_i es la cantidad de veces que el usuario participa en el hilo de discusión con rol R_i ; pR_i es el puntaje asignado al usuario con rol R_i , y n es la cantidad de roles/subroles posibles.

De esta manera, el puntaje establecido para el usuario u se obtiene a partir de considerar la presencia o no de cada uno de los 8 roles/subroles posibles producto el valor asignado a dicho rol (pR_i). Por último, el valor obtenido estará influenciado por la sumatoria todos los roles presentes. Para determinar el *valor global* se utiliza la columna *Rango según rol* de la Tabla 1 de acuerdo al valor obtenido $P(u)$.

4. Validación

Para estudiar el comportamiento de la propuesta en un contexto real, se llevó a cabo un caso de estudio [9]. Para ello se realizó un conjunto de encuestas basadas en hilos reales de 5 foros diferentes. La elección de los hilos se realizó considerando aquellos que tuvieran más de 8 posts, en los cuales intervengan más de 3 usuarios participantes en la discusión y cuya pregunta disparadora inicial sea una duda técnica. Para cada una de las encuestas se transcribió el hilo de discusión completo elegido, asignando al azar un conjunto de hasta 5 encuestas a cada uno de los evaluadores. Para cada encuesta un evaluador debía puntuar el grado de conocimiento de cada usuario en el hilo de conversación, siguiendo una escala numérica del 1 al 5, donde 1 equivale a mucho conocimiento; 2, a bastante; 3, a impreciso; 4 a poco conocimiento y 5 a no tener conocimiento en el tema. Los evaluadores fueron un grupo de 25 estudiantes de Informática (UNComa), 11 del primer año de la carrera de Licenciatura en Ciencias de la Computación y el resto del primer año de la Tecnicatura en Desarrollo Web, el detalle de la composición de las encuestas puede observarse en la Tabla 2 donde *Id* es el identificador de la encuesta, *Referencia URL hilo* es la dirección de la misma, *Cant.Eval.* indica la cantidad de evaluadores que completaron cada encuesta, *Cant.posts* corresponde con la cantidad de posts del hilo de discusión y la *Cant.Usu* es la cantidad de usuarios que participan del hilo de discusión.

Tabla 2. Resumen de encuestas

ID	Referencia URL hilo	Cant. Eval.	Cant. posts	Cant. Usu.
1	http://www.javahispano.org/java-se/post/1612752	14	9	4
2	http://www.espaciolinux.com/foros/programacion/ayuda-con-programa-proceso-hijo-ficheros-script-t34804.html	16	9	5
3	https://www.lawebdelprogramador.com/foros/Pascal-Turbo-Pascal/251768-contar-palabras-de-un-fichero.html	12	8	4
4	http://www.espaciolinux.com/foros/programacion/problema-con-shell-script-t32515.html	14	10	4
5	http://www.javamexico.org/foros/conceptos/el_punto_que_significa	12	10	8

Para comparar los resultados obtenidos entre la heurística propuesta en la Sección 3 y el resultado de las encuestas, se cotejó el valor obtenido automáticamente, para cada usuario contra la media de los valores otorgados por los evaluadores a ese usuario, en el hilo correspondiente. El resultado puede visualizarse en la Figura 4, donde las barras representan el valor obtenido por la heurística y la línea continua representa la media otorgada por los evaluadores. Por ejemplo, el valor de la media otorgado al participante u_1h_1 (el cual corresponde al usuario 1 del hilo 1), concuerda entre el valor de la media $valormediaEvaluadores$ con R_4 (rol Novato), dado por la heurística. La misma concordancia se da para los usuarios participantes u_2h_1 , con el valor R_1 (rol Líder), al usuario u_3h_2 con el valor R_3 (rol Moderador), usuario u_4h_2 con el valor R_2 (rol Jefe), usuario u_2h_3 , con el valor R_2 (rol Jefe) y al usuario u_2h_5 con el valor R_3 (rol Moderador).

Al analizar en detalle los valores que difieren la media de los evaluadores y la heurística, se observa que la diferencia es de sólo 1 punto y que en pocos casos (6), se da una diferencia de 2 puntos, los cuales están detallados en la Tabla 3, donde se puede visualizar, para cada uno de esos usuarios, la conformación del $P(u)$, en la columna detalle, el valor de la media de los evaluadores y el de la heurística. Analizando específicamente los usuarios en los que hubo mayor diferencia entre la heurística y los evaluadores, se detectó que en el 83.4 % de los casos la media de los evaluadores asignó el valor 3, correspondiente a la opción de conocimiento *impreciso*.

Finalmente, al comparar los resultados obtenidos por los evaluadores con los hallados por la heurística en general se encuentra concordancia entre ambos resultados, en particular en los casos en que el grado de conocimiento estaba mejor definido. Otra detalle a destacar es que en muchos casos existe una

dificultad para determinar el nivel de conocimiento de los participantes cuando aparece un usuario participante con rol *Conflictivo*.

Tabla 3. Heurística vs media de evaluadores

Usuario	p(u)	Heurística	Ruf	Detalle	Media evaluador
u_2h_2	49	5	R_6	$C_3 = 1; C_8 = 1; pR_3 = 79; pR_8 = 20$	3
u_4h_3	20	5	R_8	$C_8 = 1; pR_8 = 20$	3
u_4h_5	40	5	R_6	$C_6 = 1; pR_6 = 40$	3
u_5h_5	40	5	R_6	$C_6 = 1; pR_6 = 40$	3
u_6h_5	40	5	R_6	$C_6 = 1; pR_6 = 40$	3
u_8h_5	69	4	R_4	$C_4 = 2; pR_4 = 69$	2

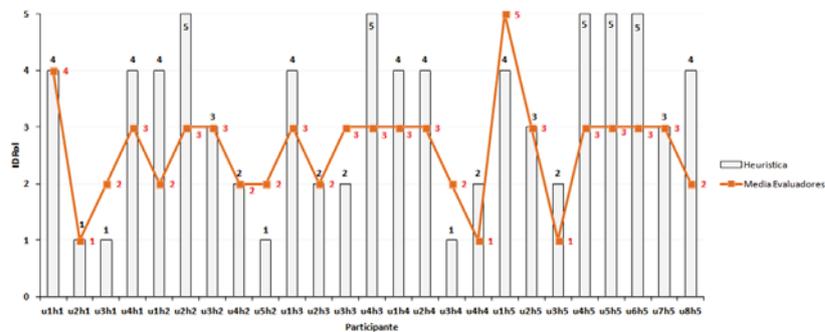


Figura 4. Heurística vs media de evaluadores

5. Trabajos relacionados

Otros trabajos han analizado los roles de los usuarios en hilos de discusión. Por ejemplo, Wever y otros [10] tienen como objetivo determinar la importancia de la asignación de roles de usuario para la construcción del conocimiento, para lo cual definen cinco roles (Started, Summariser, Moderator, Theoretician y Source Searcher), y utilizan el modelo de análisis de interacción Guanaardena [11], la diferencia es que lo utilizan en un entorno de aprendizaje colaborativo. Por otro lado, el trabajo de Lui & Baldwin [12] analiza los usuarios en base a cuatro habilidades básicas acordes al nivel de claridad de expresión en el post, de cuán positivo es el candidato, del esfuerzo de comunicación, del conocimiento, y la predicción de roles. Otra diferencia, es que estos autores utilizan redes de usuarios (grafos en donde los nodos son los participantes que emiten los posts y las aristas representan la clase de vinculación que hay entre los mismos), y redes de hilos (donde los nodos representan hilos de discusión y las aristas la similitud entre ellos) para su análisis. El trabajo, de Golder & Donath [13],

utiliza una taxonomía de roles y estrategias de interacción que describen el comportamiento que adoptan los usuarios en los foros de discusión para lograr obtener respuestas a sus consultas y, la diferencia con esta perspectiva, es que se enfocan en foros sociales de temas variados y gran volumen de información. Por su parte, Zhang y otros [14] definen su trabajo exclusivamente para la comunidad de Java, analizando el foro de discusión mediante una red social, clasificando cinco roles desde novatos hasta expertos Java. Su enfoque utiliza estadísticas costosas en términos de procesamiento, sumado a que se basa en la suposición de que si un usuario responde a otro indica un mayor conocimiento del que realizó la pregunta, lo que no siempre se cumple. Finalmente, la propuesta de Fisher y otros [15] utiliza la visualización y el análisis de redes sociales en los patrones de respuestas de cada autor, en grupos de noticias seleccionados, para encontrar diferentes tipos de participantes. Esto lo realizan por medio de redes egocéntricas calculadas sobre un usuario, usando el grado de entrada (a cuántas personas respondió) y grado de salida (cuánta gente le respondió) para identificar los roles dentro del grupo.

6. Conclusiones y trabajo futuro

En este trabajo se presenta una estrategia de clasificación de roles para participantes de hilos de discusión en foros técnicos, determinando para cada rol un grado de conocimiento y expertitud específico. Con el objetivo de evaluar la aplicabilidad de esta propuesta, se realizó un caso de estudio en base a cinco hilos de foros de discusión reales y se compararon los resultados obtenidos mediante la aplicación de la heurística con el análisis realizado por un grupo de evaluadores humanos. Al comparar los resultados, se detectó que en general los valores obtenidos mediante ambos métodos son similares. Específicamente, se encontró que en los casos que el grado de conocimiento de los usuarios está más definido (mucho, bastante, poco o ningún conocimiento) la diferencia fue mínima, y se presentaron algunas diferencias en los casos que los humanos clasificaron el nivel de conocimiento como impreciso. Dichos resultados apoyarían la propuesta por lo que se planea extender la cantidad de casos de estudio y replicar los experimentos realizados. A futuro, se planea incluir un proceso de retroalimentación para extender el conjunto de frases determinantes para cada rol y trabajar en la automatización de la misma.

Agradecimientos

Este trabajo está parcialmente soportado por el subproyecto “*Reuso de Conocimiento en Foros de Discusión, Parte II*”, correspondiente al Programa de Investigación 04/F009 “*Desarrollo Orientado a Reuso, Parte II*”, de la Universidad Nacional del Comahue (Neuquén, Argentina).

Referencias

1. G. Aranda, N. Martinez, P. Faraci, and A. Cechich, “Hacia un framework de evaluación de calidad de información en foros de discusión técnicos,” in *ASSE 2013-*

- Simposio Argentino de Ingeniería de Software, JAIIO 42^o-Jornadas Argentinas de Informática*, (Córdoba, Argentina), SADIO, 2013.
2. Aranda, Gabriela, Martínez Carod, Nadina, Roger, Sandra, Faraci, Pamela, and Cechich, Alejandra, "Una herramienta para el análisis de hilos de discusión técnicos," in *CACIC 2014, XX Congreso Argentino de Ciencias de la Computación*, (San Justo, Argentina), pp. 803 – 812, Oct. 2014.
 3. N. Martínez Carod, P. Faraci, and G. N. Aranda, "Análisis de métricas de calidad en foros de discusión técnicos," in *XXIII Congreso Argentino de Ciencias de la Computación (La Plata, 2017)*, 2017.
 4. V. Zoratto, N. M. Carod, F. Otermin, and G. N. Aranda, "Análisis de estrategias para clasificar contenidos en foros de discusión," in *XXIII Congreso Argentino de Ciencias de la Computación (La Plata, 2017)*, 2017.
 5. F. Otermin, *Mejora de un recomendador de foros de discusión: Utilización de bases de datos léxicas para evaluación de sinónimos*. Licenciatura, Facultad de Informática, Universidad Nacional del Comahue, Neuquén, Argentina, Octubre 2018.
 6. G. A. Miller, "Wordnet: a lexical database for english," *Communications of the ACM*, vol. 38, no. 11, pp. 39–41, 1995.
 7. "The stanford natural language processing group." Última modificación: 16-10-2018, Último acceso: 23-07-2019.
 8. P. Faraci, *Evaluación de Calidad en Foros de Discusión Técnicos*. Licenciatura, Facultad de Informática, Universidad Nacional del Comahue, Neuquén, Argentina, Octubre 2017.
 9. C. Wohlin, P. Runeson, M. Höst, M. Ohlsson, B. Regnell, and A. Wesslén, *Experimentation in Software Engineering*. Computer Science, Springer, 2012.
 10. B. De Wever, H. Van Keer, T. Schellens, and M. Valcke, "Roles as a structuring tool in online discussion groups: The differential impact of different roles on social knowledge construction," *Computers in Human Behavior*, vol. 26, no. 4, pp. 516–523, 2010.
 11. C. N. Gunawardena, C. A. Lowe, and T. Anderson, "Analysis of a global online debate and the development of an interaction analysis model for examining social construction of knowledge in computer conferencing," *Journal of educational computing research*, vol. 17, no. 4, pp. 397–431, 1997.
 12. M. Lui and T. Baldwin, "Classifying user forum participants: Separating the gurus from the hacks, and other tales of the internet," in *Proceedings of the Australasian Language Technology Association Workshop 2010*, pp. 49–57, 2010.
 13. S. A. Golder and J. Donath, "Social roles in electronic communities," *Internet Research*, vol. 5, no. 1, pp. 19–22, 2004.
 14. J. Zhang, M. S. Ackerman, and L. Adamic, "Expertise networks in online communities: structure and algorithms," in *Proceedings of the 16th international conference on World Wide Web*, pp. 221–230, ACM, 2007.
 15. D. Fisher, M. Smith, and H. T. Welser, "You are who you talk to: Detecting roles in usenet newsgroups," in *Proceedings of the 39th Annual Hawaii International Conference on System Sciences (HICSS'06)*, vol. 3, pp. 59b–59b, IEEE, 2006.



CACIC 2019

Congreso Argentino de Ciencias de la Computación

XIV Workshop Arquitectura, Redes y Sistemas Operativos

Coordinadores

Ardenghi, Jorge (UNS)
Buckle, Carlos (UNPSJB)

Comité Científico

Acosta, Nelson (Argentina)	Fischer, Stefan (Alemania)	Molinari, Lía (Argentina)
Ardenghi, Jorge (Argentina)	García Garino, Carlos (Argentina)	Padovani, Hugo (Argentina)
Bevilacqua, Roberto (Argentina)	García Villalba, Luis Javier (España)	Steinmetz, Ralf (Alemania)
Buckle, Carlos (Argentina)	Marrone, Luis (Argentina)	Tarouco, Liane (Brasil)
Castro Lechtaler, Antonio (Argentina)	Matrángolo, Carlos (Argentina)	Ultreras, Florencio (Chile)
Díaz, F. Javier (Argentina)	Mercado, Gustavo (Argentina)	Vendrell Vidal, Eduardo (España)
	Micolini, Orlando (Argentina)	

Temas

Administración y monitoreo de redes y servicios
 Análisis y detección de vulnerabilidades
 Arquitecturas de computadoras
 Arquitecturas orientadas a servicios (SOA)
 Calidad de Servicio (QoS, SLAs)
 Comunicaciones unificadas
 Criptografía
 Infraestructura para firma digital y certificados digitales
 Infraestructura para Grid
 Middleware
 Redes activas
 Redes de comunicaciones
 Redes de nueva generación
 Redes heterogéneas
 Redes inalámbricas
 Redes móviles
 Seguridad Informática y autenticación, privacidad
 Servicios de integración (Web Services o Net)
 Sistemas embebidos
 Sistemas operativos
 Sistemas P2P

Estudio experimental de tráfico de video en redes IPv6 multicast IEEE 802.11ac

Higinio Facchini¹, Santiago Pérez¹, Alejandro Dantiacq¹, Gabriel Quiroga Salomón^{1,2},
Fabián Hidalgo¹, Gastón Cangemi¹, Mauricio Muñoz¹, Adrián Cárdenas¹

¹ CeReCoN – Departamento de Electrónica – Facultad Regional Mendoza – UTN
Rodríguez 273, Ciudad Mendoza
CP (M5502AJE) República Argentina
{higiniofac, santiagocp}@frm.utn.edu.ar

² Escuela de TICs – Universidad Nacional de Chilecito
9 de Julio 22 – Chilecito, La Rioja, Argentina
CP (5360) República Argentina
hgquiroga@undec.edu.ar

Abstract. The growing number of multimedia users involves an exponential increase in bandwidth requirements on networks. Consumers are increasingly demanding in terms of quality and performance of products based on video. This paper proposes an experiment of multicast video traffic in a network of a real laboratory as test bed, with clients working in the IEEE 802.11ac standard, and using IPv6. The videos were encoded in H.264, H.265, VP8 and Theora. And the experiments were carried out using a trailer of the movie Star Trek, and alternately for contrast, a video of a videoconference using Adobe Connect, of the same duration. This work is a continuation of experience made on wired networks for video traffic. The conclusions of the work will help to determine the configurations to be taken into account for the adequate management of similar networks and efficient use of resources available, without compromising the performance and QoS expected.

Keywords: Multicast traffic, Codecs, IEEE 802.11ac, IPv6 Networks

1 Introducción

En los últimos años, el tráfico de video ha crecido en forma exponencial, especialmente con el uso de los dispositivos móviles. Las redes inalámbricas basadas en el protocolo IEEE 802.11 (Wi-Fi) son, en la actualidad, los puntos de acceso a redes y servicios más utilizados en la mayoría de los ámbitos conocidos (laborales, académicos y de esparcimiento). Se espera que antes de 2020, los servicios y aplicaciones de video superen el 70% del tráfico total en redes de datos.

Sin embargo, para las aplicaciones exigentes, como las de video, hay que tener en cuenta la disponibilidad del ancho de banda y demás recursos compartidos de las redes Wi-Fi. Las mismas han mejorado su performance de acuerdo a las nuevas normas IEEE 802.11n e IEEE 802.11ac. Pero el incremento del tráfico de video, que

proporcionalmente representa una mayor carga para la red, y la cantidad de dispositivos móviles, supera muchas veces las mejoras obtenidas con estas tecnologías de mayores prestaciones. El uso del tipo de tráfico de multidifusión o multicast puede ser una solución interesante para acotar el impacto de una transmisión de video simultánea, a un grupo de usuarios, dado que ahorra recursos de la red, enviando un único flujo de datos igual a todos los receptores.

Además, se observa el creciente uso del protocolo de Internet en su versión 6 (IPv6), por lo que es interesante conocer el comportamiento del tráfico de video en redes Wi-Fi y, bajo este nuevo protocolo, comprender los requisitos necesarios, para ayudar a los administradores de la red a una mejor implementación.

En este trabajo se describe un experimento de tráfico de video multicast en una red de laboratorio real como test bed. Se usó una topología de redes cableadas e inalámbricas, con clientes inalámbricos trabajando en las normas IEEE 802.11ac, usando IPv6, el Software FFmpeg Server y Client, como servidor y clientes de video, y el analizador de tráfico WireShark. El tráfico de video se codificó en H.264, H.265, VP8 y Theora, para evaluar y comprender el impacto de distintos codecs sobre el tráfico de la red. Y los experimentos se realizaron usando un tráiler de video de la película Star Trek (Video 1) y, alternativamente para contraste, un video de una videoconferencia usando Adobe Connect (Video 2), de la misma duración y resolución. Este trabajo es una continuación de experimentaciones realizadas sobre redes cableadas, para tráfico de video con codecs H261, H263 y H264 en IPv4.

2 Estado del arte

En este apartado se plantea en forma resumida las principales características de los protocolos y herramientas a utilizadas.

2.1 Multidifusión IP

La multidifusión IP es una tecnología para conservar el ancho de banda, específicamente diseñada para reducir el tráfico, transmitiendo un único flujo de información a miles de destinatarios potenciales.

Los routers (enrutadores) emplean protocolos multidifusión que construyen árboles de distribución para transmitir el contenido multidifusión, que aseguran la mayor eficiencia para el envío de datos a múltiples receptores. En IPv6 se utilizan protocolos como PIM-SM, PIM-SSM u otros. Para nuestro trabajo se utilizó PIM-SM (Protocol Independent Multicast – Sparse Mode), para continuar la línea de investigación utilizada en IPv4.

2.2 Codec de video

La compresión de video es un facilitador fundamental para estas aplicaciones. Existe un número importante de algoritmos o codecs, ya sean estándares o de propiedad exclusiva, para que sea práctico almacenar y transmitir video en formato digital. La compresión a menudo implica compromisos entre los requisitos de calidad de imagen y otras necesidades de la aplicación.

Los Codecs de video utilizados en este trabajo son:

- H.264/MPEG-4 AVC: Es una norma promovida conjuntamente por la UIT y la ISO, que ofrece un gran avance significativo en la eficiencia de compresión para lograr una reducción de alrededor de 2 veces en la velocidad de bits en comparación con MPEG-2 y MPEG-4 de perfil simple.
- H.265/ MPEG-H Parte2/ High Efficiency Video Coding (HEVC): Define un formato de compresión de video, sucesor de H.264/MPEG-4 AVC, desarrollado conjuntamente por la ISO/IEC Moving Picture Experts Group (MPEG) y ITU-T Video Coding Experts Group (VCEG), como ISO/IEC CD 23008-2 High Efficiency Video Coding. Este estándar puede utilizarse para proporcionar mejor calidad de videos de bajo bitrate con la misma tasa de datos. Es compatible con la televisión en ultra-alta definición y resoluciones hasta 8192x4320
- VP8: Es un códec de video de On2 Technologies anunciado el 13 de septiembre de 2008. El 19 de mayo de 2010, Google, que adquirió On2 Technologies en 2009, liberó el códec VP8 como código abierto (bajo una licencia permisiva similar a la licencia BSD).
- Theora: Es un códec de video libre desarrollado por la Fundación Xiph.Org, como parte de su proyecto Ogg. Basado en el códec VP3. Google, en 2010, empezó a financiar parte del proyecto de Ogg Theora Vorbis. Theora es un códec de video de propósito general con bajo consumo de CPU.

2.3 Flujo (Streaming) de video

Existe una gama muy diversa de transmisiones de video, que tienen diferentes propiedades y/o condiciones de funcionamiento. Por ejemplo, las aplicaciones de transmisión de video pueden ser tipo punto a punto, de multidifusión, o difusión, y el video puede estar pre-codificado (almacenado), o puede codificarse en tiempo real (por ejemplo, con videoteléfono interactivo o en videoconferencias). Los canales de video para la transmisión también pueden ser estáticos o dinámicos, con conmutación de paquetes o de circuitos, pueden soportar una velocidad de transmisión de bits constante o variable, y se puede considerar alguna forma de Calidad de Servicio (QoS), o sólo puede proporcionar tráfico de mejor esfuerzo.

Hay una serie de problemas básicos que afectan a la transmisión de video a través de una red, ya que, en general, sólo se ofrece servicio de mejor esfuerzo. Por lo tanto, un objetivo clave de la transmisión de video es diseñar un sistema fiable para entregar video de alta calidad a través de una red, teniendo en cuenta los siguientes aspectos: ancho de banda, variación de retardo (jitter) y tasa de pérdidas de paquetes o tramas.

2.4 Protocolo IEEE 802.11ac

La norma IEEE 802.11ac ofrece velocidades de datos de hasta 1.3 Gigabits por segundo (Gbps), lo que permite a los dispositivos manejar aplicaciones exigentes como Ultra HD y video 4K, transmisión multimedia y transferencia rápida de archivos en tabletas, en los dispositivos de juegos, en los teléfonos y muchos otros dispositivos. Además, proporciona una experiencia de movilidad de mayor rendimiento. Las principales características son: la comunicación tipo MIMO (multiple entrada-múltiple salida) para múltiples usuarios y mayor ancho de banda.

2.5 Protocolo de Internet versión 6 – Ipv6

IPv6 es una evolución de IPv4. El protocolo se instala como una actualización de software, en la mayoría de los dispositivos y sistemas operativos. Las estadísticas muestran que a principios de 2011 (cuando se agotó el grupo IANA IPv4), el porcentaje de usuarios nativos de Internet IPv6 era de, aproximadamente, 0.2%. En noviembre de 2018 se estaba en el 22.25%. La cantidad de usuarios IPv6 se ha estado duplicando, aproximadamente, cada nueve meses. Las principales diferencias con respecto a IPv4 son: espacio de direccionamiento extendido, autoconfiguración, simplificación del encabezado y soporte mejorado para opciones y extensiones.

3 Escenarios y recursos experimentales

Los trabajos de investigación, relacionados con la temática, muestran, en general, una falta de uniformidad de los escenarios de experimentación, en la cantidad y tipos de codecs, en los videos utilizados, etc. Estos y otros aspectos complican, en su conjunto, los contrastes entre trabajos contemporáneos entre sí, y los realizados previamente. Además, se observa que hay pocos trabajos disponibles con estudios experimentales y de simulación de redes Wi-Fi, como propuestas que combinen en un laboratorio de prueba las limitaciones de estas redes, la problemática de las variantes de los codecs para tráfico de vídeo, y el uso de la multidifusión IPv6 con las normas IEEE 802.11ac.

3.1 Protocolo de Internet versión 6 – Ipv6

La topología genérica propuesta incluye un servidor de streaming y dispositivos móviles y PCs de escritorio con conectividad Wi-Fi, como clientes conectados en los extremos de una red. Dicha red está formada por routers y switches (conmutadores) con distintos tipos de enlaces interconectando a los mismos. La Figura 1 muestra la topología de trabajo.

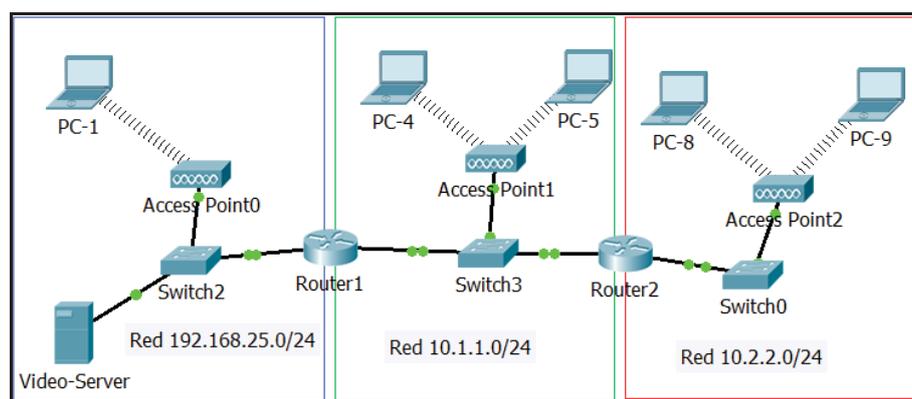


Fig. 1. Topología de red utilizada

En esta topología, los enlaces indicados con línea continua son del tipo FastEthernet con una velocidad de transmisión de 100 Mbps, mientras que los dispositivos clientes se conectan bajo norma 802.11ac. Para el funcionamiento entre los routers se configuró el protocolo de enrutamiento unicast OSPFv2, y el protocolo de enrutamiento multicast PIM-SM. Los routers utilizados son Cisco 2811, y los APs fueron Cisco Linksys LAPAC 1200. El software utilizado como servidor de streaming y los clientes receptores estaban basados en FFmpeg.

3.2 Video utilizados

Se utilizaron dos archivos de video codificados con los codecs mencionados previamente, en la modalidad on demand. Un video es un tráiler de la película Star Trek, y el otro es un segmento de una videoconferencia, usando el software Adobe Connect, con la misma duración y calidad que el video anterior. Las Tablas 1 y 2 muestran la comparación de las características propias de cada códec para cada video.

3.3 Metodología de trabajo

Se realizaron una serie de experimentos con las siguientes consideraciones y secuencia de pasos comunes para cada video:

- a) Se configuraron los archivos de video codificados en los 4 formatos mencionados en el servidor de streaming,
- b) Antes de comenzar las mediciones se sincronizaron en tiempo todos los equipos involucrados en la topología, usando un servidor NTP local.
- c) Desde el servidor se envió el archivo del primer video en un codec en particular a la red en formato multidifusión (multicast).
- d) Posteriormente, se cambió el formato de video a los otros tres codecs de video, repitiendo el paso c.

Durante cada ensayo, las mediciones se realizaron capturando tráfico en el servidor y en cada una de las PCs conectadas por Wi-Fi, usando el software sniffer Wireshark [14]. Con este software se generó un archivo con la captura de todas las tramas o paquetes de Capa 2, para cada ensayo, tanto del server como de las PCs. En estos archivos se dispuso de toda la información de cada una de las tramas capturadas durante el ensayo, como el día y hora exacta de la captura de cada trama, las direcciones MAC e IP origen y destino, el protocolo de capa de transporte y/o de aplicación, el tamaño de trama, etc.

3.4 Métricas

En base a la experimentación realizada, se obtuvieron una serie de métricas, tanto para el servidor como para los clientes. Se construyeron tablas con datos individuales netos para cada dispositivo y, luego, se determinaron los promedios de las mediciones. Las métricas obtenidas fueron: Tiempo total del vídeo [Tt], Número total de tramas o paquetes [NP], Número total de bytes [NB], Tamaño promedio de los paquetes [Tp], Espacio intertrama o Tiempo entre tramas o paquetes [EI], y Tasa de transferencia efectiva [BR]. Además, se registraron otras métricas y mediciones,

como: Retardos de tiempo involucrados en la transmisión [Rt], Diferencia de retardo (o jitter) [DR], Número de errores obtenidos [Eo], y las Distribuciones estadísticas.

Tabla 1. Propiedades del Video 1 – Trailer Star Trek

Video 1	H.264	H.265	Theora	VP8
<i>Format</i>	MPEG-4	MPEG-4	Ogg	WebM v2
<i>File size</i>	79,9 MiB	72,3 MiB	83,3 MiB	78,6 MiB
<i>Duration</i>	2 min 11 s			
<i>Bit rate mode</i>	Variable	Variable	Variable	Variable
<i>Bit rate</i>	5 109 kb/s	4 620 kb/s	5 329 kb/s	5 028 kb/s
Video				
<i>Format</i>	AVC	HEVC	Theora	VP8
<i>Bit rate</i>	5 011 kb/s	4 514 kb/s	5 010 kb/s	4 721 kb/s
<i>Width [pixeles]</i>	1 280 pixeles	1 280 pixeles	1 280 pixeles	1 280 pixeles
<i>High [pixeles]</i>	528 pixeles	528 pixeles	528 pixeles	528 pixeles
<i>Aspect ratio</i>	2,4:1	2,4:1	2,4:1	2,4:1
<i>Frame rate mode</i>	constant	constant	constant	constant
<i>Frame rate [fps]</i>	23,976 FPS	23,976 FPS	23,976 FPS	23,976 FPS
<i>Bits/(pixel*frame)</i>	0.309	0.279	0.309	0.291
Audio				
<i>Format</i>	AAC LC	AAC LC	Vorbis	Vorbis
<i>Bit rate mode</i>	Variable	constant	Variable	Variable
<i>Bit rate</i>	98,7 kb/s	99,7 kb/s	98,7 kb/s	98,7 kb/s
<i>Maximum bit rate</i>	167 kb/s	167 kb/s	167 kb/s	167 kb/s
<i>Channel</i>	2 canales	2 canales	2 canales	2 canales
<i>Sampling rate</i>	44,1 kHz	44,1 kHz	44,1 kHz	44,1 kHz
<i>Track size</i>	1,54 MiB (2%)	1,56 MiB (2%)	1,54 MiB (2%)	1,54 MiB (2%)

Tabla 2. Propiedades del video 2 – Videoconferencia Adobe

Video 2	H.264	H.265	Theora	VP8
<i>Format</i>	MPEG-4	MPEG-4	Ogg	WebM - v2
<i>File size</i>	6,02 MiB	6,10 MiB	8,84 MiB	11,5 MiB
<i>Duration</i>	2 min 11 s	2 min 11 s	2 min 11 s	2 min 11 s
<i>Bit rate mode</i>	Variable	Variable	Variable	Variable
<i>Bit rate</i>	385 kb/s	390 kb/s	565 kb/s	733 kb/s
Video				
<i>Format</i>	AVC	HEVC	Theora	VP8
<i>Bit rate</i>	256 kb/s	256 kb/s	407 kb/s	568 kb/s
<i>Width [pixeles]</i>	1 280 pixeles	1280 pixeles	1 280 pixeles	1 280 pixeles
<i>High [pixeles]</i>	720 pixeles	720 pixeles	720 pixeles	720 pixeles
<i>Aspect ratio</i>	16:9	16:9	16:9	16:9
<i>Frame rate mode</i>	constant	constant	constant	constant
<i>Frame rate [fps]</i>	30,000 FPS	30,000 FPS	30,000 FPS	30,000 FPS
<i>Bits/(pixel*frame)</i>	0.009	0.009	0.015	0.021
Audio				
<i>Format</i>	AAC LC	AAC LC	Vorbis	Vorbis
<i>Bit rate mode</i>	Variable	Variable	Variable	Variable
<i>Bit rate</i>	126 kb/s	126 kb/s	127 kb/s	127 kb/s
<i>Maximum bit rate</i>	257 kb/s	127 kb/s	257 kb/s	257 kb/s
<i>Channel</i>	2 canales	2 canales	2 canales	2 canales
<i>Sampling rate</i>	44,1 kHz	44,1 kHz	44,1 kHz	44,1 kHz
<i>Track size</i>	43,066 FPS	43,066 FPS	1,99 MiB (22%)	1,99 MiB (17%)

4 Resultados de la experimentación

En la Tabla 3, y como resumen, se muestran los datos netos promedios para las más importantes métricas definidas previamente. Dado que se trabajó en una topología de laboratorio controlada, no hubo errores de transmisión ni tramas perdidas en forma significativa. Se observa que existe una alta dependencia sobre las características del tipo de video transmitido, y del códec utilizado. La videoconferencia representa una carga de tráfico del orden del 80-90% menor que una película de iguales características de resolución y, por lo tanto, tendrá una menor cantidad de tramas, un mayor espacio intertramas promedio o una menor tasa de bits, en similar proporción. Además, se observa que para el video 1, el promedio de los codecs, requiere 94.000 tramas para transmitir el trailer. Mientras que para el video 2, el promedio de los codecs demanda 8.220 tramas para transmitir la videoconferencia de la misma duración. Es decir, casi 90% menos. Dado que los videos tienen la misma duración es de esperar que para el video 1 el espacio intertrama sea sustancialmente menor. Efectivamente, el tiempo intertrama promedio de los codecs para el video 1 es de 1,445 mseg, mientras que para el video 2 es de 15,7345 mseg. Aproximadamente un 90% más.

Por cuestiones de espacio, se presenta con más detalle el comportamiento de la compresión usando la codificación H265. Las Figuras 2 y 3 muestran la distribución de los tamaños de las tramas y de los espacios intertramas para el video 1, respectivamente. En la Figura 2 se observa una alta concentración de tramas, del orden de las 44.000 tramas para cada caso, para una longitud de 95 bytes y de 1510 bytes (la casi totalidad de las tramas). Mientras que en la Figura 3 se destaca que casi el 90% de las tramas tiene un espacio intertrama por debajo de los 5 mseg.

Las Figuras 4 y 5 presentan la distribución de los tamaños de las tramas y de los espacios intertramas para el video 2, respectivamente. En la Figura 4 se observa una distribución mayor de las tramas según el tamaño; un 26 % corresponden a las tramas de más de 1500 bytes, otro 25 % en el orden de los 1150 bytes, y un 35 % alrededor de los 200 bytes. En la Figura 5 se observa que un 32 % de las tramas tiene un espacio intertrama del orden de un mseg, un 35% alrededor del 20 mseg, un 15 % en los 30 mseg, y el resto se distribuye mayoritariamente entre los 10, 41 y 50 mseg.

Tabla 3. Métricas promedio

Codec	Número de paquetes NP	Tiempo intertrama promedio EI [s]	Número de bytes NB [Mbytes]	Tamaño de paquete promedio Tp [bytes]	Tasa de bits BR [Mbits/s]
Video 1 – Trailer Star Trek					
<i>H264</i>	102659	0,00128	82,80746	806,62	5,04901
<i>H265</i>	87903	0,00148	70,85176	806,02	4,34782
<i>Theora</i>	75218	0,00174	60,37944	802,72	3,69813
<i>VP8</i>	110202	0,00128	82,28519	802,38	5,01721
Video 2 – Videoconferencia Adobe					
<i>H264</i>	8067	0,015736	6,908887	856,44	0,421429
<i>H265</i>	7943	0,015971	6,910678	870,03	0,421507
<i>Theora</i>	6782	0,018576	9,785508	1442,86	0,596617
<i>VP8</i>	10090	0,012655	12,761038	1264,72	0,778314

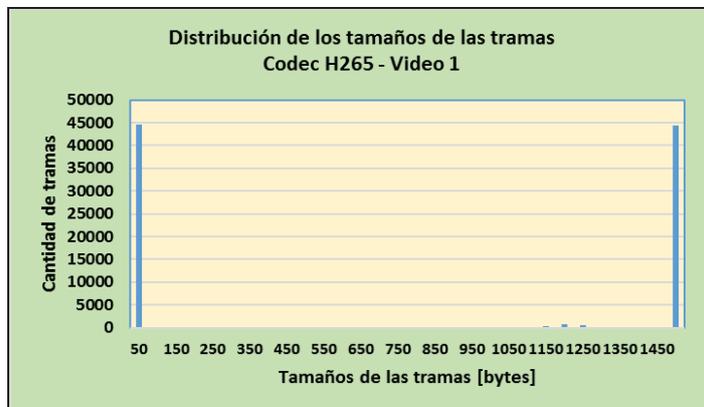


Fig. 2. Distribución de tamaños de tramas del video 1 usando el codec H265

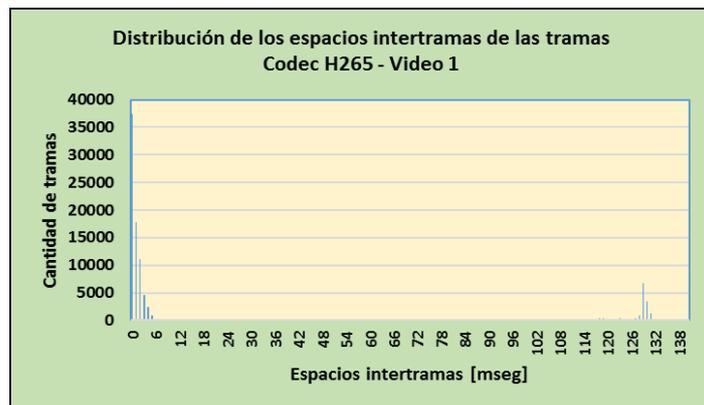


Fig. 3. Distribución de espacios intertramas del video 1 usando el codec H265

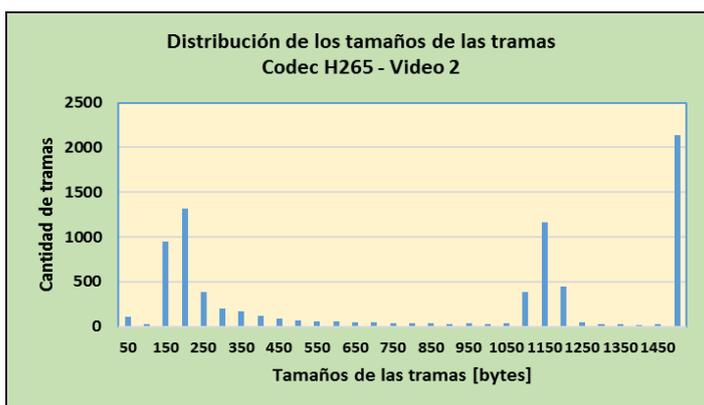


Fig. 4. Distribución de tamaños de tramas del video 2 usando el codec H265

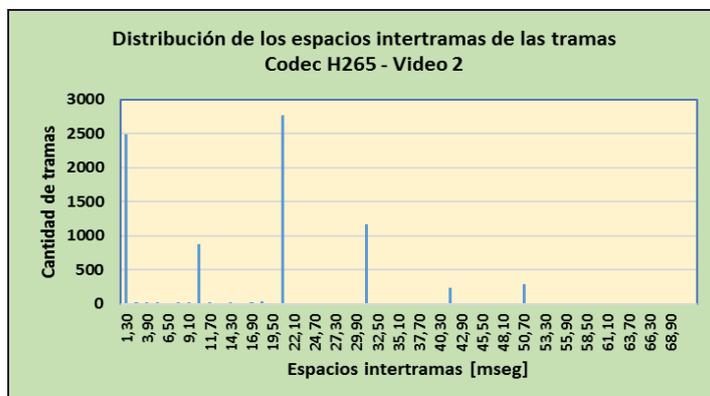


Fig. 5. Distribución de espacios intertramas del video 2 usando el codec H265

5 Conclusiones

La más importante motivación para este trabajo fue conocer el comportamiento del tráfico de video en redes Wi-Fi, usando el protocolo IPv6, y comprender los requisitos necesarios de la red contrastando dos casos claramente diferenciados: el tráiler de una película y el video de una videoconferencia. Estos casos de estudio ayudarán a los administradores, diseñadores, planificaciones, analistas y simuladores del tráfico de las redes para una mejor implementación.

Utilizando el escenario detallado, se realizaron una serie de experimentos. Entre los resultados se encuentran los valores de las métricas directas y promedio para cada uno de los 4 codecs utilizados para el Video 1 y 2.

Los Videos en comparación presentan características similares en cuanto a la duración y a la resolución, pero son bastante diferentes en otros aspectos (dinámica, peso en Mbytes del archivo, etc.).

Se concluyó cuantitativamente que existe una alta dependencia sobre las características del tipo de video transmitido (película o videoconferencia), y en menor medida del códec utilizado. Además, que la topología de la red y el tipo de equipamiento no influye notoriamente, ya que para equipos clientes ubicados en distintos nodos de la red, el comportamiento es prácticamente idéntico

Esta comparación, a nuestro entender, es muy útil para comprender el conocimiento que debe tenerse en cuenta en la gestión de la red Wi-Fi, que posea tráfico de video. Evidentemente, existe una gran diversidad de carga en la red, dependiendo del tipo de video. Esta diversidad debe tenerse en cuenta en la planificación de la red. Además, muchos analistas pueden usar esta información para la parametrización de sus estudios de simulación.

Referencias

1. Cisco Mobile VNI, 2017, "Cisco Visual Networking Index: Global Mobile Data Traffic Forecast Update, 2016–2021 White Paper, [online] Available: <https://www.cisco.com/c/en/us/solutions/collateral/service-provider/visual-networking-index-vni/mobile-white-paper-c11-520862.html>
2. Santiago Pérez, Javier Campos, Higinio Facchini and Alejandro Dantiacq, "Experimental study of unicast and multicast video traffic using WAN test bed", [online] Available: <https://ieeexplore.ieee.org/document/7585260>.
3. Santiago Pérez, Luis Marrone, Higinio Facchini and Fabian Hidalgo, "Experimental Study of Multicast and Unicast Video Traffic in WAN Links", IEEE Latin American Transactions, Vol. 15, issue 10, ISSN: 1548-0992, pp. 1847-1855, octubre 2017.
4. John G. Apostolopoulos, Wai-tian Tan, Susie J. Wee - Mobile and Media Systems Laboratory HP Laboratories Palo Alto - Video Streaming: Concepts, Algorithms, and Systems - [online] Available: <http://www.hpl.hp.com/techreports/2002/HPL-2002-260.pdf>.
5. A. Abdalla, A. Mazhar, M. Salah and Sahar Khalaf, "Comparative Study of Compression Techniques for Synthetic Videos," The International Journal of Multimedia & Its Applications (IJMA), vol. 6, no.2, April 2014.
6. D. Grois, D. Marpea, T. Nguyena and O. Hadarb, "Comparative Assessment of H.265/MPEG-HEVC, VP9, and H.264/MPEG-AVC Encoders for Low-Delay Video Applications", SPIE Proceedings, vol. 9217, Applications of Digital Image Processing XXXVII, California, USA, September, 2014
7. Y. Sun, I. Sheriff, E. Belding-Royer and K. Almeroth, "An Experimental Study of Multimedia Traffic Performance in Mesh Networks," Workshop on Wireless Traffic Measurements and Modeling, pp.25-30, Seattle, Washington, USA, June, 2005.
8. R. Addu and V. Potuvaranam, "Effect of Codec Performance on Video QoE for videos encoded with Xvid, H.264 and WebM/VP8," Master Thesis, Blekinge Institute of Technology, Karlskrona, Sweden, August 2014
9. Y. Shin, M. Choi, J. Koo, Y. Kim, J. Ihm and S. Choi, "Empirical Analysis of Video Multicast over WiFi," Third International Conference on Ubiquitous and Future Networks (ICUFN), pp. 381-386, China, 2011
10. S. Tanwir, H. Perros and B. Anjum, "A QoS evaluation of video traffic models for H.264 AVC video," Fifth International Conference on Next Generation Networks and Services (NGNS), pp. 313-320, Casablanca, Morocco, May, 2014
11. Available: <https://www.ffmpeg.org/>
12. Video 1, Star Trek, unpublished Available: https://www.youtube.com/watch?v=iLbnDWS3f_g
13. Video 2, Videoconference Adobe Connect, Available: https://www.youtube.com/watch?v=iLbnDWS3f_g
14. Available: <https://www.wireshark.org/>

Evaluación de Convergencia del Protocolo OSPF en Redes Definidas por Software

Juan Ángel González¹, Fabio López Pires²

¹ Facultad de Politécnica. Universidad Nacional de Asunción

² Parque Tecnológico Itaipu
Paraguay

angelo_py@hotmail.com, fabio.lopez@pti.org.py

Resumen. El protocolo *Open Shortest Path First* (OSPF) es el protocolo de enrutamiento más utilizado en la actualidad. Las Redes Definidas por *Software* permiten implementar nuevos enfoques en el flujo de tráfico y enrutamiento. Este trabajo presenta una evaluación experimental del tiempo de convergencia del protocolo OSPF en Redes Tradicionales en comparación con Redes Definidas por *Software*, simulando diferentes condiciones de fallas simples y dobles en una red de datos, considerando una topología en anillo, con 5 nodos terminales, 5 enrutadores y 11 enlaces. Los resultados experimentales indican que en promedio, el tiempo de convergencia en las Redes Definidas por *Software* es hasta 82% menor ante fallas simples y hasta 78% ante fallas dobles.

Palabras Claves: Redes Definidas por *Software*, Enrutamiento Dinámico, Evaluación Experimental, Tiempo de Convergencia, OSPF.

1 Introducción

EL auge de las Tecnologías de la Información y de Comunicación (TIC), así como la expansión de las aplicaciones de banda ancha imponen estrictos requerimientos en términos de rendimiento de equipos y protocolos de enrutamiento en redes de datos, las cuales deben contar con rutas que garanticen el alcance a todos los nodos destinos, de manera a adaptarse a posibles cambios de topología, producidos por fallas en los enlaces o por flujo intenso de tráfico. Los enrutadores representan el punto clave en la infraestructura de red, debido a que a través de ellos, son interconectadas las redes y los protocolos de enrutamiento del tráfico son tradicionalmente ejecutados en los mismos.

Una de las características de un protocolo de enrutamiento, es su impacto en el rendimiento de extremo a extremo, adaptándose a cambios de topología cuando éstos ocurren. El factor de rendimiento que distingue a diferentes protocolos de enrutamiento es el **tiempo de convergencia** o tiempo de restablecimiento del enlace ante una falla o corte del mismo [1]. Cuanto menor es el **tiempo de convergencia**, se puede considerar que mejor es el protocolo de enrutamiento.

El objetivo de este trabajo es evaluar experimentalmente el **tiempo de convergencia** del protocolo OSPF en redes SDN en comparación con redes tradicionales, considerando diferentes escenarios de prueba. El resto de este trabajo está organizado de la siguiente manera: la Sección II, presenta una breve descripción del protocolo OSPF, en la Sección III, se presenta los trabajos relacionados más relevantes, mientras que la Sección IV describe el entorno experimental, propone la arquitectura y topología de ambos esquemas evaluados. La Sección V resume los resultados experimentales. Finalmente, las conclusiones y trabajos futuros son presentados en la Sección VI.

2 Protocolo *Open Shortest Path First* (OSPF)

El *Open Shortest Path First* (OSPF), es actualmente el protocolo de enrutamiento más utilizado en redes de datos, ya que ofrece simplicidad, adaptabilidad y buen rendimiento ante cambios de topología [3]. En las redes de datos tradicionales, los protocolos de enrutamiento se ejecutan en cada enrutador, donde se administran el plano de datos y el plano de control.

Cuando ocurren fallas de enlaces de red, la topología lógica existente en cada enrutador debe actualizarse, generando así tráfico de control que inunda la red, mediante los anuncios *Link State Announcement* (LSA). Esto ocurre constantemente de manera a mantener las tablas de rutas siempre actualizadas (ver Fig. 1).

Con la aparición de las *Software Defined Networks* (SDN) [10], el modelo de actualización de las tablas de rutas son más ágiles, ya que las SDN separan el plano de datos y el plano de control, gestionando ambos planos por separado.

Este nuevo modelo de gestión, incorpora al controlador, el nodo principal de la red, responsable de administrar y mantener comunicación constante con todos los conmutadores de la red, obteniendo así una visión centralizada y global de la red. [10].

Los diferentes tipos de mensajes que intercambian los enrutadores, son gestionados por el controlador, ya que trata estos mensajes como flujos de datos en una red SDN, cada enrutador necesita intercambiar información con otros enrutadores de manera a mantener actualizada la tabla de rutas, en las SDN éstos mensajes son captados por los conmutadores y lo envían al controlador. De ésta manera el controlador recibe todos los mensajes, actualiza la tabla de rutas y la topología lógica de la red.

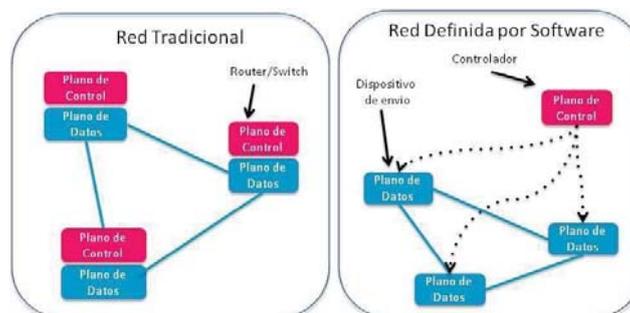


Fig. 1. Comparativo de Redes Tradicionales y SDN.

3 Trabajos Relacionados

En esta sección se presentan varias técnicas y métodos ejecutados para determinar el **tiempo de convergencia** y el rendimiento del protocolo OSPF en redes tradicionales y redes definidas por software los cuales son presentados en trabajos relacionados [1].

El **tiempo de convergencia** del protocolo OSPF es directamente influenciado por los tiempos de envío del paquete de saludo (*HELLO*) y el tiempo de cálculo del algoritmo de *Shortest Path First* (SPF), que son ejecutados en cada enrutador [1]. Estos parámetros pueden ser ajustados en cada enrutador de manera a obtener el menor tiempo posible de ejecución. Por ejemplo, el paquete de saludo (*HELLO*) es enviado periódicamente entre los enrutadores vecinos con la frecuencia establecida en el enrutador por el Intervalo de Saludo (*Hello-Interval*) [1]. Otros parámetros de rendimiento, pueden ser ajustados en el enrutador, de manera a mejorar el **tiempo de convergencia** del protocolo OSPF, teniendo en cuenta diferentes escenarios de fallas.

El enrutamiento en Redes Definidas por *Software* incluye un modelo de enrutamiento centralizado, el descubrimiento de la topología y el cálculo de las rutas son finalizados en el controlador [2]. El modelo de enrutamiento centralizado nos brinda mayor flexibilidad, debido a que nos ofrece condiciones más favorables para la optimización de la red [2]. La convergencia, representa el proceso de sincronización de las tablas de enrutamiento después de un cambio en la topología. La cantidad de tiempo es definido como “Intervalo” y es basado en el máximo tiempo esperado en la estabilización de la red después de un cambio en la topología.

La Ecuación (1) es aplicada a redes tradicionales y muestra que el tiempo de entrega de los mensajes LSA [2] y la ejecución del Algoritmo SPF [1], son factores claves en el **tiempo de convergencia** [1]. De acuerdo al tamaño de la red y retardo de los mensajes LSA [2], influyen directamente al **tiempo de convergencia** [1].

$$B + C + D + E = TCRT \quad (1)$$

dónde:

B = tiempo de descubrimiento del corte;

C = tiempo de entrega de mensajes LSA;

D = tiempo de ejecución del Algoritmo SPF;

E = tiempo de actualización de la tabla de rutas.

TCRT = tiempo de convergencia en Red Tradicional

La Ecuación (2) aplicada a redes SDN [10], muestra que el factor clave para el **tiempo de convergencia** [1], incluye el tiempo de actualización de la topología, el tiempo de ejecución del Algoritmo de SPF [1], y el tiempo de actualización de la tabla del flujo de rutas. Típicamente el controlador en las redes SDN [10], disponen de un enlace de alta velocidad con los conmutadores y esto mejora ampliamente el **tiempo de convergencia** [1].

La reproducción de video en tiempo real, exige de la red alto rendimiento y excelente calidad servicio QoS (*Quality of Service*), éstas métricas son influenciadas por el retardo, la pérdida de paquetes y la fluctuación del retardo [3].

$$B + F + D + E = TCSDN \quad (2)$$

dónde:

F = tiempo de actualización de la topología.

$TCSDN$ = tiempo de convergencia en SDN

En [1] se analiza el tiempo de convergencia de rutas en un entorno únicamente SDN, compuesto por *Mininet* y *Quagga*, con una topología *mesh* con 4 *routers* y 4 *hosts*. Se realizaron 10 experimentos por cada tipo de falla, optimizando algunos parámetros de rendimiento como el Intervalo de Saludo y el Tiempo de Ejecución del Algoritmo *SPF Delay*.

En [2] se analiza principalmente el tiempo de convergencia del protocolo OSPF, en redes tradicionales simuladas y redes SDN. Los entornos son simulados con varias máquinas virtuales VMWARE utilizando *Mininet* y *Quagga*. Disponen 2 topologías de red, una de 16 nodos y otra de 120 nodos, ambos con topología *mesh*. Se realizaron experimentos bajo diferentes parámetros de enlace, optimizando algunos parámetros de rendimiento del protocolo OSPF.

En [3] se analiza el tiempo de convergencia y calidad de servicios cuando se transmite tráfico multimedia sobre una red tradicional y una SDN. El entorno SDN está compuesto por *Mininet* y *Quagga*, con una topología en anillo, 4 máquinas físicas, interconectadas mediante túneles GRE, simulando máquinas virtuales en cada una de ellas, donde se tiene 4 *routers* y 6 *hosts*. El entorno tradicional, compuesto de *routers*, *switches* Cisco y PCs. Se realizaron experimentos a fin de determinar el rendimiento del protocolo OSPF en ambos entornos experimentales.

4 Entorno Experimental

En esta sección se describe la topología que se utilizarán en las pruebas experimentales, pero antes de describirlo, se expondrán tres subsecciones. La primera describe la arquitectura, plataforma y *software* utilizado para simular una red tradicional en un entorno virtualizado. La segunda describe la arquitectura de las redes SDN, mientras que la tercera describe la topología a ser simulada en ambos esquemas.

Las diferencias fundamentales de este entorno de trabajo, en comparación con los trabajos relacionados, son (1) topología de red y cantidad de nodos y (2) simulación de la red tradicional mediante virtualización de enrutadores, enlaces y nodos.

4.1 Simulación de Redes Tradicionales

De manera a contar con una infraestructura y plataforma que simule una red tradicional, se ha montado un servicio de simulación de redes virtuales Linux, sobre

plataforma Linux conocida como Virtual Network – User Mode Linux (VNUML) [8].

Esta plataforma simula un servidor Linux con las siguientes características: memoria base de 4Mb, memoria de video de 128 Mb, 2 discos virtuales de 16 Gb y de 8 Gb, soporte de red, audio y USB, con soporte VirtualBox. Utilizando ésta plataforma se ha construido el entorno virtual para simular una red tradicional, del cual obtenemos los siguientes beneficios: el usuario puede crear su propio escenario, varias máquinas virtuales pueden correr en una sola pc, compartición de configuraciones comunes entre máquinas virtuales, muy ligero y opera solo en modo consola.

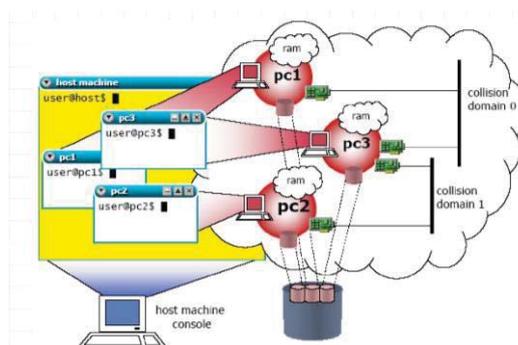


Fig. 2. Ejemplo de Simulación de Redes con VNUML.

Cada máquina virtual VNUML, puede desempeñar roles como host, enrutador entre otros, de manera a simular una red tradicional, ciertas máquinas virtuales deben ejecutar el rol host y otras como router. Asignar el rol de enrutador a una máquina virtual implica que esa máquina virtual deberá contar con procesos de enrutamiento, lectura, reenvío de paquetes, administración de mensajes de control, actualización de tablas de enrutamiento, entre otros, que son tareas inherentes a un enrutador en las redes tradicionales.

Quagga [6], es un conjunto de herramientas que permiten implementar en un sistema Linux los protocolos de enrutamiento (OSPF), (RIP) y (BGP), gestionando, para ello, la tabla de enrutamiento del propio núcleo del sistema.

Quagga [6] permite asignar el rol de enrutador, mediante la activación de demonios de enrutamiento y así poder intercambiar información de enrutamiento con otras máquinas del mismo rol. Quagga [6], puede configurar la interfaz de cada máquina virtual con direcciones, banderas, rutas estáticas y otros.

4.2 Redes Definidas por Software

En la Sección II se ha descrito como las SDN administran los mensajes del protocolo OSPF, en la Fig. 3 se puede ver la separación de las capas de la arquitectura SDN y su interconexión mediante una interfaz como **OpenFlow**, que comunica la capa de control con la capa de infraestructura. **OpenFlow**, es una interface de comunicaciones definidas entre la Capa de Control con la Capa de Infraestructura (Datos) en una arquitectura SDN. Admite el acceso directo a la manipulación de datos presente en los dispositivos por ejemplo, *switches*, enrutadores, entre otros [14].

OpenFlow utiliza el concepto de flujos para identificar el tráfico de red basado en reglas predefinidas que pueden ser programadas de forma estática o dinámica por el controlador SDN, lo cual permite definir como debe fluir el tráfico hacia los dispositivos de red [14].

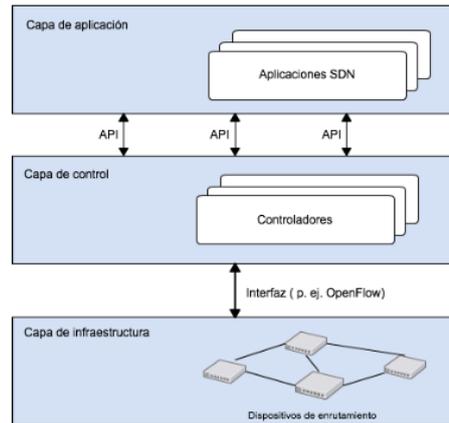


Fig. 3. Modelo de Arquitectura SDN.

4.3 Topología para Pruebas Experimentales

Se ha diseñado una topología que será simulada en ambos esquemas, presentados en las sub-secciones anteriores. A cada máquina virtual se han asignado roles específicos. Las máquinas virtuales con rol de router, utilizan el software de enrutamiento Quagga [6], y con el proceso activo OSPFD, de manera a utilizar el protocolo de enrutamiento dinámico OSPF.

En la simulación para redes tradicionales, cada máquina virtual desempeña un rol específico, el de *host* y *router*. La topología fue diseñada mediante el VNUML de Linux, con enlaces virtuales a los nodos vecinos [8]. Todos los *routers* están configurados en el área "0". La Fig. 4 representa gráficamente la topología de la red tradicional.

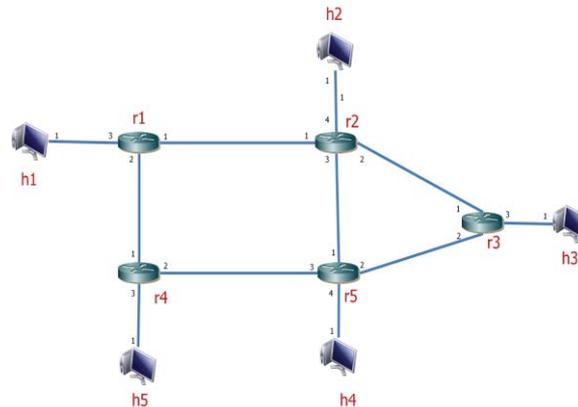


Fig. 4. Topología de red tradicional considerada en pruebas experimentales.

En la simulación para SDN, la topología fue diseñada con el software Mininet [4], que es una herramienta open-source de simulación de redes SDN, que permite crear host, switches, controladores y enlaces virtuales. Mininet corre sobre el estándar Linux, en este caso en particular, se ha instalado sobre Ubuntu v16.01, cada nodo de la red, es un switch virtual (Open vSwitch) y soportan **OpenFlow** y se han aplicado los mismos roles asignados en el modelo tradicional [14].

Todos los switches de la red son conectados al **controlador**, elemento central de las redes SDN. De manera a completar el entorno SDN [10], se ha utilizado una versión extendida de la herramienta Mininet denominada MiniNExT (Mininet ExTended) [12], que incluye una capa de extensión para facilitar la creación de redes complejas [4]. El motor de enrutamiento para cada nodo MiniNext es software de enrutamiento Quagga [6], en el cual se ha configurado y activado el proceso OSPFD de manera a utilizar el protocolo OSPF, similar al modelo tradicional.

Para la simulación en ambos esquemas, se ha implementado el protocolo IPv4, con dirección de red 10.10.0.0/24 y se han asignado direcciones IP a cada interface de la red propuesta.

5 Resultados Experimentales

En ésta sección se resumen los resultados de las pruebas experimentales, focalizadas en el **tiempo de convergencia** del protocolo OSPF, tanto en redes tradicionales como en redes SDN, bajo diferentes condiciones y parámetros que influyen en el rendimiento del protocolo OSPF.

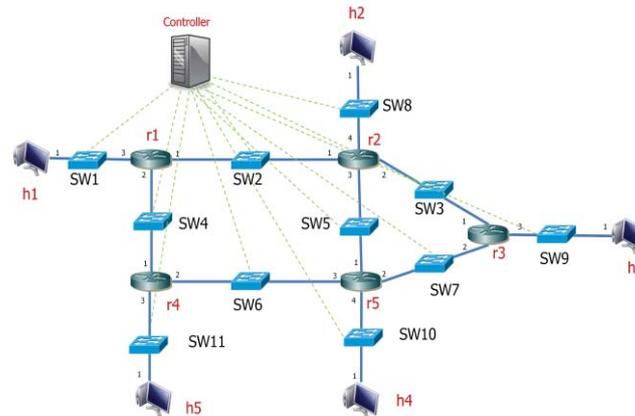


Fig. 5. Topología de la red SDN

5.1 Parámetros de rendimiento del protocolo OSPF

El protocolo OSPF es influenciado por diferentes parámetros de rendimiento, entre los cuales se pueden citar:

- **Intervalo de Saludo (*Hello Interval*):** Cuando un enrutador inicia un proceso de enrutamiento OSPF envía un paquete de saludo (*Hello*) que contiene información acerca de las redes conectadas al enrutador y sigue enviando paquetes de saludo (*Hello*) a intervalos regulares. El valor por defecto del intervalo de saludo es de 10 segundos [1].
- **Algoritmo SPF:** El algoritmo SPF, representa una gran exigencia para la CPU del enrutador y el tiempo que lleva realizar los cálculos depende del tamaño del área. Si en la red hay un enlace inestable, que alterna entre los estados up y down, esto causa que los enrutadores OSPF de un área estén ejecutando constantemente el algoritmo para actualizar el árbol SPF, lo que dificulta una convergencia adecuada del enrutamiento. Esto se lo conoce como retardo del protocolo OSPF, (*SPF Delay*) [1].
- **Intervalo Muerto (*Dead Interval*):** determina si un vecino adyacente está desactivado. Por defecto, el intervalo muerto es de cuatro veces el valor del intervalo de saludo, por defecto 40 segundos [1].
- **Retardo de Transmisión (*Transmit Delay*):** es el tiempo estimado para enviar un paquete de actualización del estado del enlace (LSA), el valor por defecto es 1 segundo [1].
- **Intervalo de Retransmisión (*Retransmit Delay*):** es el tiempo entre envíos de paquetes de actualización de del estado del enlace (LSA) entre enrutadores, el valor por defecto es de 5 segundos [1].

Parámetros	Falla Simple - 01 enlace cortado					Falla Doble - 02 enlaces cortados				
	Valor (A)	Valor (B)	Valor (C)	Valor (D)	Valor (E)	Valor (F)	Valor (G)	Valor (H)	Valor (I)	Valor (J)
Intervalo Hello < s >	10	8	6	4	3	10	7	5	4	3
SPF Delay										
Pérdida < ms >	400	200	100	80	50	400	150	80	50	30
Initial - Holdtime < ms >	600	400	300	200	100	600	300	200	120	80
Max - Holdtime < ms >	10000	10000	10000	10000	10000	10000	10000	10000	10000	10000
Transmit Delay < 1 - 65530 > s	1	1	1	1	1	1	1	1	1	1
Dead Interval < 1 - 65530 > s	40	40	40	40	20	40	40	40	40	20
Retransmit Interval < 1 - 65530 >	5	5	5	5	5	5	5	5	5	5
Ancho de Banda < mBits >	10	10	10	10	10	10	10	10	10	10
Pérdida < ms >	0	0	0	0	0	0	0	0	0	0
Pérdida de Paquetes < % >	0	0	0	0	0	0	0	0	0	0

Tabla 1. Detalle de parámetros de rendimiento considerados.

En la tabla 1 vemos dos secciones en columnas que indican el tipo de falla o corte del enlace, para cada tipo de falla se ha modificado los mismos parámetros de rendimiento, de manera a verificar el comportamiento del protocolo, ante el tipo de falla y su rendimiento. Tenemos entonces desde la columna “Valor(A)” hasta la columna “Valor(E)”, indican los valores aplicados a los parámetros del protocolo OSPF que se utilizaron para realizar las pruebas de Falla Simple. Las siguientes columnas, desde “Valor(F)” hasta la columna “Valor(J)”, indican los valores aplicados a los parámetros del protocolo OSPF que se utilizaron para realizar las pruebas de Falla Doble.

Falla Simple indica que existe un corte de enlace entre el origen y el destino.

Falla Doble indica que existen dos cortes del enlace entre origen y el destino.

5.2 Pruebas Experimentales

Todas las pruebas fueron realizadas utilizando el envío de paquetes ICMP de solicitud (ICMP echo request) y de respuesta (ICMP echo reply) de manera a comprobar el estado de la comunicación de cada host local con los demás host de la red. Antes de realizar la prueba se verificó el camino establecido por el protocolo, mediante el comando “traceroute”. Luego se configuró cada enrutador según los parámetros de rendimiento indicados en la Tabla 1.

Posteriormente, se eligió aleatoriamente un enrutador ubicado en la ruta trazada desde el host origen al host destino; mediante línea de comandos del software Quagga se cortó el enlace, deshabilitando la interfaz del enrutador elegido. En ese momento se produce la convergencia de rutas del protocolo. Cada solicitud ICMP consistió en el envío de 64 bytes y secuencia de 20 envíos y cada test es guardado en un archivo de texto. El corte fue realizado en la secuencia ICMP 10 para falla simple y para falla doble, el corte fue realizado en la secuencia ICMP 8 y en la secuencia ICMP 15.

En la Tabla 2, se resumen resultados obtenidos, que incluyen en total 400 pruebas experimentales, compuestos de 200 pruebas de fallas simples (1 corte del enlace) y 200 pruebas de fallas dobles (2 cortes del enlace) para ambos esquemas (i.e. redes tradicionales y SDN). Los parámetros de rendimiento citados, influyen en el rendimiento y mejoran el **tiempo de convergencia** del protocolo OSPF.

Fallas del Enlace	Red Tradicional			Red SDN		
	Test hechos	ICMP Perdido	Tiempo Promedio (ms)	Test hechos	ICMP Perdido	Tiempo Promedio (ms)
Falla Simple - Parámetro (A)	20	8	0,904	20	2	0,138
Falla Simple - Parámetro (B)	20	1	1,013	20	0	0,120
Falla Simple - Parámetro (C)	20	1	1,008	20	0	0,222
Falla Simple - Parámetro (D)	20	1	1,622	20	2	0,244
Falla Simple - Parámetro (E)	20	2	1,996	20	1	0,456
Falla Doble- Parámetro (F)	20	16	0,752	20	1	0,205
Falla Doble- Parámetro (G)	20	2	1,098	20	0	0,165
Falla Doble- Parámetro (H)	20	3	0,958	20	1	0,208
Falla Doble- Parámetro (I)	20	7	0,517	20	0	0,141
Falla Doble- Parámetro (J)	20	2	0,500	20	0	0,103

Tabla 2. Resumen de resultados experimentales

6 Conclusiones y Trabajos Futuros

Este trabajo presenta algunos los parámetros que influyen en el rendimiento y la convergencia del protocolo OSPF. El **tiempo de convergencia** del protocolo OSPF [1], es influenciado por intervalos de tiempo del paquete de saludo (*HELLO*) y otros parámetros de rendimiento del protocolo, ante diferentes condiciones de fallas.

En las SDN, sin embargo, la congestión del tráfico por inundación de anuncios LSA, tráfico de control y mecanismo de enrutamiento, disminuyen debido a la incorporación del controlador.

Como se puede ver en la Tabla 2, la pérdida de paquetes ICMP en redes tradicionales totalizan 43 paquetes perdidos en las 200 pruebas realizadas, mientras que en las SDN totalizan solo 7 paquetes perdidos, en la misma cantidad de pruebas realizadas, lo que representa una disminución del 84% de paquetes perdidos.

Los tiempos de convergencia presentados en la Tabla 2, indican que en promedio, el tiempo de convergencia en las Redes Definidas por *Software* es hasta 82% menor ante fallas simples y hasta 78% ante fallas dobles, en ésta implementación.

Las SDN ofrecen mayor flexibilidad y alto rendimiento para adecuarse a eventuales cambios de topologías y repentinos que requieren las redes de datos actuales.

Como trabajo futuro se analizará el **tiempo de convergencia** de otros protocolos de enrutamiento como el BGP, con distintas topologías y cantidad de nodos, considerando ambos esquemas de redes tradicionales y SDN.

Referencias Bibliográficas

1. Cristina Loredana Duta, Laura Gheorghe, Nicolae Tapus “Analyse OSPF Convergence Time in the Presence of Single and Multiple Failures”, EMERGIN 2015: The Seventh International Conference of Emerging Networks and Systems Intelligence, pp 3 –7, 2015.
2. Hailong Zhang, Jinyao Yan, “Performance comparison with Legacy Routing Protocol”, 2015 International Conference on Cyber-Enabled Distributed Computing and Knowledge Discovery, pp 2 – 4, 2015.
3. Albert Rego, Sandra Sendra, Jose Miguel Jimenez, Jaime Lloret, “OSPF Routing Protocol Performance in Software Defined Networks”, 2017 Fourth International Conference on Software Defined Systems (SDS), pp 2 – 6, 2017.
4. Mininet, <http://mininet.org>.
5. RouteFlow, <http://routeflow.github.io/RouteFlow>.
6. Quagga Routing Suite, <http://www.nongnu.org/quagga/>.
7. Escuela Politécnica Federal de Lausana, Curso TCP/IP Networking, <http://www.epfl.ch>.
8. Virtual Network over Linux, http://web.dit.upm.es/vnumlwiki/index.php/Main_Page.
9. Chinmay Abhay Joglekar, “Route Manipulation using SDN in Quagga”, Master Thesis, University of Colorado at Boulder
10. SDN Hub Tutorials, <http://www.sdnhug.org>
11. VirtualBox, <http://www.virtualbox.org>
12. MiniNExT, <http://github.com/USC-NSL/miniNExT>
13. Enrutamiento OSPF, <https://sites.google.com/site/redeslocalesyglobales/4-configuracion-de-red/2-configuracion-de-routers/6-configuracion-del-encaminamiento/2-encaminamiento-dinamico/6-protocolo-ospf/4-problema-de-convergencia-en-ospf>
14. Openflow, <https://www.opennetworking.org/sdn-definition/>.

Experiencias de Análisis de Consumo Energético en Redes de Sensores

Medina Santiago¹, Romero Fernando¹, De Giusti Armando^{1,2}, Tinetti Fernando G.^{1,3}

¹Instituto de Investigación en Informática LIDI (III-LIDI), Facultad de Informática
Universidad Nacional de La Plata, 50 y 120 2do piso, La Plata, Argentina

²CONICET – Consejo Nacional de Investigaciones Científicas y Técnicas

³CIC – Comisión de Investigaciones Científicas de la Pcia. de Buenos Aires
{smedina, fromero, degiusti, fernando}@lidi.info.unlp.edu.ar

Abstract. En este artículo se presenta la experimentación para evaluación de consumo energético de microcontroladores y módulos de comunicación utilizados o propuestos en el mercado para la construcción de y operación con redes de sensores inalámbricas o Wireless Sensor Network (WSN). Se identifican y analizan tanto los denominados modos de períodos de consumo mínimo (o *sleep mode/s*) de los módulos/hardware utilizado (en los no se transmiten datos) como los períodos en los cuales se transmiten datos se necesita y consume energía. Estos consumos se deben identificar tanto en los microcontroladores como en los propios módulos de comunicación específicos utilizados, dado que una gran cantidad de WSN combina ambos tipos de dispositivos en (inter)operación. En particular, los módulos de comunicación inalámbrica son utilizados para la transmisión propiamente dicha de los datos, utilizando diferentes estándares y protocolos, como WiFi, Bluetooth, ZigBee, LoRa, etc. Este tipo de redes generalmente tienen un despliegue en ambientes en donde no existe alimentación eléctrica constante por lo tanto es necesario optimizar el consumo energético de cada nodo ya que deben ser alimentados con baterías con una capacidad limitada.

Keywords: Red de Sensores, LoRa, WiFi, WSN.

1 Introducción

Las redes de sensores inalámbricos constituyen uno de los sistemas de procesamiento de información con mayor crecimiento y ámbito de aplicación de los últimos tiempos. Son denominados usualmente también como WSN (Wireless Sensor Network) o WSAN (Wireless Sensor & Actuator Network) [1] [2], y son un concepto alternativo de las MANET (Mobile Ad hoc NETWORK), enfocadas directamente en la interacción con el ambiente en donde son desplegadas, en lugar de la interacción con las personas.

El inicio de la evolución de las redes de sensores inalámbricas está dado principalmente por diferentes aplicaciones militares. Entre algunos de los primeros proyectos con similitudes a las actuales características de las WSN se pueden nombrar: el proyecto Chain Home, un anillo de radares para alerta temprana que podía detectar y rastrear

aeronaves durante la Segunda Guerra Mundial; el proyecto SOSUS, del acrónimo Sound Sourviellance System, utilizado en la guerra fría por el ejército de Estados Unidos para rastrear submarinos soviéticos y el proyecto NORAD (North American Aerospace Defense Command), también desarrollado durante la guerra fría, para en control y la defensa aérea en los Estados Unidos.

En la estructura básica de una Red de Sensores se pueden definir 3 partes: los nodos sensores, la red inalámbrica de comunicación y nodos centralizadores o gateways [3]. Los nodos sensores y los centralizadores están basados en pequeños dispositivos de hardware, los microcontroladores, que utilizan transceptores inalámbricos para comunicación y un módulo de alimentación.

El crecimiento exponencial en el desarrollo de hardware del último tiempo ha generado nuevos microcontroladores de pequeño tamaño, con mayor capacidad de procesamiento y menor consumo energético. Estos dispositivos, combinados con el avance sobre nuevas tecnologías de comunicación inalámbrica dan como resultado la capacidad de desplegar diferentes tipos de nodos que conforman Redes de Sensores aplicadas en diferentes ámbitos, para obtener datos y realizar operaciones. Entre las características de las redes de sensores se pueden identificar: Baja tasa de transferencia, Limitación en la alimentación o bajo consumo de los nodos sensores, alimentación por baterías, escalabilidad, gran cantidad de nodos. Estas Redes tienen diferentes ámbitos de aplicación debido a que dan la posibilidad de desarrollar los nodos de manera personalizada para su utilización en cada área específica, y adaptar la topología de red a las necesidades del problema. Entre los ejemplos más comunes de aplicación de las Redes de Sensores Inalámbricos están, desarrollos militares, monitoreo ambiental, agricultura, seguridad, domótica y ciudades inteligentes [3] [4] [5].

Uno de los aspectos importantes a considerar en el desarrollo de los nodos de una red de sensores es el consumo energético. Dadas las características de los ambientes en donde son desplegadas este tipo de redes, no se cuenta en la mayoría de los casos, con una conexión a la red eléctrica que sea constante. Por lo tanto, es necesaria la utilización de baterías que se encarguen de proveer alimentación a cada nodo de la red. Si bien el avance en el desarrollo informático/electrónico dio como resultado microcontroladores y tecnologías de comunicación de bajo consumo, es necesario hacer un análisis preciso en donde se combinen diferentes aspectos a considerar:

- La elección de los componentes electrónicos que conforman a un nodo.
- El tipo de transmisor/receptor de radio.
- Las posibilidades de modos de ahorro de energía que da el microcontrolador.
- El tipo de batería.

Este trabajo es una continuación de la experimentación realizada en [6], haciendo foco en el consumo energético de los nodos que componen una red de sensores inalámbricos y las técnicas disponibles para reducir dicho consumo.

2 Consumo Energético en una WSN

Los principales campos de aplicación de las WSN requieren que los nodos que la conforman tenga autonomía energética, y esto se logra utilizando pilas o baterías (algunas veces recargadas con paneles solares). Estos componentes electrónicos de alimentación

tienen su duración en función del consumo que se les requiere, y por lo tanto es de vital importancia realizar un análisis de las características de consumo energético de cada nodo de la red para poder estimar la autonomía de funcionamiento sin necesidad de un cambio o recarga en las baterías.

Los componentes principales de un nodo son el microcontrolador y el transceptor inalámbrico para transmisión. Para lograr una estimación correcta de consumo se suele realizar un análisis individual de cada parte y su funcionamiento en conjunto. En el caso del microcontrolador existen dos aspectos de análisis. El primero es a nivel de hardware, en el que por ejemplo se evalúan los componentes electrónicos necesarios para nivelar la alimentación de entrada (un regulador de tensión) en función de los niveles que maneja la batería utilizada. También en este nivel hay que considerar las optimizaciones de hardware la temperatura del ambiente, ya que este factor podría hacer que los componentes electrónicos se vean afectados y requieran un mayor consumo para funcionar correctamente. El segundo aspecto a evaluar son las opciones de optimización energética que se pueden lograr a nivel de software, en este aspecto entran los modos *sleep* o modos bajo consumo, la posibilidad de variar la velocidad (frecuencia) del reloj del procesador en ejecución y la desactivación de componentes que no se utilicen.

Por el lado del transceptor de radiofrecuencia, el consumo va a depender de la tecnología que se utilice, existen varias de baja consumo como ser LoRa o ZigBee. También hay disponibles tecnologías que no son de bajo consumo como WiFi o Bluetooth. En el análisis de consumo a nivel comunicación es importante determinar la cantidad de paquetes que necesitan ser enviados por el nodo en un tiempo específico, ya que a mayor cantidad de paquetes habrá mayor consumo. Otro punto a considerar es la potencia que se utiliza en la transmisión, donde se debe lograr una optimización entre la distancia y la potencia, de tal manera que se configure la transmisión con la menor potencia posible (que implica menor consumo), pero que alcance la distancia necesaria para que el nodo siga siendo parte activa de la red.

3 Hardware Utilizado

A los fines prácticos del análisis de este artículo se eligieron placas de desarrollo y módulos que son de alta disponibilidad y que ya mostraron buen funcionamiento como parte de una WSN en la experimentación realizada en [6].

3.1 NodeMCU ESP8266

NodeMCU es un firmware y una placa de desarrollo de código abierto, basada en el chip de la empresa Espressif ESP8266-12E. La placa agrega al microcontrolador el chip para la conexión UART a través de un puerto micro USB, que sirve como alimentación y punto de comunicación [7]. El ESP8266 es un microchip WiFi de bajo consumo que contiene como CPU al procesador Tensilica LX106 de 32-bit RISC, a 80MHz; con 64 KB de memoria de instrucciones y 96KB de datos [8] [9]. Tiene integradas varias características de hardware que lo hacen apropiado para una variedad muy amplia de aplicaciones: 1 ADC de 10-bit, 17 GPIO pins, Interfaces: 3 SPI, 1 I2C, 1 I2S y 2 interfaces

UART, PWN, Protocolo WiFi 802.11 b/g/n, entre otras. Las dos características más importantes, sin embargo, no son necesariamente de hardware: bajo costo y amplio soporte de programación. En particular, la programación utilizando el IDE y las bibliotecas de Arduino han generado una gran cantidad de ejemplos disponibles gracias al uso masivo. En la Fig. 1 se puede apreciar, por ejemplo, que los pines GPIO son identificados como los del Arduino UNO (D0, D1, etc.).



Fig. 1 – Placa de desarrollo NodeMCU

El ESP8266 tiene predefinidos 3 tipos de modos sleep y un consumo teórico para cada uno de ellos [10], tal como se muestra en la Fig. 2. Cada uno afecta a diferentes partes de la electrónica del micro.

Item	Modem-sleep	Light-sleep	Deep-sleep
Wi-Fi	OFF	OFF	OFF
System clock	ON	OFF	OFF
RTC	ON	ON	ON
CPU	ON	Pending	OFF
Substrate current	15 mA	0.4 mA	~ 20 µA
Average current	DTIM = 1	16.2 mA	1.8 mA
	DTIM = 3	15.4 mA	0.9 mA
	DTIM = 10	15.2 mA	0.55 mA

Fig 2 - Tabla de Modos Sleep ESP8266

3.2 Heltec WiFi LoRa 32

La Fig. 3 muestra la placa de desarrollo construida por la empresa Heltec. Esta placa incluye dos chips principales, el ESP32 y el transceptor LoRa SX1278 (con conector I-PEX, para antena externa), sumados a un convertidor CP210x MicroUSB a UART, una pantalla 0,96” OLED y un conector Molex PicoBlade de 2 pines para batería LiPo/Li-Ion [11]. El módulo WiFi puede transmitir a una tasa de hasta 150Mbps. El módulo Bluetooth, tiene funcionamiento normal y también BLE (Bluetooth Low Energy).

El ESP32 es una solución integrada de WiFi-Bluetooth con radiofrecuencia de 2.4GHz. Basado en el microprocesador Tensilica Xtensa LX6 de 32 bit, a 160MHz, con 448 KB de memoria ROM y 520 SRAM [12] [13]. Algunas de sus características más

importantes de hardware son: 34 GPIOs programables, 1 SAR ADC de 12bit, 2 DAC de 8 bit, varias interfaces (SPI, I2S, I2C), PWM y Protocolo 802.11 b/g/n. EL SX1278 es un chip transceptor de radio frecuencia LoRa desarrollado por Semtech, dentro de la placa de desarrollo de Heltec, tiene dos versiones de frecuencia: 433MHz y otro que soporta las frecuencias 868MHz y 915MHz [14]. Este módulo se conecta vía SPI al ESP32 (conexión integrada en la propia placa).



Fig. 3 – Placa de Desarrollo de Heltec

El ESP32 tiene 4 modos sleep que desactivan los componentes internos del micro según explica la tabla de la Fig. 4.

Power mode	Active	Modem-sleep	Light-sleep	Deep-sleep	Hibernation
Sleep pattern	Association sleep pattern			ULP sensor-monitored pattern	-
CPU	ON	ON	PAUSE	OFF	OFF
Wi-Fi/BT base-band and radio	ON	OFF	OFF	OFF	OFF
RTC memory and RTC peripherals	ON	ON	ON	ON	OFF
ULP co-processor	ON	ON	ON	ON/OFF	OFF

Fig 4 -Tabla de Modos Sleep ESP32

3.3 Adafruit INA219

Para la medición del consumo energético se seleccionó el sensor de corriente Adafruit INA219 como el que se muestra en la Fig. 5. Este sensor es de característica invasivo ya que se conecta en serie al elemento y de esta manera se obtiene la corriente que pasa por ese conductor. El módulo sensor está compuesto por un amplificador de precisión que mide el voltaje que cae en la resistencia de detección de 0.1ohm que tiene la placa. Dado que la diferencia máxima de entrada del amplificador es $\pm 320\text{mV}$, da un rango de medición de hasta $\pm 3.2\text{A}$. Con el ADC interno de 12bits, la resolución en el rango $\pm 3.2\text{A}$ es $0,8\text{mA}$. Con la ganancia interna establecida al mínimo, la corriente varía entre $\pm 400\text{mA}$ y la resolución es de $0,1\text{mA}$. Se puede alimentar con 5V o 3V y maneja una comunicación I2C para la salida de los datos medidos.

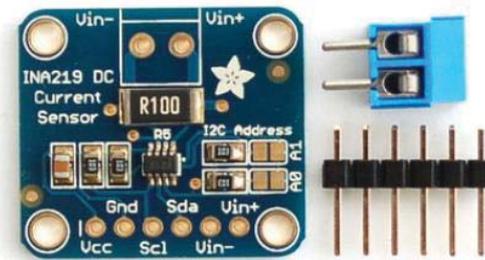


Fig 5 - Placa del Sensor INA219 [15]

4 Experimentación

En todas las pruebas de la experimentación se utilizó el mismo sistema para la medición de consumo [16], mostrado en la Fig. 6: se captura la corriente con el sensor INA219 y se envía por I2C a un Arduino UNO [17]. En el código que corre el Arduino se utilizó la librería que fue desarrollada por el fabricante del módulo sensor (Adafruit) para esta plataforma. La alimentación al microcontrolador del cual se censa el consumo fue a través del puerto micro USB de cada placa con un voltaje constante de 5V.

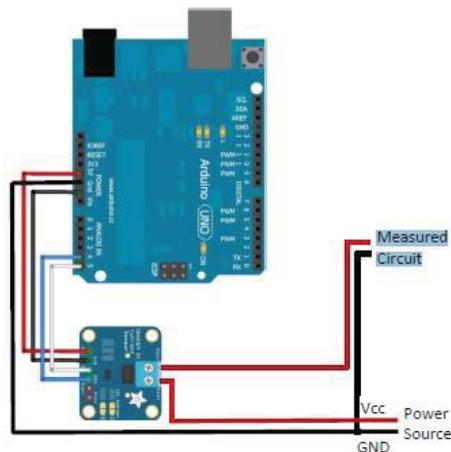


Fig. 6 - Esquema de conexión de la experimentación [15]

La experimentación se dividió en dos partes en función de la tecnología de comunicación estudiada en cada caso. Primero se hicieron experimentos de la conexión WiFi en cada una de las plataformas de desarrollo, en donde también se consideran los modos sleep, y segundo en la conexión inalámbrica LoRa.

I-WiFi

Los escenarios planteados fueron similares para cada placa de desarrollo, NodeMCU ESP8266 y Heltec WiFi LoRa 32. Todos se desarrollaron en scripts dentro del IDE

Arduino de programación:

- Se analizó la transmisión y recepción continua de paquetes WiFi.
- Se analizaron los modos sleep específicos de cada una de las placas y sus posteriores procesos de despertar.
- Se analizó el proceso de conexión de un cliente a una red WiFi.

Como agregado de las mediciones recién listadas, se consideraron las 4 variaciones que presenta el ESP8266 al momento de despertarse del modo DeepSleep:

- WAKE_RF_DEFAULT: cuando se reinicia el microprocesador, únicamente se calibra la señal de radio si el chequeo da error.
- WAKE_RFCAL: cuando se reinicia el microprocesador siempre se calibra la señal de radio.
- WAKE_NO_RFCAL: cuando se reinicia el microprocesador no se calibra la señal de radio.
- WAKE_RF_DISABLED: cuando se reinicia el microprocesador se deshabilita la señal de radio (como en el modo modem sleep). No permite ni enviar ni recibir datos vía WiFi.

II – LoRa

En el caso de LoRa, los experimentos se realizaron solamente sobre la placa de Heltec, ya que cuenta con el transceptor LoRa SX1278. Para el desarrollo de los códigos de prueba se utilizó el IDE Arduino, con la última versión de la librería desarrollada por Heltec para las comunicaciones LoRa de su placa. En este caso las pruebas se dividieron entre las mediciones de recepción y las de transmisión de paquetes. En transmisión, la nueva versión de la librería permite variar el factor de potencia de la comunicación inalámbrica (entre 0 y 20), para lograr un mayor alcance. Por lo tanto se probaron cada uno de los factores disponibles.

5 Resultados

Se muestran en este punto los resultados de los experimentos comentados en la sección anterior con las muestras obtenidas en cada caso. En la conexión WiFi para cada tipo de placa utilizada se muestran los modos de operación como Access Point y como cliente respetivamente. Finalmente se muestran los resultados correspondientes al módulo de comunicaciones LoRa en recepción y transmisión. En la siguiente sección se mencionan las consideraciones como análisis de los resultados obtenidos.

5.1 NodeMCU

Las Tablas 1 y 2 a continuación muestran los resultados obtenidos para la placa de desarrollo NodeMCU. Los modos de sleep corresponden a cada modo de operación. Los picos de consumo se producen en períodos relativamente cortos de tiempo en algunos estados de operación del módulo.

Tabla 1. Consumo de cada por estado en modo Access Point

Operación	Consumo promedio (mA)	Consumo pico (mA)
Transmisión		360
Recepción	76	
ModemSleep	15	
DeepSleep	0.55	
DeepSleep - WAKE_RF_DISABLED	14	
DeepSleep - WAKE_RFCAL	76	360
DeepSleep - WAKE_NO_RFCAL	70	310
DeepSleep - WAKE_RF_DEFAULT	72	320
LightSleep	2	

Tabla 2. Consumo por estado en modo cliente

Operación	Consumo promedio (mA)	Consumo pico (mA)
Conexión		330
Conectado	68	
Transmisión		270
DeepSleep	0.6	
ModemSleep	15	

5.2 Heltec WiFi LoRa 32

Las Tablas 3, 4, y 5 a continuación muestran los resultados obtenidos para la placa de desarrollo de Heltec. Los modos de sleep corresponden a cada modo de operación WiFi. Los picos de consumo se producen en períodos de tiempo cortos en algunos estados específicos.

Tabla 3. Consumo por estado en modo Access Point

Operación	Consumo promedio (mA)	Consumo pico (mA)
Transmisión		310
Recepción	150	
Inicio	130	280
DeepSleep	10	
LigthSleep	11	
Hibernation	10	
ModemSleep	15	

Los valores de la operación de Inicio que muestra la tabla anterior son de los milisegundos que tarda el modulo en iniciar y activar la red WiFi como Access Point, durante ese proceso llega a picos de 280mA que corresponden a la calibración de la antena WiFi.

Tabla 4. Consumo por estado en modo cliente

Operación	Consumo promedio (mA)	Consumo pico (mA)
Conexión		150
Conectado	56	
Transmisión		162
DeepSleep	10	
ModemSleep	15	

Tabla 5. Consumo por estado utilizando el transceptor de LoRa

Operación	Consumo promedio (mA)	Consumo pico (mA)
Recepción	80	
Transmisión Tx Power 0		102
Transmisión Tx Power 5		107
Transmisión Tx Power 10		114
Transmisión Tx Power 15		132
Transmisión Tx Power 20		177

6 Conclusiones y Trabajo Futuro

Este trabajo se basa en una experimentación para la obtención de valores de referencia del consumo energético de dos placas de desarrollo con dos tecnologías de comunicación inalámbrica distintas, para tomar decisiones en cuanto al armado del nodo de una red de sensores. Los resultados obtenidos muestran que ambas placas de desarrollo presentan herramientas efectivas para lograr un consumo global bajo. Considerando que en una aplicación de red de sensores el tiempo concreto de funcionamiento es muy poco y la placa pasaría la mayoría del tiempo en un estado de reposo (modo sleep) con su transceptor inalámbrico desactivado, por lo tanto, son una buena elección a la hora de desarrollar un nodo basándose en cada una de ellas.

En la comparación con los valores teóricos que dan los fabricantes se puede ver que los resultados obtenidos están por encima. Esto se debe a que los valores teóricos están definidos para el chip solo, y en nuestro caso de experimentación estamos trabajando sobre una placa de desarrollo entera, que incluye electrónica extra para la regulación de tensión, el manejo de la conexión USB y leds de encendido, entre otros componentes.

Respecto a la placa NodeMCU se obtiene un excelente valor de consumo al llevarla al modo de DeepSleep, en ambos modos de operación WiFi; esto es muy útil si se tienen conexiones con transmisiones esporádicas. Se debe considerar como muestran las tablas que tanto las transmisiones, como la conexión a la red inalámbrica tienen picos altos de consumo, que duran milésimas de segundo, por lo tanto la batería que alimenta a esta placa debe poder resistir estos picos.

En la placa de Heltec puede llegar a dos conclusiones. La primera respecto a las conexiones WiFi, que muestran una leve ventaja frente a las de la placa anterior respecto a la transmisión y recepción de paquetes, pero no presenta mejores valores en los modos

sleep. Y la segunda conclusión se da con los valores obtenidos de las pruebas realizadas da la comunicación LoRa, cabe aclarar que estos valores solo cuentan el consumo en estado básico de la placa (con la conexión WiFi desactivada) para hacer foco en el transceptor SX1278. Se puede ver en los resultados que hay una curva creciente en consecuencia directa al aumento de la potencia de transmisión. Solo se consideraron los valores de pico ya que el valor promedio siempre se mantuvo alrededor de los 80mA como en el estado de recepción, pero se deben considerar estos picos para elegir el módulo de alimentación correcto.

Como trabajos futuros se plantean las pruebas específicas de las distancias de funcionamiento de LoRa en las variaciones de la potencia de transmisión y la posibilidad de programar a más bajo nivel el módulo ESP8266 para variar la potencia en las transmisiones WiFi. También es de interés para complementar el análisis funcional de una WSN realizar pruebas de performance a nivel de seguridad y encriptación de datos con las placas utilizadas en este artículo.

Bibliografía

1. Estándar IEEE 802.11 Wireless Local Area Network, <http://www.ieee802.org/11/>
2. Estándar IEEE 802.15 Wireless Personal Area Network, <http://www.ieee802.org/15/>
3. Sohraby, K., Minoli, D., Znati T.: Wireless sensor networks: technology, protocols, and applications. John Wiley & Sons (2007).
4. Raghavendra, C. S., Sivalingam, K. M., Znati, T.: Wireless sensor networks. Springer (2006).
5. Akyildiz, I., Vuran, M.: Wireless sensor networks(Vol. 4). John Wiley & Sons (2010).
6. Medina, S., Romero, F., De Giusti, A. E., & Tinetti, F. G. (2018). Análisis para despliegue de una red de sensores heterogénea. In XXIV Congreso Argentino de Ciencias de la Computación (La Plata, 2018).
7. Documentación NodeMCU, www.nodemcu.com
8. ESP8266EX Datasheet Version 5.8, Espressif Systems. Año 2018. Disponible en https://www.espressif.com/sites/default/files/documentation/0a-esp8266ex_datasheet_en.pdf
9. ESP8266 Technical Reference Version 1.3, Espressif Systems. Año 2017. Disponible en https://www.espressif.com/sites/default/files/documentation/esp8266-technical_reference_en.pdf
10. ESP8266 Low Power Solutions Version 1.1, Espressif Systems. Año 2016. Disponible en https://www.espressif.com/sites/default/files/9b-esp8266-low_power_solutions_en_0.pdf
11. Documentación WIFI LoRa 32, <http://www.heltec.cn>
12. ESP32 Datasheet Version 2.3, Espressif Systems. Año 2018. https://www.espressif.com/sites/default/files/documentation/esp32_datasheet_en.pdf

13. ESP32 Technical Reference Manual Version 3.4, Espressif Systems. Año 2018.
https://www.espressif.com/sites/default/files/documentation/esp32_technical_reference_manual_en.pdf
14. SX1276/77/78/79 Datasheet Rev 5, Semtech. Año 2016.
https://www.semtech.com/uploads/documents/DS_SX1276-7-8-9_W_APP_V5.pdf
15. Adafruit INA219 Current Sensor Breakout, Adafruit Systems. Año 2019.
Disponible en <https://cdn-learn.adafruit.com/downloads/pdf/adafruit-ina219-current-sensor-breakout.pdf>
16. Fernando G. Tinetti, Santiago Medina, Mariano Méndez, Fernando L. Romero, “Energy Measurements in IoT/ Maker Development Cards”, III-LIDI, Facultad de Informática, UNLP, Sept. 2016, Technical Report TR-RT-01-2016, disponible en <http://fernando.bl.ee/>
17. Arduino LLC, “Arduino – ArduinoUnoRev3”,
<https://store.arduino.cc/usa/arduino-uno-rev3>

Filtering Useless Data at the Source

Pablo Pessolani¹, Constanza Quaglia¹, and Ramón Nou²

¹Departamento de Ingeniería en Sistemas de Información
Universidad Tecnológica Nacional - Facultad Regional Santa Fe
Santa Fe, Argentina
{pessolani,cquaglia}@frsf.utn.edu.ar

²Barcelona Supercomputing Center
Departamento de Arquitectura de Computadores
Universitat Politècnica de Catalunya. Barcelona, España
ramon.nou@bsc.es

Abstract. There are some processing environments where an application reads remote sequential files with a large number of records only to use some of them. Examples of those environments are servers, proxies, firewall and intrusion detection log analysis tools, sensor log analysis, large scientific datasets processing, etc.

To be processed, all file records must be transferred through the network, and all of them must be processed by the application. Some of the transferred records would be discarded immediately by the application because it has no interest in them, but they just consumed network bandwidth and operating system's cache buffers.

This article proposes to filter records from the source of data but without changing the application. Those records of interest will be transferred without modifications but only references to the other records will be transferred from the source to the consuming application. At the application side, the sequence of records is rebuilt, keeping the content of records of interest and filling the others with dummy values which will be discarded by the application. As the number and length of records are preserved (and therefore the file size too), it is not necessary to modify the application. Once a filtering rule is applied to a file, only the useful records and references to unuseful ones will be transferred to the application side reducing network usage, transfer time, and cache utilization. A modified (but compatible) version of NFS protocol was developed as a proof of concept.

Keywords: Logging, Network File System, NFS

1 Introduction

Some remote applications, servers, network devices, and IoT sensors produce a large number of data which will be processed by other applications in a central location. Online transactional applications, backup servers, mail servers, web

server, network proxies, routers, firewalls, IDS/IPS, environment sensors, etc. are a sample of that. Often, an application is used to analyze the content of these data to search records of interest as suspicious transactions, abnormal behavior, patterns of security attacks, malformed network packets, excessive temperature detected by a sensor, etc. To perform the data analysis the complete dataset must be transferred from the source to the central application. As the application would not be interested in the content of all data records, several records will be transferred from the source to the client but later, they would be ignored/discarded by the application. This behavior misspends network bandwidth, operating system cache buffers and, in some cases, increases the transfer time. An additional drawback occurs in cases where the network is leased and the services are paid for each packet or amount of transferred data.

This article proposes to filter records from the source of data but without changing the application. Those records of interest will be transferred without modifications but only references to the other records will be transferred from the source to the application. At the application side, the sequence of records is rebuilt, keeping the content of matching ones and filling the others with dummy values which will be discarded by the application. As the number and length of records and the file size are preserved, it is not necessary to modify the application.

The “*filtering useless data at the source*” approach is not new. It is often used by Client/Server applications where the Client request data using and Database (like SQL) query to filter those records of interest. Then, the Database Server replies sending only the requested records to the Client application.

As a proof of concept of the “*filtering useless data at the source*” approach, a modified (but compatible) version of NFS protocol version 3 was developed, based on UNFS3 [1] and HSFS [2]. The same approach could be used on other network or distributed file systems, FTP-like servers or storage devices. Seeing the processing power that controllers of storage devices currently have, a portion of these power could be used to perform filtering at the source [3].

The remainder of this paper is organized as follows. Section 2 describes the authors’ motivation for this project. Section 3 outlines FNFS design and implementation. Section 4 presents some experimental results. Finally, section 5 presents conclusions and future work.

2 Motivation

Data traffic reduction is essential when it comes to high latency links, limited bandwidth, high packet loss or when the system uses a paid communication service in which amount of transferred data impacts on the costs, or when an autonomous device needs to save energy to keep its batteries charged for as much time. Low Earth Orbit (LEO) satellite and mobile networks like 4G are examples of leased links with appreciable latency, eventually packet loss, and limited bandwidth. IoT devices must be specially considered, as they are being increasingly used for different purposes. These devices normally have a low cache

or buffer storage (due to having low memory in general) and energy usage is a critical issue since they are battery-powered. In this sense, reducing packet transmission also helps to reduce battery usage.

This article proposes a methodology to mitigate these negative issues of data transfer through these kinds of networks.

3 Filtered Network File System

This section presents FNFS design goals and implementation details that includes:

1. How to specify filtering rules outside the application which consumes the data.
2. How the data are encoded and transferred from the Server to the Client.
3. How data compression can be made on the transferred data.

The prototype implements Filtered NFS (FNFS) using a user-space NFS server [1] and client [2] as base source code. Even though the prototype was implemented using user-space NFS server and client, the same approach would be applied to kernel versions and to other network/cluster filesystems.

Although FNFS can be used in production environments, it was developed as a proof of concept of the “filtering useless data at the source” approach. Currently, it has some constraints as it can only filter sequential ASCII text files.

3.1 FNFS Design goals

The following are the properties of FNFS established as design goals:

1. *Compatibility*: Keep client and server compatibility with unmodified NFS servers and clients.
2. *UID and File specification*: The filters will be applied considering the file-name (complete path) and the UID of the requester client.
3. *External filtering rule specification*: The application does not need to be modified. Therefore, an external filtering rule specification mechanism must be used.
4. *Efficiency*: The filtering mechanism must have high efficiency related to network bandwidth usage, count of packet transfers, cache buffers savings and total transfer time.
5. *Simple filtering rule specification*: Filtering rules must be specified by a simple and well-known language.
6. *Fixed and Variable length records*: The length of records must be kept and it must support fixed length records and variable length records (Line Feed terminated).
7. *Discretionary data encoding*: Encoding only will be applied if appropriate.

8. *Include compression:* To improve network bandwidth and packet transfer savings, apply data compression on records of filtered files.

FNFS lets specify filtering rules outside an application that will be applied by the Server before sending file records to the Client. This feature is very useful because different filtering rules could be applied for each user/application which uses the data or when modifying the application is expensive or when the application source programs are not available.

As it is well known, NFS is a stateless file-server protocol. The Client side of FNFS gets the filtering rules, encodes them on each read request afterwards it sends them to the Server. The Server decodes the *Read Request* with the filtering rules and applies them to the file records. Those records which match the filtering rules are sent from the Client to the Server without modifications. For those records which do not match the filtering rules, only the length and position of them are encoded and sent to the Client.

Later, when the Client receives the reply, it decodes the data received from the server, and it rebuilds the sequence of file records filling the not-matching records with specified dummy data that will be ignored or discarded by the application. An example can clarify FNFS operation. Suppose that the application only processes records with the text “**ARGENTINA**” in position 10, discarding the other records.

```
char *country; // point to the country name first character
country = &record[10]; // record just read
if (strncmp(country, "ARGENTINA", 9) == 0) {
    // process the matching record
} else {
    // discard the record
}
```

When FNFS Client receives the code of a not-matching record from the Server, it creates a dummy record with dummy text in position 10 as “**123456789**” whose will be discarded by the application *if* sentence.

As the application processes the same number of records as the original file and record lengths are preserved, there is no need to modify it. In addition to FNFS data filtering features, data compression was added to improve the usage of network bandwidth and to reduce the transfer time. Data compression is a feature that the user can disable. In some use cases, it is not desired due to the higher CPU and memory usage required (i.e. IoT devices).

Data encryption is another feature considered to be added in the near future. The current version presented here only supports sequential ASCII text files with fixed or variable record length and text filters.

3.2 Filtering Specifications

As was mentioned earlier, filters are specified outside the application. FNFS adopts a simple approach by using shell environment variables.

- *FNFS_FILTER*: Specifies if the filter will be applied (YES/NO).
- *FNFS_FILENAME*: The pathname of the file to be read and filtered.
- *FNFS_RLEN*: The record length. It must be a positive number for fixed length records or zero for variable length records.
- *FNFS_FILTER*: The text search on each record.
- *FNFS_MATCH*: If the value is "YES" it means that the matching registers will be transferred to the Client. If the value is "NO", unmatching registers will be transferred to the client.
- *FNFS_DUMMY*: It specifies the character which replaces all characters within useless registers. i.e. "#".
- *FNFS_COMPRESS*: Specifies if compression will be applied (YES/NO).

These environment variables must be set and exported before starting FNFS Client. Once the Client checks correct values, it codifies them into newly added fields of the NFS read request.

3.3 Data Encoding

Data encoding must meet design goals #4, #6, #7. The server sends to the client a block of data which consists of a variable size list followed by a sequence of matching records as shown in Figure 1.

Figure 1 presents the data encoding of a block of 10 fixed length records of 80 bytes each (records #0 to #9). The records that match the filtering rules are #2, #4, #5, #6, #8 and #9. The first two rows represent a list of 10 items, one for each record. Each list element with a value greater than zero represents the record length of a non-matching record. If the value is 0, it means that the record in this position matches the filtering rule, and it is stored in order and without modifications within the block. Figure 1 represents 10 records of 80 bytes or 800 bytes, but only 6 records of 80 bytes (480 bytes) plus the size of the array (10 elements of 4 bytes) are sent to the client, saving about 35% of data to be sent through the network. Figure 2 presents the data block encoding of variable length records terminated with Line Feed character.

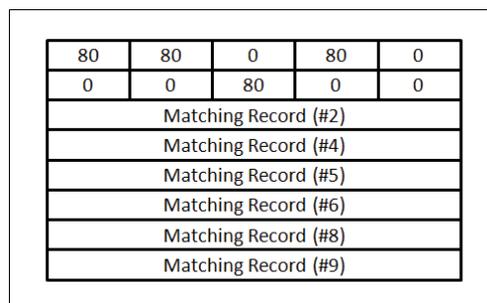


Fig. 1: Fixed Length Records Data Block Encoding

The last character LF on each matching record is a Line Feed ASCII code. It is used by the FNFS client to compute the record length. Figure 2 represents the encoding of a block of 10 records totalizing 521 bytes, but only 6 records totalizing 348 bytes plus the size of the array (10 elements of 4 bytes) are sent to the client, saving about 26% of data to be sent through the network.

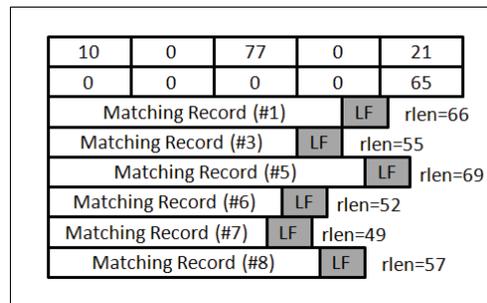


Fig. 2: Variable Length Records Data Block Encoding

Let $rlen$ the length (bytes) of each record and $nrecs$ the total amount of records of the file.

For fixed length records, without filtering, the total bytes is defined as:

$$T_{bytes} = rlen * nrecs \quad (1)$$

With filters, and a matching rate k of the filtering rule ($0 \leq k \leq 1$), the number of bytes to be transferred is:

$$F_{bytes} = (rlen * nrecs * k) + (nrecs * 4) = n_{recs} * ((rlen * k) + 4) \quad (2)$$

For variable length records, without filtering:

$$T_{bytes} = \sum rlen_i \quad (3)$$

With filters and $K_i = \{0, 1\}$ where 1 means that the record match:

$$F_{bytes} = \sum (rlen_i * K_i) + (nrecs * 4) \quad (4)$$

Only if $F_{bytes} < T_{bytes}$ the block is encoded by FNFS. The saving ratio is $1 - \frac{F_{bytes}}{T_{bytes}}$.

3.4 Data Compression

Data filtering reduces the total data sent from Server to Client. Applying compression to encoded data introduces another level of efficiency improvement on network usage. The simple additions of compression functions to the filtering code produce important savings.

The resulting block of encoded data on the Server is compressed using the zlib[4] compression library functions.

4 Evaluation

Several benchmarks have been performed with simulated networks, considering different latency and bandwidth scenarios, and with and without data compression. The tests consist of filtering records without the occurrence of an ASCII string. Only records with the occurrence of the string will be transferred without modification. On the server-side, several files were created with different rate of occurrence of the string to be filtered. The x-axis in charts represents the hit rate of the filtering rule. A hit rate of 128 means that the probability of string matching is 1/128, and so on. A hit rate of 1 means that the probability of string matching is 1/1 (the entire file).

For each simulation, the charts show the amount of transmitted packages and transmission time, under the specified conditions. Each line is labeled as follows:

- *NFS*: NFS protocol without modifications.
- *FNFS*: Filtered NFS without compression.
- *FNFS w/c*: Filtered NFS with compression.
- *GZIP*: NFS standard protocol with the file previously compressed with GZIP at the server and decompressed at the client.

Simulations were made using the Traffic Control tool (*tc*) on the server side. *tc* allows modifying traffic parameters like latency, bandwidth limit, and error rate. In this way, the server was set to simulate VSAT, Low-Earth Orbit (LEO) and Mid-Earth Orbit (MEO) satellite, and mobile 4G communication behaviours.

4.1 VSAT

VSATs (Very Small Aperture Terminals) consist of small satellite stations, that work with Geostationary Satellites (GEO)[5]. Each VSAT station can be configured in point-to-point or mesh topology. Usually, they are configured in a star topology with a special station with a high-gain antenna, called Hub. In this configuration, the signal must go through two round trips to the satellite increasing the latency time.

4.2 LEO and MEO satellite constellations

A constellation of LEO satellites is placed near the surface of the earth (about 2000 to 5000 Km). Due to their closeness, stations do not need much power, and the communication delay is around a few milliseconds (30-50 [ms])[6]. MEO satellite constellation moves across the sky at a height of 8000 Km with a delay of 70 to 150 [ms].

4.3 Mobile 4G communications

4G mobile networks[7] are intended to provide services like video and voice streaming and mobility support. Packet loss must be especially considered here, due to the kind of services these networks are thought to bring.

4.4 Simulation Results

To evaluate the performance and the operation of FNFS, two nodes were used with the following hardware configurations: Server) AMD A6-3670, 2.7 GHz, 8GB RAM. Client) Intel(R) Celeron(R) CPU G1820, 2.7 GHz, 4 GB RAM. The nodes were connected through a dedicated 1 Gbps LAN Switch. Both nodes run Debian 9.4 (called “stretch”) with a Linux kernel version 4.9.88.

The benchmarks performed to evaluate the amount of packet transferred which would affect the network and battery usage, eventually the cost of the transfer, and the total transfer time. Different hit rates of the filtered string, between 0 (no matches) and 1 (100% matches) were used for the benchmarks. Table 1 shows the configuration parameters (delay, bandwidth and packet loss) considered for each technology. Fig. 3 presents the amount of packets transferred

	Delay [ms]	Bandwidth [kbps]	Packet loss [%]
MEO	150	492	0
4G/LEO	40	5120	2
VSAT	540	1024	0

Table 1: Summary of the Evaluated Configurations

(a) and the total transfer time (b) on a MEO satellite link. For a hit rate lower than $1/3$, the amount of packets is reduced, even more if compression is used ($1/2$). The total transfer time is greater than compressing the file with gzip, only for a hit rate lower than $1/64$ w/o compression and lower than $1/16$ with compression. Only when there are no hits, the transfer with the standard NFS protocols is better than FNFS. The amount of packets transferred (Fig. 4(a)) for 4G/LEO links is lower for FNFS when the hit rate is lower than $1/3$ without compression and lower than 1 when FNFS uses compression. The total transfer time (b) is lower for FNFS only for a hit rate of lower $1/64$ w/o compression and $1/16$ with compression against zipping the file.

Fig. 5 shows the amount of packets transferred (a) and transfer time (b) on a VSAT link. For a hit rate lower than $1/4$, the amount of packets is reduced, even more if compression is used ($1/2$).

Finally, the transfer time for a VSAT link results better when the file is previously compressed and FNFS has not appreciable advantages over NFS.

As shown in figures 3-5, FNFS demonstrated to reduce the amount of packets transmitted and, in some cases the total transfer time with the exception of a hit rate of 1. In this case, NFS performs better due to the extra work (compression/encryption) made by FNFS. Even though these few cases, FNFS is appropriated to reach the proposed goals.

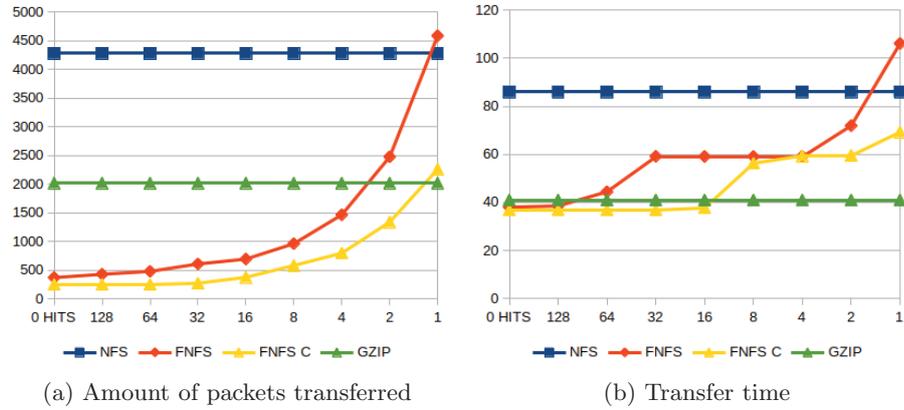


Fig. 3: Results for MEO satellite.

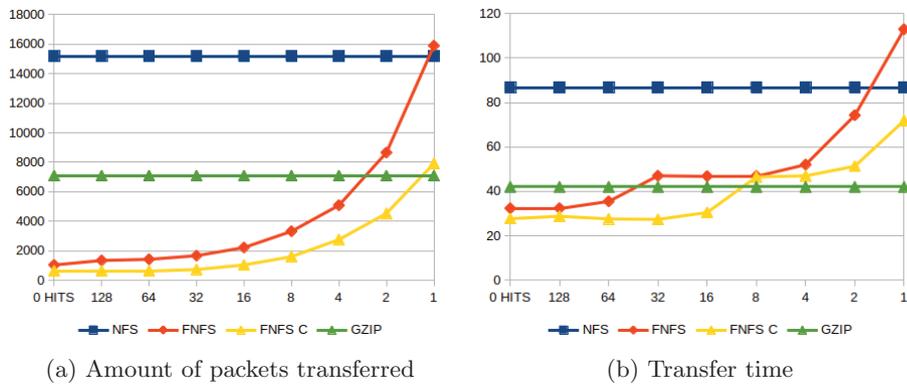


Fig. 4: Results for 4G/LEO.

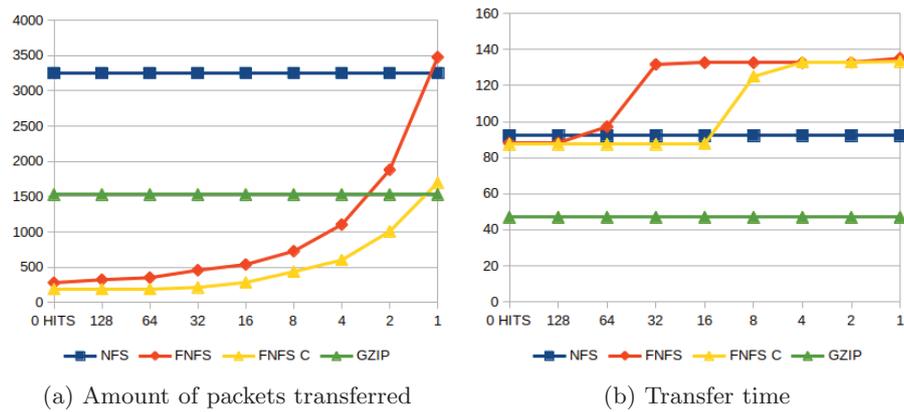


Fig. 5: Results for VSAT.

5 Conclusions and Future Works

This work proposes to apply the “*filtering useless data at the source*” approach on applications which read a large amount of raw data from a remote server. The filtering rules are specified outside the application and are sent to the server which applies them, afterwards it sends only useful records and encoded useless records to the client. The Client rebuilds useless records with dummy data which will be discarded by the application. This approach reduces network usage, cache buffer utilization, transmission costs in per-packet leased links and battery usage on remote autonomous IoT devices.

To evaluate the proposed approach, popular user-space NFS Client and NFS Server were modified to include filtering capabilities resulting in FNFS. FNFS was designed to keep compatibility with unmodified NFS versions. Moreover, applications that use the protocol don't have to be modified to work with it.

On the other hand, different environment benchmarks reveal a better use of bandwidth, and in some cases lower transfer time. Also, as a consequence of data compression, it reduces the amount of packet transferred. These features translate into buffer saving, smaller numbers of packets to be processed by client and server kernels and fastest communications. In addition, it also helps to keep battery energy when talking about IoT devices, since they usually have limited battery supply. Finally, it may also contribute to decrease communication costs due to decreasing packet transmission for those services whose cost is proportional to the data transferred.

Currently, FNFS is a prototype which can be improved to be used in production environments and only supports a single preloaded filter. Future versions should allow dynamic and multiple filters configurations and data transfer encryption.

References

1. User-space NFSv3 Server, <https://unfs3.github.io/>
2. HSFS - an NFS client via FUSE, <https://github.com/openunix/hsfs>
3. J. Kreyig, H. Schukat, H.Ch. Zeider, H. Diel, H. Weber, An intelligent disk controller - A processor system for file management and query functions, *Microprocessing and Microprogramming*, Volume 25, Issues 1-5, Pages 55-60, ISSN 0165-6074, 1989.
4. ZLIB, A Massively Spiffy Yet Delicately Unobtrusive Compression Library-<http://www.zlib.net/>.
5. Bruce R. Elbert, *The Satellite Communication Applications Handbook*, Artech House Space Applications Series, ISBN 1580534902, 2003. Pag. 327 and 406.
6. Satellite Internet access. https://en.wikipedia.org/wiki/Satellite_Internet_access
7. N. J. Upadhyay, *Analyzing VoIP for 3G / 4G Wireless Networks*, 2009.

Gestión remota de dispositivos IOT mediante técnicas de mensajería instantánea empleando Bots

Carlos Binker¹, Hugo Tantignone¹, Guillermo Buranits¹, Eliseo Zurdo¹, Diego Romero¹, Rubén Darío Moreira¹, Maximiliano Frattini¹

¹Universidad Nacional de La Matanza, Florencio Varela 1903 (B1754JEC) -- San Justo, Buenos Aires, Argentina

{cbinker, htantignone, gburanits, djromero, rmoreira}@unlam.edu.ar; {ezurdo, mfrattini}@alumno.unlam.edu.ar

Abstract. Este trabajo plantea como eje principal gestionar un hardware IOT remotamente mediante mensajería instantánea de texto, empleando *Telegram* [1]. Telegram permite definir *bots* [2] (aféresis de Robots) y administrarlos a través de un *token* que la misma aplicación genera en forma aleatoria a través de su bot principal, denominado *BotFather* [3]. Telegram brinda una *API* [4], que permite que los bots interactúen con nuestro sistema. El bot estará alojado en el chip ESP32. El ESP32 es un SOC (System on chip) [5]. Este SOC incorpora WIFI, bluetooth, sensores, conversores AD y DA, etc. Como caso de estudio se propone controlar mediante un bot a un conjunto de leds y a un sensor; este último elemento será simulado mediante un pulsador. Será necesario para ello el empleo de interrupciones. Además se requerirá el uso de Telegram (versión web o móvil) y el entorno de programación Arduino. Los comandos del bot se activarán anteponiendo el símbolo “/”, o bien configurando teclas para cada comando.

Keywords: Telegram, API, Bot, ESP32, IOT

1 Introducción

Cuando se quiere acceder a un host en internet se lo puede hacer a través de su dirección IP o de su nombre, que se extrae de una tabla donde se encuentran mapeadas las direcciones IP con sus respectivos nombres de Host (El Domain Net System o DNS, se encarga de actualizar las tablas de direcciones y nombres). Con el crecimiento de internet y la proliferación de computadoras personales y dispositivos dentro de las redes privadas con servicios de comunicación y acceso a internet brindados por ISPs se hizo necesario el desarrollo de un método para acceder desde internet a los hosts de la red local conectada al router. La asignación de la dirección IP para los clientes se puede hacer en forma estática (se mantiene la dirección IP a lo largo del tiempo) o en forma dinámica (la dirección IP puede variar en el tiempo). La asignación dinámica le impediría a un cliente externo tener la dirección actualizada a la cual comunicarse, por lo cual será necesario un servicio de publicación de las direcciones IP actualizadas. Ver Fig. 1. Para evitar esto, ya que no siempre es fácil tener una IP pública fija se propone

la solución de mensajería por medio de Telegram mediante el empleo de bots, ya que nos independizamos de la capa de red. En nuestro caso nuestro host es un SOC ESP32, que a través de su IP privada accedemos a él a través de la red WIFI local conectada a Internet. El SOC (ESP32) estará configurado en modo STATION y albergará el bot.

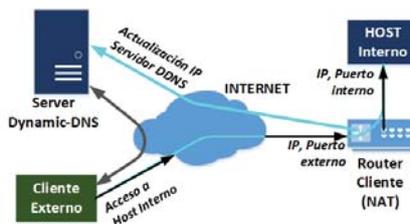


Fig. 1. Esquema de un servicio NO-IP usando una técnica de DNS dinámico

1.1 Esquema de la Plataforma Telegram

Telegram está creado sobre una serie de servidores distribuidos. Dichos servidores, para comunicarse entre sí, utilizan un protocolo propio llamado *MTPProto* [6]. La razón del uso de este protocolo propietario es la mejora en seguridad como también el envío de mensajes, sobre todo vídeo e imagen. Haciendo un poco de historia podemos ver que hay dos versiones de *MTPProto*. La primera versión se utilizó en 2014. Los mensajes en *MTPProto* eran cifrados con el algoritmo SHA-1 [7]. Por un reporte en enero de 2015, en donde el investigador Juliano Rizzo reveló un error en el funcionamiento de SHA-1 que originó una vulnerabilidad al interceptar los mensajes, en 2016 se señaló un posible reemplazo del SHA-1 por el SHA-2. Por lo tanto en 2017, se lanza la segunda versión. El cifrado se reemplazó a SHA-256 con mayor cantidad de bytes de carga útil. Para complementar lo dicho observar la Fig. 2.



Fig. 2. Encriptación de mensajes Telegram mediante SHA-1. Ubicación del MTPProto en el stack de protocolos de comunicaciones

1.2 API de Telegram asociada para el manejo de bots

La API de Telegram permite la interacción de los bots con un sistema. En nuestro caso ese sistema será un hardware a controlar remotamente a través de una interfaz de

usuario, que es en concreto el bot. Los bots de Telegram son cuentas especiales que no están ligadas a un número de teléfono. Al igual que la cuenta de un usuario, el acceso a la misma es por medio del *Alias*, que se accede anteponiendo @ al nombre del bot, o bien se accede por el propio nombre. Estas cuentas sirven como una interface para albergar código que puede estar ejecutándose en cualquier lugar del mundo en un servidor. Su uso es totalmente transparente, ya que los usuarios no necesitan conocer nada absolutamente sobre cómo el protocolo de encriptación MTProto funciona. Los servidores de Telegram manejarán todo lo referente a la encriptación de los mensajes y la comunicación con la API de una manera muy sencilla. Esta comunicación con los servidores de Telegram a través de la API se da vía una simple conexión https. Todas las consultas a la API Telegram Bot deberán realizarse de esta manera:

https://api.telegram.org/bot<token>/METHOD_NAME.

Se crearán dos tipos de bots, uno que suministra los comandos anteponiendo el símbolo “/” y el otro bot suministra los comandos a través de un teclado. El bot creado para la primera situación se llama CACIC_2019, mientras que el bot creado para la segunda situación (comandos a través de teclas) se llama CACIC_2019_Key. A título de ejemplo con el token suministrado para CACIC_2019, la forma de acceder desde la web sería a siguiente:

<https://api.telegram.org/bot904682755:AAHG6B1SsF3j6jJdMXJxUklcGBaUg2lEcaU/getme>

La API soporta los métodos http GET y POST. Ante una petición (request), la respuesta (reply) de la API bot es un objeto JSON como el siguiente:

```
{"ok":true,"result":{"id":904682755,"is_bot":true,"first_name":"CACIC_2019",
"username":"cbinker5bot"}}
```

Tal como se observa, la API siempre devuelve un campo de tipo Boolean denominado “ok” y otro campo denominado “result”, en donde puede tener un campo opcional String denominado “description” con una descripción del resultado.

Si la petición hacia el bot fue satisfactoria, el campo “ok” recibido en el objeto JSON es igual a *true* y si además el resultado de la consulta fue localizado se nos devolverá el campo “result” con todos los resultados, tal como se indica en el ejemplo precedente. En cambio, si la respuesta del campo “ok” es igual a *false*, se devolverá un código de error, tal como el siguiente ejemplo.

```
{"ok":false,"error_code":401,"description":"Unauthorized"}
```

En la Fig. 3 se puede observar claramente la interacción entre todos los componentes, es decir entre el cliente, el bot, la API de Telegram y el hardware remoto a controlar.

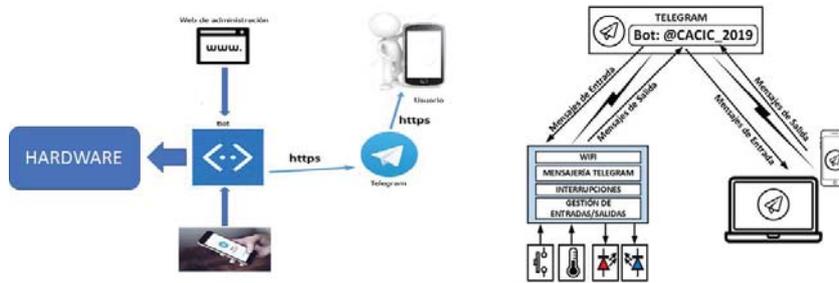


Fig. 3. –Interacción entre usuario (cliente), el bot, la API de Telegram y el hardware remoto

2 SOC ESP 32

Este dispositivo es un SOC que está adquiriendo una enorme popularidad debido a su bajo costo y la implementación de diferentes entornos de programación, tales como LUA y Python. Pero su enorme potencial radica en el hecho de que soporta también el IDE de Arduino y trabajar con él es como si realmente estuviéramos trabajando sobre alguna de las placas Arduino, que como ya sabemos goza de una enorme comunidad abocada al desarrollo de código y de librerías. Todo esto puede ser utilizado en forma transparente para este dispositivo y así evitamos la utilización de un doble microcontrolador. Este dispositivo se destaca por sobre todo porque ya resuelve todo lo concerniente a la comunicación de dispositivos a través de redes, ya que soporta WIFI, Bluetooth, Hardware criptográfico, SPI, I2C, I2S y hasta Ethernet. Además incorpora un sensor de temperatura y sensores Touch capacitivos, además de conversores DAC, ADC, etc. Posee un microprocesador con dos núcleos de 32 bits, lo que lo hace ideal para procesamiento paralelo. Cualquier pin del ESP32 puede configurarse para el control de interrupciones. A continuación exponemos el diagrama en bloques para dar una idea general de su potencialidad, como así también el correspondiente PIN-OUT, (ver Fig.4 para mayor detalle).

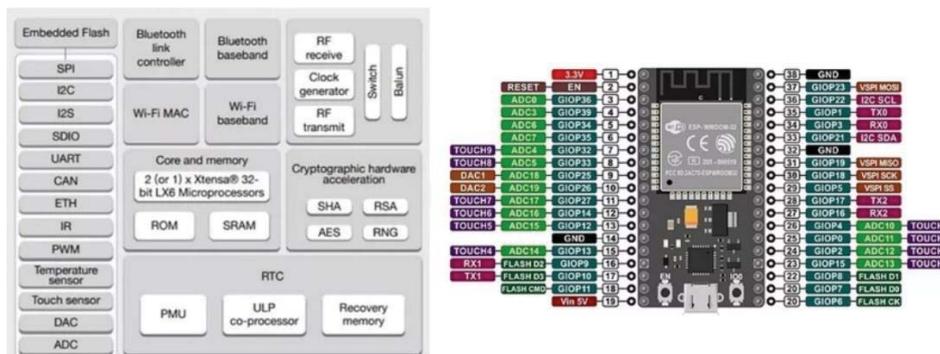


Fig. 4. Diagrama en bloques del ESP32

3 Caso de estudio empleando ESP32, Telegram y Arduino IDE

El prototipo de hardware que se utilizará es el mostrado en la Fig. 5:

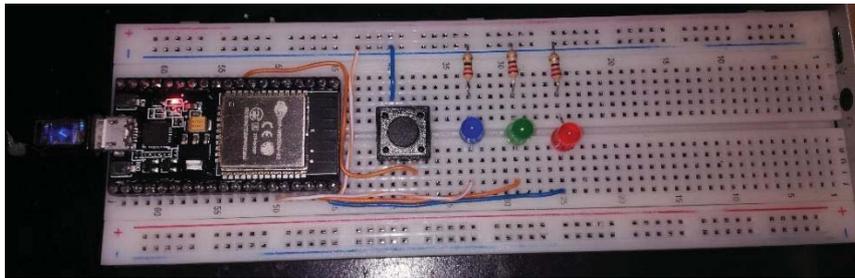


Fig. 5 - Hardware a controlar por el bot de Telegram empleando el NodeMCU ESP32

El bot que lo controlará tiene el siguiente aspecto (ver Fig. 6), el mismo fue configurado a través de los comandos del BotFather, y lo apreciamos tanto en su versión web cómo móvil. También, puede configurarse el bot para que los comandos se introduzcan a través de un teclado, tal como puede observarse en la Fig. 7

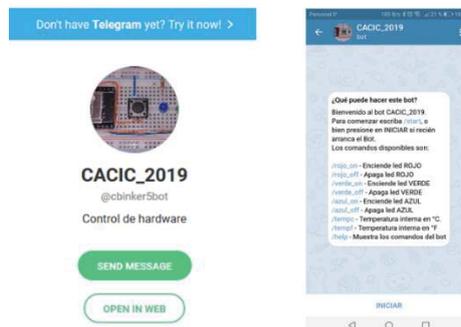


Fig. 6. - Bot que controlará el hardware (CACIC_2019), version web y móvil por comandos

Recordemos que para invocar al bot se usa el alias y éste se corresponde con el nombre del bot. Para acceder en forma web debe escribirse el siguiente comando:

t.me/cbinker5bot (acceso al bot por comandos utilizando “/”), donde en este caso *cbinker5bot* es el username del bot y se corresponde con el Alias (*@cbinker5bot*). En cambio para el acceso al bot que emplea un teclado para los comandos, el acceso web es *t.me/cacickeybot*. Notar también que obligatoriamente el administrador del bot debe terminar en bot.



Fig. 7. Bot que controlará el hardware (CACIC_2019_Key), version web y móvil por teclado

4 Desarrollo de la experiencia

4.1 Software empleando el IDE de Arduino

El software que controlará remotamente el hardware por medio del bot es el siguiente (se exponen las partes más sobresalientes y se omiten algunos saltos de líneas), dada su extensión:

4.1.1. Bot que utiliza los comandos anteponiendo el símbolo “/”

1. Para ambos fuentes con sendos bots, las librerías a emplear son las siguientes:

```
#include <WiFi.h>
#include <WiFiClientSecure.h>
#include <TelegramBotClient.h>
```

Para el correcto funcionamiento, debe estar instalada la librería ArduinoJson version 5.13.5, ya que es una dependencia del resto de las librerías intervinientes.

2. Variable de entorno para el sensor de temperatura interno del ESP32

```
#ifdef __cplusplus
extern "C" { uint8_t temprature_sens_read(); }
#endif
```

3. Definición de variables globales:

PULSADOR en el GIOP 25, y pongo los leds en los pines 23, 22 y 21.

```
const int pulsador=25; const int led_rojo=23;
```

```
const int led_verde=22; const int led_azul=21;
```

4. Código para manejo de interrupciones:

```
portMUX_TYPE mux = portMUX_INITIALIZER_UNLOCKED;
bool pirTriggered = false; // Al pulsar pasa a true
long chatId = 464201191; // chatId del usuario que
interactúa con el bot
String msg = "Pulsador presionado"; // Mensaje al bot
```

5. Manejo de la interrupción por flanco descendente "Rising"

```
void IRAM_ATTR handleInterruptRising() {
portENTER_CRITICAL_ISR(&mux); Serial.println("Rising");
pirTriggered = true; portEXIT_CRITICAL_ISR(&mux); }
```

6. Instanciación de las credenciales para la conexión WIFI (ssid y password)

```
const char* ssid = "IPLAN-306232"; // nombre del ssid
const char* password = "NDUND4XYGYK3";
```

7. Instanciación del Bot de Telegram

```
const String botToken =
"904682755:AAHG6B1SsF3j6jJdMXJxUklcGBaUg2lEcaU";
```

8. Instanciación del cliente ssl para comunicarse con la web API de Telegram.

```
WiFiClientSecure sslPollClient;
```

9. Instanciación de la clase client (argumentos: el Token y un cliente seguro ssl)

```
TelegramBotClient client(botToken, sslPollClient);
```

10. Declaración de función destinada a recibir mensajes desde el Bot de Telegram.

```
void onReceive (TelegramProcessError tbcErr,
JwcProcessError jwcErr, Message* msg)
{Serial.println("onReceive") ...;
```

11. Ejemplo de tratamiento de un comando (por ejemplo encender el led rojo)

```
if (msg->Text == "/rojo-on" ) {
client.postMessage(msg->ChatId, String("Encendiendo el
led rojo")); digitalWrite(led_rojo, HIGH); }
```

12. Arranque del WIFI utilizando las credenciales definidas

```
void setupWiFi()
{
Serial.println();Serial.printf("Intentando conectar a
la red %s ",ssid); Serial.println();
WiFi.begin(ssid, password); //Método que arranca WiFi.
while (WiFi.status() != WL_CONNECTED) {
```

```

delay(500); Serial.print(".");} // Mientras no conecta
imprimo ....
Serial.println(); Serial.println("OK");
Serial.print("IP address");
Serial.println(WiFi.localIP());

```

13. Imprimo OK y muestro la IP asignada mediante DHCP por el router WIFI.

```

// Imprimo la intensidad de la señal en dbm.
Serial.print("Strength ...: ");
Serial.println(WiFi.RSSI()+ String(" dbm"));
Serial.println();}

```

14. Función setup, se ejecuta una sola vez.

```

// Habilito como salida el pin GIOP23. Lo mismo para los otros leds.

```

```

void setup() {pinMode(led_rojo, OUTPUT);

```

15. Manejo de interrupciones

```

pinMode(pulsador, INPUT_PULLUP); // Resistencias
internas de Pull Up.
attachInterrupt(digitalPinToInterrupt(pulsador),
handleInterruptRising, RISING);

```

16. El cliente llama a las funciones para el procesamiento de datos o errores.

```

client.begin( onReceive, onError);

```

17. Pregunto si se disparó un evento, en este caso si se oprimió un pulsador en loop.

```

void loop() {
if (pirTriggered) {
client.postMessage(chatId, msg); // ChatId user
pirTriggered = false; // pirTriggered en true}

```

18. Para procesar los datos recibidos, este método deberá ser invocado en forma continua dentro del main loop.

```

client.loop();

```

4.1.2. Bot que utiliza los comandos ingresados mediante teclado

En esta situación el código se modifica de la siguiente manera:

En la función onReceive, para instalar el teclado se ejecuta la siguiente sentencia:

```

if (msg->Text == "/start") client.postMessage(msg-
>ChatId,String("Bienvenido. Ingrese los comandos indicados.
Escriba /help para mas información.\n"), board);

```

La clave es la palabra reservada **board**. El teclado ya se instala al darle *start* sólo una vez. El teclado se activa con las siguientes sentencias:

```
// Añadiendo cuatro filas al teclado
String row1[] = {"rojo-on", "verde-on", "azul-on"};
String row2[] = {"rojo-off", "verde-off", "azul-off"};
String row3[] = {"TempC", "TempF"};
String row4[] = {"help"};

// Configurando el teclado, cantidad de teclas por fila. Observar sólo va “;” al final
push(3, row1)
push(3, row2)
push(2, row3)
push(1, row4);
```

4.2 Resultados obtenidos

Se indican en la Fig. 8 los resultados obtenidos desde el bot para el control del hardware. Este bot emplea comandos anteponiendo el símbolo de “/”. En la Fig. 9 se ve el resultado de aplicar el bot con los comandos a través de un teclado en la pantalla.

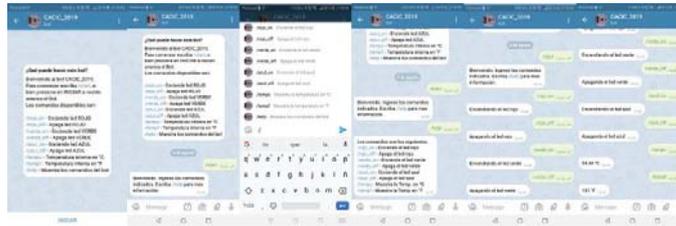


Fig. 8. Resultados arrojados por el bot con comandos precedidos por “/”



Fig. 9. - Resultados arrojados por el bot con comandos dados por teclas en la pantalla.



Fig. 10. – Información recepcionada en ambos bots al activarse un sensor (PULSADOR). Se muestra el mensaje Pulsador presionado.

Ejemplo de Capturas del monitor serie del JSON recibido al prender y apagar el led rojo:

```
{
  "ok": true,
  "result": {
    "message_id": 1464,
    "from": {
      "id": 904682755,
      "is_bot": true,
      "first_name": "CACIC_2019",
      "username": "cbinker5bot",
      "chat": {
        "id": 464201191,
        "first_name": "Carlos",
        "username": "Pqj23",
        "type": "private"
      },
      "date": 1565008733,
      "text": "Encendiendo el led rojo"
    }
  }
}
```

```
{
  "ok": true,
  "result": {
    "message_id": 1466,
    "from": {
      "id": 904682755,
      "is_bot": true,
      "first_name": "CACIC_2019",
      "username": "cbinker5bot",
      "chat": {
        "id": 464201191,
        "first_name": "Carlos",
        "username": "Pqj23",
        "type": "private"
      },
      "date": 1565008746,
      "text": "Apagando el led rojo"
    }
  }
}
```

5 Conclusiones y trabajo futuro

1. La mensajería constituye un excelente método de control a través de Internet sin la necesidad de caer en un esquema de un servicio NO-IP como lo explicado en la introducción.
2. Cada vez es mucho más el interés por el empleo de bots y se desarrollan constantemente nuevos frameworks a fin de facilitar la escritura del código.
3. En un futuro trabajo se prevee la utilización de bots pero para que se comuniquen directamente entre dispositivos sin intervención humana.

6 Referencias

1. <https://telegram.org/>
2. Designing bots. 1st Edition. Amir Shevat. Publicación: mayo del 2017. Editorial OREILLY. ISBN-13: 978-1491974827
3. <https://core.telegram.org/bots#6-botfather>
4. <https://core.telegram.org/bots/api>
5. Internet of Things Projects with ESP32. Agus Kurniawan. Publicación: 30 de marzo de 2019. Editorial Packt. ISBN-13: 978-1789956870
6. <https://core.telegram.org/mtproto>
7. INTERNATIONAL JOURNAL FOR ADVANCE RESEARCH IN ENGINEERING AND TECHNOLOGY (Paper). CHAITYA B. SHAH, DRASHTI R. PANCHAL. Volume 2, Issue X, Oct 2014. ISSN 2320-6802.

Implementación de Red de Internet de las Cosas Usando OpenMote y Raspberry Pi

Matías Gastón Casteluccio, Sacha Agustín Niemetz,
Juan Carlos Taffernaberry y Rodrigo Gonzalez

GridTICs, Universidad Tecnológica Nacional, Rodriguez 273, Mendoza, M5502AJE,
Argentina.

casteluccio.m@gmail.com, sanietz@hotmail.com, {cartaffe,
rodrazlez}@frm.utn.edu.ar

Resumen Las redes de Internet de las Cosas (IoT) han ido ganando popularidad en los últimos años, debido a un amplio campo de aplicación y uso. En el presente trabajo se propone una mejora respecto a un desarrollo anterior basado en la Computadora Industrial Abierta Argentina (CIAA). Se propone utilizar una computadora embebida Raspberry Pi para centralizar la implementación de diferentes componentes del sistema IoT. Los resultados de las pruebas llevadas a cabo en laboratorio y en campo demostraron la correcta operación del sistema IoT propuesto.

Keywords: IoT · Raspberry Pi · 6LoWPAN · CoAP.

1. Introducción

La comunicación autónoma entre diferentes elementos sin la participación de los seres humano se conoce hoy como la Internet de las Cosas (IoT, por sus siglas en inglés)[1]. Estos son dispositivos electrónicos embebidos dentro de objetos de uso cotidianos que permiten su conexión a Internet, con el objetivo de lograr el acceso remoto a los datos generados, o el accionamiento de actuadores a distancia. Son implementados generalmente con microcontroladores con bajo poder de procesamiento, poseen comunicación inalámbrica y sensores que permiten obtener valores de parámetros del medio ambiente tales como temperatura, humedad, posición, entre otros. Estos objetos normalmente están interconectados y forman redes de área personal (PAN, por sus siglas en inglés).

Las redes de área personal pueden ser cableadas o inalámbricas. En este último caso se denominan redes inalámbricas de área personal (WPAN), siendo el estándar IEEE 802.15 el más utilizado. Además, la IEEE desarrolló el estándar IEEE 802.15.4 [2] para ser aplicado a redes inalámbricas de baja potencia y baja tasa de transferencia, características típicas de las WPANs, denominadas redes LLNs (Low-Power and Lossy Networks). Este estándar define las capas física y de acceso al medio. Las LLNs tienen requerimientos de transporte de datos mucho menores en comparación con las redes WLAN IEEE 802.11.

Contemplando los requerimientos de la creciente IoT y las restricciones de las LLNs, la IETF desarrolló el protocolo 6LoWPAN [3] en el estándar RFC-4944.

Este posibilita el uso del protocolo de Internet IPv6 en redes LLNs y define un conjunto de adaptaciones de la pila de protocolos estándar IPv6 entre la capa de enlace y red, nombrada como *Adaptation Layer*.

Adicionalmente, IETF desarrolló estándares para las capas superiores, como Constrained Application Protocol (CoAP)[4], con bibliotecas disponibles en numerosos lenguajes de programación. Es deseable el desarrollo de aplicaciones con estándares abiertos, pues asegura interoperabilidad entre diferentes sistemas.

El grupo de investigación GridTics está trabajando en el desarrollo de tecnologías IoT desde hace varios años. Se pueden mencionar los siguientes proyectos: "Livres: Análisis y evaluación de características relevantes de las WSN aplicadas al manejo y sensado en agricultura de precisión"(2010-2012), "SIPIA: Estudio de campo de red de sensores inalámbricos para adquisición de parámetros ambientales, de uso en investigaciones agronómicas y biológicas"(2012-2014) y "GW-CIAA-IoT: Gateway con CIAA para red inalámbrica de IoT"(2015-2016). <https://www.overleaf.com/6881371211tgsnhnjcxpk>

Este último proyecto realizaba la interconexión entre una red de sensores y una red con soporte IPv6, está completamente desarrollado en los siguientes trabajos científicos [5][6]. El gateway, realizado con la placa CIAA [7], ejecuta el sistema operativo FreeOSEK [8]. Al no contar con soporte para protocolos IoT, el trabajo implicó la implementación de los protocolos IPv6 y SLIP6. Por otra parte, el almacenamiento y presentación de los datos obtenidos en los nodos fue realizada en un host con arquitectura x86 y sistema operativo GNU/Linux conectado a la red IPv6. La debilidad de este esquema es la pérdida de información en caso de no haber conectividad entre el gateway y el host, pues los datos adquiridos no pueden ser almacenados. Este problema recién se puso de manifiesto en el despliegue de una red IoT en producción, al dejar el host de almacenamiento y presentación en el laboratorio del GridTics, y el gateway en campo. Para solucionar este inconveniente se realizó el traslado de dicho host a campo, pero esto retrazó su despliegue, pues se debió hacer acondicionamientos físicos y energéticos para instalarlo en campo, sin contar que sería necesario tener un proveedor de Internet con soporte IPv6 en el lugar.

Otra alternativa analizada para solucionar este inconveniente fue instalar todas las aplicaciones del host en el gateway. FreeOSEK no soporta lenguajes de programación de alto nivel, y solo se pueden desarrollar aplicaciones en lenguaje C. Tampoco existen bibliotecas para manejo de periféricos ni protocolos IoT, como CoAP, para facilitar el desarrollo. Además la comunidad que da soporte a este sistema operativo no es muy activa. Debido a lo anterior, esta alternativa fue descartada.

Por lo expuesto, el presente desarrollo es una mejora del trabajo anterior, solucionando los inconvenientes mencionados. El objetivo principal de este proyecto es desplegar un sistema de IoT compuesto por una red de sensores inalámbricos de bajo consumo empleando protocolos abiertos y estándares, y concentrar en un solo dispositivo todo el software necesario para implementar exitosamente los servicios requeridos.

El resto del trabajo se desarrolla de la siguiente manera. La sección 2 detalla los componentes con los que cuenta el sistema, en la sección 3 se comparten los resultados obtenidos y finalmente las conclusiones a las que se arribó en el presente trabajo.

2. Componentes del Sistema

Se enumeran y describen a continuación todos los componentes utilizados para constituir el sistema completo de IoT.

2.1. Red de sensores Inalámbricos

Los nodos que componen estas redes suelen estar limitados por el consumo energético al estar usualmente alimentados por baterías, lo que permite desplazarse en el área a controlar. La distancia entre estos nodos puede llegar hasta 100 metros con visión directa entre antenas, así se abarcan grandes extensiones con un número reducido de dispositivos. Cada nodo debe comunicarse al menos con otro de los nodos de la red hasta llegar al nodo coordinador, haciendo múltiples saltos y formando una red tipo malla. Debido que un nodo puede encontrar diferentes rutas para alcanzar el coordinador, este tipo de redes posee redundancia, y si algún nodo falla o agota su energía, el resto puede encontrar una ruta alternativa para llegar al coordinador.

Para el diseño de la red fueron usados nodos sensores OpenMote CC2538, los cuales poseen sensores de temperatura, humedad y luz, entre otros. A su vez disponen de zócalos adaptadores para conexión USB y alimentación por baterías [9]. Los mismos ya estaban disponibles por haber sido usados anteriormente en otro proyecto.

Los nodos utilizan el estándar IEEE 802.15.4 el cual define el nivel físico y el control de acceso al medio de redes inalámbricas de área personal con tasas bajas de transmisión de datos para el ahorro máximo de energía. Se utilizó el estándar 6LoWPAN (IPv6 over Low power Wireless Personal Area Networks) en la subcapa de adaptación de red, lo que posibilitó el uso de IPv6 sobre redes basadas en el estándar IEEE 802.15.4. Esto permite que los nodos de la red inalámbrica puedan comunicarse directamente y extremo a extremo con otros dispositivos IPv6.

El protocolo usado en capa de red fue IPv6, donde su gran capacidad de direccionamiento es particularmente útil en este tipo de redes dada la cantidad de nodos sensores que pueden ser utilizados. Por otra parte, el uso de este protocolo elimina la necesidad de servicios como DHCP y NAT.

En la actualidad existen desarrollos y soluciones que permiten el despliegue de redes de sensores basadas en los protocolos seleccionados. Entre las posibles soluciones, se seleccionó el sistema operativo Contiki OS [10] para ser ejecutado en cada nodo de la red. Sus principales ventajas comparativas son compatibilidad con 6LoWPAN, stack completo de TCP/IP y RPL.

La figura 1 muestra los nodos sensor y coordinador.

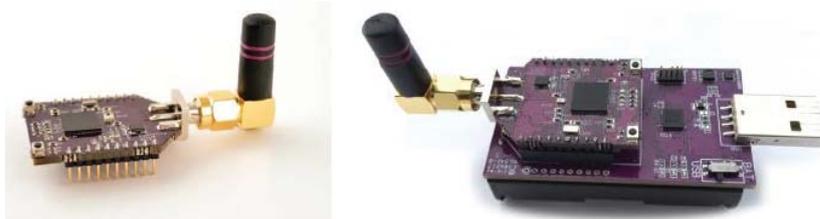


Figura 1. Nodo sensor (izquierda) y nodo coordinador (derecha).

2.2. Servidor de datos

Esta aplicación es ejecutada en cada uno de los nodos. El objetivo es proveer valores de temperatura ambiente, humedad e iluminación cada vez que un cliente le haga una petición.

Inicialmente se utilizó un stack completo de protocolo TCP/IP y soporte para HTTP-REST, pero los resultados obtenidos no fueron satisfactorios. Este tipo de redes tiene un alto porcentaje de probabilidad de pérdida. Por tanto, el uso de HTTP-REST sobre TCP/IP resulta desaconsejable.

Se sustituyó por una implementación de protocolo CoAP/UDP por su rapidez y eficiencia. Es un protocolo con arquitectura REST, heredado de HTTP. Utiliza las mismas primitivas de servicio (GET, POST, PUT, DELETE). A diferencia de HTTP, CoAP distingue mensajes confirmados y no confirmados. Los mensajes confirmados requieren un reconocimiento por parte del receptor, los no confirmados no. Fue seleccionada la primera opción para asegurar que los mensajes lleguen a destino a pesar de funcionar sobre un protocolo no orientado a la conexión como es UDP.

Adicionalmente, CoAP codifica los mensajes en formato binario en lugar del formato texto empleado en HTTP, lo que supone una reducción significativa de la longitud de las cabeceras y del propio mensaje. Esto permite un mejor aprovechamiento del ancho de banda disponible.

En capa de transporte, se utilizó UDP por su sencillez y reducido tamaño de cabecera. No fue necesaria la utilización de un protocolo orientado a conexión ya que no era necesario disponer control de flujo y confirmación de entrega o recepción. Cabe aclarar que Contiki provee bibliotecas para utilizar UDP y CoAP.

2.3. Nodo Coordinador o Router de Borde

El nodo coordinador es un nodo más de la red inalámbrica, pero posee la funcionalidad de mantener actualizadas las rutas para llegar a cada uno de los nodos sensores de la red. El nodo coordinador difunde el prefijo de la red IPv6, administra y registra los dispositivos de la WPAN mediante radiofrecuencia.

Los nodos OpenMote permiten programar uno de los nodos con adaptador USB como router de borde. El software consulta a los nodos sensores su dirección

MAC y a partir de un prefijo de red establecido, construye y difunde las diversas direcciones IPv6 globales. Tiene la capacidad de soportar la topología de red en malla, por lo que cada nodo está conectado con uno o más nodos de la red. Esta topología reduce el riesgo de fallos, y por ende el mantenimiento periódico. En caso que un nodo falle, los nodos adyacentes propagarán un cambio en la tabla de rutas, notificando a nodos contiguos del cambio en la red, y así sucesivamente, dando confiabilidad.

Se destaca que Contiki utiliza RPL (Routing Protocol for Low Power and Lossy Networks) [11], un protocolo de enrutamiento para redes inalámbricas de bajo consumo de energía y típicamente susceptibles a pérdidas de paquetes. Es un protocolo proactivo basado en vectores de distancia y opera sobre IEEE 802.15.4. Se basa en el modelo multisalto, en el que un nodo que transmite a la estación base, lo hace reenviando sus datos a uno de sus vecinos hasta que llega al destino.

La figura 2 muestra un ejemplo de una red IoT con RPL.

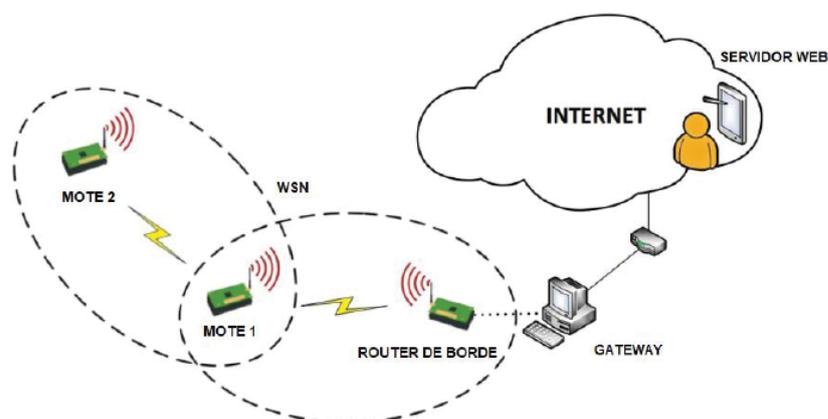


Figura 2. Ejemplo de red IoT con RPL.

Finalmente, Contiki soporta Neighbor Discovery, el cual es un protocolo que mediante la utilización de mensajes especiales ICMPv6 cada dispositivo conoce las direcciones IP del resto de los miembros de la red usando un mecanismo de autoconfiguración, sin necesidad de servicios como DHCP.

2.4. Gateway

Este dispositivo está conectado al coordinador utilizando la interfaz USB, y a Internet (IPv6) por medio de una interfaz Ethernet o WiFi. Su función es adaptar el protocolo IPv6 de Internet a la capa de enlace de la red sensores (802.15.4).

La implementación del gateway se realizó en una placa Raspberry Pi [12], seleccionada de acuerdo a los objetivos de este proyecto. Raspberry Pi es una computadora de bajo costo y tamaño reducido, del orden de una tarjeta de crédito, desarrollada con el objetivo de estimular la enseñanza de la informática en las escuelas. Está formada por una placa que soporta varios componentes necesarios en un ordenador común y es capaz de comportarse como tal. En este proyecto se utilizó la Raspberry Pi 3 B+ que cuenta con las siguientes características: Processor Cortex-A53 (ARMv8) SoC de 64 bits a 1,4 GHz, 1 GB de RAM, LAN inalámbrica IEEE 802.11 de 2.4GHz y 5GHz, Bluetooth 4.2, Gigabit Ethernet, video HDMI , 4 puertos USB 2.0 y micro SD para cargar su sistema operativo y almacenar datos, entre otras cosas.

Esta placa no incluye sistema operativo alguno, se optó por instalar Raspbian-OS basado en Debian, por estar optimizado para el hardware de Raspberry Pi.

La figura 3 muestra una Raspberry Pi con un nodo actuando como router de borde.

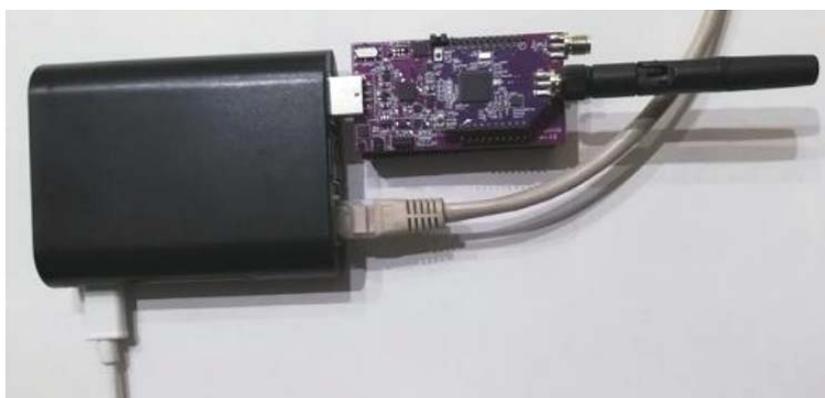


Figura 3. Raspberry Pi con nodo coordinador o router de borde.

La aplicación `tunslip6` [14] se ejecuta en el gateway. Establece una conexión hacia el puerto serie (USB) y genera una interfaz virtual IPv6 que habilita la comunicación de paquetes desde y hacia la red de sensores. Además, permite administrar los parámetros de red IPv6 global que le serán otorgados al router de borde.

2.5. Cliente de datos

Este programa se ejecuta en el gateway y realiza periódicamente una petición de datos a cada nodo, con un periodo que puede ser ajustado por el usuario, como se detallará más adelante. El servidor de datos que se ejecuta en los nodos, al recibir esta solicitud, mide las variables requeridas y luego le responde con dicha información en un formato preestablecido (CoAP). Esta comunicación puede

ser punto a punto, en el caso de que el nodo esté en el rango de visión del coordinador, o multisaltos en caso que el nodo se comunique con el coordinador por medio de otros nodos.

El cliente se desarrolló usando el lenguaje de programación Python [13], cuya filosofía hace hincapié en una sintaxis que favorezca un código legible. Posee una licencia de código abierto compatible con la Licencia pública general de GNU. Este lenguaje es interpretado, multiparadigma y multiplataforma. Contiene una gran cantidad de bibliotecas, tipos de datos y funciones incorporadas en el propio lenguaje, que facilitan la realización de muchas tareas comunes. Las bibliotecas más importantes usadas en este proyecto fueron Coapthon, para implementar el protocolo CoAP, InfluxDB para manejo de bases de datos, y Request para trabajar HTTP/1.1 en Python.

Con la ayuda de estas bibliotecas, el programa cliente configura los nodos asociados a la red, realiza periódicamente peticiones CoAP a los nodos, solicitando los datos adquiridos por los sensores de los mismos, interpreta la trama de respuesta de los nodos, identifica los valores de los parámetros deseados y guarda dicha información en la base de datos influxDB.

El periodo de estas peticiones, es programable por parte del usuario a través de la plataforma de visualización.

2.6. Almacenamiento de datos

Cada vez que la aplicación cliente de datos obtiene información de los nodos sensores, se guarda en un base de datos no relacional. Esta base de datos también está instalada y ejecutandose en el gateway. Se seleccionó InfluxDB [15] debido a que es una base de datos basada en series de tiempo (time-series database), no relacional y de código abierto. Está optimizada para el almacenamiento rápido y de alta disponibilidad de datos de series temporales en campos como el monitoreo de operaciones, la métrica de aplicaciones, los datos de sensores de Internet de las cosas y el análisis en tiempo real. El formato de almacenamiento de datos en InfluxDB es JSON (JavaScript Object Notation), el cual es un formato de texto ligero para el intercambio de datos.

2.7. Servidor de visualización

El servidor permitir la visualización y control de los datos adquiridos por los nodos sensores, y la programación de los tiempos de adquisición de datos, alertas. Además, permite hacer comparaciones entre las variables medidas.

Grafana [16] es la plataforma de visualización seleccionada. Es un software para la visualización, análisis y supervisión de datos. Se trata de un servidor web de código abierto y multiplataforma. Permite crear paneles personalizados para la visualización de datos, configurar alertas y enviar notificaciones. Grafana soporta un variado tipo de bases de datos para la adquisición de información, incluyendo InfluxDB. El servidor de visualización también fue instalada en el gateway.

2.8. Cliente de visualización

Como cliente de visualización se utiliza un navegador o browser con soporte de IPv6. Por ello, la máquina en la que se ejecute debe poseer una dirección IPv6 global, para poder acceder al gateway.

3. Resultados

En la figura 4 se muestra esquemáticamente el diseño final de la red. Se detallan la pila de protocolos y aplicaciones utilizados en cada componente del sistema.

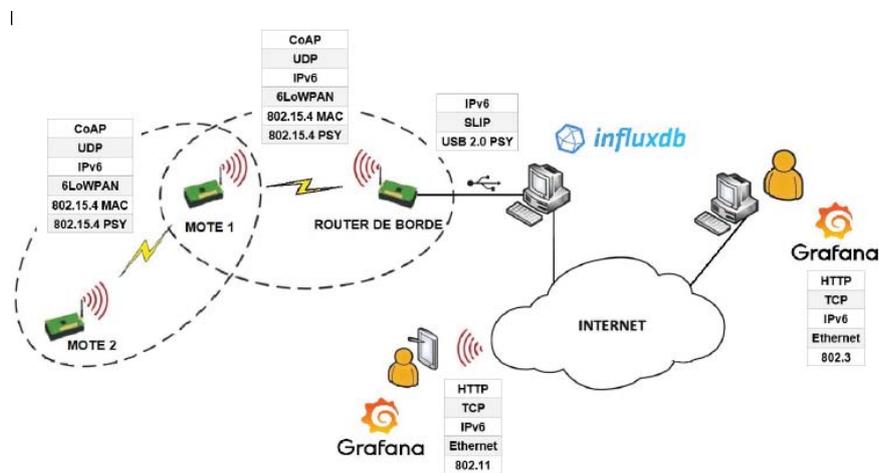


Figura 4. Sistema completo de la red IoT.

Se ensayó exitosamente una red compuesta por 4 nodos sensores OpenMote, un coordinador OpenMote y un gateway implementado con Raspberry Pi. Se obtuvieron distintos gráficos y se observaron las variables ambientales almacenadas.

A continuación, la figura 5 muestra algunas capturas de los paneles creados en Grafana para la visualización de las magnitudes sensadas por la red IoT.

La figura 6 muestra cómo se puede modificar el tiempo de refresco para la adquisición de datos modificando el valor en la solapa *refreshing every*.

Se pudo verificar, tanto en un test bed, como en un despliegue en campo, que las pérdidas de conectividad no provocaron pérdidas de datos, pues los mismos se siguieron almacenando en la base de datos. Adicionalmente, se ensayó el correcto funcionamiento de todo el sistema, usándolo continuamente durante aproximadamente 1 mes en el testbed de GridTics.

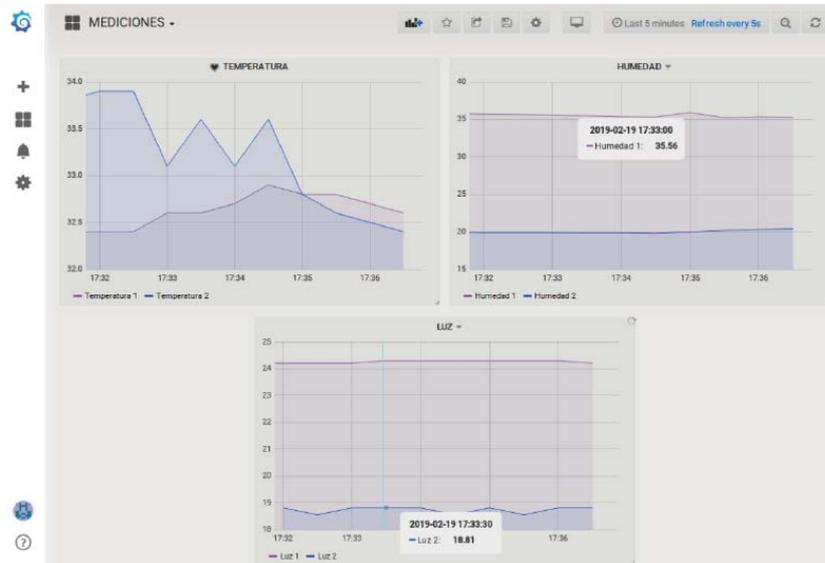


Figura 5. Paneles creados en Grafana para la visualización de diferentes variables ambientales.

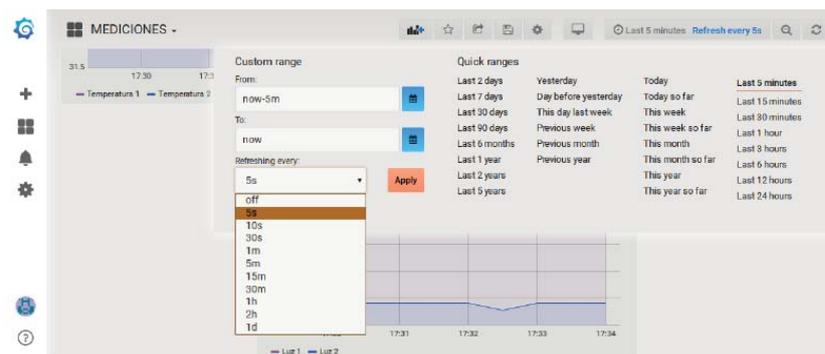


Figura 6. Panel en Grafana para ajustar el tiempo de refresco de las variables ambientales.

4. Conclusiones

A pesar del lento despliegue que está teniendo IPv6 en Internet, el número de dispositivos que demandan este servicio es cada vez mayor. Esto promueve que la transición hacia el nuevo protocolo de red sea ya una realidad inminente. Con la solución que provee este trabajo, los nodos de la red son dispositivos con direcciones IPv6 propias y cuentan a priori con conectividad extremo a extremo con cualquier sistema de la red Internet.

La selección de la plataforma Raspberry Pi, fue fundamental en el éxito del presente trabajo, pues concentra en un solo dispositivo diferentes componentes para el correcto funcionamiento y manejo de una red IoT. Los componentes integrados en ella son: el coordinador de red de sensores, un gateway IPv6, un cliente de datos, el almacenamiento de datos y un servidor de visualización.

Contiki fue escogido como sistema operativo a ejecutar en los nodos sensores. La ventaja respecto a otros software fue la posibilidad de usar un conjunto de herramientas que facilitan el despliegue de diferentes escenarios y también poder realizar simulaciones.

Configurada y establecida la red de sensores, con el propósito de ofrecer servicios sobre los nodos de la red desplegada, se estudiaron e implementaron distintas alternativas para acceder a recursos que ofrecen los nodos, así como un sistema de recolección y almacenamiento de datos. En términos de eficiencia, la solución REST basada en CoAP presenta mayor rendimiento frente a la tradicional empleando HTTP.

Los ensayos de confiabilidad de los datos frente a problemas de conectividad fueron exitosos en todos los casos. Adicionalmente se verificó que el despliegue de la red en pruebas de campo y laboratorio fue mas sencilla, debido a que una buena parte del software estaba integrado en el dispositivo gateway.

Referencias

1. Waher, P.: Learning Internet of Things, 1st Edition. Packt Publishing, UK (2015).
2. Molisch, A.F., Balakrishnan, K., Chong, C.C., Emami, S., Fort, A., Karedal, J., Kunisch, J., Schantz, H., Schuster, U. and Siwiak, K.: IEEE 802.15. 4a channel model-final report. IEEE P802, 15(04), p.0662 (2004).
3. Kushalnagar, N., Montenegro, G., Schumacher, C.: IPv6 over low-power wireless personal area networks (6LoWPANs): overview, assumptions, problem statement, and goals (2007).
4. Shelby, Z., Hartke, K., Bormann, C.: The constrained application protocol (CoAP) (2014).
5. Taffernaberry, J.C., Mercado, G., Pecchia, M., Tobar, S., Verdejo, A., Sayago, J.: Puerta de Enlace para Internet de las Cosas usando Computadora Industrial Abierta. In: XX Workshop de Investigadores en Ciencias de la Computación (WICC 2018), pp. 174-179, ISBN 978-987-3619-27-4. Corrientes, Argentina (2018).
6. Taffernaberry, J.C., Mercado, G.: GW- CIAA-IoT: Gateway con CIAA para red inalámbrica de IoT. In: WICC 2016, ISBN: 978-950-698-377-2. San Juan, Argentina (2016).
7. Proyecto CIAA. Disponible en <http://www.proyecto-ciaa.com.ar>. Mayo 2019.
8. Proyecto FreeOSEK. Disponible en <http://opensek.sourceforge.net>. Marzo 2019.
9. Vilajosana, X., Tuset, P., Watteyne, T. Pister, K.: OpenMote: Open-source prototyping platform for the industrial IoT. In: International Conference on Ad Hoc Networks (pp. 211-222). Springer, Cham (2015).
10. Contiki: The Open Source OS for the Internet of Things. Disponible en <https://www.contiki-os.org>. Mayo 2019.
11. Winter, T.: RPL: IPv6 routing protocol for low-power and lossy networks. IETF (2012).

12. Raspberry Pi. Disponible en <https://www.raspberrypi.org>. Mayo 2019.
13. Python. Disponible en <https://www.python.org>. Mayo 2019.
14. Romkey, J.L.: Nonstandard for transmission of IP datagrams over serial lines: SLIP. RFC 1055 (1988).
15. InfluxDB open-source time series database. Disponible en <https://www.influxdata.com>. Mayo 2019.
16. Grafana: The open platform for beautiful analytics and monitoring. Disponible en <https://www.grafana.org>. Mayo 2019.

Smartcities con LoRaWAN. El Caso de Monitoreo de Condiciones Ambientales de Lagunas en Tiempo Real

Carlos Ivan Piasentini¹, Jose Statkiewicz¹, Diego Bolatti², Reinaldo Scappini²,
Sergio Gramajo²

Universidad Tecnológica Nacional – FRRe, French 414, Resistencia Chaco

¹{cipiasentini, josestatkiewicz}@gmail.com, ²{dbolatti, rscappini, sergio}@frre.utn.edu.ar

Abstract. Las necesidades de gestionar diversas condiciones ambientales en las ciudades impulsan a que se tomen mediciones en tiempo real en los reservorios de agua para ser analizada por expertos y tomar decisiones al respecto. Para ello se analizaron diferentes sensores y medios de comunicación arribando a una propuesta que cubra tales necesidades. Así, este trabajo propone un conjunto de tecnologías relacionadas a smartcities que ayudan a mantener mediciones de diversas variables del agua. Se muestra además el desarrollo del framework realizado y la interacción de los componentes electrónicos y de telecomunicaciones LoRaWAN que conforman el sistema propuesto. Para finalizar se expone un ejemplo ilustrativo de los datos censados.

Keywords: LoRaWAN, LoRa, LPWAN, IoT, Smartcities Frameworks, Mediciones Medioambientales, Multitech, ARM, Kerlink.

1 Introducción

Hoy en día los problemas de urbanización, gobernanza y nuevas exigencias de la población para requerir servicios de mayor calidad han intensificado las formas de generar soluciones innovadoras basadas en diferentes tecnologías y un impacto mínimo en el medio ambiente [1] [2]. La integración inicial de las Tecnologías de Información y de Comunicación (TICs) en las ciudades ha promovido los conceptos de ciudad de la información, ciudad digital e Internet de las Cosas (IoT) [3] [4] que se utilizan para mejorar los estilos de vida de los ciudadanos y que apoyan la toma de decisiones para la gestión de las ciudades con mínima interacción humana [1].

Estos conceptos han sido aplicados para resolver diferentes problemas (medio ambiente, salud, educación, seguridad pública, transporte, aguas, energía, entre otros) y se basan en la interconexión de dispositivos con sensores y electrónica mediante diferentes tipos de redes de comunicación y que convergen en sistemas con algún tipo de “inteligencia” o automatización [3] [5] [6] [7] [8].

Entre los medios de interconexión que se utilizan para estas soluciones, se destaca la tecnología LoRaWAN [9] que utilizamos para nuestra propuesta. Ésta es una especificación de redes LPWAN (Low Power Wide Area Network), en capa 2 por lo que permite conectar diferentes dispositivos gestionando sus canales y parámetros de conexión: canal, ancho de banda, cifrado de datos, etc. [10].

En este trabajo nos centramos en exponer un proyecto de investigación y desarrollo y mostrar el prototipo obtenido utilizando diferentes sensores conectados mediante tecnología LoRa. Los sensores miden en lagunas o reservorios de agua diferentes variables que son utilizadas para gestionar la calidad del agua. El alcance del trabajo es mostrar los aspectos tecnológicos de innovación desde el punto de vista sistémico y no se centra en el análisis de las variables químicas para determinar el comportamiento de aguas. Se muestra, además, aspectos del software desarrollado y las plataformas usadas para registrar sensores LoRa.

Para mostrar lo antedicho, en la sección 2 se realiza una breve revisión de antecedentes teóricos de smartcities y LoRaWan, en la sección 3 se exponen los principales resultados de esta investigación y los componentes que integran la propuesta; en la sección 4 se ejemplifican y analizan la interacción de las partes y el código esencial creado. Por último, se enuncian las conclusiones.

2 Antecedentes

Para clarificar los tópicos tecnológicos y de innovación en los que se centra nuestra propuesta, es necesario revisar algunos conceptos relacionados. Así, en los siguientes apartados explicaremos las principales características en las que se enmarca nuestra investigación; smartcities y la tecnología de comunicaciones LoRaWAN.

2.1 Smartcities

En términos genéricos, la ciudad inteligente es un entorno urbano que utiliza las TIC y otras tecnologías relacionadas para mejorar la eficiencia del rendimiento de las operaciones regulares de la ciudad y la calidad de los servicios (QoS) proporcionados a los ciudadanos urbanos. Como base de nuestra investigación un concepto a poner énfasis es saber cuáles son los componentes genéricos de smartcities (ver Fig. 1).



Fig. 1. Componentes

En la *Fig. 1* se observan cuatro atributos principales: i) sostenibilidad, ii) calidad de vida, iii) urbanización e iv) inteligencia. A su vez hay atributos secundarios como infraestructura, gobernanza, contaminación y desperdicio, la energía, cambio climático, cuestiones sociales y calidad de vida, economía y salud. Además, desde el punto de vista tecnológico existen múltiples alternativas para sensores, infraestructura, microcontroladores, interconexión entre componentes, seguridad y sistemas de respaldo.

En este sentido, podemos afirmar que el entorno, región o ciudad donde se aplique una determinada solución tiene que estar adaptada a diferentes criterios como cultura, estudios de factibilidad técnico-económicos, uso de TICs, normativas locales de uso de telecomunicaciones y espectro, clima, situación política, entre otros aspectos relevantes. En este trabajo se estudiaron diferentes alternativas que son útiles en estas configuraciones de ciudades del norte argentino.

2.2 LoRaWAN

Durante los últimos años, el uso de las LPWANs (redes de área amplia y de baja potencia) basadas en el protocolo de comunicación LoRaWAN, se ha incrementado exponencialmente debido a la necesidad de poder brindar infraestructura al creciente número de soluciones IoT.

Una de las principales ventajas del protocolo LoRaWAN, es que permite al usuario tener el completo control de la red (nodo, gateway y servidor de red); diferenciándose así de otros protocolos de comunicación inalámbrica como por ejemplo LTE.

A la hora de desplegar la infraestructura de una red LoRaWAN, se debe tener en cuenta dos grandes aspectos, el hardware (nodo y gateway) y el software (servidor de red) a utilizar. Respecto al hardware existen diversos fabricantes, pero se destacan entre ellos Microchip, Multitech y Kerlink, siendo estos dos últimos los utilizados y descritos en este artículo.

En lo que respecta al software de red, también existen diversas alternativas y entre ellas se destacan la plataforma TheThingsNetwork y OrbiWAN. Siendo la primera la plataforma más ampliamente utilizada, debido a su carácter open source y de partner de la Lora Alliance [10]. OrbiWAN tiene a su favor, principalmente, la facilidad de uso al tratarse de una solución premium, provee funcionamiento “out-of-the-box”. No obstante, sin importar la plataforma por la que se opte, se deben de realizar un cierto número de consideraciones y configuraciones respecto al funcionamiento de la misma, alguno de estos parámetros, que afectan a la calidad del servicio, son: ancho de banda, velocidad de transmisión de datos, factor de propagación, entre otros.

LoRa (Long Range) es una tecnología de comunicaciones de datos de largo-alcance y bajo-consumo desarrollado por Cycleo, (Grenoble, Francia) y adquirido por Semtech en 2012, miembro fundador de la LoRa Alliance.

Esta tecnología apunta a ser utilizada en dispositivos alimentados por baterías, ya que gracias a su bajo consumo, permite maximizar la vida útil de un ciclo. Es decir, es una gran solución para dispositivos IoT.

Los términos LoRa y LoRaWAN, no hacen referencia a lo mismo. La principal diferencia que existe es que el primer término se refiere normalmente a la

implementación de la capa física, mientras que el segundo es una especificación LPWAN que define un protocolo de acceso al medio y una arquitectura subyacente para este tipo de redes [11].

LoRa es un esquema de modulación de espectro expandido derivado de la modulación CSS (Chirp Spread Spectrum), donde se negocia la tasa de transmisión de datos por la sensibilidad en un canal con un ancho de banda fijo. Implementa velocidad de transmisión de datos variable, utilizando factores de expansión ortogonales, las cuales permiten al diseñador de sistema negociar tasa de transmisión de datos por rango de alcance o energía utilizada, a modo de optimizar el desempeño de la red haciendo uso de un ancho de banda constante [12].

La capa física LoRa opera en el rango de frecuencias 433, 868 y 915 MHz dependiendo de la región (en Argentina opera en el rango 915-928MHz, rango de bandas no licenciadas). Permite la operación de bajo consumo energético, tasa de transmisión de datos máxima de 27Kbps (50Kbps usando modulación FSK) y largo alcance (2-5km en áreas urbanas y hasta 15km en áreas rurales).

El protocolo de acceso al medio LoRaWAN, mantenido actualmente por la LoRa Alliance, permite la gestión de la comunicación entre gateways y dispositivos finales o nodos. Esta comunicación es bidireccional desbalanceada (la tasa de transmisión de subida, desde los nodos al gateway, es mayor que la de bajada).

La arquitectura típica de una red LoRa es la de una estrella de estrellas que incluye tres tipos de dispositivos: nodos finales LoRa, gateways LoRa y servidor de red LoRa (Fig. 2) [13].

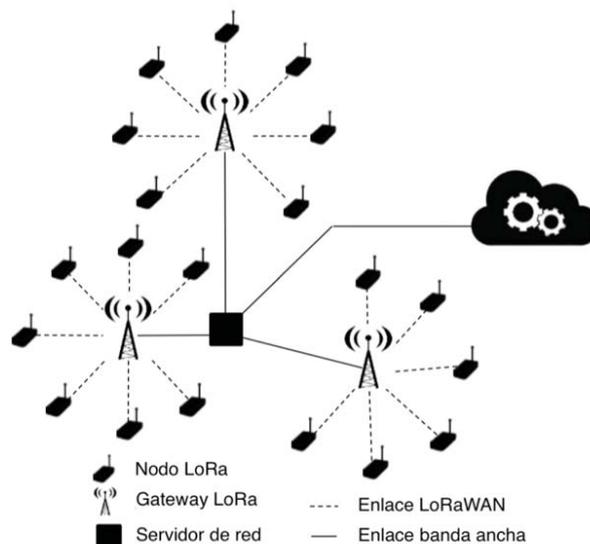


Fig. 2. Arquitectura típica de una red LoRaWAN

Los dispositivos finales LoRa, o nodos LoRa, se comunican con los gateways mediante el protocolo LoRaWAN. Los gateways reenvían la trama LoRaWAN de estos

dispositivos al servidor de red utilizando una interfaz de mayor performance operando sobre el protocolo TCP/IP, como por ejemplo Ethernet, WiFi o 3G.

El servidor de red es el responsable de enrutar al correspondiente servidor de aplicaciones, donde los paquetes de subida (desde los nodos al gateway) son decodificados, y los de bajada (desde el gateway a los nodos) son generados. Para adaptar la manera de realizar esta labor y poder suplir a las demandas de las diversas aplicaciones, existen tres definiciones de clases de operación:

1. **Clase A:** permite la comunicación bidireccional, y lo realiza de la siguiente manera: para transmisiones de subida, los dispositivos utilizan acceso aleatorio al medio (similar al funcionamiento de ALOHA PURO [14]). Para transmisiones de bajada, su ventana de ocurrencia es únicamente inmediatamente luego de una transmisión de subida exitosa, que es donde existe un período durante el cual se crean dos ventanas de recepción en los nodos.
2. **Clase B:** permite la comunicación bidireccional, y lo realiza por medio de la especificación previa de ventanas de recepción. Esto se realiza por medio de beacons, los cuales permiten la transmisión de datos de manera ordenada y por turnos.
3. **Clase C:** también permite la comunicación bidireccional pero es el modo de operación que mayor consumo energético posee ya que se encuentra en constante escucha del medio. El beneficio principal usar este enfoque es que provee la menor latencia de bajada de los tres modos de operación.

3. Propuesta tecnológica y componentes del sistema

El proyecto abordado tiene por objetivo la obtención y procesamiento de datos provenientes de sensores instalados en lagunas de la ciudad de Resistencia, Chaco. Se desea que la información obtenida sirva como herramienta para mantener registros históricos del estado del recurso hídrico natural de la ciudad, a fin de alertar irregularidades y en caso de detectar desechos tóxicos, informar a las autoridades. Para ello se cuenta con una aplicación web, en donde se visualizan todas las medidas de las variables. Esta aplicación se nutre del servidor de red que recibe los paquetes LoRa provenientes del nodo.

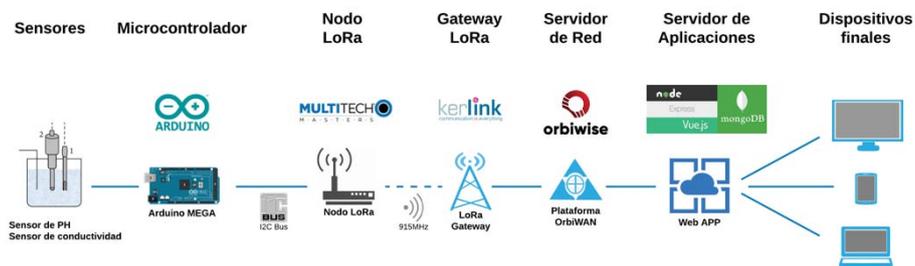


Fig. 3. Arquitectura desplegada para el proyecto de medición de variables químicas.

En la *Fig. 3* se puede observar la infraestructura completa de la solución implementada y de modo genérico puede ser útil para posibles desarrollos futuros; ya que si bien es una solución específica, se puede hacer uso de cualquier tipo de sensores, incluso utilizar actuadores activados por medio de la comunicación descendente.

En primer lugar, se encuentran los dispositivos de censado de variables químicas del agua, los cuales están conectados a un microcontrolador, como por ejemplo el Arduino Mega. Estas permiten generar los datos que luego serán enviados a un nodo LoRa.

Para nuestra propuesta hemos utilizado nodos de la marca Multitech, modelo mDot [15]. La ventaja que poseen estos nodos es la posibilidad de utilizar una plataforma de sistemas embebidos bastante amplia y documentada como ser ARM mBed [16]. Y más importante aún, la existencia de librerías para el manejo del protocolo y ejemplos de uso y adaptación a los nodos mDot de Multitech.

Antes de transmitir los datos, se debe determinar cómo y con qué periodicidad se transmitirán los datos desde el Arduino al nodo LoRa. Para eso se decidió utilizar el protocolo de comunicación i2c, utilizando al nodo como maestro y al Arduino como esclavo. De esta manera, es posible solicitar, desde la aplicación web, la captura y envío de datos por enlace descendente desde el gateway al nodo.

Una vez dispuesta la información en el nodo, es necesario enviarla al servidor de aplicaciones, para ello se debe hacer uso de la red LoRaWAN. Los datos viajan desde los nodos hacia el gateway, y una vez que se obtiene un acuse de recibo por parte de este último, finaliza la transmisión. Es por esto que la tecnología es de bajo consumo energético ya que, si solamente se necesita enviar datos de manera ascendente, se puede poner en modo “*reposo*” al equipo y “*despertarlo*” solamente cuando sea necesario.

4. Ejemplo de Funcionamiento de la Propuesta

Para realizar las pruebas reales, se colocó el nodo a aproximadamente 360 metros de distancia del Gateway (ubicado en la sede de la Facultad), luego también a 620 metros, y no hubo pérdida alguna de paquetes (*Fig 4. Derecha e izquierda respectivamente*).



Fig. 4. Distancias de prueba de comunicación.

La intensidad del paquete entrante medida por el gateway para ambas distancias es de -62dB aproximadamente (*Fig. 5*), intensidad es típica de una conexión buena y confiable.

Direction	Time	FCNT	Port	Status	Data Rate	RSSI
05/08/2019 - 05/08/2019			Port			
Up	31/07/2019 22:11:23	36	15		SF10BW125	-62
Up	31/07/2019 21:32:23	35	15		SF10BW125	-60
Up	29/07/2019 20:02:24	34	15		SF10BW125	-62
Up	28/07/2019 03:08:24	33	15		SF10BW125	-62

Fig. 5. Señal de paquete recibido a ~620 m.

La ubicación de trabajo de la boya de medición de variables químicas será en la Laguna Argüello, ubicada a aproximadamente 620 metros del gateway (*Fig. 6*).

Una vez instalada la boya en la laguna, desde una aplicación web, se procedió a obtener los datos que recibe el Gateway Lora ubicado en la facultad, a través de una API que éste ofrece. Las respuestas proveídas por la API son de tipo HTTP [17] y el formato utilizado es JSON [18].

Cuando se obtienen los datos en la aplicación web, se comparan con una Base de Datos NoSQL implementada con MongoDB [19], para verificar si dichos datos varían de los últimos registrados en la BD. Si varían en un rango determinado, dependiendo del sensor, se registran las nuevas mediciones en la BD, de lo contrario no se realiza ninguna acción.

LORAWAN - MONITOREO DE VARIABLES QUÍMICAS					GATEWAY DIRECTAMENTE	MONGODB ATLAS	GRAFICOS
Fecha	pH	Nitratos (ppm)	Conductividad (mS/m)	Temperatura (°C)			
31/7/2019 22:11:23	7.00	6	5.50	28			
31/7/2019 21:10:23	7.60	30	5.30	29			
31/7/2019 20:09:24	7.50	31	5.60	27			
31/7/2019 19:08:24	8.00	32	5.40	11			
31/7/2019 18:06:23	6.60	2	5.50	15			
31/7/2019 17:05:24	8.00	8	5.60	20			
31/7/2019 16:04:23	7.00	37	5.70	18			

Fig. 6. Tabla de las últimas mediciones obtenidas del Gateway mediante la API.

En las siguientes figuras se puede observar la interfaz de la aplicación web, donde se detallan los datos de las últimas mediciones realizadas (*Fig. 6*), los datos registrados en la BD (*Fig. 7*) y un gráfico de variabilidad en el tiempo de uno de los sensores (*Fig. 8*), en este caso, el sensor de Temperatura.

LORAWAN - MONITOREO DE VARIABLES QUÍMICAS				
Fecha	pH	Nitratos (ppm)	Conductividad (mS/m)	Temperatura (°C)
31/7/2019 22:11:23	7.00	6	5.50	28
31/7/2019 21:32:23	7.60	30	5.30	29
29/7/2019 20:02:24	7.50	31	5.60	27
28/7/2019 03:08:24	8.00	32	5.40	11
28/7/2019 01:55:02	6.70	12	5.50	11
27/7/2019 18:33:23	6.60	2	5.50	15
25/7/2019 22:05:24	8.00	8	5.60	20

Fig. 7. Tabla de las mediciones obtenidas de la BD.

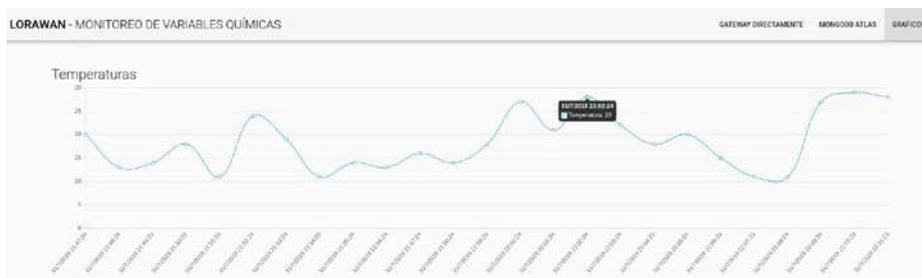


Fig. 8. Gráfico de mediciones del sensor de Temperatura obtenidas de la BD.

Conclusiones

IoT es un conjunto de tecnologías que día a día logra cubrir más campos (ganadería, agricultura, sociedad, obras, etc.) a través de mediciones, controles y automatizaciones. En este sentido, específicamente al hablar de las nuevas necesidades de las ciudades, se han abordado diversas soluciones tecnológicas en pos del medio ambiente y la gestión. Es en este aspecto en donde desarrollamos nuestra propuesta o framework para relevar variables químicas útiles en la región y, como logro importante, dicho framework permitirá que los nodos puedan escalar hasta completar diversos espejos de agua dulce que existen, incluso ríos y arroyos. Esto se logró gracias a la conectividad que permite a LoRaWAN obtener datos de sensores a distancias sumamente importantes y facilita generar desarrollos innovadores que son de vital importancia, como la gestión de las variables que indican contaminación ambiental.

Referencias

[1] Sekhar N. Kondepudi, et. al. An overview of smart sustainable cities and the role of information and communication technologies. Set of ITU-T's Technical Reports and Specifications, 2016.

[2] Bhagya Nathali Silva, Murad Khan, Kijun Han, Towards sustainable smart cities: A review of trends, architectures, components, and open challenges in smart cities. Sustainable Cities and Society, 38, 2018.

- [3] Silva, B. N., Khan, M., & Han, K. (2017a). Internet of things: A comprehensive review of enabling technologies, architecture, and challenges. *IETE Technical Review*, 1–16.
- [4] Gubbi, J., Buyya, R., Marusic, S., & Palaniswami, M. (2013). Internet of Things (IoT): A vision, architectural elements, and future directions. *Future Generation Computer Systems*, 29, 1645–1660.
- [5] Islam, S. R., Kwak, D., Kabir, M. H., Hossain, M., & Kwak, K.-S. (2015). The internet of things for health care: A comprehensive survey. *IEEE Access*, 3, 678–708.
- [6] Hollands, R. G. Will the real smart city please stand up? Intelligent, progressive or entrepreneurial? *City*, 12, 303–320, 2008.
- [7] Kondepudi, S. Smart sustainable cities analysis of definitions. The ITU-T Focus Group for Smart Sustainable Cities, 2014.
- [8] Mohanty, S. P., Choppali, U., & Koungianos, E. Everything you wanted to know about smart cities: The internet of things is the backbone. *IEEE Consumer Electronics Magazine*, 5, 60–70, 2016.
- [9] Lain-Chyr Hwang, Chao-Shun Chen, Te-Tien Ku, Wei-Cheng Shyu, A bridge between the smart grid and the Internet of Things: Theoretical and practical roles of LoRa, *International Journal of Electrical Power & Energy Systems*, 113, 2019, 971-981, ISSN 0142-0615.
- [10] LoRa Alliance. Recuperado de: <https://www.lora-alliance.org/what-is-lora>.
- [11] LoRa Alliance (2017, October). “LoRaWAN Backend Interfaces 1.0 Specification”. Recuperado de: <https://lora-alliance.org/sites/default/files/2018-04/lorawantm-backend-interfaces-v1.0.pdf>.
- [12] Semtech Corp. (2015, May). “AN1200.22 LoRa Modulation Basics”. Recuperado de: <https://www.semtech.com/uploads/documents/an1200.22.pdf>.
- [13] Performance evaluation of LoRa considering scenario conditions. Ramon Sanchez Ibarrola, Jesus Sanchez Gomez, Juan Ballesta Viñas, Maria Dolores Cano y Antonio F. Skarmeta, Diciembre 2017.
- [14] N. Abramson. Development of the ALOHANET (1985). *IEEE Transactions on Information Theory*.
- [15] Multitech mDot. Recuperado de: <https://os.mbed.com/platforms/MTS-mDot-F411/>
- [16] ARM Mbed. Recuperado de: <https://www.mbed.com/en/>
- [17] W3 HTTP. Recuperado de: <https://www.w3.org/Protocols/>
- [18] JSON. Recuperado de: <https://www.json.org/json-en.html>
- [19] MongoDB. Recuperado de: <https://www.mongodb.com/>

**CACIC 2019**

Congreso Argentino de Ciencias de la Computación

XI Workshop Innovación en Sistemas de Software

Coordinadores

Fillottrani, Pablo (UNS)
Estayno, Marcelo (UNLZ)
Mon, Alicia (ITBA)
Zanarini, Dante (UNR)

Comité Científico

Bursztyn, Andrés (Argentina)
Estayno, Marcelo (Argentina)
Fillottrani, Pablo (Argentina)
Flores, Andrés (Argentina)

Gómez, Sergio (Argentina)
Meles, Judith (Argentina)
Montejano, Germán (Argentina)
Riesco, Daniel (Argentina)

Sposito, Osvaldo (Argentina)
Vitturini, Mercedes (Argentina)
Vivas, Luis (Argentina)
Zanarini, Dante (Argentina)

Temas

Dispositivos móviles (celulares, smartphones, PocketPCs, TabletPC, etc)
Software para la Web Semántica
Nuevas herramientas y métodos para ingeniería de software
Computación ubicua
Interfaces emergentes
Ingeniería de ontologías
Nuevas tendencias en ingeniería de software
Computación colaborativa (e-commerce, e-government, e-business, etc)

A Framework for OBDA: Current State and Perspectives

Sergio Alejandro Gómez^{1,2} and Pablo Rubén Fillottrani^{1,2}

¹Laboratorio de I+D en Ingeniería de Software y Sistemas de Información (LISSI)
Departamento de Ciencias e Ingeniería de la Computación

Universidad Nacional del Sur

San Andrés 800 - Campus Palihue – Bahía Blanca, Buenos Aires, Argentina

Email: {sag,prf}@cs.uns.edu.ar

²Comisión de Investigaciones Científicas de la Provincia de Buenos Aires (CIC-PBA)

Abstract. Ontology-based Data Access (OBDA) is concerned with providing end-users and applications with a way to query legacy databases through a high-level ontology that models both the business logic and the underlying data sources. The bridge between the ontology and the data sources is addressed by mappings that define how to express records of the database as ontological assertions. In this research, we are concerned with providing with tools for performing OBDA with relational and non-relational data sources. We developed a tool, which nowadays is in a prototypical state, that is able to access an H2 database, allowing the user to explicitly formulate mappings, and populating an ontology that can be saved for later querying. In this paper, we report on the advances we have made on the development of such a tool, which includes adding the functionality of creating, loading, saving a global ontology that can be populated with a database. Also the system allows the user to visually express mappings from the database to the ontology and the ability of creating databases for testing the behavior of the system in the presence of increasing workloads. The tests we performed indicate that the system is able to handle a moderate workload of tables of tens of thousands of records but fails to handle tables of millions of records.

Keywords. Ontology-based data access, Ontologies, Description Logics, Web Ontology Language, Relational databases.

1 Introduction

Ontology-based Data Access (OBDA) [1, 2] is concerned with providing end-users and applications with a way to query legacy databases through a high-level ontology that models both the business logic and the underlying data sources. Modern knowledge-based applications have replaced the representation of business logic by using a high-level representation of the business intelligence which is decoupled from the application code. This allows for improved flexibility. In Semantic Web applications [3], the business intelligence is represented by ontologies expressed in the Web Ontology Language 2 (OWL 2) [4]. Briefly, an

ontology is a logical theory formed by a collection of concepts and roles and also a set of concept and role assertions [5]. The relationship holding among the concepts and roles in the ontology are described in terms of inclusion and equality axioms. Ontologies used to represent business logic are then used by ontology reasoners to draw conclusions. The conclusions that can be got include making explicit the implicit terminology of concepts defined by the ontology, determining if a certain individual is a member of a concept, or determining if two individuals are related through a role, determining if a concept is subsumed by other concept, or if a role is subsumed by other role.

Thus, the classic OBDA architecture is composed of a global database, a legacy database and a bridge between the ontology and the database. The bridge between the ontology and the data sources is addressed by mappings that define how to express records of the database as ontological assertions. Relational databases are comprised of relations (tables), that in term are defined by data schemas, which define the names and domains of table attributes as well as any integrity constraints that might apply to them, and are composed of records. Ontologies, on the other hand, are composed of axioms, and concept and roles assertions. The mappings define how to populate the ontology in terms of the elements of the database. Basically, the concept and role fillers are defined by SQL queries that indicate how to populate them. Notice that in the case of having several databases, a federation system can be used that allows to see the set of databases as a unified database. In this work, however, we will not take this possibility into account.

In this research, we are concerned with providing with tools for performing OBDA with relational and non-relational data sources. Several tools have been developed by other research groups (see for instance [6–9] that we reviewed in [10]). Some of those tools are closed-source while others are open-source, some are downloadable and can be used as stand-alone applications or as programming libraries. While many times they are a good starting point for building applications, many times they are not flexible enough. In that regard, we are developing a tool, which nowadays is in a prototypical state, that is able to access an H2 database, allowing the user to explicitly formulate mappings, and populating an ontology that can be saved for later querying and visualization. See [11, 10] for previous reports on the functionality of the application and its prospective application areas.

In this paper, we report on the advances we have made on the development of such a tool. In particular, we have added a form that allows end-users to fully specify in a high-level manner the nature of mappings and by writing SQL queries as well. We also added a module that allows to test how our application behaves in the presence of increasing demands. We assume that the reader has a basic knowledge of Description Logics (DL) [12], relational databases [13] and the Web Ontology Language [4].

The rest of the paper is structured as follows. In Sect. 2, we briefly recapitulate the concepts associated with materializing ontologies from tables. In Sect. 3, we present a novel development in the system that allows a naïve user to define

a mapping from tables to ontologies in a visual manner. In Sect. 4, we show an empirical evaluation of the performance of the prototype creating tables and ontologies. In Sect. 5, we review related work. In Sect. 6, we conclude and foresee future work.

2 Materialization of OWL Ontologies from Relational Databases

Materializing an OWL ontology from a relational database requires exporting the database contents as a text file in OWL format. For doing this, we need to export the schema information of each table as Tbox axioms and the instance data of the tables as Abox assertions. Here, we review the formalization for exporting database relations as ontologies as we presented it in [10] according to the directions given by [1, 14]. Building an ontology from a database requires creating at least a class C_T for every table T , and for every attribute a of domain d in T we need two inclusion DL axioms $C_T \sqsubseteq \exists a$ and $\exists a^- \sqsubseteq d$. Primary key values k_i serve the purpose of establishing the membership of individuals to classes as DL Abox assertions of the form $C_T(C_T\#k^j)$. For indicating that a^j is the value of attribute a , we will use a role expression of the form $C_T\#a(C_T\#k^j, C_T\#a^j)$. When it is clear from context, we might drop the prefix $C_T\#$ for simplifying our notation. A foreign key fk in table T_1 referencing a primary key field in table T_2 will also require to add two Tbox axioms $C_{T_1} \sqsubseteq \exists \text{ref_}fk$ and $\exists \text{ref_}fk^- \sqsubseteq C_{T_2}$ and an Abox assertion $\text{ref_}fk(k^j, fk^t)$ for expressing that the individual named k^j in C_{T_1} is related to the individual named fk^t in C_{T_2} . Besides, in any case, if we want to consider a subset of a table for its mapping into an ontology, we might define an SQL query that will act as an SQL filter. In this work, we will only deal with the translation of single tables into OWL, as defined next (for details see [10]):

Definition 1 (Mapping of a table with a single primary key). *Let T be a table with schema $T(\underline{k}, a_1, \dots, a_n)$ and instance $\{(k^1, a_1^1, \dots, a_n^1), \dots, (k^m, a_1^m, \dots, a_n^m)\}$. To map T into a DL terminology \mathcal{T} , we have to create a class T and for each attribute a_i of domain D_i we have to add two axioms: $T \sqsubseteq \exists a_i$, indicating that every T has an attribute a_i , and $\exists a_i^- \sqsubseteq D_i$, meaning that the domain of a_i is D_i . The assertional box \mathcal{A} for T will contain $\{T(k^1), \dots, T(k^m)\}$. Given a key value k_j , $j = 1, \dots, m$, for every attribute a_i , $i = 1, \dots, n$, of the schema and instance value a_i^j (i.e. the value of i -th attribute of the j -th individual), produce a property $a_i(k^j, a_i^j)$.*

Example 1. Consider a table for representing people with schema

Person(personID, name, sex, birthDate, weight)

and instance as in Fig. 1. This table is created by the SQL script presented in Fig. 2.

<i>personID</i>	<i>name</i>	<i>sex</i>	<i>birthDate</i>	<i>weight</i>
1	John	true	2010-01-01	100.0
2	Mary	false	2009-01-01	60.0

Fig. 1. Relational instance of the table *Person*

```

create table "Person" (
"personID" int unsigned not null auto_increment primary key,
"name" varchar(20) not null,
"sex" boolean,
"birthDate" date,
"weight" real,
);

insert into "Person"("name", "sex", "birthDate", "weight") values ('John', true, '2010-01-01', 100.0);
insert into "Person"("name", "sex", "birthDate", "weight") values ('Mary', false, '2009-01-01', 60.0);

```

Fig. 2. SQL script for creating the table *Person*

The table *Person* is interpreted in Description Logics according to Def. 1, as $\Sigma = (\mathcal{T}, \mathcal{A})$ where:

$$\mathcal{T} = \left\{ \begin{array}{ll} \text{Person} \sqsubseteq \exists \text{personID}, & \exists \text{personID}^- \sqsubseteq \text{Integer}, \\ \text{Person} \sqsubseteq \exists \text{name}, & \exists \text{name}^- \sqsubseteq \text{String}, \\ \text{Person} \sqsubseteq \exists \text{sex}, & \exists \text{sex}^- \sqsubseteq \text{Boolean}, \\ \text{Person} \sqsubseteq \exists \text{birthDate}, & \exists \text{birthDate}^- \sqsubseteq \text{Date}, \\ \text{Person} \sqsubseteq \exists \text{weight}, & \exists \text{weight}^- \sqsubseteq \text{Real} \end{array} \right\} \text{ and}$$

$$\mathcal{A} = \left\{ \begin{array}{ll} \text{Person}(\text{Person}\#1), & \text{personID}(\text{Person}\#1, 1), \\ \text{name}(\text{Person}\#1, \text{John}), & \text{sex}(\text{Person}\#1, \text{true}), \\ \text{birthDate}(\text{Person}\#1, 2010-01-01), & \text{weight}(\text{Person}\#1, 100.0), \\ \text{Person}(\text{Person}\#2), & \text{personID}(\text{Person}\#2, 2), \\ \text{name}(\text{Person}\#2, \text{Mary}), & \text{sex}(\text{Person}\#2, \text{false}), \\ \text{birthDate}(\text{Person}\#2, 2009-01-01), & \text{weight}(\text{Person}\#2, 60.0) \end{array} \right\}.$$

Description Logic ontologies are implemented in the OWL language, which includes an XML serialization which we partially present in Fig. 3 by showing the representation for John.

3 Visual Mapping Specification

The specification of the mappings for obtaining a the fillers of concept from a table is usually a complex matter for naïve end-users. Remember that a mapping is basically a SQL query that defines how the fillers of concept, property or role are computed in terms of the contents of a database. When there is no support for composing mappings, the user has to write such SQL from scratch. We believe that adding support for building the mappings will improve the user experience of a prospective user of OBDA technology.

With the idea of providing support to end-users in their quest of creating concepts for populating ontologies from database contents, we created a module

```

<owl:Class rdf:about="http://cs.uns.edu.ar/~sag#Person"/>
<!-- http://cs.uns.edu.ar/~sag/Person/personid=1 -->

<owl:NamedIndividual rdf:about="http://cs.uns.edu.ar/~sag/Person/personid=1">
  <rdf:type rdf:resource="http://cs.uns.edu.ar/~sag#Person"/>
  <Person:birthDate rdf:datatype="http://www.w3.org/2001/XMLSchema#dateTime">
    2010-01-01T00:00:00</Person:birthDate>
  <Person:name rdf:datatype="http://www.w3.org/2001/XMLSchema#string">John</Person:name>
  <Person:personID rdf:datatype="http://www.w3.org/2001/XMLSchema#integer">1</Person:personID>
  <Person:sex rdf:datatype="http://www.w3.org/2001/XMLSchema#boolean">true</Person:sex>
  <Person:weight rdf:datatype="http://www.w3.org/2001/XMLSchema#double">100.0</Person:weight>
</owl:NamedIndividual>

```

Fig. 3. Part of the OWL code for the definition of the class `Person`

that allows to visually specify a mapping from a table. The module retrieves the tables from the database, and allows to select a table. Once the table is selected, its fields can be selected too. The user can then introduce what conditions each field of the table has to satisfy. Besides, one field (usually the key field of the table) has to be selected to fill the concept. The module then will automatically generate the SQL filter for filling the concept by extracting the records from the table, and will also add a subclass axiom to the ontology.

Example 2. Consider again the table `Person` from Ex. 1 and suppose that some user of the system wants to define the concept “heavy, young, male individual”. Suppose also that the user models a heavy individual as somebody who weighs at least a hundred kilograms, a young individual as someone who was born after 2001, and a male individual as someone of male sex. People of male sex are codified as having the column named `sex` as `true` while females are codified as `false`. Although this is a trivial example, it shows the complexities that run into database modeling that produce a degradation of the representation of the world and that are unretrievable afterwards. The user will then visually specify the conditions for an individual to be a member of the concept `YoungHeavyMalePerson` in a form like the one presented in Fig. 4. Notice how the user specifies which database field corresponds to the key (i.e. the name of the individuals), in this case `personID`. In turn, the system will generate a SQL query as follows:

```

SELECT "Person"."personID" FROM "Person"
WHERE "Person"."birthDate" >= '2001-01-01'
AND "Person"."weight" >= 100 AND "Person"."sex" = true

```

After the execution of the query that will compute the individuals that fill the concept, the system will add to the current ontology the triples expressing that those individuals are the fillers of the concept `YoungHeavyMalePerson`. Besides, in order to relate this concept to its superconcept, the axiom

$$\text{YoungHeavyMalePerson} \sqsubseteq \text{Person}$$

will be added to the current ontology as well.

This will lead to the situation presented in Fig. 5. The new class `YoungHeavyMalePerson` is defined as a subclass of `Person` and `John`, whose `personID` role is “1” becomes

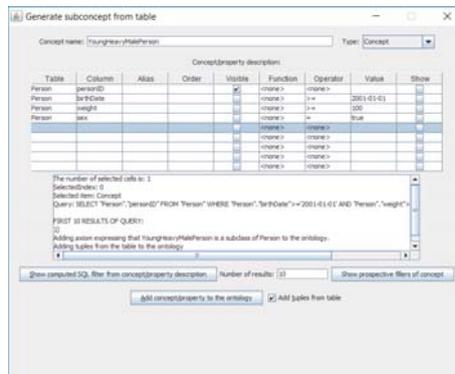


Fig. 4. Visual concept specification of the concept YoungHeavyMalePerson

a member of YoungHeavyMalePerson. Notice also that no new individuals are defined as John is already present in the ontology because he is a Person. In this sense, we adhere to the unique name assumption as much as we can although this is not required by the formalism. Also notice how the intensional definition of the concept is lost in the ontology (other than being a subclass of Person) and only its extension is maintained in the ontology (as the set of its fillers).

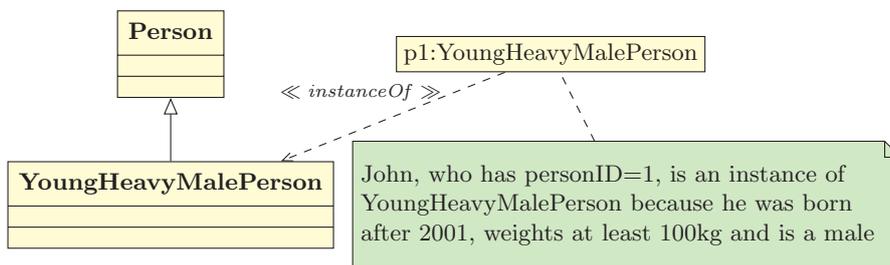


Fig. 5. Situation arisen by specifying a subclass of Person named YoungHeavyMalePerson

Another feature that the current version of the system includes is the possibility of specifying a subclass by means of an explicit SQL query.

Example 3. Continuing Ex. 2, the concept FemalePerson (which defines a subset of the table Person formed by women) is specified by means of the SQL query:

```
select "personID" from "Person" where "sex" = false
```

This can be done by using the form presented in Fig. 6. Notice the additional OWL code in the ontology generated by out tool which is presented in Fig. 7

expressing that a female person is a person (i.e. $\text{FemalePerson} \sqsubseteq \text{Person}$ is an axiom in the ontology) and that Mary is both a female person and a person (i.e. $\text{FemalePerson}(\text{Mary})$ and $\text{Person}(\text{Mary})$ are assertions in the ontology).

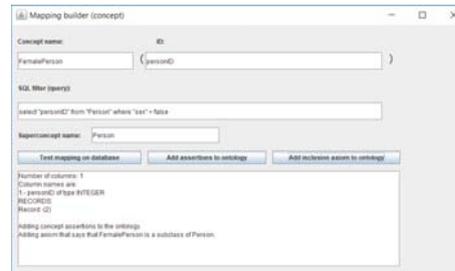


Fig. 6. Specification of the subclass FemalePerson of Person by a SQL query

```
<owl:Class rdf:about="http://cs.uns.edu.ar/~sag#FemalePerson">
  <rdfs:subClassOf rdf:resource="http://cs.uns.edu.ar/~sag#Person"/>
</owl:Class>
...
<owl:NamedIndividual rdf:about="http://cs.uns.edu.ar/~sag/FemalePerson=2">
  <rdf:type rdf:resource="http://cs.uns.edu.ar/~sag#FemalePerson"/>
...
</owl:NamedIndividual>
```

Fig. 7. Portion of OWL code for introducing subconcept FemalePerson

4 Experimental Evaluation

We now discuss some of the tests we have performed in order to test how our application handles increasing demands in database size. The performance of our system is affected mainly by the fact that we chose to materialize tables as triples (i.e. class membership, property and roles assertions) and also by three factors: (i) the system is implemented in the JAVA programming language; (ii) the database management system that we use is H2¹, and, (iii) the handling of the global ontology is done via the OWL API [15, 16].

Our tests were conducted on an ASUS notebook having an Intel Core i7, 3.5GHz CPU, 8GB RAM, 1TB HDD, Windows 10. They involved the creation of simple databases composed by a single table containing 100 fields of text type filled with an increasing number of records. Table 1 summarizes our results.

¹ See <http://www.h2database.com>.

Table 1. Running times for ontology generation

<i>Number of records</i>	<i>Database file size in Megabytes</i>	<i>Time for creating the ontology in seconds</i>	<i>Size of the ontology in Megabytes</i>	<i>Time for loading the ontology from disk in seconds</i>
1,000	0.80	1.029	8.65	4.014
10,000	8.82	5.345	87.26	15.106
100,000	98.11	66.48	19,053.36	out of memory error
1,000,000	1,080.60	out of memory error	-	-

As it can be seen, our implementation starts having problems at tables with 100,000 records; although an ontology can be generated and saved to the disk, when we try to load the ontology we saved previously, we get an error inside the code of the OWL API, indicating that that library cannot handle such a data load. When running a test for creating a database of a million records, the H2 database produces an error (which is understandable as it is maintained in RAM). Therefore, we conclude that our application can only handle tables with a size tens of thousands records and is not able of handling tables of a hundred thousand records. Because of this limitation, we think that we will be forced to use query-rewriting techniques [1] for delegating the evaluation of queries to the database management system instead of the ontology management library.

5 Related Work

With ViziQuer, Cerans et al. [17] provide an open source tool for web-based creation and execution of visual diagrammatic queries over RDF/SPARQL data. The tool supports the data instance level and statistics queries, providing visual counterparts for most of SPARQL 1.1 select query constructs, including aggregation and subqueries. A query environment can be created over a user-supplied SPARQL endpoint with known data schema. ViziQuer provides a visual interface for expressing user queries in SPARQL posed against an ontology. In contrast, we provide the user with an interface for describing subclass expressions and inclusion axioms by means of restrictions imposed on records of a relational table with the aim of populating an ontology that could later be exposed and queried as an SPARQL endpoint.

Christodoulou et al. [18] make the case that structural summaries over linked-data sources can inform query formulation and provide support for data integration and query processing over multiple linked-data sources. To fulfil this aim, they propose an approach that builds on a hierarchical clustering algorithm for inferring structural summaries over linked-data sources. Thus, their approach takes as input an RDF repository and then reverse engineers an ontology using clustering techniques to detect prospective classes. In contrast, we take a database and the user proposes SQL queries to express subconcepts intensionally; when the SQL queries are executed, the fillers of the concept populate the ontology building an extensional de facto definition of the concept. In that

regard, the work of Christodoulou et al. can be considered complementary to ours.

Barrasa et al. [19] propose R2O, an extensible and declarative language to describe mappings between relational DB schemas and ontologies implemented in RDF(S) or OWL. R2O provides an extensible set of primitives with well defined semantics. This language has been conceived expressive enough to cope with complex mapping cases arisen from situations of low similarity between the ontology and the DB models. R2O allows the user to express complex queries in terms of ontologies in a language that is similar to the relational algebra but aimed at ontologies. Therefore, this approach is complementary to ours because it allows to query the ontology once it is published in OWL format.

6 Conclusions and Future Work

We presented the advances we have made in our framework for performing ontology-based data access by means of performing a materialization approach. Our implementation is JAVA-based and relies on a H2 database management system and a JAVA library called OWL-API for accessing and querying databases and maintaining an OWL database in main memory, respectively. We presented several enhancements that we have made to the previous iteration of our prototype implementation, which now includes a visual mapping specification functionality and allows to maintain a global database that can be either created from scratch or loaded from disk, modified and later saved again to disk. From the experimental evaluation to which we subjected our system, we conclude that our application is able to handle a moderate workload of a table of tens of thousands of records but fails to handle table of the order of millions of records. In this regard, we think that we will be forced to use query-rewriting techniques for delegating the evaluation of queries to the database management system instead of the ontology management library. Part of our current research efforts are aimed in this direction. Other form of improvement lies in the possibility of addressing the federation of databases for performing integration of multiple heterogeneous data sources.

Acknowledgments. This research is funded by Secretaría General de Ciencia y Técnica, Universidad Nacional del Sur, Argentina and by Comisión de Investigaciones Científicas de la Provincia de Buenos Aires (CIC-PBA).

References

1. Kontchakov, R., Rodríguez-Muro, M., Zakharyashev, M.: Ontology-Based Data Access with Databases: A Short Course. In: Reasoning Web: Semantic Technologies for Intelligent Data Access of the LNCS. Volume 8067. Springer (2013) 194–229
2. Xiao, G., Calvanese, D., Kontchakov, R., Lembo, D., Poggi, A., Rosati, R., Zakharyashev, M.: Ontology-Based Data Access – A Survey. In: Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence (IJCAI-18). (2018) 5511–5519

3. Berners-Lee, T., Hendler, J., Lassila, O.: The Semantic Web. *Scientific American* (2001)
4. Bao, J., Kendall, E.F., McGuinness, D.L., Patel-Schneider, P.F.: OWL 2 Web Ontology Language Quick Reference Guide (Second Edition) W3C Recommendation 11 December 2012 (2012)
5. Gruber, T.R.: A translation approach to portable ontologies. *Knowledge Acquisition* **5**(2) (1993) 199–220
6. Calvanese, D., Giacomo, G.D., Lembo, D., Savo, D.F.: The MASTRO system for ontology-based data access. *Semantic Web* **2**(1) (2011) 43–53
7. Jimenez-Ruiz, E., Kharlamov, E., Zheleznyakov, D., Horrocks, I., Pinkel, C., Skjæveland, M.G., Thorstensen, E., Mora, J.: BootOX: Practical Mapping of RDBs to OWL 2. In: the 14th International Semantic Web Conference. (2015) 113–132
8. de Medeiros, L.F., Priyatna, F., Corcho, O.: MIRROR: Automatic R2RML mapping generation from relational databases (2015)
9. Pinkel, C., Binnig, C., Jiménez-Ruiz, E., Kharlamov, E., May, W., Nikolov, A., Bastinos, A.S., Skjæveland, M.G., Solimando, A., Taheriyan, M., Heupel, C., Horrocks, I.: RODI: Benchmarking Relational-to-Ontology Mapping Generation Quality. *Semantic Web* (2016) 1–26
10. Gómez, S.A., Fillottrani, P.R.: Towards a Framework for Ontology-Based Data Access: Materialization of OWL Ontologies from Relational Databases. In Pesado, P., Aciti, C., eds.: X Workshop en Innovación en Sistemas de Software (WISS 2018), XXIV Congreso Argentino de Ciencias de la Computación CACIC 2018. (2018) 857–866
11. Gómez, S.A., Fillottrani, P., Estévez, E., Pesado, P., Pasini, A., Muñoz, R., Thomas, P.: Desarrollo de herramientas para acceso a bases de datos heterogéneas basado en ontologías en el contexto de la entrega de servicios públicos digitales. Primer Encuentro de Centros Propios y Asociados de la Comisión de Investigaciones Científicas de la Provincia de Buenos Aires (2018) 235–238
12. Baader, F., Calvanese, D., McGuinness, D., Nardi, D., Patel-Schneider, P., eds.: *The Description Logic Handbook – Theory, Implementation and Applications*. Cambridge University Press (2003)
13. Silberschatz, A., Korth, H.F., Sudarshan, S.: *Database System Concepts* (6th edition). McGraw-Hill Education (1983)
14. Arenas, M., Bertails, A., Prud’hommeaux, E., Sequeda, J.: A Direct Mapping of Relational Data to RDF. W3C Recommendation 27 September 2012 (2012)
15. Horridge, M., Bechhofer, S.: The OWL API: A Java API for OWL Ontologies. *Semantic Web* **2**(1) (2011) 11–21
16. Matentzoglou, N., Palmisano, I.: An Introduction to the OWL API. Technical report, The University of Manchester (2016)
17. Cerans, K., Sostaks, A., Bojars, U., Ovcinnikova, J., Lace, L., Grasmanis, M., Romane, A., Sprogis, A., Barzdins, J.: ViziQuer: A Web-Based Tool for Visual Diagrammatic Queries Over RDF Data. In et al., G.A., ed.: *The Semantic Web: ESWC 2018 Satellite Events. ESWC 2018. Lecture Notes in Computer Science. Volume 11155.*, Springer, Cham (2018) 158–163
18. Christodoulou, K., Paton, N.W., Fernandes, A.A.: Structure Inference for Linked Data Sources using Clustering. In: EDBT ’13 Proceedings of the Joint EDBT/ICDT 2013 Workshops. (2013) 60–67
19. Barrasa, J., Corcho, Ó., Gómez-Pérez, A.: R2O, an extensible and semantically based database-to-ontology mapping language. *Proceedings of the Second Workshop on Semantic Web and Databases, SWDB 2004* **3372** (2004)

Accesibilidad de edificios por medio de modelado y simulación con aplicaciones 3D.

Nicole Denon¹, Santiago Doti¹, Lucas Olivera¹, Martín Morales^{1,2}, Diego Encinas^{1,3}

¹Programa TIC en aplicaciones de interés social - Instituto de Ingeniería y Agronomía - UNAJ

²Unidad CodApli - Facultad Regional La Plata - UTN

³Instituto de Investigación en Informática LIDI (III-LIDI) - Facultad de Informática - UNLP – Centro Asociado CIC

{nicole.a.denon, luksolivera10, santiagoandresdoti}@gmail.com, {martin.morales, dencinas}@unaj.edu.ar

Resumen. En esta publicación se presenta el desarrollo de una aplicación 3D con la finalidad de simular la estructura interna de un edificio público, concretamente la estructura de una universidad, permitiendo a los usuarios interesados recorrerla. El objetivo principal de la aplicación es ofrecer a aquellas personas, que posean movilidad reducida, recorrer el edificio a través de la aplicación y poder conocer su infraestructura antes de acudir al lugar.

Palabras Clave: Modelado y Simulación, Aplicaciones 3D, Accesibilidad.

1 Introducción

A lo largo de la historia el ser humano ha creado diversos dispositivos con el objetivo de ayudar, a personas que poseen algún tipo de discapacidad, a que su vida se realice de manera más simple y digna. Basándose en esta visión, se desarrolló una aplicación de entorno 3D con el objetivo principal de brindar una herramienta a las personas que sufrieron la pérdida parcial o total del movimiento y que particularmente dependen, para su traslado personal, de una silla de ruedas.

La problemática planteada para la confección de este trabajo fue que muchas personas con movilidad reducida deben asistir a diversos edificios públicos (juzgados, municipalidades, universidades, etc.) y estos, en ciertos casos, no cuentan con la infraestructura adecuada para permitir un correcto desplazamiento considerando las limitaciones mencionadas. Por lo tanto, surgió la necesidad de brindar a estas personas la posibilidad de conocer el lugar antes asistir.

Se propuso una solución, la cual fue generar una aplicación que ofrezca la posibilidad de conocer el lugar. La aplicación cuenta con el modelado de la estructura de un edificio público, en este caso la Universidad Nacional Arturo Jauretche, y permite de manera virtual recorrerla. El usuario puede elegir un lugar de interés dentro del edificio y la aplicación le mostrará en tiempo real el camino más corto para llegar.

En otros edificios públicos se han implementado soluciones parecidas a la propuesta. La Universidad Nacional de La Plata (UNLP) tiene actualmente habilitado en su página web un recorrido virtual de la Facultad de Informática [1] [2]. Por otro lado, la Universidad Nacional Abierta y a Distancia (UNAD) de Bogotá también ha desarrollado su propio recorrido virtual de la Sede Principal [3].

La principal diferencia entre la implementación presentada en este trabajo y los anteriores mencionados, es el enfoque de los mismos. El objetivo del trabajo de Bogotá es más publicitario, su intención es mostrar un recorrido virtual de una forma atractiva e interactiva. Por otro lado, el proyecto de La Plata propone guiar a los estudiantes noveles y posee los puntos de interés más importantes para los ingresantes. En cambio, el objetivo del desarrollo presentado es guiar a aquellas personas con movilidad reducida, por lo que se enfoca en los caminos ideales para ellos, como las trayectorias con rampas o sin escaleras.

2 Implementación

En este apartado se detallarán las herramientas, como así también las metodologías que se utilizaron para el desarrollo de la aplicación.

Herramientas:

- Unity: plataforma para la creación de contenidos interactivos en 2D y 3D. Incluye un entorno de desarrollo, una tienda de assets, un motor gráfico, un repositorio de trabajo y una amplia gama de herramientas que facilitan el proceso de creación del proyecto [4].
- Sweet Home 3D: Es un editor CAD de ingeniería, arquitectura y construcción bajo licencia General Public License (GNU) para el diseño de muebles de una vivienda en un plano 2D que permite generarlo y exportarlo en un plano 3D [5].

3 Desarrollo

En este apartado se detalla, la implementación de las herramientas además de las decisiones tomadas para el desarrollo.

Cuando se determinó el alcance que tendría el software se planteó que para su correcto funcionamiento eran necesarios tres módulos fundamentales:

En primer lugar, un mundo diseñado en tres dimensiones en el que se desarrolle toda la acción, es decir, un espacio en donde se muestre al usuario por dónde es más accesible el ingreso a un edificio público. Y que ésto suceda de una forma virtual pero que sea lo más parecido a la realidad de dicho inmueble. En segundo lugar, se precisó que todo lo que suceda en dicho entorno se presente de una manera real, en otras palabras, que respete las leyes de la física para contribuir a la inmersión y credibilidad del usuario final cuando necesite las indicaciones para guiarse dentro del edificio público. Finalmente, se requirió del algoritmo de búsqueda capaz de calcular cuál iba a ser el camino más corto para llegar al destino que el usuario solicitaba. Analizando estos pasos como partes de un proceso para lograr el producto final se puede expresar como se ve en la Figura 1.



Figura 1. Diagrama de los procesos que componen el software

Se puede apreciar en dicho diagrama que cada proceso es cuasi dependiente del otro, en otras palabras, cada uno necesita una característica del otro para poder funcionar con normalidad. Si no existiese un mundo virtual 3D el algoritmo de búsqueda no tendría donde mostrarse y las físicas no actuarían sobre ningún tipo de objeto, y así sucedería lo mismo con los restantes procesos.

A continuación se detalla cada proceso por separado:

3.1 Diseño entorno 3D

Haciendo uso de la herramienta Sweet Home se diseñó uno de los edificios de la Universidad [6]. Se escogió este edificio ya que es uno de los más frecuentados por los estudiantes y posee una especie de rampa en su hall permitiendo que una persona con silla de ruedas puede subir al primer piso como se puede ver en la Figura 2. Una vez elaborado el diseño se exporta a formato OBJ (Wavefront 3D Object File), el cual es una extensión de archivo usado para un objeto tridimensional que contiene las

coordenadas 3D. Esto permite hacer uso de este diseño dentro de la plataforma de Unity. En la Figura 3 se visualiza el diseño 3D desde Unity una vez agregada la iluminación y los colores de los objetos.



Figura 2. Diseño 3D del primer piso en SweetHome 3D



Figura 3. Vista en primera persona de la aplicación

3.2 Simulación Física

Como el proyecto se basó en una simulación del mundo real se necesitaban de las reglas físicas que lo rigen. Para esto se hizo uso del motor físico integrado que provee el mismo Unity, el cual es el encargado de dar los correspondientes comportamientos físicos a cada objeto (GameObjects dentro del sistema) que se encuentra en el entorno virtual.

Los comportamientos de los GameObject son controlados por los Components, éstos no son sólo funciones físicas, sino que pueden ir desde funciones de renderizado, de navegación, de sonido hasta estadísticos, entre muchos otros. Inclusive se pueden llegar a crear los propios en forma de Scripts, éstos pueden ser desarrollados en lenguaje C# o UnityScript.

Centrándose en los Components de físicas más importantes que fueron utilizados en este proyecto, se encuentran:

- **Rigidbody:** componente básico que le permite a cada GameObject actuar bajo el control de la física. Gracias a éste cada uno puede recibir fuerza y torque para moverse de una manera realista, en el proyecto fue aplicado a la silla de ruedas para que cuando detecte colisión detenga su posición y no traspase al objeto en cuestión.
- **Capsule Collider:** como su nombre lo indica, crea una cápsula de dos semi-esferas unidas por un cilindro alrededor de cada GameObject para que detecte colisiones con otros objetos en el entorno (se

puede ver en la Figura 4). Cuando detecta colisiones se lo indica al Rigidbody para que éste actúe según lo programado.

- **Mesh Collider:** funciona al igual que el anterior componente pero en este caso funciona como una malla, la cual se acopla casi perfectamente al objeto asignado por lo que es mucho más precisa pero consume muchos recursos.
- **Movimiento del personaje:** en este caso se trata de un script desarrollado en este trabajo para controlar la silla de ruedas en el entorno virtual. Fue desarrollado en C# y básicamente lo que hace es capturar cuando se presiona la tecla W mientras se recorre el entorno virtual. Cuando detecta dicha acción se mueve al protagonista hacia delante, dependiendo de hacia donde esté enfocada la vista (manejada por el mouse de la computadora), ya que se trata de una silla de ruedas y la única dirección de movimiento permitida es esa.



Figura 4. Personaje y el componente Capsule Collider

3.3 Algoritmo de Búsqueda [7]

Para este desarrollo se analizaron diferentes algoritmos de búsqueda, los cuales se describirán brevemente en este apartado resaltando sus virtudes y falencias.

Algoritmo de Dijkstra: Es un algoritmo que permite recorrer un grafo a través de sus vértices y conocer el camino más óptimo entre dos nodos. La ventaja de este algoritmo es que permite conocer cuál es el camino más corto o de menor costo para moverse de un punto a otro. La desventaja principal es que recorre el grafo por amplitud, es decir, visita a todos los nodos vecinos, esto provoca mayores iteraciones. La Figura 5 muestra cómo trabaja el algoritmo. El punto rosa representa el origen y el violeta el destino, los cuadrados pintados de celeste son los que el algoritmo visitó hasta llegar al destino.

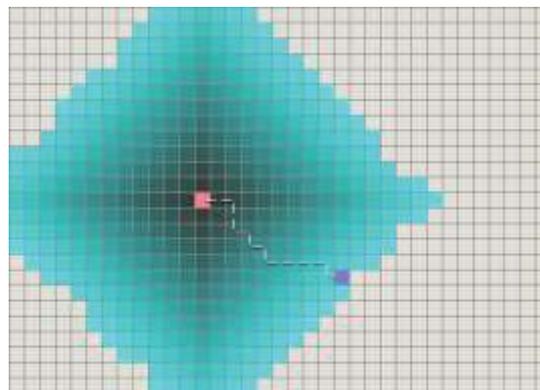


Figura 5. Algoritmo de Dijkstra.

Algoritmo Greedy Best-First-Search (GBFS): Este algoritmo es un ajuste al método tradicional de BFS, la principal diferencia es que utiliza la Búsqueda Heurística (BH) para recorrer el grafo. La gran ventaja de este algoritmo es que gracias a la BH permite en menos interacciones encontrar el nodo destino. Pero a diferencia del algoritmo anterior, como se mencionó es un ajuste de BFS por lo tanto el recorrido que devuelve no es el más óptimo por lo tanto esta es la desventaja al usar este algoritmo. Como puede verse en la Figura 6 el algoritmo visita menos nodos que el algoritmo anterior.

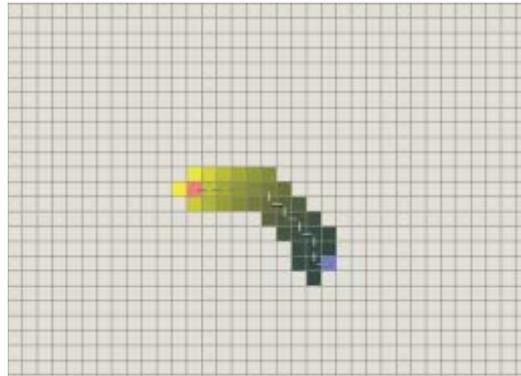


Figura 6. Algoritmo de GBFS.

En las Figura 7 y Figura 8 se puede ver una comparativa entre los dos algoritmos en donde el algoritmo de Dijkstra itera más pero devuelve un camino óptimo. Y como el GBFS itera menos pero su camino no es muy óptimo.

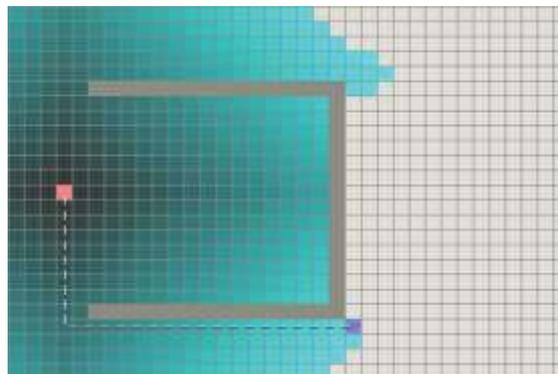


Figura 7. Algoritmo de Dijkstra con obstáculo.

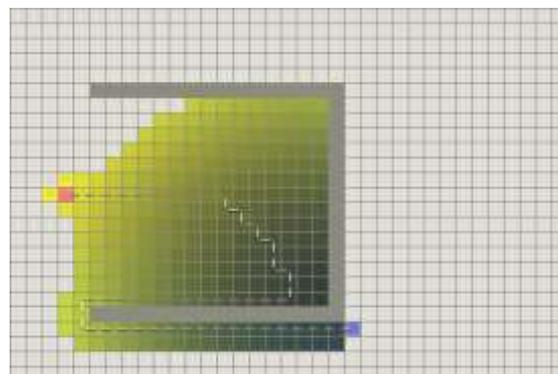


Figura 8. Algoritmo de GBFS con obstáculo.

Algoritmo Estrella o Star (A*): Este es el algoritmo que se seleccionó para realizar el trabajo. A* resuelve los problemas causados por los algoritmos anteriores ya que vincula la búsqueda óptima de Dijkstra y la BH de GBFS.

En las Figura 9 y Figura 10 puede verse cómo se comporta este algoritmo en las mismas condiciones que los casos anteriores y se puede notar que el algoritmo A* itera en menor medida y devuelve el camino óptimo.

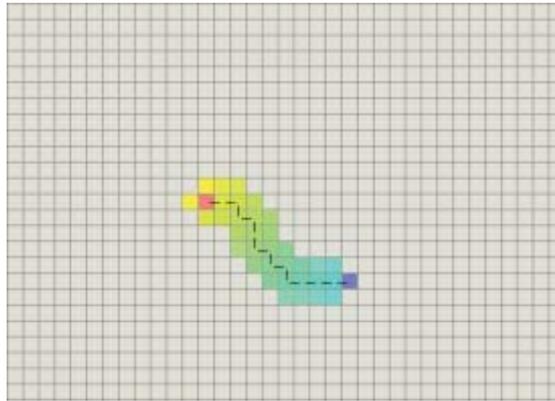


Figura 9. Algoritmo A*

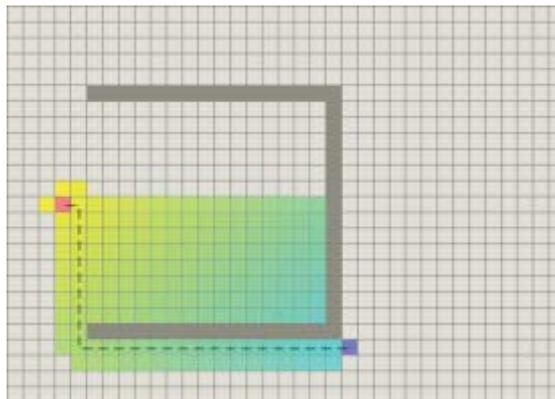


Figura 10. Algoritmo A* con obstáculo.

Búsqueda Heurística

Es una función que se utiliza para la estimación de costos desde cualquier vértice n hasta el destino. Hay diversas funciones dependiendo el problema que se desee solucionar, para este problema en particular se utilizaron funciones que trabajan sobre grillas o cuadrículas:

1. Distancia Manhattan: Este algoritmo sólo permite desplazarse en 4 direcciones, es decir no permite desplazamientos en diagonal por lo tanto el camino devuelto no es del todo el óptimo si en el sistema real existe la posibilidad de moverse diagonalmente. Este funcionamiento puede visualizarse en la Figura 11 en donde el camino solo se traza de forma horizontal o vertical.

Cálculo realizado: sea I y D nodos de la cuadrícula; x, y sus posiciones en los ejes, P un valor de movimiento y C el camino devuelto:

$$Dx = |I_x - D_x| \quad (1)$$

$$Dy = |I_y - D_y| \quad (2)$$

$$C = P * (Dx + Dy) \quad (3)$$

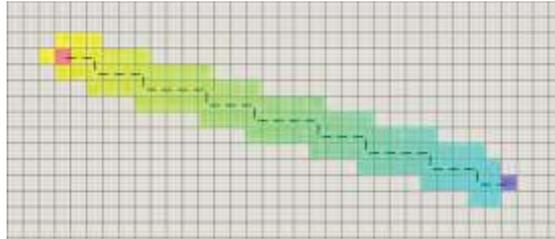


Figura 11. Camino con distancia Manhattan.

2. Distancia Diagonal: Este algoritmo ofrece la posibilidad de desplazamiento en diagonal, es decir se tiene 8 direcciones para poder moverse, y se consigue simplemente agregando unos cambios a la ecuación de la distancia Manhattan. Esto se aprecia fácilmente en la Figura 12. Gracias a las ventajas que provee este algoritmo fue posible el desarrollo en este trabajo.

Tomando (1) y (2), se aplica un ajuste a (3). donde P_2 es un valor de movimiento.

$$C = P * (Dx + Dy) + (P_2 - 2 * P) * \min(Dx, Dy) \quad (4)$$

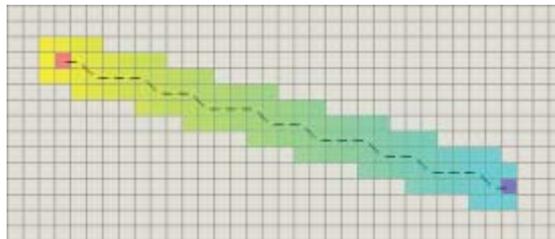


Figura 12. Camino con distancia Diagonal.

3. Distancia Euclidiana: Al igual que la anterior la euclidiana permite movimiento por las diagonales, pero también se puede utilizar si se necesita moverse por cualquier ángulo de la cuadrícula y su función es similar a Pitágoras. Tomando (1) y (2)

$$C = P * \sqrt{Dx^2 + Dy^2} \quad (5)$$

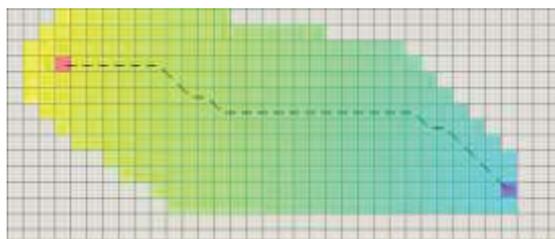


Figura 13. Camino con distancia Euclidiana.

El problema al aplicar este algoritmo es que aumenta la cantidad de iteraciones que realiza el Algoritmo de Búsqueda como se puede observar en la Figura 13.

Vinculación de Componentes:

Un desafío grande para el desarrollo fue intentar que el algoritmo de búsqueda pueda ser implementado por Unity independientemente del modelo 3D creado por Sweet Home 3D con el objetivo de poder incluir diversas arquitecturas de edificio. Para eso se desarrolló el algoritmo con la habilidad de reconocer,

independientemente de la estructura, los objetos a los cuales no se puede atravesar (paredes, mesas, etc). Gracias a ello, y con la debida configuración a los correspondientes modelos 3D, se puede diseñar cualquier edificio y obtener el mejor camino.

4 Resultados

Al iniciar con la aplicación el usuario se encuentra un menú principal en donde puede seleccionar la habitación a la cual quiere ir. Estas habitaciones son en su mayoría aulas u oficinas de administración. Una vez elegido el lugar se pasa a la escena principal del edificio, para este ejemplo se eligió un aula del piso superior. En la Figura 14 se puede observar que al iniciar movimiento por el mapa aparecen círculos verdes que indican el mejor camino, en este caso el camino indica la rampa ya que el lugar de destino se encuentra en el primer piso.



Figura 14. Señalización del mejor camino en ejecución

Una vez que se sube por la rampa y se llega al piso superior, el mejor camino vuelve a marcarse como se muestra en la Figura 15. Ahora el camino indica el aula de destino que se ha ingresado en el menú principal.



Figura 15. Señalización hacia el aula de destino

Por último en la Figura 16 se muestra la llegada al destino final del software, el cual está marcado con un cuadrado rojo para poder visualizarlo. Puede volverse a cambiarse el destino si se presiona "ESC" y se vuelve a seleccionar un nuevo destino en el menú principal. Si esto se hace se repetirá el mismo proceso especificado anteriormente



Fig. 16. Señalización hacia el aula de destino

La aplicación puede verse en funcionamiento en el siguiente link: <https://youtu.be/4CgL2okrR68>

5 Trabajos futuros

El proyecto ha sido diseñado para que sea escalable y que permita implementar de forma sencilla con otros edificios. Aun así existen muchas mejoras que pueden aplicarse, estas son las siguientes:

- Implementación de la aplicación para dispositivos móviles: El smartphone se ha convertido con el paso del tiempo en el medio de comunicación masivo más utilizado por las personas [8]. Esto se debe en gran parte a la movilidad y comodidad que brinda un teléfono celular. Por lo tanto la aplicación desarrollada podría tener un muy buen uso social si contará con una implementación móvil, ya que una persona con capacidad de movimiento reducida podría necesitar una visualización de un edificio justo antes de adentrarse al mismo. Aunque un punto muy importante que hay que tener en cuenta para poder lograrlo es el rendimiento que tendrá la aplicación actual corriendo en un dispositivo móvil. No será una tarea fácil puesto que se necesitarán celulares muy potentes para que funcione de una manera fluida. Si se tienen en cuenta los requisitos que necesita un smartphone para ejecutar aplicaciones no tan complejas desarrolladas en Unity [9] se requiere de Android 4.1 o posterior, arquitectura ARMv7 CPU con soporte NEON o CPU Atom y soporte para OpenGL ES 2.0 o posterior. En cuanto a celulares iPhone se tendrá que contar con iOS 9.0 o posterior.
- Modelado de más edificios de interés: Como se ha mencionado antes, el algoritmo de búsqueda desarrollado se adapta a cualquier edificio que se quiera modelar. Por lo que sólo es necesario importar un diseño 3D de una infraestructura al programa para poder representarla.
- Sumar obstáculos, tales como personas en movimiento o muebles si se quiere simular el edificio tal cual es, es necesario implementar más objetos, como mesas o computadoras, y personas. Esto agrega más obstáculos y dificulta mínimamente el cálculo de la búsqueda del mejor camino.
- Agregar sonidos ambientales: Los efectos de sonido permiten dar una mejor inmersión en el mundo virtual desarrollado.

6 Conclusiones

Se desarrolló una aplicación que cumple con los objetivos planteados inicialmente, inclusive con una alta escalabilidad, es decir, aplicando las mejoras mencionadas previamente se podrá conseguir una aplicación mucho mayor, más integral y de mayor valor si así se desea.

Con el fin de implementar algunas de las mejoras planteadas en los trabajos futuros se debería hacer uso de Android Studio para el desarrollo de una aplicación móvil orientada a los smartphones que cuenten con sistema operativo Android, como así también contar con los planos de los edificios correspondientes que se quieren modelar para lograr un resultado lo más fiel al establecimiento real.

Para finalizar, se entiende que la implementación y uso de esta aplicación podría ayudar a muchas personas que requieran tener esta información previamente y actualmente no poseen acceso a una tecnología de esta índole.

7 Referencias

1. Thomas P., Cristina F., Dapoto S., Pesado P. “Desarrollo de Aplicaciones Móviles 3D Orientadas a Educación.”, XVIII Workshop de Investigadores en Ciencias de la Computación. 2016.
2. Recorrido virtual de la Universidad Nacional de La Plata <https://www.info.unlp.edu.ar/recorrido-virtual-3d-facultad-informatica/> - Fecha de consulta: 29/07/2019
3. Nieto Acevedo Y., López Quintero J., González Clavijo C. “Recorrido Virtual en Tercera Dimensión de la Sede Principal en una Universidad de Bogotá”. Corporación Unificada Nacional (CUN). Universidad Nacional Abierta y a Distancia (UNAD). Bogota, Colombia. 2015.
4. Página oficial de Unity3D, <http://www.unity3d.com/es> - Fecha de consulta 24/11/2018.
5. Página oficial de Sweet Home 3D <http://www.sweethome3d.com/es/> - Fecha de consulta: 24/11/2018.
6. Edificio diseñado para la aplicación, <https://www.unaj.edu.ar/edificio-manuel-savio/> - Fecha de consulta 24/11/2018.
7. Algoritmos de búsqueda, <http://theory.stanford.edu/~amitp/GameProgramming/> - Fecha de consulta 19/11/2018.
8. La venta de celulares vuelve a romper el record en Argentina, <https://www.eldia.com/nota/2017-9-18-2-37-2-la-venta-de-celulares-vuelve-a-romper-el-record-en-argentina-informacion-general> - Fecha de consulta 24/11/2018.
9. Requisitos del sistema para la versión de Unity, <https://unity3d.com/es/unity/system-requirements> - Fecha de consulta 20/02/2019

An Automated Technique for Analysis of Orthogonal Variability Models based on Anti-patterns Detection using DL reasoning

Angela Oyarzun¹ and Germán Braun^{1,3} and Laura Cecchi¹ and Pablo Fillottrani^{2,4}

¹UNIVERSIDAD NACIONAL DEL COMAHUE ²UNIVERSIDAD NACIONAL DEL SUR

³Consejo Nacional de Investigaciones Científicas y Técnicas (CONICET)

⁴Comisión de Investigaciones Científicas de la Provincia de Buenos Aires (CIC)

angela.oyarzun@est.fi.uncoma.edu.ar

{german.braun, lcecchi}@fi.uncoma.edu.ar

prf@cs.uns.edu.ar

Abstract During a Software Product Line (SPL) variability management, model validation is crucial so as to detect faults in early development stages and avoid affecting derived products quality. Therefore, the automated variability analysis has emerged for translating and validating variability models. In this work, we present a catalogue of anti-patterns, which describes scenarios associated to the detection of problems in a SPL. Moreover, we extend *crowd-variability*, a novel graphical tool designed for modelling and validating Orthogonal Variability Models (OVM), for detecting such anti-patterns using Description Logics (DL)-based reasoning services.

Keywords: Software Product Lines, Orthogonal Variability Models, Description Logics, Graphical tools for modelling variability

1 Introduction

Currently, the increasing complexity of Information Systems, derived from a Software Product Line (SPL), results in more complex variability models (VM). Consequently, the tasks involved in variability management become more costly in time and error prone, making it impossible to guarantee the quality of final products without the assistance of automated tools. In this regard, we developed *crowd-variability*¹, a graphical tool for designing, visualising and validating orthogonal variability models (OVM). This novel client-server tool, presented in [1], provides graphical support for users modelling their diagrams and is integrated with automatic DL-based reasoning.

In variability management of SPLs, checking consistency of variability models (VM) is a critical problem. On this line, the most common mistake committed when modelling variability of a SPL is invalidating its services, i.e. creating dead services. This problem directly affects the quality of products as it can invalidate a whole model. Therefore, we need a way of capturing this error-prone modelling decision. For this purpose, there are anti-patterns [2].

¹ <http://crowd.fi.uncoma.edu.ar/crowd-variability/web-src/>

In this work, we extend *crowd-variability* automated variability analysis support by detecting a series of anti-patterns which are based on a set of instances of dead services. Each of these anti-patterns corresponds with one or more dependencies of variability and/or restrictions of a model. Furthermore, we simplified its original DL *ALCI* encoding and we extended it in order to be able to classify services and dependencies according to its type. A prototype of this tool already runs on a client-server architecture and enables anti-patterns detection on OVM graphical diagrams.

This work is structured as follows. Section 2 introduces orthogonal variability models, describes automated variability analysis and compares *crowd-variability* with other existing tools. Section 3 presents anti-patterns and their detection algorithms. Section 4 presents the prototype developed together with a simple example of use and exposes a preliminary evaluation and discussions. To conclude the paper, section 5 elaborates on final considerations and directions for future works.

2 Context

Orthogonal Variability Models (OVM) A software product line is a means to develop a set of products in which variability is a central phenomenon captured in variability models [3]. Among these models, there are Orthogonal Variability Models comprised by services, variants and variation points, related by variability and constraint dependencies. Both elements and interactions are depicted in Table 1 using *crowd-variability*'s graphical language which is based on the one presented in [4]. We expanded JointJS library functionality by developing a plugin for these OVM primitives.

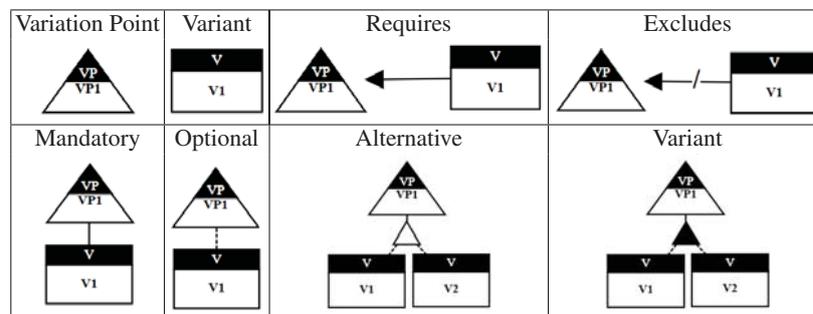


Table 1. OVM Graphical Components

Services

- *Variation Point*: It represents a variable object in real world.
- *Variant*: It represents different forms in which a variation point varies.

Constraint Dependencies: Interactions between variation points, variants or a variant and a variation point.

- *Requires*: It describes the selection of a service requires the selection of its associated service.

- *Excludes*: It describes the selection of a service excludes the selection of its associated service.

Variability Dependencies: Interactions between a variation point and a group of variants and/or variation points.

- *Mandatory*: It describes the selection of a variation point requires the selection of all associated services.

- *Optional*: It describes the selection of a variation point requires the selection of 0 or more associated services.

- *Alternative*: It describes the selection of a variation point requires the selection of only one of the associated services.

- *Variant*: It describes the selection of a variation point requires the selection of at least one of the associated services.

Automated Variability Analysis Variability Management is one of the most important activities in a SPL as it has a great impact on the way software is developed, extended and maintained. Therefore, automated variability analysis has emerged [5, 6, 7, 8], which bases on the extraction of information from models using automated mechanisms. During this process, a variability model is translated into a formal representation, that is sent to a reasoner in order to validate such model, taking into consideration the anomalies and incompatibilities it may contain. The results of this process answer some of the following queries:

- **Valid Model (VM)**: Given a model, it determines whether at least one product can be derived from it.

- **Valid Instantiation (VI)**: Given a model and one of its possible instantiations, it determines whether such instantiation is a valid product.

- **All Products (AP)**: Given an OVM, it returns all possible instantiations.

- **Instantiation Number (IN)**: Given an OVM, it indicates the number of possible instantiations.

- **Problem Detection (PD)**: It identifies the causes or anomalies that invalidate the model and returns this information.

- **Cross Validation (CV)**: It evaluates the validity between models that use the same service.

In particular, during the detection of problems, this undesirable information [6] about models should be identified:

- **Dead Service**: It is a service that will never be derived as part of a product [9].

- **False Optional Service**: Given the selection of its parents, it is a service that is included in all derived products of a product line in spite of not being modelled as mandatory.

- **Wrong Cardinality**: A group or range of cardinality is wrong if it can never be instantiated.

- **Redundancy**: Occurs when semantic information is modelled in different ways.

Comparison with other tools We have surveyed existing tools for automated variability analysis taking into consideration graphical support, variability model, formalisa-

tion method and automatic reasoning support. FaMa-OVM [10] is an extensible tool for automated analysis of OVM diagrams, integrated with multiple off-the-shelf reasoners such as SAT4j, JavaBDD and Choco-Solver. However its input model is specified in a textual format, which makes the model specification a difficult and error-prone task as users have to learn this new syntax. Furthermore, due to the fact that it works with SAT reasoners, it presents certain limitations related to more restrictive logics and thus fail to reflect the finer logical structure of variability models. On the other hand, VariaMos [11] is a tool that allows defining modelling languages and automatically analysing the generated models. This tool consists of a graphical “front-end” and a “back-end” that implements all required functionalities. It is independent from the variability modelling language and works with two reasoners, SWI Prolog and GNU Prolog. Nonetheless, it does not use the standard OVM graphical language, forcing users to learn this new language in order to properly design their models.

In contrast with these tools, *crowd-variability* is a tool that provides users with a graphical interface with all the necessary functionalities for representing OVM diagrams, using the standard graphical language, and is integrated with DL-based reasoning systems which are in charge of carrying out a more precise variability analysis.

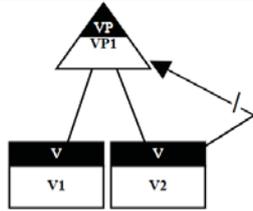
3 An Automated Technique for OVM Analysis

According to the previously described problems, we focus on *dead services* detection for presenting a technique for automated analysis of OVM based on anti-patterns. We emphasise on dead services because they are the most influential in the quality of derived products of a SPL [12]. A service is dead if it cannot appear in any of the generated products of the software product line. Additionally, we also consider the validation scenario named *cross validation* as another modelling failure, where a variant belongs to two or more variation points in the same model. So in order to find these described problems in the context of more complex OVM models, a set of anti-patterns capturing modelling scenarios has been proposed: Mandatory Exclude, Parent Exclude, Alternative Ambiguity, Transitivity Contradiction, Constraint Contradiction, and Cross Validation. These instances were extracted from [12] and renamed for this work. For detecting them, we have extended DL *ALCI* encoding proposed in [9], so as to be able to query the type of the concepts and roles defined in an OVM. According to the encoding in [9], each service is encoded by a concept and each relation between them in a constraint or a variability dependency is encoded by a role. The extended encoding includes new concepts and axioms, so as to classify services and dependencies. Furthermore, we have designed and implemented a series of algorithms that have as input the reasoner answers for queries on services and their relationships. For instance, for the mandatory variability dependency as indicated in Table 1, our proposal identifies such dependency modelled as a DL role and after that it gets both domain (VP1) and range (V1) of such mandatory role.

Now we present our catalogue of anti-patterns, explain and illustrate each one of them, and provide a DL encoding and an algorithm for detecting them in an OVM model. Such anti-patterns detection algorithms are tractable. The time complexity is nm for each one, where n is the cardinality of object properties and m is the number of

relations in the model.

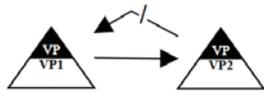
Mandatory Exclude A service excludes a mandatory service. In this scenario, a derived product from this model must include the services VP1, V1 and V2, however, VP1 and V2 are mutually excluded.



$$\begin{aligned}
 &VP1 \text{ MAND}\{V1, V2\} \\
 &\exists vp1v1 \sqsubseteq VP1 \quad \exists vp1v2 \sqsubseteq VP1 \text{ (Domain)} \\
 &\exists vp1v1^- \sqsubseteq V1 \quad \exists vp1v2^- \sqsubseteq V2 \text{ (Range)} \\
 &vp1v1 \sqsubseteq \text{MandatoryDep} \quad vp1v2 \sqsubseteq \text{MandatoryDep} \\
 &\text{If } V2 \text{ EXC } VP1, \text{ then:} \\
 &\exists E_{v2vp1} \sqsubseteq V2 \text{ (Domain)} \quad \exists E_{v2vp1}^- \sqsubseteq VP1 \text{ (Range)} \\
 &v2vp1 \sqsubseteq \text{ExcludesDep}
 \end{aligned}$$

How to detect it? Both domain and range of the excludes dependency should belong to the set of domain and range of the mandatory dependency. This is done by Algorithm 1, where the set of object properties (*ObjectPropertySynset*) for mandatory and excludes are revised and compared.

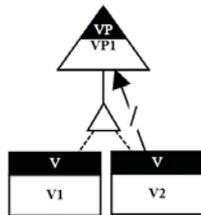
Transitivity Contradiction A service includes another one that excludes the first one. In this scenario, a derived product from this model applying the *requires* dependency must include the services VP1 and VP2, however, VP2 excludes VP1 according to the respective dependency.



$$\begin{aligned}
 &\text{If } VP1 \text{ REQ } VP2, \text{ then:} \\
 &\exists R_{vp1vp2} \sqsubseteq VP1 \text{ (Domain)} \quad \exists R_{vp1vp2}^- \sqsubseteq VP2 \text{ (Range)} \\
 &vp1vp2 \sqsubseteq \text{RequiresDep} \\
 &\text{If } VP2 \text{ EXC } VP1, \text{ then:} \\
 &\exists E_{vp2vp1} \sqsubseteq VP2 \text{ (Domain)} \quad \exists E_{vp2vp1}^- \sqsubseteq VP1 \text{ (Range)} \\
 &vp2vp1 \sqsubseteq \text{ExcludesDep}
 \end{aligned}$$

How to detect it? The domain and range of the excludes dependency should be equal to the range and domain of the requires dependency, respectively. This is done by Algorithm 2, where the set of object properties (*ObjectPropertySynset*) for requires and excludes are revised and compared.

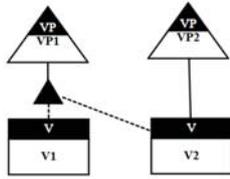
Parent Exclude One or more alternative, variant, or optional services exclude its parent. In this scenario, a derived product from this model must include VP1 and only one of the services V1 and V2, however, V2 excludes VP1.



$$\begin{aligned}
 &VP1 \text{ ALT}\{V1, V2\} \\
 &\exists vp1v1 \sqsubseteq VP1 \quad \exists vp1v2 \sqsubseteq VP1 \text{ (Domain)} \\
 &\exists vp1v1^- \sqsubseteq V1 \quad \exists vp1v2^- \sqsubseteq V2 \text{ (Range)} \\
 &vp1v1 \sqsubseteq \text{AlternativeDep} \quad vp1v2 \sqsubseteq \text{AlternativeDep} \\
 &\text{If } V2 \text{ EXC } VP1, \text{ then:} \\
 &\exists E_{v2vp1} \sqsubseteq V2 \text{ (Domain)} \quad \exists E_{v2vp1}^- \sqsubseteq VP1 \text{ (Range)} \\
 &v2vp1 \sqsubseteq \text{ExcludesDep}
 \end{aligned}$$

How to detect it? Both domain and range of the excludes dependency should belong to the set of domain and range of the alternative (or optional or variant) dependency. This is done by Algorithm 3, where the set of object properties (*ObjectPropertySynset*) for alternative and excludes are revised and compared.

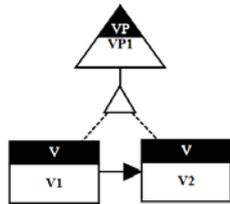
Cross Validation A same service belongs to two or more variation points. In this scenario, a derived product from this model must include the services VP2, V2, VP1, however, V1 may or not be included.



$$\begin{aligned}
 &VP1 \text{ VAR}\{V1, V2\} \\
 &\exists vp1v1 \sqsubseteq VP1 \quad \exists vp1v2 \sqsubseteq VP1 \text{ (Domain)} \\
 &\exists vp1v1^- \sqsubseteq V1 \quad \exists vp1v2^- \sqsubseteq V2 \text{ (Range)} \\
 &vp1v1 \sqsubseteq \text{VariantDep} \quad vp1v2 \sqsubseteq \text{VariantDep} \\
 &VP2 \text{ MAND}\{V2\} \\
 &\exists vp2v2 \sqsubseteq VP2 \text{ (Domain)} \quad \exists vp2v2^- \sqsubseteq V2 \text{ (Range)} \\
 &vp2v2 \sqsubseteq \text{MandatoryDep}
 \end{aligned}$$

How to detect it? The range of the variability dependencies should match in at least one element. This is done by Algorithm 4, where the set of object properties (*ObjectPropertySynset*) for variant and mandatory are revised and compared.

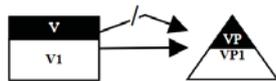
Alternative Ambiguity A service includes an alternative of itself or an alternative of its parents. In this scenario, a derived product from this model must include the services VP1 and only one of the services V1 and V2, however, V1 requires V2 to be included.



$$\begin{aligned}
 &VP1 \text{ ALT}\{V1, V2\} \\
 &\exists vp1v1 \sqsubseteq VP1 \quad \exists vp1v2 \sqsubseteq VP1 \text{ (Domain)} \\
 &\exists vp1v1^- \sqsubseteq V1 \quad \exists vp1v2^- \sqsubseteq V2 \text{ (Range)} \\
 &vp1v1 \sqsubseteq \text{AlternativeDep} \quad vp1v2 \sqsubseteq \text{AlternativeDep} \\
 &\text{If } V1 \text{ REQ } V2, \text{ then:} \\
 &\exists R_{v1v2} \sqsubseteq V1 \text{ (Domain)} \quad \exists R_{v1v2}^- \sqsubseteq V2 \text{ (Range)} \\
 &v1v2 \sqsubseteq \text{RequiresDep}
 \end{aligned}$$

How to detect it? Both domain and range of the requires dependency should belong to the set of range of the alternative dependency. This is done by Algorithm 5, where the set of object properties (*ObjectPropertySynset*) for alternative and requires are revised and compared.

Constraint Contradiction A service includes and excludes the same service, presents a transitive relationship of this case, or presents a dependency with itself. In this scenario, a derived product from this model must include the services V1 and VP1, however, V1 excludes VP1.



$$\begin{aligned}
 &\text{If } V1 \text{ REQ } VP1, \text{ then:} \\
 &\exists R_{v1vp1} \sqsubseteq V1 \text{ (Domain)} \quad \exists R_{v1vp1}^- \sqsubseteq VP1 \text{ (Range)} \\
 &v1vp1 \sqsubseteq \text{RequiresDep} \\
 &\text{If } V1 \text{ EXC } VP1, \text{ then:} \\
 &\exists E_{v1vp1} \sqsubseteq V1 \text{ (Domain)} \quad \exists E_{v1vp1}^- \sqsubseteq VP1 \text{ (Range)} \\
 &v1vp1 \sqsubseteq \text{ExcludesDep}
 \end{aligned}$$

How to detect it? The domain and range of the restriction dependencies should be equal. This is done by Algorithm 6, where the set of object properties (*ObjectPropertySynset*) for requires and excludes are revised and compared.

Algorithm 1: Mandatory Exclude

```

1: AntiPattern = []
2: Mandatories = []
3: for all ObjectPropertySynset mand in GetSubObjectProperties(MandatoryDep) do
4:   Add Domain(mand) → Range(mand) to Mandatories
5: end for
6: for all ObjectPropertySynset exc in GetSubObjectProperties(ExcludesDep) do
7:   for all origin → targets in Mandatories do
8:     if {Domain(exc), Range(exc)} is included in {origin, targets} then
9:       Add {origin, targets} to AntiPattern
10:    end if
11:  end for
12: end for
13: return AntiPattern

```

Algorithm 3: Parent Exclude

```

1: AntiPattern = []
2: Optionals = []
3: Alternatives = []
4: Variants = []
5: for all ObjectPropertySynset opt in GetSubObjectProperties(OptionalDep) do
6:   Add Domain(opt) → Range(opt) to Optionals
7: end for
8: for all ObjectPropertySynset alt in GetSubObjectProperties(AlternativeDep) do
9:   Add Domain(alt) → Range(alt) to Alternatives
10: end for
11: for all ObjectPropertySynset var in GetSubObjectProperties(VariantDep) do
12:   Add Domain(var) → Range(var) to Variants
13: end for
14: for all ObjectPropertySynset exc in GetSubObjectProperties(ExcludesDep) do
15:   for all origin → targets in Optionals do
16:     if Range(exc) = origin and {Domain(exc)} is included in targets then
17:       Add {origin, targets} to AntiPattern
18:     end if
19:   end for
20:   for all origin → targets in Alternatives do
21:     if Range(exc) = origin and {Domain(exc)} is included in targets then
22:       Add {origin, targets} to AntiPattern
23:     end if
24:   end for
25:   for all origin → targets in Variants do
26:     if Range(exc) = origin and {Domain(exc)} is included in targets then
27:       Add {origin, targets} to AntiPattern
28:     end if
29:   end for
30: end for
31: return AntiPattern

```

Algorithm 2: Trans. Contradiction

```

1: AntiPattern = []
2: for all ObjectPropertySynset exc in GetSubObjectProperties(ExcludesDep) do
3:   for all ObjectPropertySynset req in GetSubObjectProperties(RequiresDep) do
4:     if Domain(exc) = Range(req) and Domain(req) = Range(exc) then
5:       Add {Domain(exc), Range(exc)} to AntiPattern
6:     end if
7:   end for
8: end for
9: return AntiPattern

```

Algorithm 4: Cross Validation

```

1: AntiPattern = []
2: Targets = []
3: for all ObjectPropertySynset mand in GetSubObjectProperties(MandatoryDep) do
4:   Add Range(mand) to Targets
5: end for
6: for all ObjectPropertySynset opt in GetSubObjectProperties(OptionalDep) do
7:   Add Range(opt) to Targets
8: end for
9: for all ObjectPropertySynset alt in GetSubObjectProperties(AlternativeDep) do
10:  Add Range(alt) to Targets
11: end for
12: for all ObjectPropertySynset var in GetSubObjectProperties(VariantDep) do
13:  Add Range(var) to Targets
14: end for
15: Add getDuplicates(Targets) to AntiPattern
16: return AntiPattern

```

Algorithm 5: Alt. Ambiguity

```

1: AntiPattern = []
2: Alternatives = []
3: for all ObjectPropertySynset alt in GetSubObjectProperties(AlternativeDep) do
4:   Add Domain(alt) → Range(alt) to Alternatives
5: end for
6: for all ObjectPropertySynset req in GetSubObjectProperties(RequiresDep) do
7:   for all origin → targets in Alternatives do
8:     if {Domain(req), Range(req)} is included in targets then
9:       Add {origin, targets} to AntiPattern
10:    end if
11:  end for
12: end for
13: return AntiPattern

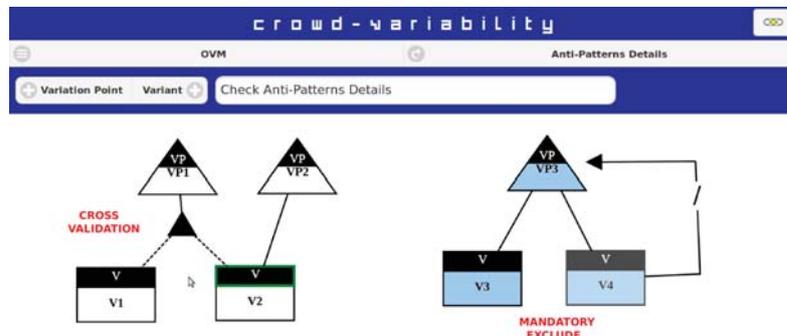
```

Algorithm 6: Cons. Contradiction

```

1: AntiPattern = []
2: for all ObjectPropertySynset exc in GetSubObjectProperties(ExcludesDep) do
3:   if (Domain(exc) = Range(exc)) then
4:     Add Domain(exc) to AntiPattern
5:   end if
6: for all ObjectPropertySynset req in GetSubObjectProperties(RequiresDep) do
7:   if ( (Domain(exc) = Domain(req) and Range(req) = Range(exc)) or (Domain(req) = Range(req)) ) then
8:     Add {Domain(req), Range(req)} to AntiPattern
9:   end if
10: end for
11: end for
12: return AntiPattern

```

4 crowd-variability for Anti-patterns Detection**Figure 1.** crowd-variability Front-End

New prototype of *crowd-variability* is integrated with Racer [13] and SPARQL-DL [14] reasoners. For this reason, it offers users the possibility of requesting an anti-pattern detection of a model by pressing the semaphore presented in Figure 1. By doing so, a JSON representation of the model is sent to the server, through a HTTP request, in order to be formalized in *ALCI* DL by the Formal Language Translator. This module generates a document in OWL syntax and another one in OWLlink syntax with the formal representation of the model. Later, The Query Generator creates a series of queries for each document. The OWLlink document, comprised of the model's formalisation and its queries, is sent to RACER reasoner and the OWL document with its queries are sent separately to SPARQL-DL reasoner. The results generated by the reasoners are later processed by the Answer Analyser. This module produces a JSON file with information about anti-patterns occurrences in the diagram. Finally, the client side receives this JSON file and graphically presents such information to the user by colouring those services involved in anti-patterns. Each anti-pattern has a corresponding colour associated. Figure 2 presents the path followed by a model starting in the

GUI for its creation and ending in the GUI with all its services involved in anti-pattern coloured and listed (see Figure 1). Other tests can be found by following this link: <https://bitbucket.org/gilia/crowd-variability/wiki/>.

Algorithms for anti-patterns detection were designed for recognising the scenarios presented in the catalogue. Therefore, their recognition is made at only one level of depth of the models. However, they can be generalised in order to detect dead services in larger diagrams. Likewise, new anti-patterns can be added to the catalogue and their associated algorithms for detection can be implemented in *crowd-variability*.

Lastly, *crowd-variability* has been developed using expansible graphical libraries and Web technologies. Therefore, it can be extended to other variability modelling proposals by creating or adding new plug-ins to the JointJs graphics library, incorporating new formalisation methods, or connecting new back-end reasoners. Hence, a relevant feature of crowd variability is to consider users' preferences and usages, allowing the selection of different approaches to model variability and distinct reasoners for the reasoning service.

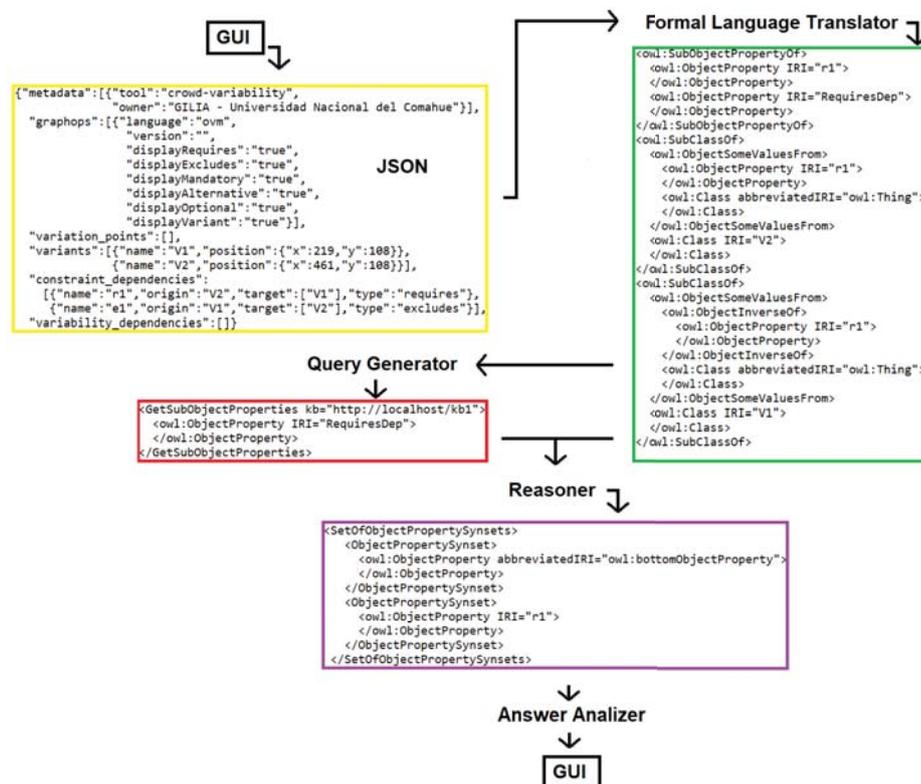


Figure 2. *crowd-variability* Back-End

5 Conclusions and Future Work

In this work, we presented a catalogue of anti-patterns for automated analysis of variability models. Such anti-patterns match specific modelling scenarios obtained from the literature. Moreover, we extend our tool *crowd-variability* for detecting them by extending the DL encoding of OVM and providing tractable algorithms for processing reasoning outputs. In spite of the fact that this is a starting catalogue, our technique detects dead services, which are the most influential in the quality of derived products of a SPL. However, the approach presents some limitations associated to the complexity of models because the anti-patterns detection is made at only one level of depth in the models.

In future works, we plan to extend its support for automated variability analysis by generalising anti-patterns detection algorithms, expanding the catalogue and considering other queries about the model. Furthermore, we will continue evaluating *crowd-variability* so as to enhance and extend its functionalities.

References

1. A. Oyarzun, G. Braun, L. Cecchi, and P. Fillottrani. A graphical web tool with dl-based reasoning support over orthogonal variability models. In *XXIV CACIC.*, 2018.
2. T. Sales and G. Guizzardi. Ontological anti-patterns: Empirically uncovered error-prone structures in ontology-driven conceptual models. *Data & Knowledge Engineering*, 2015.
3. M. Raatikainen, J. Tiihonen, and T. Männistö. Software product lines and variability modeling: A tertiary study. *Journal of Systems and Software*, 2019.
4. K. Pohl, G. Böckle, and F. van der Linden. *Software Product Line Engineering: Foundations, Principles and Techniques*. 2005.
5. F. Roos-Frantz, J. Galindo, D. Benavides, A. Ruiz-Cortés, and J. Garcia-Galán. Automated analysis of diverse variability models with tool support. *Jornadas de Ingeniería del Software y de Bases de Datos (JISBD 2014)*, 2014.
6. D. Benavides, S. Segura, and A. Ruiz-Cortés. Automated analysis of feature models 20 years later: A literature review. *Inf. Syst.*, 2010.
7. M. Kowal, S. Ananieva, and T. Thüm. Explaining anomalies in feature models. In *ACM SIGPLAN, GPCE 2016*. ACM, 2016.
8. A. Metzger, K. Pohl, P. Heymans, P. Y. Schobbens, and G. Saval. Disambiguating the Documentation of Variability in Software Product Lines: A Separation of Concerns, Formalization and Automated Analysis. In *15th IEEE IREC*, 2007.
9. G. Braun, M. Pol'la, L. Cecchi, A. Buccella, P. Fillottrani, and A. Cechich. A DL Semantics for Reasoning over OVM-based Variability Models. In *Description Logics*, 2017.
10. F. Roos Frantz, J. Galindo Duarte, D. Benavides Cuevas, and A. Ruiz Cortés. FaMa-OVM: A tool for the automated analysis of ovm. In *16th ISPLC*, 2012.
11. R. Mazo, J. Muñoz-Fernández, L. Rincón-Perez, C. Salinesi, and G. Tamura. VariaMos: an extensible tool for engineering (dynamic) product lines. In *SPLC 2015*, 2015.
12. M. Pol'la, A. Buccella, and A. Cechich. Automated analysis of variability models: The SeVaTax Process. In *ICCSA*. Springer, 2018.
13. V. Haarslev and R. Möller. RACER System Description. In Rajeev Goré, Alexander Leitsch, and Tobias Nipkow, editors, *Automated Reasoning*, pages 701–705, Berlin, Heidelberg, 2001. Springer Berlin Heidelberg.
14. E. Sirin and B. Parsia. SPARQL-DL: SPARQL Query for OWL-DL. In *In 3rd OWL Experiences and Directions Workshop (OWLED-2007)*, 2007.

Arquitectura de segmento terreno satelital adaptada para el control de límites de telemetría dinámicos

Pablo Soligo, Jorge Salvador Ierache 

Universidad Nacional de La Matanza
Departamento de Ingeniería e Investigaciones Tecnológicas.
Florencio Varela 1903, B1754JEC San Justo, Buenos Aires.
psoligo@unlam.edu.ar
<http://www.unlam.edu.ar/>

Resumen El presente trabajo muestra la implementación de un prototipo funcional para el monitoreo del estado de salud de un satélite artificial científico. Aunque aplicable a otros sistemas, la implementación inicial ha sido probada utilizando como fuente de datos telemetría satelital y como prototipo inicial de desarrollo el Prototipo de Segmento Terreno de la UNLaM (UGS) derivado de la misión académica Formador Satelital 2017 (FS2017). Utilizando herramientas de estantería en cualquiera de sus variantes Off-the-Shelf (OTS), Commercial-Off-The-Shelf (COTS) u Open Source Software (OSS), el prototipo analiza los datos históricos para obtener patrones de comportamiento que permitan ajustar los límites de control y efectuar un monitoreo mas cercano y preciso de la telemetría, en tiempo real, manteniendo los costos operativos acotados y utilizando hardware de propósito general.

1. Introducción

Existen tres métodos extendidos para analizar la telemetría de un satélite y controlar el estado de salud; control de límites, sistemas expertos y sistemas basados en modelos [1].

Control de límites: El control de límites es el sistema más simple y difundido, consiste en establecer, con la ayuda de un experto, los valores máximos y mínimos aceptables para un sensor (temperaturas, voltajes, corrientes) o conjunto de sensores y verificar que los mismos no sean violados. Estos valores usualmente deben ser ajustados con el sistema en vuelo y la cantidad de sensores puede volver impracticable por este método el control de todos los sensores. La simplicidad vuelve a este sistema el mas popular, aunque se puede deducir que el mismo es absolutamente insensible al contexto.

Sistemas expertos: Durante los 80 y inicio de los 90 sistemas expertos han sido probados en diferentes misiones (Ej: GEOTAIL, NOZOMI y HAYABUSA). Estas técnicas si bien han mejorado el rendimiento de los sistemas

basados exclusivamente en control de límites no están exentas de problemas. Los sistemas expertos no pueden encontrar desperfectos desconocidos o no contemplados y requieren una laboriosa configuración y el conocimiento incorporado puede volverse inconsistente por cambios de diseño o por desconocer el comportamiento del satélite después del lanzamiento.

Sistemas basados en modelos: En esos sistemas la idea principal es detectar las anomalías y sus razones comparando simulaciones obtenidas por modelos computacionales contra el comportamiento actual del sistema.[1].

Cualquiera de los métodos mencionados requieren un asesoramiento casi continuo del experto en el área, ya sea actualizando límites, generando reglas o parametrizando simulaciones.

Por otro lado el aprendizaje automático ofrece un amplio rango de posibilidades para la predicción de comportamientos. Las técnicas de aprendizaje automático intentan servirse de la datos disponibles para predecir comportamientos futuros. En lugar de requerir a personal experto que infiera reglas y construya modelos, el aprendizaje automático ofrece una forma mas eficiente para capturar el conocimiento, ha salido de los laboratorios y hoy es utilizado en el reconocimiento de voz e imágenes, en la búsquedas en Internet, en el ofrecimiento de productos y las fronteras de aplicación todavía no están establecidas.

Específicamente en el área de machine learning y telemetría satelital existen algunos antecedentes que abren el camino a estos estudios, los trabajos de [2], [1] y mas recientemente [3] ofrecen alternativas distintas para la solución del mismo problema.

Los trabajos citados se concentran establecer conclusiones, mayormente prometedoras, de las posibilidades que tiene el aprendizaje automático en el monitoreo del estado de salud de un satélite sin especificar una implementación práctica, objetivo principal de este trabajo.

2. Antecedentes

Durante la experiencias obtenidas durante la primera cohorte de la Maestría en Desarrollos Informáticos de Aplicación Espacial (MDIAE) (2015-2017) se genero el primer prototipo funcional del segmento terreno ahora denominado UGS para la misión FS2017. Se utilizo como segmento de vuelo un modelo de ingeniería de cubesat de 2U de la compañía ISIS (<https://www.isispace.nl/>) [4], [5] y [6]. En particular para el control del estado de salud se utilizo un sistema de control de límites, durante la MDIAE se tuvieron experiencias con multiples sistemas del área espacial que hacen uso de esta técnica. Se han priorizado sobre las decisiones de diseño las recomendaciones que publicara en 2003 el Instituto Norteamericano de Aeronáutica y Astronáutica, del inglés Institute of Aeronautics and Astronautics (AIAA) [7], intentando cumplir los siguientes objetivos:

- Independencia de la arquitectura de hardware
- Independencia del fabricante del hardware
- Tolerancia a fallas

- Implementable en computadoras personales comerciales
- Independencia del sistema operativo

Con estas premisas y priorizando soluciones comúnmente utilizadas en los sistemas de propósito general se ha optado por:

1. Utilizar interpretes de propósito general para la decodificación de telemetría y los scripts de comandos en lugar de desarrollar interpretes propios (Solución extendida en el área espacial)
2. Utilizar base de datos relacionales (Sistema de base de datos relacional, del inglés Relational Database Management System (RDBMS)) y no relacionales para el almacenamiento de datos y metadatos.
3. Utilizar herramientas de máxima penetración para la distribución y ejecución de tareas.
4. Utilizar capas de acceso a los datos (Object Relational Mapper (ORM)) que faciliten la portabilidad entre motores de RDBMS y otras tecnologías de almacenamiento.
5. Utilizar protocolos de comunicación a nivel de aplicación comúnmente aceptados en el ámbito informático de propósito general.

El UGS, desde su primera versión, opta por el uso de un lenguaje de propósito general para el procesamiento de telemetría como para la generación de scripts de comandos. Por popularidad [8], productividad y facilidad de uso la opción elegida fue python. Al tratarse de un interprete, se ha utilizado las capacidades de reflexión de software como herramienta para la carga dinámica de procedimientos y posterior el procesamiento en tiempo real de los datos satelitales [5] y [4]. Puntualmente respecto al uso de lenguajes de propósito general, ya en el 2008, trabajos en la industria privada explicitaban las ventajas de este enfoque, considerando: [9]:

Procesamiento: XML, SOAP, XTCE parsing

Comunicaciones: Sockets, Acceso a Internet, Llamada a procedimiento remoto, del inglés Remote Procedure Call (RPC), Correo Electrónico

Mediciones de performance: Contadores, acceso al hardware y sistema operativo.

Otros: Acceso a bases de datos, funciones matemáticas, compresión de datos, hilos, criptografía.

Procesar telemetría utilizando carga en tiempo de ejecución y reflexión de software y la opción de utilizar un lenguaje e interprete de propósito general habilitan a una fracción de costo la incorporación de bibliotecas para la minería de datos y el aprendizaje automático. Todo esto suma un argumento mas a la lista expresada en [9].

3. Arquitectura e implementación

Originalmente el primer prototipo desarrollado de segmento terreno desarrollado para el FS2017 [6] contaba de un servidor de base de datos (RDBMS), una

aplicación web y una serie de productos a ejecutar manualmente. Actualmente el sistema trabaja sobre procesos distribuidos. Un subconjunto variable de procesos se ejecuta de manera distribuida dentro de la infraestructura de hardware disponible. Cada nueva ingesta de telemetría, independientemente de la fuente, dispara un proceso distribuido encargado de procesar el paquete, generar y persistir las variables de telemetría que el paquete contenga. Se generan todas las variables de telemetría que el satélite tiene configuradas, activas y se corresponden con el tipo de paquete. Actualmente el prototipo puede trabajar con telemetría de múltiples misiones incluida Satélite Argentino Científico - D (SAC-D) ([10]) utilizado en esta prueba. Desde su primera versión, el prototipo verifica el estado de salud del satélite mediante un control de límites (1), tanto para sus variables directas como derivadas, estas últimas son variables que se crean artificialmente a partir de la combinación de otras. La adaptación mostrada en este trabajo propone crear una entidad asociada al tipo de telemetría donde almacenar un modelo de predicción automáticamente creado. La figura 1 muestra el diagrama de clases (Solo se muestran los campos sensibles al análisis) donde se pueden encontrar los campos para almacenamiento del modelo (Clase **TlmyPrediction**, atributo *data*) y fecha de expiración (Clase **TlmyPrediction**, atributo *expiration*), esta entidad esta asociada al tipo de telemetría **TlmyVarType**.

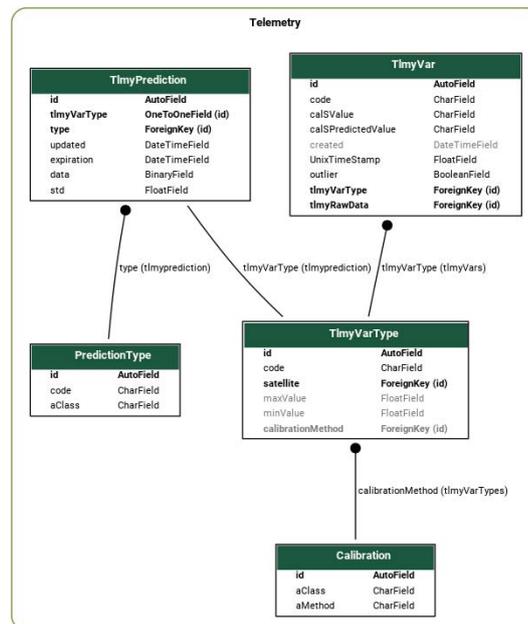


Figura 1: Diagrama de clases

La predicción puede no estar disponible, no ser conveniente o no presentar relevancia para el tipo de telemetría, por esta razón la relación no es obligatoria. Si se recibe un nuevo paquete de datos (independientemente de la fuente) y este contiene un tipo de telemetría determinado y este tipo tiene un modelo para la predicción, el mismo es aplicado modificando los límites máximos y mínimos de ese tipo de telemetría en un valor desplazado desde el valor predicho, positivamente y negativamente una distancia configurada. En las pruebas desarrolladas se ha tomado un σ (Sigma) como desplazamiento (1 y 2).

$$max = p + \sigma \quad (1)$$

$$min = p - \sigma \quad (2)$$

Estableciendo los máximos y mínimos dinámicamente en función de una predicción se pueden obtener controles sensibles al contexto actualizado del satélite. En el caso de análisis de este trabajo se ha creado una variable derivada denominada **eclipseElapsedTime**, esta variable contiene el tiempo en segundos que el satélite permanece eclipsado o la cantidad de segundos (negativos) hasta el próximo eclipse. La figura 2 muestra las variable directa **vBatAverage** (Voltaje medio de batería) del SAC-D y la variable derivada **eclipseElapsedTime** calculada en *runtime* dentro del mismo sistema utilizando pyephem [11] (Esta última variable podría ser provista por los servicios del área de determinación orbital en una agencia espacial [12]).

En la figura 2 se puede observar como la tensión decrece (variable **vBatAverage**) durante el eclipse, los paneles solares quedan parcial o totalmente tapados por la tierra durante algunos minutos.

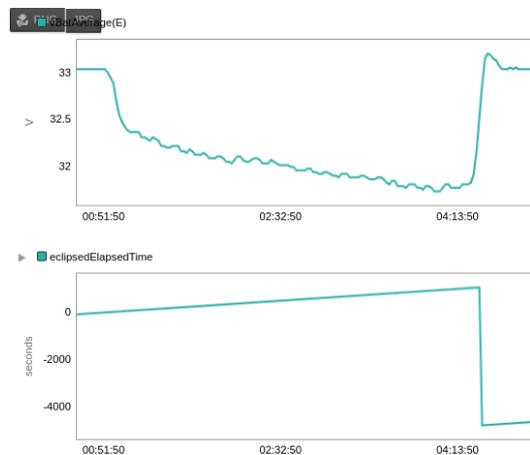


Figura 2: Caída de tensión al entrar en eclipse, ambos gráficos apilados muestran el mismo periodo de tiempo. La figura superior muestra el voltaje y la figura inferior la variable derivada **eclipseElapsedTime**.

Una vez que los paneles vuelven a tener la iluminación solar ocurre una rápida recuperación durante el periodo entre umbra y penumbra. La figura 3 muestra que el patrón se repite durante varios eclipses.

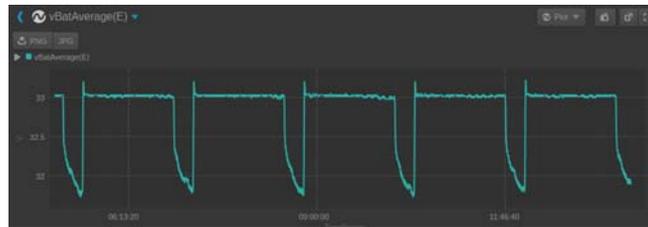


Figura 3: Variable directa **vBatAverage** visualizada por OPENMCT

Utilizando como técnica el control únicamente límites estáticos(1), para el ejemplo dado, se deberían establecer límites máximos y mínimos del orden de los 30v y 34v respectivamente. Mediciones de 31v durante los periodos entre eclipses no serían tomadas como una anomalía cuando en realidad se podría estar presentando un problema grave en la carga y un comportamiento anormal. Un conjunto de reglas explicitadas por un experto pueden salvar esta situación, aunque es necesario el experto y por otro lado los valores finales solo se conocen con el sistema en vuelo.

En el sistema propuesto, un proceso distribuido (Figura 4, Tareas Periódicas) revisa los tipos de telemetría que tienen modelos de predicción vencidos, es decir, superan su fecha expiración (propiedad *expiration*, entidad **TlmyPrediction**). Previamente ese tipo de telemetría fue configurado como sensible a predicción. Los modelos tienen una fecha de vencimiento que puede forzar su regeneración. Para cada tipo de telemetría en esta situación se realiza una nueva preparación de datos según configuración, se genera un nuevo modelo de predicción y en el caso de ser lo suficientemente preciso se persiste, caso contrario se informa mediante una alarma.

Al utilizar un lenguaje de propósito general para el procesamiento de telemetría se puede hacer uso de las múltiples bibliotecas disponibles para aprendizaje automático y normalmente existen herramientas para la persistencia (Almacenamiento en el algún medio permanente) de los modelos de predicción. Cuando un nuevo paquete de telemetría ingresa al sistema mediante la Interfaz de programación de aplicaciones, del inglés API: Application Programming Interface (API) se procesa el paquete y se llama a la función de calibración para cada uno de los tipos de variables de telemetría configuradas para ese paquete (Figura 4, Procesador de telemetría). Si el tipo de telemetría tiene asociado un modelo de predicción entonces este también se aplica y se utilizan el resultado para ajustar los máximos y mínimos tolerables, si el valor calibrado está por fuera de los límites dinámicos entonces se genera una alarma siguiendo el flujo establecido en el sistema.

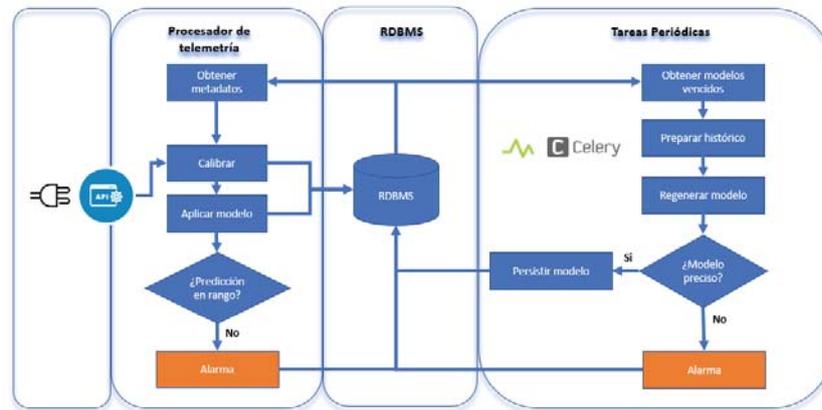


Figura 4: Arquitectura conceptual para la verificación de límites en la telemetría

4. Resultados

La figura 5 muestra los resultados aplicando regresión polinómica y aplicando un árbol de regresión lineal (Biblioteca scikit-learn [13]). En ambos casos los valores predichos están dentro del rango de un σ establecido. Se puede observar tanto en los periodos entre eclipses como dentro del eclipse como los máximos y mínimos se ajustan permitiendo un control mas estricto y por sobre todo, sensible al contexto.

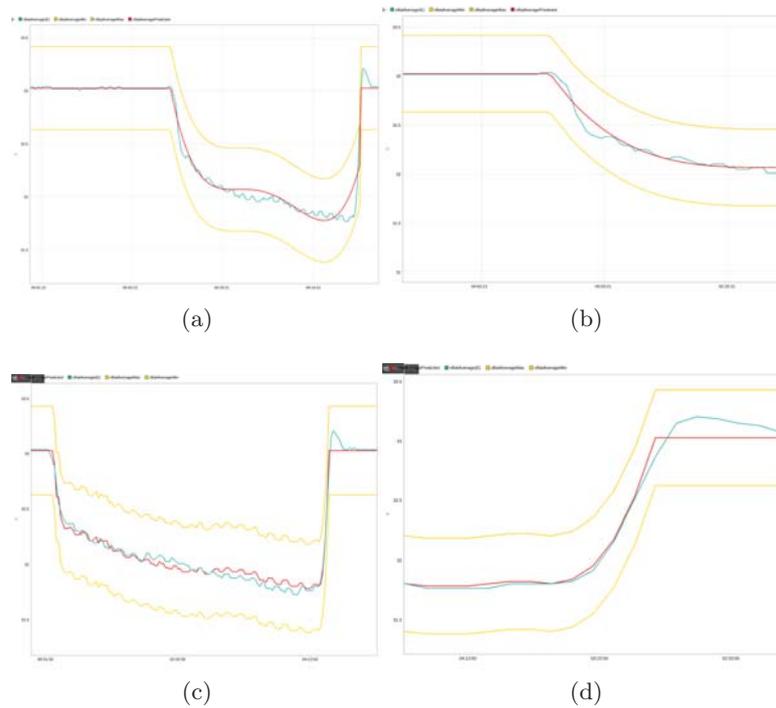
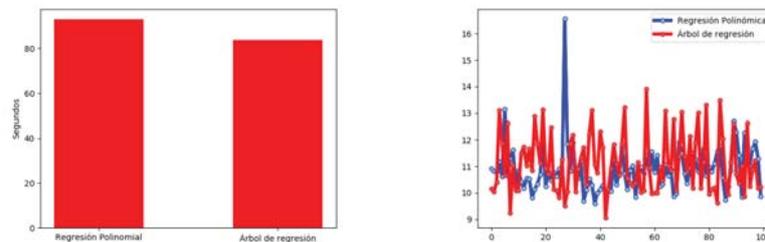


Figura 5: (a) Predicción utilizando regresión polinómica (Eclipse completo) (b) Predicción utilizando regresión polinómica (Entrada en eclipse) (c) Predicción utilizando árbol de regresión (Eclipse completo) y (d) Predicción utilizando árbol de regresión (Salida del eclipse), donde celeste es valor real, rojo valor predicho y amarillos máximos y mínimos

Los máximos y mínimos son ahora sensibles al contexto, entendiendo por contexto a la variable derivada que indica el periodo entre eclipses en segundos. La figura 6 (a) muestra los tiempos de generación del modelo incluyendo la búsqueda de los datos (12 horas), la preparación y persistencia de los datos para alimentar el algoritmo y la generación y persistencia del modelo en sí. La figura 6 (b) muestra (En milisegundos) los tiempos necesarios para la generación, utilizando el modelo persistido, del valor predicho y el ajuste de los límites para la variable **vBatAverage**. Se muestran los tiempos para 100 paquetes SAC-D, los tiempos no contemplan el control de límites que es independiente de si el límite es modificado dinámicamente o establecido por configuración.



(a) Tiempo de generación del modelo de predicción

(b) Tiempo en milisegundos para la predicción de variables **vBattAverage** pertenecientes a satélite SAC-D (100 muestras).

Figura 6

5. Conclusiones

El presente trabajo muestra una propuesta de implementación que permite aplicar las herramientas contemporáneas de aprendizaje automático a un segmento terreno desarrollado bajo la premisas expresadas en trabajos anteriores [6], [5] y [4]. La capacidad añadida no implicó modificación alguna al núcleo del segmento terreno original y su implementación es transparente. Como se mostró en las figuras 6 (a) y 6 (b), sin generar un coste relevante en cuanto a tiempo de procesamiento, la idea permite un control más cercano de la telemetría de housekeeping sin necesidad de intervención humana durante la operación. El costo y riesgo de este tipo de misiones justifican el desarrollo estas técnicas y suman un argumento adicional a favor del trabajo con herramientas de propósito general en el área espacial.

6. Trabajo a Futuro

La propuesta presentada en este trabajo no está exenta de problemas o puntos a profundizar. La búsqueda de relaciones, o variables contextuales aun requieren del asesoramiento del un experto para su correcta identificación. En próximos trabajos se abordará esta problemática en función de encontrar correlaciones y explotaras de manera automática, buscando explotar tanto como sea posible el poder de computo actual, no como remplazo del experto, sino como un aliado en la búsqueda e identificación de patrones ocultos.

Por otro lado, aunque se ha contemplado la eliminación de *outliers* durante la preparación de los datos de entrenamiento, la aparición de estos durante la operación y su impacto en la generación de falsos positivos no ha sido tenida en

cuenta en esta primera versión, la incorporación de un filtro que detecte *outliers* en tiempo real esta pendiente.

Referencias

1. Takehisa Yairi, Yoshinobu Kawahara, Ryohei Fujimaki, Yuichi Sato, and Kazuo Machida. Telemetry-mining: a machine learning approach to anomaly detection and fault diagnosis for space systems. In *Space Mission Challenges for Information Technology, 2006. SMC-IT 2006. Second IEEE International Conference on*, pages 8–pp. IEEE, 2006.
2. Takehisa Yairi, Minoru Nakatsugawa, Koichi Hori, Shinichi Nakasuka, Kazuo Machida, and Naoki Ishihama. Adaptive limit checking for spacecraft telemetry data using regression tree learning. In *2004 IEEE International Conference on Systems, Man and Cybernetics (IEEE Cat. No. 04CH37583)*, volume 6, pages 5130–5135. IEEE, 2004.
3. L.Amoruso C.Brighenti, A.Brighenti D.Evans, M.Ricatto D.Moretto, and M.Carbone F.Ferrari. Advances in context aware spacecraft telemetry checking. In *2018 IAC*. IAC, 2018.
4. Pablo Soligo and Jorge Salvador Ierache. Segmento terreno para misiones espaciales de próxima generación. *WICC 2019*.
5. Pablo Soligo and Jorge Salvador Ierache. Software de segmento terreno de próxima generación. In *XXIV Congreso Argentino de Ciencias de la Computación (La Plata, 2018)*., 2018.
6. Pablo Soligo, Ezequiel González, Eduardo Sufán, Emmanuel Arias, Ricardo Barbieri, Pablo Estrada, Alfonso Montilla, José Robin, Javier Uranga, M Cecilia Valenti, et al. Misión cubesat fs2017: Desarrollo de software para una misión satelital universitaria. *WICC 2017*, page 843.
7. Ken Galal and Roger P Hogan. Satellite mission operations best practices. 2001.
8. TIOBE Software. TIOBE programming community index, september 2017, 2017. [Online; accessed 26-September-2017].
9. Gonzalo Garcia. Use of python as a satellite operations and testing automation language. In *GSAW2008 Conference, Redondo Beach, California*, 2008.
10. Amit Sen, Yunjin Kim, Daniel Caruso, Gary Lagerloef, Raul Colomb, Simon Yueh, and David Le Vine. Aquarius/sac-d mission overview. In *Sensors, Systems, and Next-Generation Satellites X*, volume 6361, page 63610I. International Society for Optics and Photonics, 2006.
11. Brandon Craig Rhodes. Pyephem: astronomical ephemeris for python. *Astrophysics Source Code Library*, 2011.
12. S. Hackel, Y. Wasser, M. Meinel, and R. Kahle. Flight dynamics microservices. In *Proceedings of the International Astronautical Congress, IAC*, volume 2018-October, 2018.
13. scikit-learn machine learning in python. <https://scikit-learn.org/stable/>. Accessed: 2019-23-07.

Baboon, Framework Conducido por Red de Petri para Sistemas Reactivos Dirigidos por Eventos

Ing. Ariel Iván Rabinovich¹, Ing. Luis Orlando Ventre¹, Dr. Ing. Orlando Micolini¹

¹Laboratorio de Arquitectura de Computadoras, FCEfyn-Universidad Nacional de Córdoba
Av. Velez Sarfield 1601, CP-5000, Córdoba, Argentina
{ariel.rabinovich, luis.ventre, orlando.micolini}@unc.edu.ar

Abstract. Ensuring that a concurrent program is correct means that during its execution no thread suffers starvation, no set of threads falls into a deadlock and their successive states lead to the solution. To assert this, it is necessary to appeal to formal methods.

This paper proposes using a Petri Net as the logic of a reactive system, and its execution within a framework based on a concurrency monitor. It is put forward that the logical execution flow of the implemented system is driven by the execution of the model. Thus, the model transfers its properties to the system, which has been verified.

Keywords: Red de Petri, Sistema Reactivo, Programación Concurrente, Concurrencia, Modelo, Framework

1 Introducción

Este trabajo propone un framework para implementar sistemas críticos, reactivos (RS) y guiados por eventos (EDA) [1, 2]. Estos sistemas tienen que cumplir requerimientos no funcionales específicos debido a que interactúan con variables y eventos del propio sistema y del mundo exterior, donde los datos y eventos son heterogéneos y no determinísticos. Existe un gran número de propuestas dedicadas a estudiar cómo implementar estos sistemas a partir de máquinas de estados en diferentes lenguajes de programación en una amplia variedad de contextos de aplicación, tales como sistemas de control modernos [3], sistemas de control de producción [4], sistemas de control de misiones espaciales de la NASA [5], etc.

En particular, cuando se trata de sistemas paralelos y concurrentes, la generación de código a partir de máquinas de estados constituye una tarea compleja, debido a que muchos de los conceptos de especificación no están correctamente soportados, tales como eventos concurrentes, estados locales y globales, pseudo-estados históricos, o pseudo-estados de bifurcación [6, 7]. Puesto que éstos son aspectos centrales en los sistemas reactivos a implementar en este framework; es de gran importancia que el modelo los identifique. Dado que las redes de Petri (RdP) no autónomas [8] evidencian estas características, son ejecutables [9], y escalables cuando se las expresa con la ecuación de estado extendida [10] se ha considerado que son el formalismo más conveniente.

La originalidad de éste trabajo es el diseño de un framework que integra la ejecución de una RdP no autónoma y las acciones con un monitor de concurrencia [11]. El framework ejecuta la red como el modelo lógico del sistema, basándose en la ecuación de estado extendida, de esta manera permite desarrollar RS confiables compuestos por una arquitectura probada y una lógica verificada formalmente.

La importancia del framework aquí presentado radica en la reusabilidad que se logra con la inclusión de *tópicos* para desacoplar la lógica y la política de las acciones. Reflejando la desagregación de las vistas funcionales y de comportamiento, con el fin de obtener un sistema resultante modular, simple, mantenible, validable y verificable formalmente. Para alcanzar estas características, la lógica es modelada con RdP no autónomas y la política no es integrada en la RdP. La política es determinada a partir de los conflictos presentes en la red [8].

Como antecedente a este trabajo se encuentra en [9] un estudio detallado de distintas propuestas que utilizan las RdP para la solución de RS y EDA. Asimismo, en [12] se ha desarrollado un procesador que ejecuta la RdP por hardware.

En este trabajo se ha extendido la propuesta realizada en [13], integrando en un framework las vistas del modelo con la metodología para el desarrollo de sistemas haciendo uso de la ecuación de estado generalizada. También, se pueden encontrar numerosos casos de simulación resueltos con RdP en [14].

En el siguiente apartado se presentan conceptos y notaciones fundamentales sobre RS y modelos. Luego, en las siguientes secciones, se describe en detalle la arquitectura desarrollada, los resultados obtenidos y finalmente las conclusiones.

2 Marco Teórico

2.1 Sistemas Reactivos

Un sistema reactivo típico presenta las siguientes características: interactúa continuamente con su entorno, utilizando entradas y salidas que son continuas o discretas. Las entradas y salidas son a menudo asíncronas, lo que significa que pueden arribar o cambiar valores de forma impredecible. El sistema debe ser capaz de responder a las interrupciones, es decir, a eventos de alta prioridad. Su funcionamiento y reacción a las entradas frecuentemente refleja los estrictos requisitos de tiempo [2]. La mayoría de los sistemas embebidos de tiempo real, control, supervisión, procesamiento de señales, protocolos de comunicación e interfaces hombre-máquina son sistemas reactivos.

La implementación de sistemas reactivos resulta compleja debido a que es imposible predecir o controlar el orden de llegada de los eventos externos. Esto implica la necesidad de estructuras de control y manejo de eventos, para asegurar el correcto funcionamiento según la lógica, política y acciones del sistema.

2.2 Modelado y Vistas de un Sistema Reactivo

La construcción de un modelo puede considerarse como una transición de ideas y descripciones informales a descripciones concretas que utilizan conceptos y terminología predefinida. En este trabajo se consideran las siguientes descripciones utilizadas en [1] para capturar la especificación del sistema:

La *vista funcional* captura el "qué": Describe las funciones, los procesos o los objetos del sistema, también llamados actividades, precisando así sus capacidades. Esta vista también incluye las entradas y salidas de las actividades, es decir, el flujo de información hacia y desde el entorno externo del sistema, así como la información que fluye entre las actividades internas.

La *vista de comportamiento* captura el "cuándo": describe el comportamiento del sistema a lo largo del tiempo, incluyendo la dinámica de las actividades, su control y comportamiento temporal, los estados y modos del sistema, y las condiciones y eventos que provocan el cambio de modos y otros sucesos. Por lo tanto, proporciona respuestas a preguntas sobre causalidad, concurrencia y sincronización. Es descrito por el *Gráfico de Estados* y en este trabajo por la RdP no autónoma.

La *vista estructural* captura el "cómo": describe los subsistemas, módulos u objetos que constituyen el sistema real y la comunicación entre ellos. Es la arquitectura del framework y todos los objetos que constituyen a las acciones. Se considera que esta vista es el modelo físico del sistema, incluye términos y nociones obtenidos del dominio de los problemas. Este modelo conduce al dominio de las soluciones.

3 Arquitectura de la Solución

El framework desarrollado en este trabajo, presenta una arquitectura reutilizable para el desarrollo de sistemas reactivos. Por lo que provee las interfaces necesarias para

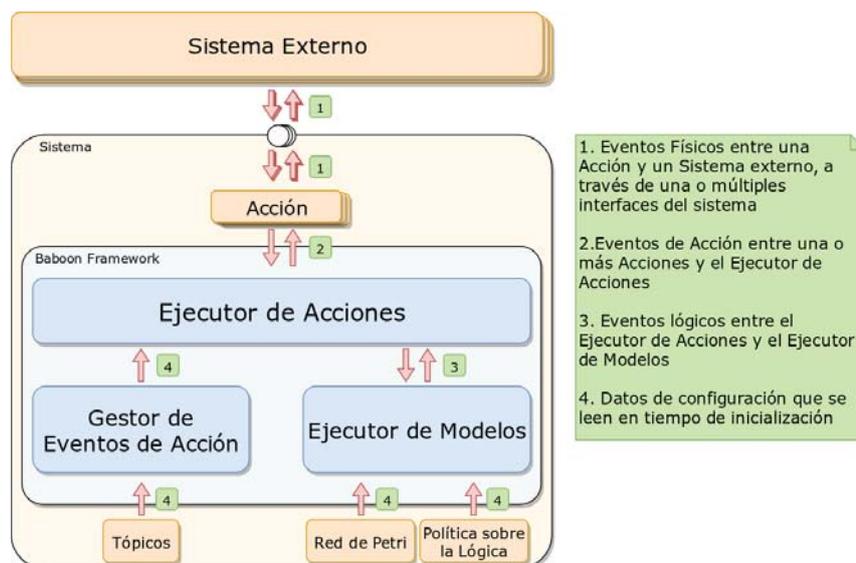


Fig. 1. Arquitectura de Baboon Framework (vista estructural)

interactuar con el modelo de comportamiento (RdP y política) y las acciones (vista funcional). En la Fig. 1 se observan los módulos principales de la arquitectura los que son descriptos a continuación, estos implementan las distintas vistas del modelo.

3.1 Ejecutor de Modelos

Las responsabilidades son: interpretar la RdP y ejecutarla según la política.

Interpretar la RdP. La responsabilidad de este submódulo, es transformar la descripción del modelo en un equivalente ejecutable. El formato aceptado por este submódulo es PNML [15], con él se generan las estructuras de datos para ejecutar la red mediante la ecuación de estado generalizada.

Ejecutar el Modelo. Si se interpreta la evolución de la RdP como una traza compuesta por una alternancia de eventos, transiciones y estados, y se asume que en cada estado se realiza una acción, y siendo que la RdP representa la lógica del sistema, estas acciones estarán coordinadas por el disparo de las transiciones habiéndose establecido una correspondencia entre los estados y las acciones.

Un conflicto emerge cuando dos o más hilos intentan hacer uso concurrente del mismo recurso para ejecutar acciones que, por la naturaleza del sistema, no pueden darse simultáneamente. Estos conflictos se resuelven aplicando una política sobre la lógica. La aplicación de la política determina la transición al estado siguiente.

Las responsabilidades de este submódulo son: computar el estado siguiente ante un evento a partir del estado actual, resolver los conflictos entre disparos y gestionar la concurrencia de acceso a la lógica y política.

Este framework está diseñado para soportar la programación de la política en tiempo de ejecución, por lo que gestiona la lógica y política de forma desacoplada.

3.2 Gestor de Eventos de Acción

Eventos. La ejecución del sistema resultante se realiza mediante el intercambio de eventos entre los distintos módulos. Estos eventos son de tres tipos:

Eventos Lógicos. Están inherentemente asociados a las evoluciones del modelo. Son los relacionados con las transiciones y las guardas de la RdP.

Eventos Físicos. Suceden en el mundo físico y representan sucesos del dominio del problema. Se comunican a través de las interfaces del software de usuario.

Eventos de Acción. Eventos de comunicación entre el framework y el software de usuario. Se corresponden al inicio o fin de una acción, desacoplan la vista funcional de la vista de comportamiento. El framework captura estos eventos para redirigir el flujo de ejecución del programa, logrando la inversión de control.

Tópicos. Un tópico es un conjunto ordenado de eventos lógicos necesarios para realizar una acción. Su responsabilidad es relacionar las acciones con los eventos lógicos, por lo cual se implementa con una estructura que contiene cuatro campos, que son: *name*, identificador unívoco; *permission*, una lista ordenada de transiciones para solicitar permiso de ejecución; *setGuardCallback*: una lista ordenada de conjuntos de guardas para gestionar la ejecución; *fireCallback*, una lista ordenada de transiciones para el aviso de finalización de ejecución.

En tiempo de inicialización, el Gestor de Eventos de Acción carga su configuración desde el archivo de tópicos provisto por el usuario.

Traducción de Eventos de Acción a Eventos Lógicos. Los eventos de acciones y los eventos lógicos deben vincularse y este vínculo puede cambiar a lo largo del desarrollo, implementación y mantenimiento del sistema. Por esto es necesario implementar una relación programable entre los mismos. Además se requiere soportar los casos donde la cardinalidad entre eventos de acciones y eventos lógicos es diferente. Para implementar esta relación en el framework se han incluido los *tópicos*. Estos le informan en tiempo de inicialización al ejecutor de modelos cómo hacerlo evolucionar ante el comienzo y el fin de cada acción, respetando el orden en que deben darse las evoluciones. Los tópicos constan de los siguientes campos: el campo *permission* es una lista de eventos lógicos para soportar el requerimiento de ejecutar una acción que requiere múltiples eventos. El campo, *fireCallback* es una lista de eventos lógicos opcionales que se desencadenan al finalizar la ejecución de una acción sólo si están las condiciones dadas para que sucedan (transiciones sensibilizadas). El campo, *SetGuardCallback* contempla el caso de las tareas complejas. Al ser una lista de listas de guardas, cada lista de guardas se establece en la finalización de la sub-tarea correspondiente dentro de la ejecución de una tarea compleja.

La decisión de introducir las guardas en los tópicos permite la toma de decisión dinámica de las acciones, esto relaciona la lógica con el estado de las entidades físicas, lo que mejora la capacidad de expresión del modelo.

3.3 Suscriptor de Acciones

Este módulo brinda al usuario las interfaces para construir una acción y suscribirla al tópico correspondiente. Esto se hace durante la fase de inicialización del sistema.

El *Ejecutor de Acciones* utiliza estas suscripciones para lanzar los eventos de acción, y ejecutar las secciones correspondientes del programa.

3.4 Ejecutor de Acciones

Este módulo se relaciona con las acciones y el ejecutor de modelos. Con las acciones a través de los eventos de acciones para coordinar su ejecución y con el ejecutor de modelos, a través de los eventos lógicos, para sincronizar con el modelo el comienzo y fin de las acciones.

Acción. Una acción es un componente de software ejecutable, desarrollado por el usuario del framework que realiza una actividad del dominio del problema. Estructuralmente está compuesta por un objeto y un método perteneciente a ese objeto. A su vez contiene todos los parámetros necesarios para realizar la llamada a este método.

El método de una acción se clasifica según dónde se origina su ejecución:

Suceso o Happening. su ejecución se origina por un evento físico que ingresa al sistema y la llama explícitamente. El ejecutor de acciones captura la llamada

(utilizando AOP) y consulta al modelo si debe permitir o no la ejecución. Debe estar anotada como *@HappeningController*

Tarea o Task. su ejecución se origina sin la necesidad de un evento físico, es decir internamente desde el ejecutor de acciones.[5]. La responsabilidad de permitir o rechazar la ejecución es del modelo y de su ejecutor. Debe estar anotada como *@TaskController*.

Tarea Compleja o ComplexSequentialTask. Si una tarea está compuesta por N tareas menores que naturalmente se realizan secuencialmente, existe la posibilidad de agruparlas en una Tarea Compleja. El framework provee las interfaces para crear una tarea compleja, agregandolas.

3.5 Proveedor de Guarda

Si el modelo construido con la RdP emplea guardas, debe existir en el programa un método por cada guarda con el fin de calcular el valor que el framework actualiza. Este método es el proveedor de la guarda y es anotado como *@GuardProvider("nombre_guarda")*. Retorna un valor booleano y es registrado en el tópic de la acción (esto se realiza por reflexión en tiempo de inicialización).

Se debe considerar que la guarda es usada por la RdP, por lo cual se abstrae al usuario de consultarla manualmente durante la ejecución del programa.

3.6 Ejecución de una Acción

Para cumplir las restricciones del sistema, al ejecutar una acción, se realizan la siguiente secuencia de pasos: el ejecutor de acciones captura el evento de acción, luego lo traduce a eventos lógicos de permiso mediante el tópic, seguidamente se emiten los eventos lógicos de permiso hacia el monitor para luego si el modelo permite la ejecución de la acción emitir un evento lógico de permiso concedido hacia el ejecutor. De otra manera queda la petición a la espera de las condiciones. A continuación el ejecutor recibe el evento de permiso concedido, se ejecuta la acción de usuario para realizar la actividad y finaliza la acción de usuario. Para concluir se emiten los eventos lógicos de finalización de ejecución al ejecutor del modelo: primero los de guardas y luego los de transiciones

4 Diseño de un Programa con Baboon Framework

Baboon Framework está diseñado para desarrollar sistemas con POO. Las principales características del diseño obtenidas con este framework son: los objetos no contienen lógica, sino que ésta está en la RdP; la concurrencia es transparente al usuario y es gestionada por el *Ejecutor de Modelos* y la interacción entre objetos se hace por el intercambio de eventos realizados por el ejecutor de acciones.

Los pasos del proceso de diseño de un programa con Baboon Framework son: primero se determinan los objetos y acciones que forman parte del sistema, luego se

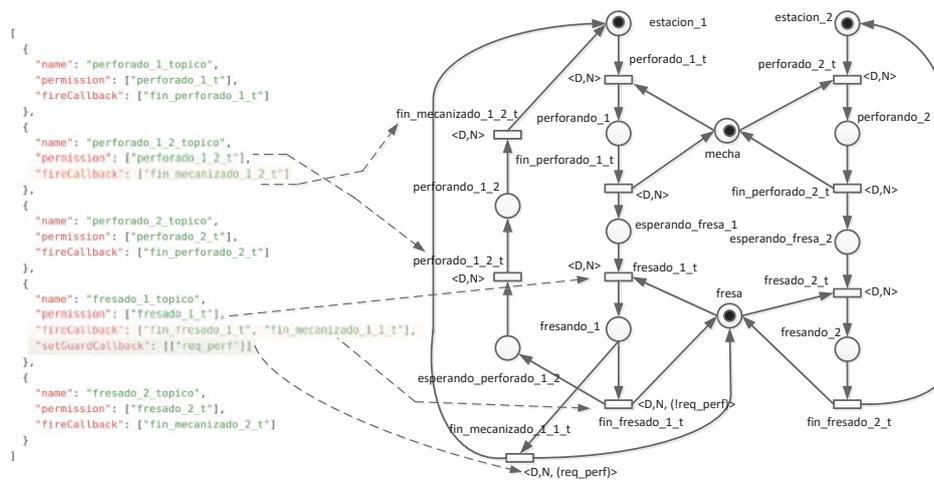
construye y valida la lógica del sistema con una RdP, a continuación se generan los tópicos y se etiquetan las acciones del sistema, en el paso siguiente se construyen los proveedores de guardas, seguidamente se define una política en las transiciones en conflicto, se escribe la inicialización del programa y finalmente se optimiza la cantidad de hilos haciendo uso de las tareas complejas.

5 Caso de Estudio

Como caso de estudio se presenta el diseño del software de un centro de mecanizado automatizado con las siguientes características: una máquina con dos estaciones de trabajo donde la primera estación puede realizar hasta dos procesos distintos. La segunda estación sólo hace un mecanizado simple.

El flujo de trabajo de la estación 1 decide en tiempo de ejecución cuál mecanizado realizar. Para esto emplea la guarda *req_perf* (requiere perforación), ambas estaciones comparten las herramientas (mecha y fresa) por lo que no las pueden utilizar simultáneamente. Si una herramienta está ocupada, la estación de trabajo que la requiera debe esperar hasta que dicha herramienta se encuentre disponible.

En la en la Fig. 2(a) se observan los tópicos del sistema, mientras que en la Fig. 2(b) se observa la RdP que modela al sistema.



(a) Tópicos del Caso de Estudio (b) RdP del Caso de Estudio
Fig. 2. Relación entre la RdP y los tópicos

En la Fig. 2(a) se observa, que para la definición de los tópicos se utiliza JSON, donde en la definición de cada artefacto se respeta su pertenencia, es decir, que para nombrar un tópico se usa la palabra tópico, para el método de interés se usa la palabra método, mientras que para los eventos se distingue su naturaleza con una F, A o L según sea un evento físico, de acción o lógico.

En la Fig. 2 se advierte la relación entre cada segmento de la RdP con el tópico y sus componentes correspondientes. Estas relaciones se muestran con las flechas de líneas de guiones.

5.1 Código del Usuario

En la Fig. 3 se observa una plantilla para la clase Fresadora. Esta clase contiene las acciones relacionadas al fresado de una pieza. En la Fig. 2 las flechas con guiones muestran cómo estas acciones están suscritas a un tópico, cuyos eventos están coordinados por la RdP.

```
public class Fresadora {
    // campos y otro métodos de la clase
    @TaskController
    public void fresado1Metodo() {
        // algoritmo de fresado sobre el mundo físico
        // emisión de eventos físicos con resultados en el mundo real
    }
    @GuardProvider("req_perf")
    public boolean requierePerforacion() {
        boolean resultado = false;
        // algoritmo para definir si se requiere hacer una perforación
        return resultado;
    }
}
```

Fig. 3. Clase de Acción

```
public class CentroDeMecanizado implements BaboonApplication {
    private Fresadora fresadora;
    @Override
    public void declare() { // tomar los recursos que requiera el sistema
        fresadora = new Fresadora();
        try {
            BaboonFramework.createPetriCore("/ruta/a/mi/red/de/petri.pnml",
                petriNetType.PLACE_TRANSITION, PoliticaPersonalizada.class);
            BaboonFramework.addTopicsFile("/ruta/a/mis/topicos.json");
        } catch (BadPolicyException | BadTopicsJsonFormat | NoTopicsJsonFileException e) {
            // manejar la excepción y terminar programa
        }
    }
    @Override
    public void subscribe() {
        try {
            // suscribir las las acciones a los tópicos usando BaboonFramework.subscribeControllerToTopic
            // crear tareas complejas usando BaboonFramework.createNewComplexTaskController
            // agregar tareas a una tarea compleja usando BaboonFramework.appendControllerToComplexTaskController
            BaboonFramework.subscribeControllerToTopic("fresado_1_topico", fresadora, "fresado1Metodo");
        } catch (NotSubscribableException e) {
            // manejar la excepción y terminar programa
        }
    }
}
```

Fig. 4. Clase de Inicialización

En la Fig. 4 se muestra un prototipo de la clase principal del programa, donde se le provee al framework el modelo de RdP, el archivo de tópicos y se realizan las suscripciones de acciones a tópicos.

6 Conclusiones

En Baboon, se extendió el trabajo desarrollado en [13] utilizando la ecuación de estado extendida de la RdP con lo cual se soportan todos los tipos de brazos. En consecuencia se obtiene mayor capacidad de expresión en la lógica de los sistemas a modelar frente a las POPN. Se desacopló la vista funcional de la vista de comportamiento, haciendo uso de los tópicos lo que simplifica la comprensión y mantenibilidad del sistema. Se ha mantenido el formalismo de la RdP no autónoma integrada en el framework, por lo tanto la lógica del sistema conserva la verificación formal realizada en el modelo. De acuerdo a los casos de estudio implementados son destacables las siguientes características alcanzadas con el framework: verificable, comprobable, mantenible, modular, basado en una arquitectura probada y además se logró reducir el tiempo de desarrollo, debido a que no se requiere codificar la lógica.

7 Resultados

Los casos implementados en [16, 17] fueron re-diseñados con el framework Baboon desarrollado en este trabajo, los que son: un sistema de control de acceso inteligente y un sistema marcapasos. Asimismo se implementó el sistema de celda flexible de producción aquí presentado.

Los resultados obtenidos han sido los siguientes: dado que el framework interpreta los modelos de RdP expresados en lenguaje PNML es directo implementar la lógica a partir del modelo realizado con un simulador. Esto facilitó el desarrollo y el mantenimiento de los sistemas.

Puesto que las propiedades de las RdP fueron verificadas formalmente con el simulador (interbloqueo, inanición, acotada, vivacidad, etc.), las acciones han sido desacopladas y no contienen lógica, y la política solo resuelve los conflictos sin modificar la lógica; éstas propiedades se han mantenido en el sistema resultante.

El uso del framework implementa la inversión de control: con lo que se obtuvo un flujo robusto y probado, y una arquitectura reutilizable.

El framework ha sido diseñado y construido considerando la relevancia de la simplicidad para su uso por lo que las interfaces de las APIs son consistentes, existe documentación online detallada y el proceso de construcción y gestión de dependencias están automatizados.

El código fuente es libre, licenciado bajo Apache 2.0 y disponible online en: https://github.com/airabinovich/java_petri_concurrency_monitor, <https://github.com/juanjoarce7456/baboon>.

Como resultado se destaca la escalabilidad de la lógica. Puesto que la lógica esta implementada a partir de la ecuación de estado extendida y ésta es una ecuación matricial (4 matrices de enteros de dimension: plazas * transiciones + 1 matriz de

floats con lógica temporal de dimensión: transiciones * 3). Para el caso de un sistema de 1000 plazas por 1000 transiciones se requieren 4 MB.

8 Trabajo Futuro

A continuación se mencionan las principales líneas de investigación sobre las que se continúa a partir de este trabajo: permitir cambios en la lógica en tiempo de ejecución, extender la ejecución a RdP coloreadas con el objetivo de explotar su capacidad de expresión manteniendo la escalabilidad del sistema resultante y re-implementar el proyecto utilizando otras arquitecturas como: web services, librería, driver, etc.

Bibliografía

1. Harel, D., Politi, M.: Modeling reactive systems with statecharts: the STATEMATE approach. McGraw-Hill, Inc. (1998)
2. Wieringa, R.J.: Design methods for reactive systems: Yourdon, statemate, and the UML. Elsevier (2003)
3. Wen, J.T., Mishra, S.: Intelligent Building Control Systems: A Survey of Modern Building Control and Sensing Strategies. Springer (2017)
4. Zhou, M., Wu, N.: System modeling and control with resource-oriented Petri nets. (2018)
5. Siewert, S.: Real time embedded components and systems. Cengage Learning (2016)
6. Niaz, I.A.: Automatic code generation from UML class and statechart diagrams (2005)
7. Niaz, I.A., Tanaka, J.: Mapping UML statecharts to java code. In: IASTED Conf. on Software Engineering, pp. 111-116 (2004)
8. David, R., Alla, H.: Discrete, continuous, and hybrid Petri nets, Springer Science (2010)
9. Micolini, O.: PhD thesis Arquitectura asimétrica multicore con procesador de Petri. vol. Doctor. Facultad de Informática, La Plata, Argentina (2015)
10. Micolini, O., Cebollada, M., Eschoyez, M., Ventre, L.O., Schild, M.: Ecuación de estado generalizada para redes de Petri no autónomas y con distintos tipos de arcos. In: XXII Congreso Argentino de Ciencias de la Computación (2016)
11. Peter A. Buhr, M.F.: Monitor Classification. ACM Computing Surveys 27, 63-107 (1995)
12. Micolini, O., Daniele, E.N., Ventre, L.O.: Modular Petri Net Processor for Embedded Systems. pp. 199-208. Springer International Publishing (2017)
13. Micolini, O., Caro, M.F., Furey, I., Cebollada, M.: Generación de Código de Sistemas Concurrentes a partir de Redes de Petri Orientadas a Procesos. In: XLIII Jornadas Argentinas de Informática e Investigación Operativa (43JAIIO)-XV Argentine Symposium on Technology (2014)
14. Universität Hamburg, <https://www.informatik.uni-hamburg.de/TGI/PetriNets/applications/>
15. PNML Org, <http://www.pnml.org/grammar.php>
16. Micolini, O., Ventre, L.O., Ludemann, M., Viano, J.I.R., Bien, C.C.: Case Study of Reactive and Embedded System Design Modeled with Petri Nets. In: 2018 IEEE International Conference on Automation/XXIII Congress of the Chilean Association of Automatic Control (ICA-ACCA), pp. 1-7. IEEE (2018)
17. Micolini, O., Ventre, L.O., Ludemann, M.: Methodology for design and development of Embedded and Reactive Systems Based on Petri Nets. In: 2018 IEEE Biennial Congress of Argentina (ARGENCON), pp. 1-7. IEEE (2018)

El modelo de Programación de Actor aplicado a Edge Computing utilizando Calvin

Nelson Rodríguez , María Murazzo, Tatiana Runco

Departamento de Informática, Facultad de Ciencias Exactas Físicas y Naturales,
Uniuersidad Nacional de San Juan, San Juan, Argentina

Resumen. Internet de las cosas (IoT) y todas las tecnologías asociadas han determinado que la computación distribuida alcance límites insospechables. A pesar de esta expansión, el desarrollo de aplicaciones en ecosistemas IoT no cumple con los requisitos específicos para estos sistemas. La pérdida de conectividad de los dispositivos, la necesidad de soportar la migración de código, la aparición de diversos errores y que los mismos no se propaguen, determinan que los desarrollos deben considerar estas dificultades. Teniendo en cuenta que el modelo de programación de actor presenta características que pueden ayudar a resolver estos problemas, se propuso utilizar este modelo para generar soluciones IoT, desarrollando aplicaciones con la entorno de aplicación Calvin. El presente trabajo analiza las ventajas de aplicar la programación de actores en ambientes IoT.

Keywords: Actor model programming, Edge Computing, Calvin

1 Introducción

Internet de las cosas (IoT) tiene un rico legado tecnológico y un brillante futuro: la conectividad ubicua ha creado un nuevo paradigma, y los sistemas cerrados, estáticos y limitados del pasado pronto serán obsoletos. Con la conexión de sensores de bajo costo, y la plataforma en el Cloud, ahora es posible realizar un seguimiento, analizar y responder a los datos operativos a gran escala [11].

Las aplicaciones de Internet de las cosas, llegan con un cambio de paradigma comparado con las aplicaciones de la Internet tradicional, debido a que ahora se generan enormes cantidades de datos en el borde de la red y fluye hacia la nube para ser procesados y potencialmente combinado con otros datos de diferentes fuentes [12].

Mientras que la primera generación de aplicaciones IoT a menudo seguían una arquitectura cliente-servidor tradicional, con clientes delgados y de baja complejidad, dejando que las tareas de alto costo computacional se realizaran en el Cloud, las aplicaciones recientes se caracterizan por clientes más ricos (teléfonos inteligentes, drones, automóviles) que utilizan datos de sensores locales como entrada para -posiblemente basado en aprendizaje automático- realizar el procesamiento de datos para respaldar la toma de decisiones a tiempo y local (por ejemplo para control

vehicular) o interacción del usuario, por ejemplo, para aplicaciones de realidad aumentada [7].

Se ha comenzado a comprobar que en la actualidad Cloud Computing está encontrando serias dificultades para satisfacer los requerimientos de IoT. Por ello, surgió como propuesta de solución llevar el almacenamiento, las funciones de red, gran parte del procesamiento hacia el borde de la red, lo que resultó en un nuevo modelo llamado edge computing [6].

En particular si se realiza en los dispositivos, sensores y actuadores se denomina mist computing, mientras que cuando se desarrolla en el extremo medio (fundamentalmente routers o switches de red) se denomina fog computing.

La programación para este tipo de sistemas se realiza utilizando lenguajes convencionales, los cuales presentan varias limitaciones para desarrollar aplicaciones para IoT. Los mismos no son de naturaleza asincrónica, no soportan aislación de errores y los mismos se pueden propagar (es común tener errores debido a descargas de baterías o falla de conexión por alcance, entre otras causas). Por lo tanto es conveniente utilizar otro tipo de lenguajes, herramientas o frameworks que permitan desarrollar aplicaciones eficientes para estos entornos.

El modelo de programación de actor presenta características que dan respuesta a esta problemática y la propuesta del presente trabajo es transmitir la experiencia en el desarrollo de aplicaciones para IoT basadas en este modelo.

El trabajo se encuentra organizado de la siguiente manera: se definen los conceptos más relevantes para el trabajo de IoT y del modelo de programación de actor, para posteriormente mostrar la experiencia en el desarrollo utilizando el entorno de aplicaciones Calvin y exponer las conclusiones y propuestas futuras pertinentes.

2 IoT

Internet de las cosas se puede definir como un conjunto de diferentes sistemas (por ejemplo, Smart Roads, Smart Buildings, Smart Grids) compuestos de componentes heterogéneos pero interactivos (por ejemplo, humanos, automóviles, teléfonos inteligentes, gateways, medidores inteligentes), tanto individual como colectivamente, que proporcionan servicios cibernéticos innovadores. Es decir, se puede describir como un ecosistema denso, a gran escala, abierto y dinámico de entidades y aplicaciones socio técnicas [22].

La idea básica de IoT es permitir una conexión e intercambio de datos autónomos y seguros entre dispositivos del mundo real y sus aplicaciones [10]. IoT vincula la vida real y las actividades físicas con el mundo virtual [16].

Hoy en día, IoT puede incluir productos industriales y comerciales, productos cotidianos como lavaplatos y termostatos, y redes locales de sensores para monitorear granjas y ciudades. En una solución de IoT, los objetos se pueden detectar y controlar a través de Internet, ya sea que se trate de dispositivos remotos, productos inteligentes o sensores que representan el estado de una ubicación física. Y la información puede estar disponible para aplicaciones, data warehouses y sistemas de negocios.

Sin embargo, a pesar de una década de investigación, el IoT aún se encuentra en una fase emergente: de hecho, está formado por unos pocos sistemas / dispositivos aislados de IoT que proporcionan servicios de computación convencionales diseñados principalmente para entornos estáticos con interacciones diseñadas a priori [5].

Los investigadores consideran que existe una tendencia clara y prominente hacia una IoT de mayor alcance, en la que posiblemente los impulsores clave serán los servicios ciberfísicos, altamente dinámicos y contextualizados.

Los bloques de construcción de la IoT son de baja potencia, integrados en dispositivos, donde las redes más grandes se construyen a partir de estos nodos pequeños y exhiben un flujo de trabajo altamente distribuido. Además de las restricciones del entorno de hardware, los desarrolladores tienen que tratar con un diseño de aplicación altamente distribuido [15].

Si bien las restricciones de hardware incluyen una baja potencia de procesamiento, paquetes de pequeño tamaño, una gran pérdida de paquetes y fallas temporales de conexión, los entornos distribuidos plantean desafíos con respecto a la escalabilidad y movilidad.

Por otro lado, los aspectos de seguridad deben considerarse muy seriamente, ya que estos dispositivos tienen acceso a una amplia gama de datos y la arquitectura de seguridad de este tipo de redes no es la más confiable. Cualquier persona que se encuentre cerca puede escuchar la transmisión inalámbrica y, por lo tanto, debe estar protegida.

Por ejemplo se podrían insertar con facilidad mensajes falsos, lo cual es una amenaza que puede ser contrarrestada si se implementa la autenticación de nodos de confianza. Un modelo de autenticación adecuado debe incluir la posibilidad de revocar la confianza de los nodos dañados o supuestamente atacados.

El uso generalizado de IoT crea nuevos desafíos, y su interacción sin precedentes con el mundo físico afecta la seguridad y la privacidad de las personas. El derecho a la privacidad debe estar protegido en el dispositivo, durante las comunicaciones y en el Cloud. Como regla general, la información debe transferirse y accederse según sea necesario y se debe cambiar, y los eventos como la creación de datos, el acceso a datos y los comandos de control deben registrarse de forma verificable [9].

Las aplicaciones pueden verse como una combinación de microservicios que se utilizan para crear un servicio. Estas aplicaciones pueden ubicarse estáticamente o migrarse dinámicamente al entorno que sea óptimo para su realización. La seguridad de las aplicaciones será el resultado del propio código de la aplicación y la plataforma que está utilizando. En los casos en que las aplicaciones pueden migrar, es importante que la migración entre plataformas se realice de forma segura [19]. En los sistemas de Cloud, las aplicaciones se pueden colocar de forma segura en plataformas confiables mediante el uso de información certificada que proviene de la confianza en la infraestructura del Cloud.

El futuro de IoT tiene el potencial ilimitado. Los avances en Internet industrial se acelerarán a través de una mayor agilidad de la red, la inteligencia artificial integrada y la capacidad de desplegar, automatizar, organizar y asegurar diversos casos de uso a nivel de hiperescala. El potencial no es solo habilitar miles de millones de dispositivos simultáneamente, sino también aprovechar los enormes volúmenes de datos procesables que pueden automatizar diversos procesos de negocios. A medida que las redes y las plataformas de IoT evolucionan para superar estos desafíos, a

través de una mayor capacidad e inteligencia artificial, los proveedores de servicios se adentran aún más en los mercados de TI y de escala web, lo que abre nuevos flujos de ingresos.

3 El Modelo de Programación de Actor

El modelo Actor es una teoría matemática de la computación que trata a los "Actores" como primitivas universales de la computación digital concurrente [18]. El modelo ha sido utilizado tanto como un framework para una comprensión teórica de la concurrencia, y como la base teórica para varias implementaciones prácticas de sistemas concurrentes.

A diferencia de los modelos de computación anteriores, el Modelo de actor se inspiró en las leyes físicas. También estuvo influenciado por los lenguajes de programación Lisp, Simula-67 y Smalltalk72, así como por fundamentos de redes de Petri, sistemas de capacidad y packet switching. El advenimiento de la concurrencia masiva a través de la computación en la nube y las arquitecturas de computadora multi núcleo ha estimulado el interés en el Modelo de actor.

Dicho modelo de programación fue definido por Carl Hewitt en 1973, y presenta ventajas considerables para el desarrollo de aplicaciones para IoT [18].

En IoT los dispositivos pueden dejar de tener conexión momentáneamente por diversas razones, por ejemplo la intensidad de la señal, razones climáticas, descarga de baterías (baterías solares) y posiblemente tiempo posterior vuelvan a conectarse, debido a la recarga de la batería solar o al alcance de la señal, si es un dispositivo móvil. Esta conexión y desconexión produce fallos que pueden propagar otros fallos en el resto de la red. En el modelo de programación de actor, cada actor se comunica de forma asíncrona con otro lo cual facilita la comunicación cuando hay conexión - desconexión de los dispositivos. Por otro lado, los actores presentan independencia en la ejecución unos de otros, lo cual en caso de falla no la propagaría al resto de la red. Además son muy livianos, por ejemplo en algunos toolkits como Akka ocupan solamente 600 Bytes, lo cual lo hace muy adecuado para entornos donde los dispositivos tienen poca capacidad de procesamiento y memoria

El modelo de programación de actor permite la concurrencia, está basado en memoria distribuida y puede ser aplicado al edge [14].

El modelo de actor [1] ofrece varias ventajas sobre el modelo de threads. Los actores modelan el mundo real de manera intuitiva, y escribir aplicaciones confiables utilizando actores es simple. La comunicación asíncrona entre actores conduce a una mejor utilización del tiempo de CPU, ya que los ciclos de inactividad no se desperdician mientras se espera una respuesta.

El modelo de programación de actor presenta mínimas diferencias con la programación reactiva.

Los objetos reactivos tienen las siguientes ventajas sobre los actores:

Los mensajes a métodos no definidos son simplemente puestos en cola.

El modelo de actor carece de mensajes síncronos.

La entrega asíncrona de mensajes no preserva el orden.

Sin embargo, la sincronización puede ser emulada por pares de mensajes asíncronos, y la mayoría de las implementaciones modernas presentan mejoras en estos los tres puntos

4 Sistemas de Actor Disponibles

Existen diversos lenguajes de programación, librerías y otras herramientas de software que permiten generar aplicaciones basadas en el modelo de programación de actor. Se van a describir las más emblemáticas.

Erlang [8] fue uno de los primeros lenguajes que soportan el modelo. Implementa actores a nivel de lenguaje y los llama procesos. Los procesos son ligeros, ya que se implementan a nivel de la máquina virtual Erlang, y no implican hilos o procesos de un sistema operativo. Cada actor se le asigna su propia memoria dinámica, y la recolección de basura se realiza para cada actor independientemente. Los mensajes enviados se copian entre montones, lo que impide compartir ellos por dos o más actores. La programación de actores es un deber de la máquina virtual Erlang, y no se necesita ningún mecanismo de sincronización, por ejemplo, semáforos. El Sistema de actores garantiza la escalabilidad (un actor necesita solo 300 bytes de memoria) y una alta confiabilidad de los programas escritos.

Es un lenguaje de programación funcional de alto nivel, que se ubica dentro del paradigma de Programación Declarativa, diseñado para escribir aplicaciones concurrentes y distribuidas de funcionamiento ininterrumpido.

Erlang usa procesos concurrentes para estructurar la aplicación. Estos procesos no comparten memoria y se comunican de forma asincrónica mediante el paso de mensajes. Los procesos Erlang son muy ligeros y pertenecen al propio lenguaje, no al sistema operativo. También posee un mecanismo para cambiar el código fuente de un programa o aplicación en ejecución (cambio en caliente de código), sin tener que detener el programa. Este mecanismo facilita la implementación de sistemas indetenibles [3,4].

Los inicios de este lenguaje toman lugar en los años 80 en los laboratorios de Ciencias de Computación de la Compañía de telefonía sueca Ericsson.

Akka [2] es un popular framework de actores al tope de la JVM escrito en Scala. Desde la versión 2.10.0, Scala usa Akka como la biblioteca de actores predeterminada [17].

El paso del mensaje es hecho en memoria compartida, si los actores se ejecutan dentro de una máquina virtual Java. Si los actores actúan en diferentes máquinas virtuales, los mensajes se serializan antes de enviarlos. La recolección de basura se aplica a todos los actores de una instancia de JVM determinada. Los actores están programados por la biblioteca Akka. No impone mensajes inmutables y evita el estado global, aunque los recomienda [24].

Al igual que Erlang, Akka adopta el modelo "Let it crash" para la resiliencia.

El cluster Akka proporciona un servicio de membresía de clúster que se basa en un protocolo Gossip, similar a Dynamo, el almacén de valores y claves distribuido de Amazon. El clúster Akka usa los latidos del corazón en un intervalo de 1 s y el Detector de fallas de acumulación de phi para calcular la probabilidad de que un nodo

sea inalcanzable. Los programadores pueden modificar la detección de fallas estableciendo un valor de umbral. Lightbend se ha ejecutado con éxito un clúster Akka con 2400 nodos en Google Compute Engine [23].

Calvin [20] está diseñado principalmente para simplificar el desarrollo de aplicaciones de Internet de las cosas y está disponible en Github <https://github.com/EricssonResearch/calvin-base>.

Fue creado por Ericsson basado en experiencia previa de Erlang, es un entorno de aplicación que permite a las cosas hablar con las cosas. Incluye un marco de desarrollo para programadores de aplicaciones y un entorno de ejecución para el manejo de la aplicación en ejecución.

Calvin se basa en la idea fundamental de que el desarrollo de aplicaciones debe ser simple y divertido. No debe haber barreras innecesarias entre una idea y su implementación. Un desarrollador de aplicaciones no debería tener que preocuparse por los protocolos de comunicación o detalles de hardware [21].

Hay que tener en cuenta, además que estos lenguajes basados en actores también son eficientes para servidores, Un resumen de las recomendaciones basadas en este pequeño conjunto de puntos de referencia es el siguiente: Para un servidor donde minimizar la latencia de mensajes es crucial Erlang es una buena opción. Por otro lado Akka reduce significativamente la latencia de comunicación en Scala y es capaz de mantener el mayor número de procesos inactivos, mientras que Erlang tiene el mejor desempeño cuando los procesos son de corta duración y el objetivo es garantizar un tiempo de generación mínimo, por ejemplo [23].

Se encuentran disponibles otros productos de menor relevancia o uso como: Pony, Concurrent ML, Anemone, Orleans, Orbit, SF Reliable Actors, entre otros.

5 Ventajas de Calvin en ecosistemas IoT

Calvin presenta diversas ventajas para el desarrollo y mantenimiento de aplicaciones IoT:

Tiene la capacidad de ejecutar partes del código de la aplicación donde sea más beneficioso por ejemplo, en un nodo central al analizar datos agregados de muchas fuentes o cerca de hardware específico para lograr baja latencia.

Calvin se basa en el modelo de actor bien establecido, utilizando una metodología a menudo denominada programación de flujo de datos.

Es escalable, la funcionalidad básica es bastante simple y funciona en dispositivos IoT pequeños, aun así utiliza la energía de cálculo completa disponible en el cloud.

El modelo de programación de Calvin no hace distinción entre el cloud y el dispositivo, smartphone y sensor, cliente y servidor, todos están representados por actores y comparten el mismo paradigma. Al escribir aplicaciones de Calvin, no hay necesidad de preocuparse por exactamente dónde desplegarlo mientras se cumplan los requisitos de hardware necesarios. El desarrollo y el despliegue son preocupaciones independientes.

Mover una aplicación a otro conjunto de nodos de cálculo no requiere cambiar el código. Por el contrario, los cambios en la aplicación no impiden la implementación en el mismo conjunto de nodos de cálculo.

Los actores de Calvin pueden incluso hacer la migración en vivo de un runtime (tiempo de ejecución) a otro. El sistema lo logra serializando el estado interno del actor en movimiento, enviándolo al tiempo de ejecución al destino en el que se deserializa y carga una copia del actor en movimiento. Esta propiedad hace posible escribir aplicaciones que se adaptan a las condiciones cambiantes. Por lo tanto, un runtime en un dispositivo que necesita entrar en modo de suspensión puede mover a los actores fuera de sí mismo sin interrumpir la aplicación de la que son parte de antes de entrar en modo suspensión.

Según especialistas de Ericsson, Calvin está pensado para todos los desarrolladores de Internet de las Cosas, debido al modelo de programación simple, la potente comunicación y a los mecanismos de despliegue que están incorporados en el sistema. Las complejidades están ocultas, el desarrollo se vuelve más eficiente, el resultado es más robusto y hay menos errores en el código.

Un modelo de programación específicamente diseñado para un entorno de ejecución distribuida es fundamental para hacer que la Internet de las cosas realmente vuele. Calvin es una tecnología muy prometedora que satisface estas necesidades.

6 Aplicaciones desarrolladas

Se desarrollaron varias aplicaciones para evaluar las ventajas de aplicar este modelo en entornos de IoT. En todos los casos se trabajó sobre la placa computadora "Raspberry Pi" conectada a sensores que recopilan los datos necesarios para la aplicación práctica a desarrollar. Esta plataforma cuenta con capacidad de procesamiento y todas las E/S necesarias, conexión Wi-Fi, Bluetooth y otras funcionalidades, además se le instaló el sistema operativo Raspbian.

La más relevante y de la cual se obtuvieron muchas conclusiones es una aplicación diseñada para mejorar la producción avícola en la incubación de huevos de aves, específicamente huevos de gallina. Debido al riesgo de las vidas de las aves en el momento del desarrollo embrionario, se presentó una propuesta de diseño para un sistema de incubación que busca cubrir aspectos principalmente de funcionalidad acordes al desarrollo embrionario del huevo de gallina, lo cual ayuda a disminuir los costos de producción antes mencionados.

Una incubadora avícola tiene como principal objetivo mantener las condiciones adecuadas para el nacimiento del ave que se pretenda incubar, basándose en un análisis fenomenológico que ocurre dentro del periodo de incubación. Para ello debe realizarse antes una revisión de los aspectos más relevantes que determinan la incubabilidad de los huevos. Como son los aspectos de temperatura y humedad relativa de incubación que competen a los fines prácticos del software a desarrollar.

Dicha aplicación se desarrolló en Calvin y permitió comprobar las ventajas del modelo para entornos de IoT como son:

- Evita el bloqueo de la programación orientada a objetos con el paso de mensajes asíncronos pero sigue conservando el principio de encapsulamiento.

- Como ya no existe una pila de llamadas compartidas entre actores que se envían mensajes entre sí, las situaciones de error se manejan por paso de mensajes y el servicio de supervisión.
- La migración de Actores se puede lograr en tiempo de ejecución entre diferentes aplicaciones.
- Facilita considerablemente el trabajo del programador a la hora de desarrollar aplicaciones gracias a la simplicidad del mismo lenguaje y gracias a que el mismo lenguaje se encarga de todo el proceso de comunicación.

7 Conclusiones

Calvin se pudo instalar sin dificultad en cada dispositivo que fue propuesto, facilita la flexibilidad y adaptabilidad al fusionar el Edge en IoT y el Cloud. La programación se distribuye sin complicaciones, combina la simplicidad y la reutilización de la funcionalidad. Por otro lado sobre Raspberry y Raspbian como sistema operativo funcionó adecuadamente y con muy baja latencia.

Implementa todos los principios y características del Modelo de Programación de Actor mencionados anteriormente, permitiendo la creación de aplicaciones que ofrecen ventajas significativas sobre el entorno de IoT: migración en caliente de Actores, sistemas más escalables, distribuidos y simples de desarrollar para los programadores (abstrayéndolos de protocolos de comunicación o detalles de hardware), las cuales no soportan los modelos de programación convencionales.

La única desventaja es que Calvin es muy sensible en sus archivos de configuración. Es decir, cada cambio que se desee realizar difiriendo de los que ya se encuentran por defecto, implica la modificación de los archivos de configuración del mismo y debiéndose considerar que tal archivo de configuración se encuentre en la ruta donde se ejecute el tiempo de ejecución que haga uso de la configuración deseada. Por ejemplo: para lograr el correcto funcionamiento de los sensores, actores y de la Raspberry Pi.

El desarrollo de aplicaciones con el modelo de actor debe ser planificado adecuadamente desde etapas iniciales, dado que si bien se dispone de herramientas de software para la construcción de las mismas que soportan esta forma de programación, no es suficiente, es decir se puede programar en Akka o Calvin y no necesariamente el resultado tiene toda la funcionalidad y las ventajas de un desarrollo basado en el modelo de actor.

Debido a las características de Calvin, se puede construir un Cloud distribuido, con importante funcionalidad en el edge, mediante esta arquitectura se puede ayudar a lograr una baja latencia mediante el uso de nodos de procesamiento paralelos.

Existen otros frameworks desarrollados para IoT, sin embargo, Calvin simplifica el trabajo para el desarrollador de la aplicación. El tiempo de ejecución oculta gran parte del trabajo complejo y el desarrollador de la aplicación no necesita preocuparse por cosas como la migración de actores o la transmisión de mensajes.

La primera generación de IoT se trató de la transición desde M2M, sin embargo actualmente IoT es una amalgama de diferentes disciplinas, como los sistemas cibernéticos, redes de sensores inalámbricos, M2M y Big Data / Inteligencia Artificial, en un contexto singular de una Internet del mundo real. A lo largo de los años, el ecosistema ha crecido de manera exponencial, y hoy, naturalmente, incluye no solo a las empresas en el dominio de TI, sino también a OT - Tecnología operacional: el nombre utilizado para las tecnologías asociadas con el monitoreo y control de dispositivos físicos y procesos en entornos industriales y de fabricación. Cualquier jugador importante de TI u OT en el mercado ahora está en desventaja si no tiene productos o servicios de soporte basados en IoT.

Una estrategia clave en la arquitectura futura, es el papel que jugará el extremo (Edge), dado que fog computing cada vez adquiere mayor relevancia y seguramente será parte del soporte para IoT. El rol del extremo es muy importante para reducir la latencia de ir al Cloud, que en casos de soluciones dependiente del tiempo puede resultar muy ventajosa, y en este caso se puede visualizar las ventajas de aplicar la programación de actor en estos contextos.

8 Trabajos futuros

Internet de las cosas (IoT) se está expandiendo rápidamente y según algunas estimaciones se espera que comprenda 18 mil millones de dispositivos conectados para 2022.

Las suposiciones de confianza y de buenas intenciones por parte de los usuarios, que formaron parte del desarrollo temprano de Internet, no se pueden aplicar en las primeras etapas del desarrollo de IoT. La privacidad y las preocupaciones de seguridad son cada vez mayores, especialmente dada la creciente importancia de IoT en los contextos corporativo, gubernamental y de infraestructura crítica. Del mismo modo, la mercantilización de los componentes de IoT incorporados en diversos rangos de productos y desplegados tanto en casos de uso administrados como no administrados presenta importantes desafíos de seguridad y crea potencial para nuevos tipos de ataques. La cooperación proactiva de todas las partes interesadas clave será necesaria para obtener los considerables beneficios económicos del IoT, al tiempo que protege la seguridad, la seguridad y la privacidad.

Se espera realizar estudios tendientes a valorar la adopción del modelo de actor para proveer software más seguro y administrar de forma más eficiente la seguridad provista.

Referencias

1. Agha G. A. :ACTORS - a model of concurrent computation in distributed systems. MIT Press series in artificial intelligence. Cambridge (1990).
2. Akka : Build highly concurrent, distributed, and resilient message-driven applications on the jvm. <https://github.com/akka/akka>, 2018.

3. Armstrong J. : Erlang- software for a concurrent world. In: Ernst E (ed.) ECOOP 2007. Object-Oriented Programming, 21st European Conference, Berlin, Germany, Proceedings, vol. 4609, p 1 (2007).
4. Armstrong J. : Erlang. Commun ACM Vol. 53 N. 9, pp. :68–75 (2010).
5. Biron J., Follett J.: Foundational Elements of an IoT Solution The Edge, The Cloud, and Application Development, O'Reilly Media, Inc. (2016).
6. Bonomi F. et al.: Fog Computing and Its Role in the Internet of Things. In: Proc. 1st Edition MCC Workshop Mobile Cloud Computing (MCC12), pp.13–15. (2012).
7. Chen K., J. Furst, Kolb J., Kim H., Jin X., Culler D., Katz R.: Snaplink: Fast and accurate vision-based appliance control in large commercial buildings. In: Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies, vol. 1, no. 4, p. 129 (2018).
8. Erlang: Sitio oficial: <https://www.erlang.org/>
9. Ericsson: IoT security - protecting the networked society. Jun 1, 2017 | White paper, <https://www.ericsson.com/en/white-papers/iot-security-protecting-the-networked-society>
10. Fan T., Chen Y.: A Scheme of Data Management in the Internet of Things. In: 2nd IEEE International Conference on Network Infrastructure and Digital Content. (2010).
11. Fortino G., Russo W., Savaglio C., Viroli M., Zhou M.: Opportunistic cyberphysical services: A novel paradigm for the future Internet of Things, in: 2018 IEEE 4th World Forum on Internet of Things (WF-IoT), 2018, pp. 488–492, <http://dx.doi.org/10.1109/WF-IoT.2018.8355174>. (2018).
12. Furst J., Fadel Argerich M., Chen K., Kovacs E.: Towards Adaptive Actors for Scalable IoT Applications at the Edge”, Open Journal of Internet of Things (OJIOT), Volume 4, Issue 1, (2018)
13. Hewitt C. : Actor Model of Computation: Scalable Robust Information Systems, Cornell University, <https://arxiv.org/abs/1008.1459> (2015).
14. Hewitt C., Bishop P. and Steiger R. : A Universal Modular ACTOR Formalism for Artificial Intelligence. In: Proceedings of the 3rd international joint conference on Artificial (1973).
15. Hiesgen R., Charousset D., Schmidt T. C. : Embedded Actors – Towards Distributed Programming in the IoT. 978-1-4799-6165-8/14/ IEEE. (2014).
16. Huang Y., Li G.: A Semantic Analysis for Internet of Things. In: International Conference on Intelligent Computation Technology and Automation (ICICTA), (2010).
17. Jovanovic V., Haller P. : The scala actors migration guide. <http://docs.scala-lang.org/overviews/core/actors-migration-guide.html>.
18. Patil S., Sruthi N. S. , Narkhede, B. E. , Pendam D. V. : A 20/20 vision of Internet of things. IOSR Journal of Business and Management. 18. 76-80. 10.9790/487X-1808027680 (2016).
19. Persson et. al. : Calvin – Merging Cloud and IoT, <http://www.sciencedirect.com/science/article/pii/S1877050915008595>
20. Persson J.: Open Source release of IoT app environment Calvin. <https://www.ericsson.com/research-blog/open-source-calvin/> (2015).
21. Persson P., Angelsmark, O.: Calvin–merging Cloud and IoT. In: Procedia Computer Science, 52, pp. 210-217 (2015).
22. Savaglio C., Fortino G., Zhou M.: Towards interoperable, cognitive and autonomic IoT systems: an agent-based approach. In: Internet of Things (WF-IoT), 2016 IEEE 3rd World Forum on, IEEE, pp. 58–63. (2016).
23. Valkov I., Chechina N., Trinder P.: Comparing Languages for Engineering Server Software: Erlang, Go, and Scala with Akka, SAC 2018: Symposium on Applied Computing. Pau, France (2018).
24. Wyatt D. : Akka concurrency. Artima Incorporation, New York (2013).

Elicitación de Requerimientos para la Construcción de Modelos Predictivos basados en Sistemas Inteligentes dentro del Ámbito Educativo

Cinthia Vegega^{1,2}, Pablo Pytel^{1,2} & María Florencia Pollo-Cattaneo^{1,2}

¹ Grupo de Estudio en Metodologías de Ingeniería en Software (GEMIS). Universidad Tecnológica Nacional. Facultad Regional Buenos Aires. Argentina.

² Programa de Maestría en Ingeniería en Sistemas de Información. Facultad Regional Buenos Aires. Universidad Tecnológica Nacional. Argentina

cinthiavg@yahoo.com.ar; pppytel@gmail.com, flo.pollo@gmail.com

Resumen. La toma de decisiones es un proceso esencial en la vida de las organizaciones siendo particularmente importante para los roles gerenciales que deben tomar decisiones sobre cómo asignar sus recursos. Estas decisiones deben basarse en predicciones sobre el tiempo, el esfuerzo y/o los riesgos de llevar a cabo sus actividades. Actualmente esta situación se encuentra acentuada por el entorno complejo que rodea a las organizaciones lo cual provoca que deban actuar más allá de los sistemas tradicionales de gestión incorporando nuevos mecanismos, como los provistos por la Inteligencia Artificial, por lo que se estaría desarrollando un Modelo Predictivo Inteligente. En este contexto, el presente trabajo tiene como objetivo proponer la aplicación de un proceso que asista al Ingeniero en Sistemas de Información en la difícil tarea de recolectar, entender, identificar y documentar la información necesaria para implementar un Modelo Predictivo basado en Sistemas Inteligentes.

Palabras Claves: Sistema Inteligente. Modelo Predictivo. Educación y Tecnología. Toma de Decisiones. Organización.

1. Introducción

La toma de decisiones implica cuestionarse la validez de las alternativas, sus ponderaciones, los criterios de selección a utilizar y su impacto a futuro [1]. Por esta razón, contar con mecanismos que permitan realizar predicciones acertadas es algo que la Humanidad siempre ha querido. Dicha necesidad no es propia únicamente de los individuos y también impacta en las organizaciones. La toma de decisiones es un proceso esencial en la vida de las organizaciones. Aunque cada miembro dentro de una organización toma decisiones, este proceso es particularmente importante para los roles gerenciales. Es por esto que a los gerentes se los conoce como “tomadores de decisiones” en sus actividades de planear, organizar, dirigir y controlar [2]. Diariamente deben decidir cómo asignar sus valiosos recursos basados en predicciones [3] sobre el tiempo, el esfuerzo y/o los riesgos que implica llevar a cabo sus actividades. Esta situación se encuentra acentuada debido al entorno altamente complejo y de difícil predicción del siglo XXI [4], lo cual genera que las organizaciones deban actuar más allá de los sistemas tradicionales de gestión e incorporar nuevos mecanismos para la “creación y potenciación del conocimiento organizativo” tales como los provistos por la Inteligencia Artificial [5; 6].

Un caso de estos nuevos mecanismos se puede encontrar en el ámbito de los Mo-

delos Predictivos. A pesar de que tradicionalmente se han aplicado Técnicas Estadísticas y Modelos Paramétricos para generar predicciones [7], en las dos últimas décadas se han incorporado diversos métodos asociados al Aprendizaje Automático [8; 3]. De esta manera, es posible construir modelos que permitan encontrar la relación entre situaciones pasadas y futuras usando los datos históricos disponibles. En este sentido, se pueden destacar las Redes Neuronales Artificiales [9; 10] y las Redes Bayesianas [11; 12] como las principales arquitecturas de Sistemas Inteligentes a ser utilizadas para este tipo de problema [13; 14; 15]. Estos Modelos Predictivos basados en Sistemas Inteligentes presentan características muy útiles, tales como son la capacidad de generalización, robustez y auto-organización [16].

No obstante, desafortunadamente los Modelos Predictivos suelen ser imprecisos [3] o, en algunos casos, se equivocan generando a menudo respuestas que son incomprensibles [17]. En este sentido, la calidad de la información requerida es sumamente importante para poder tomar decisiones bajo certeza [18]. Al reducir la ignorancia sobre el problema y su contexto es posible generar mejores predicciones. Sin embargo, contar con información completa, cierta y precisa para predecir con total certeza es casi imposible. Siempre existe un riesgo asociado en confiar en la información disponible para evaluar la situación por lo que una predicción deberá tener asociado cierto grado de probabilidad [19]. Dichas probabilidades se encuentran afectadas por lo que se conoce del problema, como por lo que no. Por consiguiente, además de recolectar los datos históricos que se aplicarán para construir el Modelo Predictivo, también es imprescindible identificar las características generales del dominio en donde se lleva a cabo la predicción, y así poder detectar situaciones o eventos de los que no se posee datos pero que el modelo deberá considerar.

En este contexto, el presente trabajo aplica un proceso propuesto para asistir al Ingeniero en Sistemas de Información en la difícil tarea de recolectar, entender, identificar y documentar la información necesaria para implementar un Modelo Predictivo basado en Sistemas Inteligentes. Para ello, en la sección 2 se presenta un breve resumen del proceso propuesto y en la sección 3 se indican los resultados de su aplicación en un caso dentro del contexto de una asignatura de grado. Finalmente, en la sección 4 se describen las conclusiones.

2. Proceso Propuesto de Elicitación de Requerimientos

El proceso propuesto busca ser una guía para los Ingenieros en Sistemas de Información involucrados en la implementación de Modelos Predictivos Inteligentes, que brinda soporte durante las etapas iniciales del proyecto teniendo presente las particularidades del mismo. Este proceso se limita a contemplar las características de dos tipos de Sistemas Inteligentes aplicables para la implementación de Modelos Predictivos: las Redes Neuronales Artificiales (RNA) de tipo Multi-Perceptrón con aprendizaje por retropropagación del error hacia atrás (también conocidas como RNA Multi-Perceptrón Backpropagation) y, las Redes Bayesianas (RB). Por lo tanto, como resultado de esta propuesta se determinan los objetivos, criterio de éxito, restricciones y suposiciones del proyecto lo que permite identificar los datos disponibles que serán necesarios utilizar para el entrenamiento del Sistema Inteligente y generar una especificación inicial del mismo. A partir de estos resultados, el equipo de desarrollo podrá comenzar a trabajar en la construcción, entrenamiento y validación del Modelo Predi-

cativo Inteligente que cumpla las expectativas de la organización. El proceso propuesto se estructura en las siguientes cinco fases:

- 1. Fase de Definición del Proyecto:** tiene como objetivo, definir los interesados que colaboran en el proyecto y el alcance del mismo a partir de los objetivos que se desean alcanzar.
- 2. Fase de Educción de Procesos de Negocio:** su objetivo es identificar y relevar los procesos de negocio que son significativos para el proyecto, así como las tareas del experto en el caso de construir un modelo que emule sus capacidades de predicción.
- 3. Fase de Educción de Datos de Procesos de Negocio:** busca identificar los repositorios de datos donde se almacena la información de los diferentes procesos de negocio y relevar las características de dichos repositorios.
- 4. Fase de Conceptualización de los Datos del Negocio:** su objetivo es identificar y evaluar la representatividad de los datos disponibles en el negocio para la construcción del modelo predictivo inteligente.
- 5. Fase de Especificación Inicial del Sistema Inteligente:** a partir de la información obtenida en las fases anteriores, se determina el tipo de arquitectura más apropiada para implementar el modelo predictivo, así como, una propuesta de la topología inicial del mismo.

Cada fase del proceso tiene definido un conjunto de actividades que se aplican a un caso de estudio dentro del contexto de una asignatura de grado en la sección 3 (la explicación detallada de cada actividad del proceso se encuentra disponible en [20]).

3. Aplicación del Proceso Propuesto en un Caso de Estudio

En esta sección se presenta la aplicación de las fases del proceso propuesto en un caso que pertenece al dominio de una universidad. En primera instancia, en la sección 3.1 se describe el contexto del caso de estudio, para luego describir la aplicación de cada fase del proceso junto con las actividades que se llevan a cabo en cada una de ellas. Así, la primera fase se describe en la sección 3.2, la segunda en la sección 3.3, la tercera fase en 3.4, la cuarta fase en 3.5 y la quinta fase en la sección 3.6.

3.1. Contexto del Caso de Estudio

Este caso de estudio se desarrolla dentro del contexto de la Facultad Regional Buenos Aires (FRBA) perteneciente a la Universidad Tecnológica Nacional (UTN) de la República Argentina. Específicamente, se realiza en la asignatura “Sistemas y Organizaciones” [21], perteneciente al primer año de la carrera de “Ingeniería en Sistemas de Información”. La asignatura analizada es anual y de cursado obligatorio para los alumnos que hayan aprobado el curso de ingreso a la carrera (con aproximadamente 800 inscriptos) y se corresponde al tronco integrador del Plan de Estudios. En este contexto, se busca implementar un Sistema Inteligente que permita predecir el desempeño de los alumnos a lo largo de la cursada de dicha materia. Debido a que cualquier error en las predicciones puede llevar a que los docentes, o alumnos, tomen decisiones equivocadas, se considera de gran importancia que el sistema presente resultados consistentes teniendo en cuenta el comportamiento normal de los alumnos de la asignatura.

3.2. Aplicación de la Primera Fase de Definición del Proyecto

Las siguientes actividades son descriptas: “Identificar los Objetivos del Proyecto”

(sección 3.2.1), “Identificar los Interesados del Proyecto” (sección 3.2.2), y “Identificar el Alcance del Proyecto” (sección 3.2.3).

3.2.1. Actividad “Identificar los Objetivos del Proyecto”

En esta actividad se lleva a cabo la primera reunión conceptual del proyecto con la jefa de cátedra de la asignatura “Sistemas y Organizaciones”, que es la principal patrocinadora del proyecto. De esta manera, se busca entender en forma general el objetivo del proyecto junto con las expectativas en relación al mismo. Asimismo, se identifican los profesores de la cátedra y los ayudantes que conformarán el conjunto de interesados del proyecto, con los cuales luego se realiza la reunión de inicio. A partir del relevamiento realizado se analiza la información obtenida e identifican los principales objetivos del proyecto, que se documentan en el formulario de objetivos del proyecto, tal como se muestra en la figura 1.

3.2.2. Actividad “Identificar los Interesados del Proyecto”

En esta actividad, el Analista Funcional, a partir de la información relevada de la organización, identifica a los participantes del proyecto y confecciona el formulario que se muestra en la figura 2.

3.2.3. Actividad “Identificar el Alcance del Proyecto”

A partir de la información recolectada, el Analista Funcional define los criterios de éxito del proyecto tal como se muestra en la figura 3 y releva los problemas a resolver a fin de establecer qué es lo que se debe incluir como resultado del proyecto. Con esta información, se construye el formulario de definición de alcance del proyecto, tal como se muestra en la figura 4. Dicho formulario debe ser validado por el patrocinador e interesados del negocio. Asimismo, identifica cuáles son los supuestos que el proyecto considera como base para la ejecución del mismo, y cuáles son las dependencias de otros proyectos o información que se consideran necesarios que estén disponibles para poder comenzar a trabajar en el proyecto. Esto se incluye en el formulario de suposiciones del proyecto, tal como se muestra en la figura 5. Por último, se definen las restricciones de información del proyecto y se confecciona el formulario de restricciones del proyecto, tal como se muestra en la figura 6.

OBJETIVOS DEL PROYECTO		
ID	Descripción del Objetivo	Prioridad
OBJ1	Implementar un Sistema Inteligente que permita predecir el desempeño de los alumnos a lo largo de la cursada de la materia “Sistemas y Organizaciones” dictada en UTN FRBA.	Alta

Figura 1. Formulario de Objetivos del Proyecto.

INTERESADOS DEL PROYECTO			
Posición	Org/Sector	Rol en el Proyecto	Áreas de Conocimiento
Jefa de Cátedra	UTN- FRBA / Asignatura	Patrocinador	Generalidades, Teoría y Práctica de la Asignatura, y Criterios de Aprobación
Docentes		Interesado	
Ayudantes			Teoría y Práctica de la Asignatura y Repositorios de Datos

Figura 2. Formulario de Interesados del Proyecto.

CRITERIOS DE ÉXITO DEL PROYECTO		
ID	Descripción Criterio	Objetivo
CE1	Predecir a partir de los datos del primer parcial y primer recuperatorio, el desempeño del alumno en el segundo cuatrimestre (incluyendo parcial y recuperatorios).	OBJ1

Figura 3. Formulario de Criterios de Éxito del Proyecto.

DEFINICIÓN DE ALCANCE DEL PROYECTO		
ID	Problemas a Resolver	Objetivo
P1	Identificar las fortalezas y debilidades de los alumnos, de forma de reforzar lo que haga falta durante la cursada.	OBJ1
Problemas Excluidos del Proyecto		
Queda excluido la nota conceptual y la evaluación clase a clase, solo se tomarán en cuenta para el análisis, las notas de parciales y recuperatorios. Asimismo, la planificación anual de la asignatura no es tenida en cuenta para el análisis que se realiza, tomándose solamente como referencia de los temas dictados.		

Figura 4. Formulario de Definición de Alcance de Proyecto.

SUPOSICIONES DEL PROYECTO		
ID	Descripción Suposición	Objetivo
S1	Se tendrá acceso a los ayudantes y docentes sin restricciones, pudiendo realizarse consultas por mail o en forma personal.	OBJ1
S2	Se consideran que los datos son correctos y están completos, teniendo la misma estructura dado que corresponde al mismo docente.	OBJ1

Figura 5. Formulario de Suposiciones del Proyecto.

RESTRICCIONES DEL PROYECTO			
ID	Tipo	Descripción	Objetivo
R1	Datos	No existen datos sobre el progreso del alumno clase a clase.	OBJ1
R2	Datos	No se pueden utilizar las notas de los trabajos prácticos dado que no se encuentran normalizados de la misma forma en todos los cursos.	OBJ1
R3	Datos	No se pueden utilizar los nombres, apellidos y legajos de los alumnos por ser considerados confidenciales.	OBJ1

Figura 6. Formulario de Restricciones del Proyecto.

3.3. Aplicación de la Segunda Fase de Educación de Procesos de Negocio

Las siguientes actividades son descriptas: “Identificar Procesos de Negocio” (sección 3.3.1) y “Relevar Procesos de Negocio” (sección 3.3.2). Asimismo, dado que se desea implementar un Modelo Predictivo en base a los conocimientos de expertos disponibles en la organización, se llevan a cabo las tareas que corresponden a la tercera actividad “Relevar Tareas del Experto” (sección 3.3.3).

3.3.1. Actividad “Identificar Procesos de Negocio”

A partir de las actas de las reuniones realizadas con los interesados del proyecto, del formulario de Objetivos del Proyecto (figura 1), del formulario de Criterios de Éxito del Proyecto (figura 3) y del formulario de Definición del Alcance del Proyecto (figura 4), el Analista Funcional define la actividad del negocio más significativa que se denomina “*Evaluar Exámenes de Alumnos*”. En esta actividad participan los docentes y ayudantes y, tiene por objetivo la corrección de los exámenes de los alumnos para luego pasar los datos a los repositorios destinados para este fin.

3.3.2. Actividad “Relevar Procesos de Negocio”

Teniendo en cuenta la información relevada, asociada al proceso del negocio identificado, el Analista Funcional realiza una nueva reunión con la Jefa de Cátedra y los interesados de la asignatura. De esta manera, se recolecta la información para registrar cómo funciona dicho proceso y cómo es su relación con los repositorios de datos. A partir de lo relevado, se formaliza la información obtenida documentándola en el formulario de procesos de negocio, tal como se muestra en la figura 7.

3.3.3. Actividad “Revelar Tareas del Experto”

A partir de las reuniones realizadas, se detecta que la tarea de determinar si el alumno aprueba, o no, la cursada teniendo en cuentas sus características y el resultado de sus

exámenes depende de muchos conocimientos que se encuentran interiorizados en la mente de los docentes. Esto significa que no existen reglas fijas ni procedimientos estándar para llevarlas a cabo, por lo que no es posible identificar un Proceso de Negocio. Por lo tanto, se trata de una tarea de expertos y se decide llevar a cabo su relevamiento con el fin de obtener los conocimientos aplicados por la Jefa de Cátedra de la asignatura. Para ello, se selecciona como técnica para educir los conocimientos al Análisis de Protocolos, realizando las etapas correspondientes a la técnica según [22]. Una vez efectuados los pasos a partir del protocolo, el Analista Funcional formalizada los conocimientos obtenidos en el formulario tareas del experto, tal como se muestra en la figura 8. Nótese que debido a su tamaño, el Árbol de Descomposición Funcional (Conocimientos Estratégicos) se encuentra disponible en <https://tinyurl.com/yxjwkrj4> y el Mapa de Conocimientos en <https://tinyurl.com/yxbyou6h>

PROCESOS DE NEGOCIO	
P0001 - Evaluar Exámenes de Alumnos	
Actores	Docente; Ayudante
Pre-Condiciones	Los alumnos resolvieron los exámenes a evaluar y ya fueron repartidos entre los docentes y ayudantes para su corrección.
Post-Condiciones	Los exámenes han sido corregidos y las notas pasadas a la planilla del curso.
Flujo Normal	
<ol style="list-style-type: none"> 1. El ayudante corrige la parte práctica de los parciales. 2. El docente corrige la parte teórica de los parciales. 3. El ayudante se reúne con el docente a fin de consultar las dudas que surgieron de la corrección. 4. El docente define la nota final del parcial. 5. El ayudante se encarga de pasar las notas a la planilla correspondiente. 	
Flujo Alternativo	
<ol style="list-style-type: none"> 1. En el caso de que algún examen se haya corregido en forma incorrecta luego de que el alumno lo analice, el docente y ayudante se vuelven a reunir a fin de corregirlo nuevamente y definir la nota final. 2. Una vez decidida la nota, el ayudante se encarga de pasarla a la planilla correspondiente. 	

Figura 7. Formulario de Procesos de Negocio.

TAREAS DEL EXPERTO		
TE001 – Evaluar Cursada del Alumno		
Conocimientos Fáclicos – Tabla Concepto – Característica – Valor		
Conceptos	Características	Valores
<i>Alumno</i>	<i>Legajo Alumno</i>	<i>Alfanumérico</i>
	<i>Apellido Alumno</i>	<i>Alfanumérico</i>
	<i>Nombre Alumno</i>	<i>Alfanumérico</i>
<i>Evaluación</i>	<i>Tipo Evaluación</i>	<i>Alfanumérico</i>
	<i>Nota Teoría</i>	<i>Alfanumérico</i>
	<i>Nota Práctica</i>	<i>Alfanumérico</i>
Conocimientos Tácticos – Pseudo-Reglas		
<ul style="list-style-type: none"> • Si el alumno aprueba la teoría del primer parcial o la teoría del primer recuperatorio del primer parcial o la teoría del segundo recuperatorio del primer parcial entonces el alumno aprueba la teoría del primer cuatrimestre. • Si el alumno aprueba la práctica del primer parcial o la práctica del primer recuperatorio del primer parcial o la práctica del segundo recuperatorio del primer parcial entonces el alumno aprueba la práctica del primer cuatrimestre. • Si el alumno aprueba la teoría del primer cuatrimestre y la práctica del primer cuatrimestre y la teoría del segundo cuatrimestre y la práctica del segundo cuatrimestre entonces aprueba la cursada. • Si el alumno no aprueba la teoría del primer parcial entonces es probable que no apruebe la teoría del segundo parcial. • Si el alumno no aprueba la práctica del primer parcial entonces es probable que no apruebe la práctica del segundo parcial. 		

Figura 8. Formulario de Tareas del Experto.

3.4. Aplicación de la Tercera Fase de Educación de Datos de Procesos de Negocio

Las siguientes actividades son descriptas: “Identificar Repositorios de Datos” (sección 3.4.1) y “Relevar Datos del Negocio” (sección 3.4.2).

3.4.1. Actividad “Identificar Repositorios de Datos”

El Analista Funcional analiza la información de las entrevistas realizadas con los interesados del negocio y el formulario que corresponde a las “Tareas del Experto” (figura 8). Como resultado, detecta que los principales datos a utilizar para la realización del proyecto se encuentran en 5 planillas en formato Excel denominadas “*planificación_<dia cursada>*” (donde *<dia cursada>* corresponde a Lunes, Martes, Jueves, Viernes A y Viernes B), que describen el comportamiento de los alumnos (en cuanto a sus notas de parciales, recuperatorios y trabajos prácticos) durante el cursado. Estas planillas fueron brindadas por la Jefa de Cátedra para proceder a su análisis en las actividades siguientes.

3.4.2. Actividad “Relevar Datos del Negocio”

El Analista Funcional confecciona el formulario de estructuras de datos, que se muestra en la figura 9. Dado que las planillas tienen el mismo formato se presenta una estructura de datos genérica que es válida para todos los repositorios de datos.

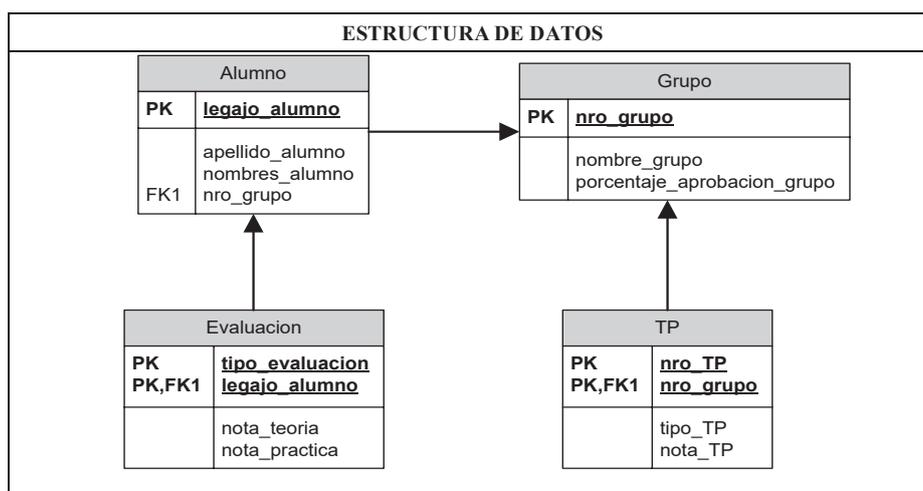


Figura 9. Formulario de Estructura de Datos.

3.5. Aplicación de la Cuarta Fase de Conceptualización de los Datos del Negocio

En esta fase se busca definir si los datos que se utilizarán en el Modelo Predictivo son representativos. En este caso de estudio se realizan dos ciclos de esta fase que se describen a continuación:

- a) En primera instancia se analiza una *Versión Inicial de los Datos* del negocio más general. Para ello, el Analista Funcional trabaja con la información relevada y las 5 planillas recolectadas en la fase anterior. Aunque las mismas constan de tres solapas (“*Planificación Anual*”, “*TPs*” y “*Parciales*”), para este proyecto se utiliza solamente los datos de los exámenes parciales, ya que los datos referidos a los trabajos

prácticos y los grupos de alumnos no se encuentran normalizados para todos los cursos. Asimismo, tampoco se pueden utilizar los nombres, apellidos y legajos de los alumnos por ser considerados confidenciales. Ambas situaciones figuran en el formulario de Restricciones del Proyecto (figura 6).

Luego, se integran las planillas en una sola que incluye 14 atributos correspondientes al desempeño de los alumnos en teoría y práctica de los exámenes (parciales y recuperatorios) así como el día del curso y el resultado de la cursada. De estos datos, se toma una muestra de 75 registros con el objetivo de analizar si son representativos del negocio para construir el Modelo Predictivo. Debido a limitaciones de espacio, todas las tareas aplicadas sobre el conjunto de datos se especifican en [23]. A partir de las conclusiones obtenidas se detecta que esta versión inicial de los datos presenta sesgos que generan diferencias con las relaciones preconcebidas por el docente de los cursos. Es por esto que, se decide generar una nueva versión que se describe a continuación.

- b) En el segundo ciclo, se evalúa una *Versión Extendida de los Datos* la cual incluye mayor cantidad de ejemplos y atributos. Para extender los datos se procede a detallar los resultados de cada tema en cada examen (así, por ejemplo, en la Práctica del primer parcial se indican los resultados para Organigrama y Cursograma). Además, se agregan 51 registros de alumnos, por lo que tienen un total de 126 filas incluyendo 23 atributos. A partir de estos datos se analiza nuevamente si son representativos del negocio para construir el Modelo Predictivo, tal como se detalla en las tareas aplicadas en [23]. Luego del análisis se concluye que la versión extendida es representativa del comportamiento de los alumnos y puede ser utilizada para la construcción del Modelo Predictivo.

3.6. Aplicación de la Quinta Fase de Especificación Inicial del Sistema Inteligente

Las siguientes actividades son descriptas: “Seleccionar el Tipo de Sistema Inteligente” (sección 3.6.1) y “Definir Topología Inicial del Sistema Inteligente” (sección 3.6.2). Asimismo, el proceso completo se encuentra especificado en [24].

3.6.1. Actividad “Seleccionar el Tipo de Sistema Inteligente”

El Analista Funcional responde las preguntas asociadas a cada característica a partir de las actas de reuniones realizadas en fases anteriores, formalizándolas en la tabla 1. Una vez asignados los valores lingüísticos correspondientes para cada característica definida en la tabla 1, el Analista Funcional obtiene los valores correspondientes para cada arquitectura y procede a seleccionar la mejor arquitectura para el proyecto. Las operaciones realizadas para esta actividad se presentan en la planilla de cálculo disponible en [25]. Tal como se muestra en la figura 10, para este caso la arquitectura seleccionada son las *Redes Bayesianas*. Esta selección es justificada, por un lado, por los datos disponibles que incluyen una cantidad suficiente de ejemplos para realizar el entrenamiento y validación de la red pudiendo definir rangos de valores para cada uno. Por otro lado, además se dispone de expertos del dominio (Jefe de Cátedra y Docentes) que pueden aportar su conocimiento y participar en el proyecto. Por lo tanto, se podrán realizar comparaciones entre las predicciones para diferentes escenarios y conocer cómo se obtuvieron los resultados así como realizar ajustes manuales entre los estados de la red que se defina.

Tabla 1. Características Evaluadas para definir la Arquitectura más adecuada.

ID	Pregunta Asociada a la Característica	Valor
D1	¿Cuánta confianza se tiene que los datos son representativos?	mucho
D2	¿En qué medida se pueden considerar a los datos como complejos y con una relación no lineal entre sus atributos?	poco
D3	¿Cuánta cantidad de ejemplos incluyen los datos?	regular
D4	¿Cuál es la proporción de datos con valores numéricos continuos con respecto a discretos?	regular
R1	¿Cuán crítica se considera la precisión de la predicción?	mucho
R2	¿En qué medida se desea conocer comparar las predicciones para diferentes escenarios?	mucho
R3	¿Con qué grado de importancia se desea que se puedan explicar cómo se obtuvieron los resultados generados?	mucho
P1	¿Qué tan estable es el problema a resolver?	mucho
P2	¿Cuánta disponibilidad tienen los expertos del dominio para participar en el desarrollo?	todo
P3	¿Qué tan deseable es poder ajustar manualmente la red a partir de los conocimientos sobre los datos?	todo

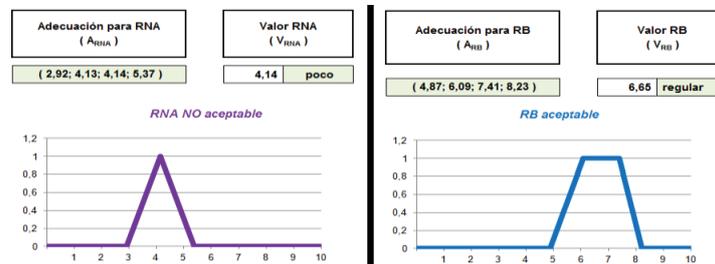


Figura 10. Valores de Adecuación para cada Arquitectura.

3.6.2. Actividad “Definir Topología Inicial del Sistema Inteligente”

Con el tipo de arquitectura ya seleccionado, el Analista Funcional busca determinar un prototipo inicial del modelo propuesto. Para ello, utiliza la herramienta conocida como GeNIe [26], donde decide aplicar una estructura simple donde cada nodo se conecta con el atributo objetivo (CURSADA) y, en el caso de los temas, se vincula el resultado del examen parcial con el primer recuperatorio y éste con el segundo recuperatorio. El motivo de esta relación es que, tal como fue explicado por el experto, se considera que el resultado que tendría un alumno en la teoría del segundo recuperatorio puede estar influenciado por los resultados en el primer recuperatorio y el examen parcial. Con dicha estructura se importan los datos disponibles (correspondientes a la versión extendida obtenida en la fase anterior), para que la herramienta determine las probabilidades correspondientes a cada nodo. Como resultado se obtiene la distribución de probabilidades, tal como se muestra en <https://tinyurl.com/y2c7h7wc>

Aunque la implementación de la Red Bayesiana final queda fuera del alcance de este proceso, con el objetivo de confirmar si esta topología inicial fue entrenada exitosamente, se procede a hacer su validación usando los mismos datos. A pesar de que en un proyecto real no tendría sentido validar un Sistema Inteligente con los mismos datos con que fue entrenado (dado que la precisión así obtenida no es confiable), aquí sólo se busca confirmar que las probabilidades brindadas son representativas de los datos utilizados. Como resultado, la precisión general de esta red es del 94%, con un 100% de precisión para predecir los estudiantes que aprueban el curso, un 87.5% para los estudiantes que no aprueban y un 96.4% para los estudiantes que no terminan la asignatura. Por lo tanto, es posible asegurar que esta topología inicial puede ser usada como un prototipo del Modelo Inteligente para predecir el desempeño de los alumnos de la asignatura.

4. Conclusiones

La implementación de un Modelo Predictivo Inteligente no es una tarea sencilla. Por lo tanto, el proceso propuesto asiste al Ingeniero en los pasos iniciales del proyecto. En este trabajo, se lo ha aplicado en un caso de estudio dentro del ámbito educativo universitario. En la primer fase ha sido posible definir el objetivo que tendrá el Modelo Predictivo. En la segunda fase se ha relevado el proceso de negocio asociado a dicho objetivo. En la tercer fase se han detectado las fuentes de datos, que luego han sido evaluadas en la cuarta fase para identificar un conjunto de datos representativo del comportamiento de los alumnos. Finalmente, en la quinta fase se ha determinado la tecnología más adecuada para construir el Modelo Predictivo.

Referencias

1. De Kohan, N. C. (2008). *Los sesgos cognitivos en la toma de decisiones*. International Journal of Psychological Research, 1(1), 68-73. ISSN 2011-7922.
2. Robbins, S. & Coulter, M. (2010). *Administración. Décima Edición*. Prentice Hall. ISBN: 978-607-442-388-4.
3. Mair, C., Kadoda, G., Lefley, M., Phalp, K., Schofield, C., Shepperd, M. & Webster, S. (2000). *An investigation of machine learning based prediction systems*. Journal of Systems and Software, 53(1), 23-29.
4. García, F. J. M., Martínez, M. A. P. & García, J. S. (2003). *Gestión Estratégica del Conocimiento*. Asociación Universitaria Iberoamericana de Postgrado.
5. Nilsson, N. J. (2014). *Principles of Artificial Intelligence*. Morgan Kaufmann.
6. Russell, S. J., Norvig, P., Davis, E., Russell, S. J. & Russell, S. J. (2010). *Artificial Intelligence: a modern approach* (Vol. 2). Englewood Cliffs: Prentice Hall.
7. Shepperd, M. & Kadoda, G. (2001). *Comparing software prediction techniques using simulation*. IEEE Transactions on Software Engineering, 27(11), 1014-1022.
8. Bontempi, G., Taieb, S. B. & Le Borgne, Y. A. (2013). *Machine learning strategies for time series forecasting*. In Business Intelligence (pp. 62-77). Springer Berlin Heidelberg.
9. Wang, S. C. (2003). *Artificial Neural Network*. In Interdisciplinary Computing in Java Programming (pp. 81-100). Springer US.
10. Wang, L. & Fu, K. (2009). *Artificial Neural Networks*. Wiley Encyclopedia of Computer Science and Engineering, 181-188.
11. Barber, D. (2012). *Bayesian Reasoning and Machine Learning*. The MIT Press.
12. Premchaiswadi, W. (2012) *Bayesian Networks*. Ed. In-Tech.
13. Chatfield, C. (2016). *The analysis of time series: an introduction*. CRC press.
14. De Gooijer, J. G. & Hyndman, R. J. (2006). *25 years of time series forecasting*. International journal of forecasting, 22(3), 443-473.
15. Zhang, G. & Hu, M.Y. (1998). *Neural network forecasting of the British pound/US dollar exchange rate*. Omega, International Journal of Management Science, 26 (4) 495-506.
16. Cohen, P. R. & Feigenbaum, E. A (2014). *The handbook of Artificial Intelligence*. Vol. 3. Butterworth-Heinemann.
17. Acquatela, H. (2006). *La predicción del futuro: desde el oráculo de Delfos hasta la medicina actual*. Gac. méd. Caracas, 114(2), 150-156. ISSN 0367-4762.
18. Rodríguez, M. & Márquez Alegría, M. (2015). *Manejo de problemas y toma de decisiones* (Vol. 8). Editorial El Manual Moderno. SBN: 9789684264670.
19. Arsham, H. (2006). *Tools for Decision Analysis: Analysis of Risky Decisions*.
20. Vegega, C., Pytel, P., Pollo-Cattaneo M. F. (2019). *Aplicación del Proceso de Elicitación de Requerimientos para la Construcción de Modelos Predictivos basados en Sistemas Inteligentes dentro del Ámbito Educativo*. Reporte Técnico GEMIS-TD-2018-05-RT-2019-03. Disponible en <https://tinyurl.com/y2pxnryb>
21. Universidad Tecnológica Nacional – Facultad Regional Buenos Aires (2008). *Programa de la carrera Ingeniería en Sistemas de Información (Plan 2008)*. DISI. Disponible en <https://tinyurl.com/y2kb36xv>
22. Gómez, A., Juristo, N., Montes, C. & Pazos, J. (1997) *Ingeniería del Conocimiento*. Centro de Estudios Ramón Areces. S.A. Madrid.
23. Vegega, C., Pytel, P., Straccia, L., Pollo-Cattaneo, M.F. (2018). *Evaluation of the Bias of Student Performance Data with Assistance of Expert Teacher*. Applied Informatics, Communications in Computer and Information Science book series (CCIS, volume 942). Proceedings of First International Conference (ICAI 2018). Págs 16-31. Online ISSN: 1865-0937.
24. Vegega, C., Pytel, P., Straccia, L., Pollo-Cattaneo, M.F. (2018). *Proceso de Selección de Arquitectura a fin de Implementar un Modelo Predictivo Inteligente*. Memorias de 6to. Congreso Nacional de Ingeniería Informática y Sistemas de Información (CONAIIISI 2018). Workshop de Aplicaciones Informáticas y de Sistemas de Información. ISSN 2347-0372. Disponible en <https://tinyurl.com/y6zjhjnv>
25. Vegega, C., Pytel, P. & Pollo-Cattaneo M.F. (2017). *Método Evaluador de Arquitectura - Desempeño de Alumnos de Sistemas y Organizaciones*. Disponible en <https://bit.ly/2YwESUM>
26. BayesFusion (2015) *BayesFusion - Home Page for users of GeNe*. BayesFusion LLC. Disponible en <https://www.bayesfusion.com/>

Evaluación de estrategias públicas de salud digital para el desarrollo de una solución de interoperabilidad local entre Sistemas de Información en Salud

Martin Correa¹, Lía Hebe Molinari², Rodolfo Bertone^{2,3}

Facultad de Informática
Universidad Nacional de La Plata

¹ martin.correa@amepla.org.ar, ² {lmolinari, [pbertone](mailto:pbertone@info.unlp.edu.ar)}@info.unlp.edu.ar

³ III-LIDI (Instituto de Investigación en Informática)

Resumen. En el ámbito asociado a salud, se observa que existe un alto grado de intercambio de información entre los diferentes actores involucrados, pero a pesar de ello los datos se encuentran altamente fragmentados y distribuidos en múltiples sistemas de información no integrados, imposibilitando la comunicación y el intercambio de información entre ellos.

Un sistema es interoperable gracias a su capacidad de transferir información de un paciente de un sistema a otro. Esta acción es imposible de lograr sin el uso de estándares que permitan interoperar efectivamente. En el estudio realizado se investiga el estado de implementación, clasificación y mecanismos para su desarrollo en Argentina, tanto para el sector público como en el privado.

A partir de ello se determinan los aspectos más relevantes para plantear una solución que permita integrar en red los sistemas de información en salud de la ciudad de La Plata, dentro del modelo de negocios de la Asociación Médica Platense (AMP) y se define una arquitectura a seguir.

Con el aporte de las Tecnologías de la Información y la Comunicación (TIC) se pretende mejorar y optimizar la relación coste-beneficio, contribuir a la seguridad, incrementar la eficacia y la eficiencia de los prestadores y de la organización. Para aprovechar el potencial que ofrecen, es fundamental acordar las normas técnicas, semánticas y sintácticas de interoperabilidad.

Palabras Clave: redes, comunicación, intercambio de datos, sistemas de información en salud, estándares.

1. Introducción

El Sector salud es uno de los que más ha aprovechado y tenido un crecimiento importante en los últimos tiempos del uso de tecnología. Tiene la particularidad de incorporar permanentemente prácticas nuevas, sin que pierdan vigencia las viejas^[1].

Paradójicamente tiene un mal manejo de la información, ya que se realizan registros manuscritos (Historias Clínicas en papel, recetas ilegibles, informes incomprensibles, mala letra, etc.), lo que deriva en una normalización e informatización deficiente.

Cada vez más existe una necesidad de intercambio de información entre los distintos actores (prestadores, financiadores, etc.) el cual mayoritariamente se realiza con soporte de papel de documentación, la mayoría de las veces manuscrita y en forma de fotocopias, fax. Existe también un intercambio de información mediante soporte electrónico pero de forma limitada y en donde el mismo se produce de una manera no normalizada, como archivos de texto, planillas de cálculo o PDF

Cada uno de los actores del sistema de salud propone un formato distinto, ya que no existe un consenso en el uso de estándares, con la consecuente dificultad en intercambiar y procesar la información.

En ese sentido la realidad de la Agremiación Médica Platense (AMP) no escapa a lo descripto, motivo por el cual desde el consejo directivo de la institución se encara un proyecto general se encuadra dentro del modelo de negocio de la institución, en el cual se detectan una serie de inconvenientes que se pretende sean resueltos aplicando TICs.

En primer lugar se puede mencionar la carga de bonos (en formato papel), que se procesan de forma centralizada en AMP luego que el profesional los presenta en la institución. Asociado a esto, existen inconvenientes respecto al tiempo desde el ingreso hasta la facturación, ya que existen plazos perentorios desde que se realiza el acto médico para presentar en el agente del seguro de salud, lo cual influye en los tiempos de procesamiento dentro de la institución.

Una cuestión no menor es el tema que deben enviarse todos los bonos facturados (que en muchos casos tiene documentación anexa) junto a la correspondiente factura e informes de facturación al agente del seguro de salud, siempre en formato papel.

Por los tiempos en diferido, existe la posibilidad que el agente del seguro de salud rechace el pago de prestaciones, también en tiempo diferido. Adicionalmente los rechazos pueden ser refutados, generando nuevas facturaciones que pueden extender los tiempos otros 60 días adicionales

2. Interoperabilidad en el ámbito de la salud

Se define como Interoperabilidad (IO) a la habilidad de dos sistemas de información para compartir información y utilizar la información compartida. Esto implica la transferencia de información estructurada, que puede descomponerse y analizarse en el sistema destino, para ser integrada con información local y de otras fuentes.

Alcanzar IO en los sistemas de información en salud es la clave para asegurar la integración de la información sanitaria, que se encuentra almacenada en los repositorios clínicos electrónicos de cada uno de los actores del sistema de salud.

2.1 Beneficios

Contar con información integrada tiene claros beneficios en la atención del paciente, donde los profesionales pueden tomar decisiones con información completa y legible. Los mecanismos de IO simplifican los procesos administrativos facilitando la integración de sistemas de turnos, recetas, órdenes médicas, etc.

Un escenario con sistemas de información interoperables asegura la transmisión completa, oportuna y segura de información, mejorando la capacidad del sistema de salud para responder a las necesidades de salud pública de la población.

2.2 Niveles de Interoperabilidad

Los niveles de interoperabilidad en salud se pueden determinar utilizando la clasificación propuesta por la European Telecommunication Standards Institute (ETSI). La figura 1 presenta la clasificación que se define a continuación ^[2].

1. *Técnica*: se asocia generalmente con componentes de hardware y/o software, que permiten la comunicación que tendrá lugar de máquina a máquina. Este tipo de interoperabilidad a menudo define protocolos de comunicación.

2. *Sintáctica*: relacionada habitualmente con los formatos de datos. Los mensajes transferidos por protocolos de comunicación necesitan una sintaxis y codificación bien definida, planteando sintaxis de transferencia de alto nivel. Esto no garantiza la correcta interpretación y uso, sino que se resuelven temas técnicos y de estándares en las áreas de protocolos y formatos de intercambio.

3. *Semántica*: nivel de IO que se asocia con el significado de los contenidos y se refiere a la interpretación humana del contenido. IO en este nivel significa que hay un entendimiento común entre personas sobre el significado del contenido (información) que se intercambia y se garantiza la correcta interpretación y uso de la misma, para lo cual se necesitan definiciones formales de cada entidad, atributo, relación, restricción y término intercambiado.

4. *Organizacional*: es la capacidad de las organizaciones para comunicar y transferir efectivamente los datos (información), sin importar los sistemas utilizados, o eventualmente diferentes infraestructuras, a través de diferentes regiones geográficas y culturas. La IO organizacional deriva del éxito de los niveles técnico, sintáctico y semántico.

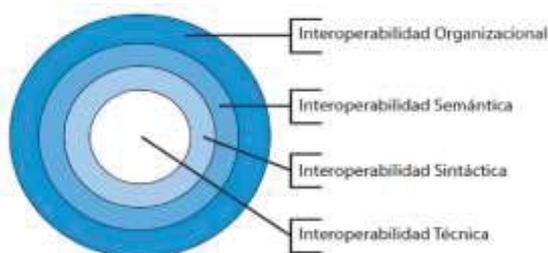


Figura 1. Niveles de Interoperabilidad

2.3 Estándares de interoperabilidad en salud

Un estándar es un documento aprobado por consenso por un organismo reconocido, que proporciona pautas, reglas y/o características para un uso común, con el objeto de obtener un óptimo nivel de resultado en un contexto dado^[3].

2.3.1 HL7

HL7 Internacional (por *Health Level Seven*) es una iniciativa que comenzó en 1987, en base a la necesidad de normalizar las interfaces entre los múltiples sistemas heterogéneos de información, y rápidamente se convirtió en el estándar de facto para el intercambio electrónico de datos clínicos y administrativos en los servicios de salud de los Estados Unidos, planteando actualmente un conjunto de 15 estándares^[4].

La estructura internacional de la organización, el procedimiento de votación balanceado, y las políticas abiertas de asociación, aseguran que todos los requerimientos sean tenidos en cuenta uniformemente y equitativamente con calidad y consistencia.

Diferentes sectores participan en forma colaborativa en la discusión y en el desarrollo de estándares por consenso, en un entorno abierto. Estos sectores pueden ser prestadores de servicios de salud, desarrolladores de software, consultores, usuarios finales, organizaciones gubernamentales y no gubernamentales, etc.

2.3.2 FHIR

FHIR (por *Fast Healthcare Interoperability Resources*), es el último estándar desarrollado y promovido por HL7. que combina las mejores características de las líneas de producto HL7 versión 2, versión 3 y CDA.

Los artefactos de IO de FHIR se generan a partir de un conjunto de componentes modulares, cuyo contenido ha sido diseñado para el intercambio de datos. Estos se denominan recursos, son representaciones de conceptos empleados en el contexto de la salud (paciente, medicación, observación, etc.), y funcionan como bloques de construcción que permiten componer estructuras de mensajes y/o documentos.

Todos los recursos comparten una manera común de definirlos y representarlos, un conjunto común de metadatos y un elemento para facilitar la legibilidad humana.

2.3.3 Nomenclatura sistematizada de Términos de Medicina Clínica (SNOMED-CT)

SNOMED Internacional es una organización compuesta por 31 países miembros (Argentina es el último en incorporarse durante 2018), que ha conformado el estándar global SNOMED-CT (por *Systematized Nomenclature of Medicine – Clinical Terms*) para la representación de información en salud^[5].

La terminología definida es para generar estándares en sistemas de historia clínica electrónica, de prescripción electrónica, laboratorios, encuestas sanitarias, registros de enfermedades y otros, y tiene el objetivo de capturar de manera precisa y unívoca referencias a conceptos clínicos. Incluye más de 330.000 conceptos clínicos,

divididos en jerarquías temáticas como síntomas, diagnósticos, antecedentes familiares, procedimientos quirúrgicos, estudios diagnósticos, etc.^[5].

Su objetivo es facilitar el registro exacto de la información médica y otros temas relacionados con la salud y la IO semántica de los registros clínicos.

3. Estado del arte en Argentina

En la actualidad existe reciente regulación para el intercambio estandarizado de información en salud, que plantea subsanar algunas situaciones como:

- Falta de regulación de tecnologías (estándares).
- No se puede compartir la información de una forma rápida y confiable.
- No se puede controlar eficientemente la validación, integridad, seguridad y disponibilidad de la información.
- Fragmentación e ineficiencia en el uso de los recursos., que no se vinculan para optimizar esfuerzos, sino que por el contrario trabajan de forma desintegrada y aislada, repitiendo en muchos casos las mismas tareas.

3.1 Marco Argentino de Interoperabilidad en Salud (MAIS)

El Marco Argentino de IO en Salud (MAIS) es un proyecto para estandarizar el intercambio de documentación clínica, el detalle de la facturación y los débitos entre prestadores y financiadores. Utilizando el estándar HL7 FHIR, CDA para especificar la estructura y semántica de documentos clínicos^[6].

Persigue el objetivo de intercambiar documentos electrónicos con información en salud entre instituciones con Sistemas de Información diferentes, sin la necesidad de una plataforma de IO central, en donde la información fluye de los prestadores a los financiadores, como adjunto de la facturación o en forma independiente.

La metodología de trabajo consiste en reuniones periódicas entre los intervinientes (actores del sistema de salud nacional), donde se proponen y consensuan las tecnologías a utilizar y las adaptaciones a realizar, vocabularios e identificadores comunes para facilitar la IO. También se decide los tipos de documentos para interoperar, que datos son obligatorios y que datos son optativos.

Se genera un consenso entre los actores, de forma tal de mejorar la eficiencia en la comunicación electrónica. Para llevar a cabo esta tarea, es necesario trabajar todos en conjunto para solucionar el problema, definiendo los estándares y tecnologías para hacer la interoperabilidad una realidad.

Con el objeto de intentar aprovechar los éxitos y evitar los fracasos o barreras para la implementación autogeneradas en proyectos anteriores, se estudian las mejores prácticas, proponiendo tener en cuenta los estándares, guías de implementación y proyectos, sin que esto implique que el proyecto se incline automáticamente por adoptar las especificaciones tal como están sino pensando en localizar y simplificar lo más posible el uso. La definición de estándar y guía de implementación FHIR incluye elementos para facturación y débitos a partir de su release 2.1.

Se intercambian documentos electrónicos con información en salud entre instituciones con Sistemas de Información diferentes, sin la necesidad de una

plataforma de interoperabilidad central. La información fluye de los prestadores a los financiadores, como adjunto de la facturación o en forma independiente.

3.2 Red de Salud Digital: Bus de Interoperabilidad

3.2.1. Introducción

El Gobierno Nacional por Decisión Administrativa N° 307/18 creó y asignó a la Dirección Nacional de Sistemas de Información la responsabilidad primaria de gestionar las condiciones para el desarrollo y coordinación de los sistemas de información en salud a nivel nacional y jurisdiccional ^[7].

La Estrategia Nacional de Salud Digital, aprobada mediante Resolución N°189 de fecha 25 de octubre de 2018, de la Secretaría de Gobierno de Salud, la cual prevé dentro de sus objetivos que los registros y programas nacionales cuenten con datos precisos y completos, y que estos sean recibidos mediante los estándares de interoperabilidad propuestos, que simplifican y homogenizan los flujos de información. Se complementa con la Resolución N° 680 de fecha 5 de diciembre de 2018 donde se aprobó la adopción de estándares de interoperabilidad en salud, lo que se realizará de manera paulatina.

3.2.2. Red de Interoperabilidad

Se dispone la creación de una Red Nacional de Interoperabilidad que posibilite la integración de los sistemas de información de todas las jurisdicciones y sectores del sistema de salud, incluyendo los subsectores público y privado.

La red de interoperabilidad está compuesta por nodos, que representan sistemas de información en salud independientes, utilizados en diferentes niveles, como sistemas provinciales, municipales, de un establecimiento asistencial, etc. Cada nodo tiene un repositorio de información sanitaria independiente, y dependiendo de los modelos de organización de cada uno pueden abarcar un único establecimiento o una red de los mismos. Estos nodos tienen autonomía tecnológica, y tienen control completo y responsabilidad sobre la información almacenada, cumpliendo su rol de depositarios de información sanitaria.

La creación de la red de interoperabilidad no modifica los usos y costumbres relacionados a los procedimientos actuales de almacenamiento de información sanitaria. La información permanece en el repositorio de cada sistema de información y solo se comunica con otros nodos en caso de existir necesidades concretas.

Los registros nacionales y provinciales que requieren información estadística, de gestión o administrativa. Hoy estos registros tienen flujos de información heterogéneos, algunos en papel, otros en archivos compartidos, formularios web. La red de interoperabilidad permite que estos flujos existentes se realicen utilizando estándares de interoperabilidad, simplificando las tareas de integración y mejorando la calidad y oportunidad de la información.

La información relacionada con prestaciones, su facturación y recupero, se comunica a financiadores cumpliendo con los requisitos de cada uno, posibilitando la implementación de procesos de reembolso sin papeles.

3.2.3. El Bus de Interoperabilidad

La comunicación entre dominios con diferentes sistemas de información requiere de la implementación de una infraestructura central que facilite la integración y la comunicación de los registros médicos en el país.

Esta infraestructura se basa en el patrón de integración conocido como “Enterprise Service Bus (ESB)” en una arquitectura orientada a servicios (SOA).

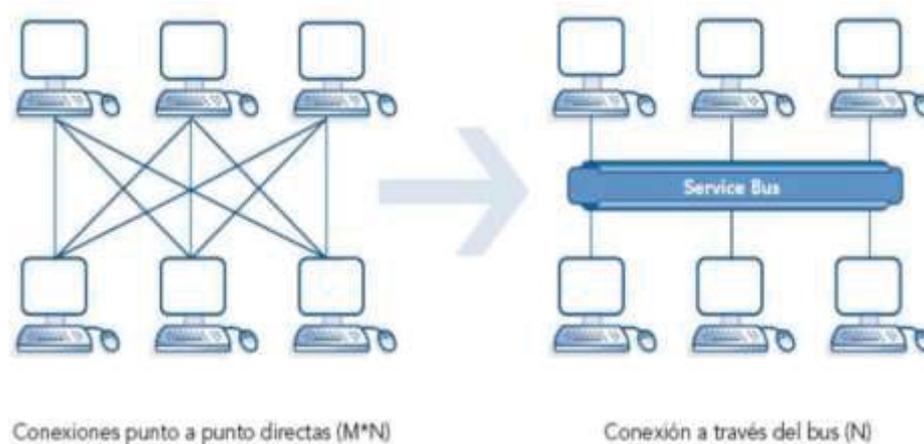


Figura 2. Patrón de Enterprise Service Bus

La implementación de este patrón en el contexto de dar soporte a una red de interoperabilidad se denomina “Bus de Interoperabilidad”.

El Bus de Interoperabilidad es provisto por el nivel central, y está disponible en una plataforma de alta disponibilidad y escalabilidad. Los diferentes programas y registros de la Secretaría de Gobierno de Salud estarán conectados al Bus de Interoperabilidad como dominios para facilitar y unificar los procesos de transferencia de información existentes hoy en día.

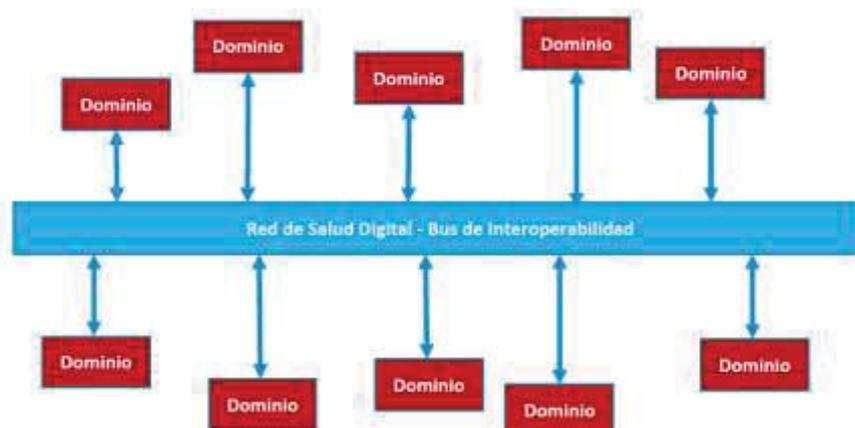


Figura 3. Componentes y funciones del Bus de Interoperabilidad

3.2.4. Objetivos de Interoperabilidad

El Bus de Interoperabilidad articula la comunicación entre los nodos de la red de interoperabilidad, con la finalidad de cumplir los ciertos objetivos destacándose:

Establecer una infraestructura de interoperabilidad a nivel nacional que permite compartir documentos e información clínica centrada en el paciente, generando una historia clínica compartida longitudinal, que integre el sistema público, de obras sociales y privado. La historia clínica compartida se almacena en forma distribuida en cada uno de los puntos de contacto del paciente

El Bus de Interoperabilidad provee servicios Web, con una tecnología REST, y con operaciones especificadas por el estándar HL7 FHIR, que define recursos e interacciones con el servidor.

La codificación de entidades clínicas se realiza con SNOMED-CT, cuando los documentos clínicos electrónicos tienen que representar referencias a componentes del sistema de salud como Profesionales, Establecimientos, Especialidades, o Financiadores. El Bus de Interoperabilidad brinda acceso a los Registros Nacionales de Referencia como por ejemplo los Registros Federales de Profesionales y Establecimientos (REFEPS y REFES).

La adherencia a estos estándares es un requisito ineludible para la participación en la red de interoperabilidad, y asegura que la información compartida pueda ser interpretada y procesada en el punto de destino.

Por último, la red de Interoperabilidad debe impulsar la aplicación de procesos y medidas que prevengan la pérdida de información y el acceso indebido o inapropiado.

4 Planteo de la solución para la organización

4.1 Estado actual

A pesar de los esfuerzos y definiciones anteriormente descriptas y analizadas en torno a la interoperabilidad en el ámbito de salud, la realidad indica que el subsector privado se encuentra lejos aún de recrear soluciones de este tipo, estando situados prácticamente en la problemática inicial de sistemas heterogéneos y fragmentados. Es por esto que se definió plantear un desarrollo que abstraiga estos inconvenientes, resolviendo un esquema que permita homogeneizar y estandarizar internamente la integración entre sistemas.

4.2 Características

La solución planteada requiere que el agente posea servicios web (WS) expuestos con tres operaciones básicas que todos proveen: validación de afiliado, autorización de atención y anulación de la misma.

Se plantea una solución que posee neutralidad tecnológica, ya que se basa en el uso de los servicios, poniendo énfasis en la interoperabilidad y no en la tecnología utilizada en las implementaciones de los agentes

Al trabajar con este enfoque, se logró un diseño del framework implementable de forma simple en cualquier lenguaje orientado a objetos. El esfuerzo de codificación se centra en los detalles específicos del lenguaje de programación de cómo resolver determinadas acciones, por ejemplo, acceso a servicios web o serialización / deserialización de objetos. De esta forma, también se logra que la arquitectura sea portable si algún avance tecnológico así lo amerita.

Dado que la incorporación de la integración con los agentes es paulatina (ya que incluye el convenio entre instituciones, que escapa a cuestiones técnicas o tecnológicas) la solución en forma de framework, tiene la capacidad de ser escalable en interoperar con agentes del seguro de salud, de una forma rápida y sencilla en términos de implementación.

Respecto a las operaciones que se ofrecen, se continúan brindando las 3 (tres) operaciones básicas identificadas. Se resuelve, en este caso, una abstracción que permita manejar dichas operaciones de forma homogénea, independientemente del agente con el cual se está interoperando, tanto a nivel las operaciones en sí mismo, como al intercambio de información.

4.3 Arquitectura

El framework se planteó con una arquitectura modelada en 2 capas. Cada una posee una funcionalidad específica y se complementan entre sí para gestionar el marco de interoperabilidad requerido. Para lograr eficiencia y simplicidad dentro de la arquitectura en general y las capas en particular, se tomó como base para la implementación el diseño orientado a objetos, aplicando patrones de diseño orientados a objetos ^[8]

4.3.1. Capa de Conectividad

Esta capa del framework se puede denominar también de “bajo nivel”, dado que a la implementación de los accesos a los servicios expuestos por los distintos agentes de salud

Se traduce en la implementación dentro del framework de objetos denominados “Proxy de conexión”, que son los que operan directamente con los WS de cada uno de los agentes de salud. La denominación se basa en el patrón de diseño *Proxy* en forma representante de un objeto remoto (en este caso el servicio web que expone el agente para interoperar).

Estos objetos cumplen el objetivo principal de abstraer las cuestiones técnicas de la interoperabilidad como pueden ser la tecnología utilizada en el WS o la estructura de mensajes intercambiados. Como resultado generan objetos con la información recibida.

4.3.2. Capa de Convergencia

Esta capa es la encargada de brindar la información que se obtiene en la capa inferior (conectividad) de forma homogénea. Abstrae al usuario del framework de los

detalles propios de cada agente para utilizar las mismas características en todas (operaciones, nombres, información). Funciona tomando la esencia del patrón de diseño *Adapter*

La capa de convergencia de forma interna utiliza el patrón de diseño *Factory* ya que dentro de la ejecución de las operaciones solicitadas al framework, el usuario debe indicar a qué agente quiere hacer el requerimiento, y a través de este patrón se genera el objeto "Proxy de conexión" correspondiente a dicho agente, para poder interoperar.

En este caso la capa expone al usuario del framework tres objetos que representan la gestión de las operaciones comunes a los agentes que se han mencionado.

5. Conclusiones

El intercambio de información entre los actores del sistema de salud es cada vez mayor y compleja. En Argentina existe un camino tanto a nivel público como privado a mejorar ese intercambio, planteando interoperar a través de estándares.

La aplicación de TIC en los procesos de la AMP como institución, viene a fortalecerlos y que sean más eficientes y eficaces. También plantea toda una redefinición de dichos procesos adecuándolos al entorno digital en el cual se sumergen.

Respecto a la solución brindada, la misma está basada en un modelo flexible, que permite implementar integraciones e interoperar con otros actores de una forma simple y robusta a la vez.

La arquitectura planteada para el framework permite reducir los tiempos de implementación de una integración entre sistemas, cubriendo las falencias actuales de interoperabilidad

Por último, se puede destacar, como con este proyecto, la AMP se posiciona estratégicamente en la relación con los distintos actores de salud de la ciudad de La Plata, promoviendo y fortaleciendo sus objetivos como institución.

6. Referencias

1. Desafíos de la interoperabilidad en el ámbito de la Salud
https://www.agesic.gub.uy/innovaportal/file/1339/1/renoir_selena_y_antonio.pdf
2. Revisión de estándares de interoperabilidad para la eSalud en Latinoamérica y el Caribe
<http://iris.paho.org/xmlui/handle/123456789/28188>
3. Principios de interoperabilidad en salud y estándares
Campos, Fernando - Kaminker, Diego - Otero, Carlos
ISBN 978-987-46878-0-7
4. HL7 Argentina
<http://www.hl7.org.ar/>

5. SNOMED Argentina
<https://www.argentina.gob.ar/salud/snomed>
6. Marco Argentino de Interoperabilidad en Salud
<https://www.mais.org.ar/>
7. Ministerio de Salud de la Nación
<https://www.argentina.gob.ar/salud>
8. Design Patterns. Elements of Reusable Object-Oriented Software
Erich Gamma, Richard Helm, Ralph Johnson, John Vlissides - Addison Wesley
ISBN: 0201633612

Hacia un marco de trabajo para desarrollar sistemas de programación de la producción utilizando servicios en la nube.

Daniel Díaz^{1,2}, Sandra Oviedo^{1,2}, Fernando Guardia^{1,2}, Juan Manuel Cuneo²,
Alejandra Otazú²

¹Laboratorio de Informática Aplicada a la Innovación
Instituto de Informática, ²Departamento de Informática
FCEF- UNSJ

{ddiaz, soviedo, fguardia, jmcuneo, aotazu}@iinfo.unsj.edu.ar
San Juan, Argentina

Resumen. Desde la perspectiva de la Ingeniería Industrial un sistema de programación de la producción es el corazón de cualquier industria dedicada a la manufactura de cualquier producto. Este tiene como responsabilidad determinar que tarea deberán realizar determinados recursos productivos, tales como máquinas u operarios, en un determinado momento, para cumplir con los objetivos que impone la dirección de la empresa. La programación de la producción es parte del sistema de planificación y control de la producción, el cual es parte del sistema logístico de una empresa. Desde una perspectiva matemática, un sistema de programación de la producción trata con un conjunto de problemas de programación de la producción. En la búsqueda de soluciones a estos problemas intervienen las Ciencias de la Computación. A lo largo de tiempo los investigadores han desarrollado muchos algoritmos para resolver este tipo de problemas, pero lamentablemente gran parte de esta investigación no ha sido reflejada en el campo industrial, este problema se conoce como el gap entre la academia y la industria. Una forma de vincular academia con industria son los sistemas de programación de la producción, por lo tanto, proponer nuevas formas para construir sistemas de programación de la producción ayuda a reducir este gap. Este trabajo propone un marco para desarrollar sistemas de programación de la producción utilizando servicios en la nube.

Palabras clave: scheduling systems, microservicios, cloud services.

1 Introducción

Las herramientas informáticas que permiten solucionar problemas de optimización industriales se denominan “optimization solvers”, son plataformas que permiten modelar, diseñar e implementar soluciones para problemas de optimización. Los problemas de scheduling son un caso particular de los problemas de optimización [1] y

en la industria se los denomina problemas de programación de la producción, de aquí en adelante cuando se utilice la palabra scheduling se estará haciendo referencia a este tipo de problemas. Un sistema de scheduling trata con muchos problemas de scheduling [2]. En una planta industrial, un sistema de scheduling es el corazón del sistema de planificación y control de la producción. Según la arquitectura jerárquica ANSI/ISA 95 que guía la organización de una planta industrial, el sistema de scheduling interactúa con otros sistemas de fabricación, tales como ERP (Enterprise Resource Planning) y MES (Manufacturing Execution System). En los últimos años, con el surgimiento de la industria 4.0 el proceso de planificación y programación de la producción está en constante cambio debido a que las funciones del MES han sido absorbidas por el CPPS (Cyber-physical production systems) [3], [4]. Existe una literatura muy extensa sobre scheduling y se ha escrito mucho sobre modelos y algoritmos que resuelven problemas de scheduling. Sin embargo, poco son los artículos que tratan de cómo traer estos modelos y algoritmos a implementaciones reales. Esto se conoce como el “hueco o gap” entre la teoría y la práctica de scheduling [5]. En [6] se propone al desarrollo de software como un camino para cerrar este hueco y se enfatiza en la necesidad de proponer nuevos caminos para diseñar e implementar sistemas de scheduling. Hoy en día, el desarrollo de software está en un constante cambio movilizado por las tecnologías de la computación en la nube. En el entorno industrial estas tecnologías están permitiendo que las pequeñas y medianas empresas utilicen las mismas tecnologías que las grandes empresas. Antes, este tipo de tecnologías solo eran accesibles a las grandes empresas debido a los costos que ellas implicaban. Los últimos avances tecnológicos han logrado que los solucionadores de optimización, orientados a resolver problemas de scheduling, se ofrezcan como servicios económicamente accesibles en la nube. Sin embargo para hacer un uso eficiente y eficaz de estos servicios se necesitan nuevos mecanismos que habiliten el desarrollo de sistemas de scheduling con estas tecnologías. Estos sistemas deben ser diseñados para aprovechar toda la potencia que brinda la nube en cuanto a escalabilidad, tolerancia a fallos, velocidad de cómputo, y economía de uso. En este artículo se propone un marco de trabajo para desarrollar sistemas de programación de la producción utilizando servicios en la nube, dicho marco está basado en el estilo arquitectónico de microservicios [7].

2 Evolución de los sistemas de scheduling

La investigación en el diseño y desarrollo de sistemas de scheduling se puede clasificar en 5 generaciones. Inicialmente, esta clase de problemas fue abordada por el área de la ingeniería industrial, y posteriormente por el dominio de la investigación operativa. En los comienzos del siglo pasado, Henry L. Gantt, fue uno de los primeros en proponer un sistema de scheduling: daily balance “el equilibrio diario” que consiste en un método de programar y de registrar el trabajo. En 1960 se desarrolló Planalog Control Board, una herramienta que constaba de una pizarra y entre otras cosas podía forzar restricciones de precedencia. Los primeros sistemas de scheduling computarizados nacieron con el MRP (Material Requirement Planning). En la segunda generación surgen los ISS (Intelligent Scheduling Systems), tales como ISIS y OPIS de [8]. Esta fue una etapa donde el problema de scheduling se comenzó a abordar con técnicas de inteligencia artificial tales como algoritmos genéticos, simulated annealing, redes

neuronales y constraint satisfaction techniques. En esta generación aparecen los sistemas de scheduling que se construyen utilizando técnicas de KBS (Knowledge Based Systems). La tercera generación se inició con el surgimiento del paradigma orientado a objetos, dando origen a sistemas de scheduling desarrollados con esta tecnología. Esto permitió el desarrollo de frameworks especializados en scheduling, los cuales encapsulan técnicas de inteligencia artificial [9]. La cuarta generación de sistemas de scheduling es la que se integra en lo que se conoce como Advanced Planning and Scheduling Systems [10]. La quinta generación está surgiendo actualmente y está basada en los sistemas ciberfísicos o CPS. En [11] y [12] se los define como sistemas de entidades que colaboran computacionalmente, que están en conexión intensiva con el mundo físico circundante y sus procesos en curso, proporcionando y utilizando, al mismo tiempo, servicios de acceso y procesamiento de datos disponibles en internet. En [12] se extiende el concepto básico de CPS a los sistemas de producción lo que da como resultado el concepto de Sistemas de Producción Ciberfísicos CPPS. En [13] se presenta una definición de Industria 4.0 que coloca a los CPS como uno de los elementos básicos que posibilitan el surgimiento de este paradigma. El proceso de scheduling de producción se ve notoriamente afectado por el surgimiento de la sistemas ciberfísicos, en [14] se discute una propuesta para tratar con los problemas de Job-shop Scheduling en el entorno de un sistema ciberfísico. En [15] se introduce un nuevo esquema de toma de decisión denominado Smart Scheduling capaz de hacer frente a los cambios propuestos por la industria 4.0 en lo que respecta a la toma de decisiones de un sistema de scheduling de producción. En [3] se discute como la arquitectura jerárquica ANSI/ISA 95 ha sido modificada por la industria 4.0 y más específicamente como estos cambios impactan en el proceso de scheduling de un industria.

3 Computación en la nube

Según [16] Cloud Computing es un modelo para habilitar acceso conveniente por demanda a un conjunto compartido de recursos computacionales configurables, por ejemplo, redes, servidores, almacenamiento, aplicaciones y servicios, que pueden ser rápidamente aprovisionados y liberados con un esfuerzo mínimo de administración o de interacción con el proveedor de servicios. Este modelo de nube promueve la disponibilidad y está compuesto por cinco características esenciales, tres modelos de servicio y cuatro modelos de despliegue.” Esta definición destaca que cloud es un modelo de prestación de servicios cuya principal orientación es la escalabilidad. Desde una perspectiva económica los servicios son elásticos, donde el usuario paga por lo que consume. Este modelo está compuesto por cinco características esenciales, tres modelos de servicio y cuatro modelos de despliegue. Las características esenciales son las que describen al modelo de cloud computing, en [17] se presentan como Auto-servicio por demanda, Acceso amplio desde la red, Conjunto de recursos, Rápida elasticidad, Servicio medido. Los Modelos de Servicios se refieren a la forma en la cual están organizados los servicios de cloud computing, son tres a saber, *IaaS* (Infrastructure as a Service), *PaaS* (Platform as a Service), *SaaS* (Software as a Service), una descripción detallada se puede ver en [18]. Los Modelos de Despliegue hacen referencia a cómo se

monta y pone en funcionamiento el sistema. Estos se presentan en [17] son *Nube privada*, *Nube comunitaria*, *Nube pública*, *Nube híbrida*.

4 Despliegue y Arquitecturas de aplicaciones en la nube

El desarrollo de aplicaciones basadas en los modelos IaaS o PaaS al momento del despliegue de las mismas se debe considerar los requerimientos sobre las capacidades de la computación en la nube que necesitan de las aplicaciones, tales como escalado automático, performance, despliegue en caliente, disponibilidad, monitoreo dinámico entre otros. Una arquitectura de una aplicación en la nube, entre otras cosas, está influenciada por las capacidades de la computación en la nube que se desean aprovechar. Generalmente, el desarrollo de aplicaciones en la nube es llevado a cabo ya sea siguiendo el estilo arquitectónico monolítico o siguiendo el estilo arquitectónico de microservicios [7] que se describen a continuación.

4.1 Arquitectura monolítica

Una aplicación para la nube basada en el estilo monolítico normalmente se compone de tres partes [7], en el lado del cliente está la interfaz de usuario, la cual normalmente se compone por un archivo HTML, un archivo javascript y una hoja de estilo. Estos archivos se ejecutan sobre un browser instalado en la máquina del usuario. El segundo componente es una base de datos, que contiene todos los datos con los que trabaja la aplicación. Por último, en el lado del servidor existe una aplicación que gestiona las solicitudes HTTP, ejecuta la lógica del dominio, lee, escribe y modifica datos de la base de datos, y construye los archivos HTML que son enviados al cliente. Esta aplicación es monolítica, todo está empaquetado como un único ejecutable. Cualquier cambio en el sistema implica construir y desplegar una nueva versión de la aplicación en el servidor. En [19] se enumeran una serie de inconvenientes de las aplicaciones monolíticas:

- Las aplicaciones monolíticas de gran tamaño son difíciles de mantener y evolucionar debido a su complejidad.
- La localización de errores requiere tiempo y un esfuerzo considerable de análisis del código.
- Sufren el "infierno de dependencia", una problemática que sucede cuando al agregar o actualizar bibliotecas da como resultado sistemas inconsistentes que no compilan / ejecutan o, peor aún, tienen mal comportamiento.
- Cualquier cambio en un módulo requiere reiniciar toda la aplicación. En proyectos de gran tamaño implica dejar a los usuarios sin sistema por tiempo considerables. Esto dificulta las pruebas y el mantenimiento del sistema.
- Dificultad para optimizar recursos de infraestructura de la nube, debido a que los requerimientos entre algunos módulos que componen la aplicación pueden ser contradictorios.
- Dificultan la escalabilidad, al escalar es necesario escalar toda la aplicación, lo que deviene en el consumo de grandes cantidades de recursos que normalmente no son utilizados

4.2 Arquitectura de microservicios

Un microservicio es una pequeña aplicación que se puede implementar de forma independiente, escalar de forma independiente y probar de forma independiente y que tiene una única responsabilidad [20]. Con responsabilidad única por un lado se refiere a que tiene una sola razón para cambiar y / o una sola razón para ser reemplazada. Pero el otro lado, una responsabilidad única en el sentido de que solo hace una cosa y una sola cosa y así puede entenderse fácilmente.

Una arquitectura de microservicios es una aplicación distribuida donde todos sus módulos son microservicios. Un análisis de las características más sobresalientes de una arquitectura de microservicios se puede encontrar en [19], a continuación se enumeran algunas de ellas:

- Los microservicios se pueden desplegar por separado permitiendo gestionar independientemente sus respectivos ciclos de vida.
- Nuevas versiones de microservicios se pueden introducir gradualmente en el sistema, desplegándolas junto a la versión anterior y los servicios que dependen de la última se pueden modificar gradualmente para interactuar con la primera. Esto fomenta la integración continua y facilita enormemente el mantenimiento del software.
- Como consecuencia del anterior ítem, desplegar cambios en un módulo de una arquitectura de microservicios no requiere un reinicio completo del sistema. Solamente se reinicia el microservicio afectado.
- Los componentes pueden ser más especializados, ya que pueden escribirse en diferentes tecnologías, siempre y cuando estas tecnologías admitan la interacción con las otras tecnologías utilizadas en la misma arquitectura de microservicios.
- Escalar una arquitectura de microservicio no implica una duplicación de todos sus componentes, y los desarrolladores pueden convenientemente desplegar o eliminar instancias de microservicios en función de la carga de trabajo que tengan en un determinado momento temporal.
- El camino para desplegar microservicios son los contenedores [21]. Esto le otorga al desarrollador un alto grado de libertad en cuanto a la selección y configuración del entorno de despliegue en la nube.

5 Marco de trabajo para desarrollar sistemas de programación de la producción

A continuación se presenta el marco de trabajo para desarrollar sistemas de programación de la producción utilizando servicios en la nube (ver Fig. 1). El marco está constituido por un conjunto de servicios integrados que tienen una alta cohesión y un bajo acoplamiento que facilita la construcción de un sistema de scheduling. La idea principal que guía el diseño del marco es el estilo arquitectónico de microservicios. Tal como se describió en apartado anterior este estilo permite concebir un sistema como una colección de servicios pequeños y autónomos. De esta forma cada servicio en el marco es independiente e implementa una funcionalidad bien definida de un sistema de

programación de la producción. Los microservicios se pueden comunicar de dos formas, con mensajes sincrónicos o mensajes asincrónicos. Los mensajes sincrónicos se utilizan cuando los microservicios esperan por una respuesta de otro microservicio para continuar con su tarea. Los mensajes asincrónicos se utilizan en situaciones donde los microservicios clientes no esperan por la respuesta de otro microservicio. El Gestor de Eventos es el encargado de la comunicación.

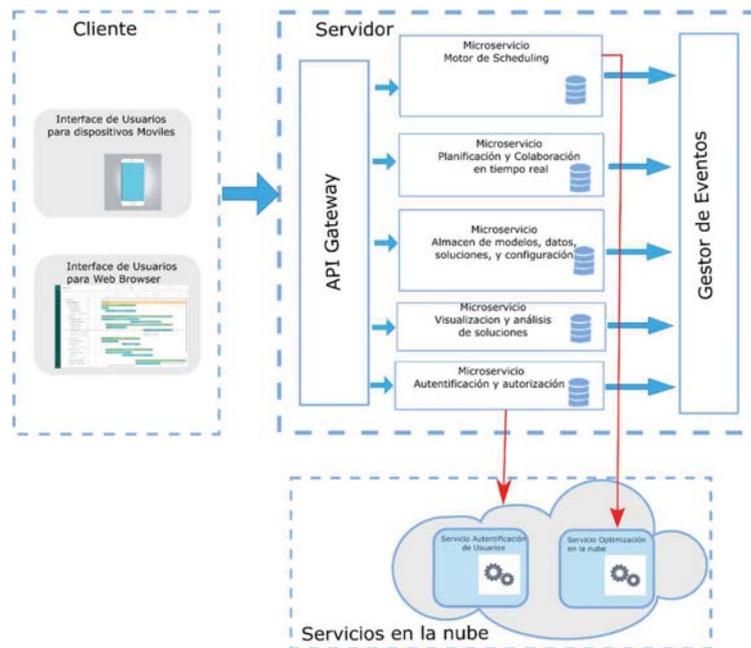


Fig. 1. Arquitectura del marco de trabajo para desarrollar sistemas de programación de la producción.

Los clientes, aplicaciones móviles o para browser, no llaman directamente a los servicios sino que lo hacen a través de una puerta de entrada que es interfaz de programación de aplicaciones, denominada “API Gateway” [22]. Esta puerta de entrada dirige la solicitud del cliente al servicio que atiende esta solicitud, desacoplando de esta manera la solicitud del cliente de los servicios que la atienden, de esta manera varios servicios pueden coordinarse para atender una determinada solicitud. Otra ventaja es que los servicios pueden ser modificados, versionados o refactorizados sin necesidad de modificar nada en el cliente. También permite que los servicios internamente utilicen otros protocolos de comunicación que no sean amigables con la web. Hay una serie de tareas (cross-cutting concerns) que pueden ser llevadas a cabo el API Gateway tales como, registro y descubrimiento de servicios, balanceo de carga, monitoreo, y seguridad, tolerancias a fallo. El *Microservicio Motor de scheduling*, es el microservicio núcleo del sistema. Este se encarga de obtener una solución a un

determinado problema de scheduling. En este contexto un problema de scheduling, en su caso más simple, queda definido por un archivo modelo y un archivo de datos que aplica al modelo. En un dominio dado es posible que para resolver un determinado problema de scheduling se requiera más de un modelo. En el lado del cliente se construyen las diferentes interfaces para el sistema de scheduling. Así, el proceso de planificación comienza con el usuario haciendo la carga de datos de un determinado problema, este proceso tiene lugar en el lado del cliente, ayudado por la interfaz de usuario, el resultado de esta tarea es remitido al servidor. El módulo API Gateway recibe la solicitud para resolver el problema enviado, analiza la solicitud y solicita al servicio *Microservicio Autenticación* las credenciales correspondientes para poder acceder a las API de los microservicios implicados en el proceso de obtener la solución al problema. Con las credenciales el API Gateway determina el tipo de solicitud del cliente y remite los datos al *Microservicio Motor de scheduling*, con los datos remitidos desde el API Gateway determina que archivos debe solicitar al *Microservicio Almacén de modelos, datos, soluciones y configuraciones*. Es posible que en las bases de datos del *Microservicio Almacén* ya exista una solución para el problema en cuestión, en este caso es comunicada y enviada al *Microservicio Motor de Scheduling* el cual envía el archivo de solución al *Microservicio de Visualización y análisis de soluciones* el cual confecciona los datos necesarios para los reportes y la visualización gráfica (diagrama de Gantt) que serán enviados al cliente. Pero si no existe tal solución, el *Microservicio Motor de Scheduling* remite los archivos modelo, datos, configuración, y credenciales al Servicio de Optimización en la Nube, el cual es un servicio externo y normalmente tarifado. Este servicio retorna un archivo con la solución al problema de scheduling, luego el *Microservicio Motor de Scheduling* envía este archivo de solución al *Microservicio de Visualización y análisis de soluciones*, y finalmente este retorna la solución en forma de reportes y gráficos al cliente.

El *Microservicio Autenticación y autorización* es el encargado de gestionar la seguridad del sistema, este autentifica los usuarios del sistema y determina a que parte del sistema están autorizados a acceder.

El *Microservicio planificación y colaboración en tiempo real*, en una arquitectura jerárquica ANSI/ISA 95 tiene por objeto asistir a la planificación a corto plazo, donde el horizonte de los scheduling no va más allá de los 10 días. Son scheduling que deben considerar más restricciones que los de planificación a largo plazo. Normalmente se trata de scheduling multiobjetivo en donde el cumplimiento de algunos objetivos se hace a costa de no cumplir otros y viceversa. Esta situación conduce a conflictos de intereses entre los departamentos involucrados en scheduling. Es aquí donde la colaboración en tiempo real para scheduling interactivo puede asistir a la búsqueda de consensos para establecer en un programa de producción consensuado. El grupo de usuarios involucrados puede ver e interactuar en tiempo real de web proponiendo o aceptando los cambios en el programa de producción. En un enfoque más moderno, es decir planificación y programación en industria 4.0, el CPPS (Cyber-physical production systems) es el encargado de las funciones del MES. En este tipo de contexto la planificación es del tipo Smart Scheduling [15]. Los eventos disruptivos que suceden en tiempo real en el proceso de producción deben ser atendidos por el sistema de scheduling. Para esta situación el *Microservicio de planificación y colaboración en*

tiempo real es el encargado del monitoreo de las actividades de producción en tiempo real y de elevar las peticiones de nuevos programas (re-scheduling) al *Microservicio Motor de Scheduling* con el objeto de cumplir los objetivos del programa de producción original.

6 Prototipando un sistema de scheduling mediante el marco propuesto

Con el objeto de ilustrar el uso del marco propuesto se describe a modo general, y sin entrar en detalle, la construcción de un muy simple sistema de scheduling utilizando el marco. La entrada al sistema de scheduling está compuesta por un conjunto de actividades, sus duraciones y sus precedencias. También se establecen los recursos que son necesarios para ejecutar las actividades, tales como máquinas y operarios. La relación entre actividades y recursos se describe mediante un conjunto de restricciones tales como, una actividad x requiere del recurso y para ejecutarse. Por último, el objetivo del scheduling puede ser de diversa naturaleza, por ejemplo: optimizar el tiempo de entrega, u optimizar el uso de los recursos. Para desarrollar el sistema siguiendo el marco se proponen las siguientes tecnologías: Meteor.js [23] para implementar todos los microservicios. Mientras que los servicios en la nube de optimización y de autenticación son IBM Decision Optimization on Cloud [24] y Auth0 [25] respectivamente. MongoDB [26] como motor de base de datos para todos los Microservicios, y como Gestor de Eventos el mecanismo reactivo publicar / suscribir de Meteor.js. Ambassador como API Gateway [27]. Cada microservicio será desplegado en un contenedor Docker [28] y la gestión de contenedores se realizara con Kubernetes [29]. La plataforma cloud IaaS para despliegue de todo el sistema será digitalOcean [30]. El corazón del sistema es el *Microservicio Motor de scheduling* que es el encargado de transformar los datos que envía la interfaz de usuario (objetivo, actividades, recursos y restricciones) en la entrada que necesita el motor de optimización de IBM, luego, la respuesta enviada por el este motor, sirve para gestionar el trabajo de los demás microservicios. Meteor.js adhiere a la programación web reactiva por lo que resulta de gran ayuda en el desarrollo del *Microservicio de Planificación y Colaboración en tiempo real*. La gestión de la seguridad se realiza mediante tokens de acceso, los cuales brindan acceso a los diferentes microservicios, y son gestionados por la plataforma Auth0. Escalabilidad, balanceo, registro y descubrimiento de servicios y tolerancia a fallos son gestionados por Ambassador API Gateway.

7 Conclusiones

En este artículo se ha presentado un marco de trabajo para desarrollar sistemas de programación de la producción utilizando servicios en la nube. El marco hace uso de un conjunto de servicios de la nube entre en los que se destaca el motor de soluciones IBM Decision Optimization. Está basado en el estilo arquitectónico de microservicios y cuyas características han sido revisadas en el apartado 4.2. En el apartado 6 se ha

esbozado de manera muy general la aplicación del marco propuesto a la construcción de un sistema de scheduling en el contexto de una problemática común del dominio industrial. El objetivo de dicho apartado no ha sido demostrar como el marco puede ayudar a resolver una problemática en concreto, sino demostrar qué tecnologías pueden ser utilizadas en los diferentes módulos que este marco presenta. Una aplicación basada en el marco presentado está pensada para ser desplegada en cualquier plataforma de la nube que presente el modelo de servicios IaaS. Este tipo de plataformas permite aprovechar toda la potencia que brinda la nube en cuanto a escalabilidad, tolerancia a fallos, velocidad de cómputo, y economía de uso. Por último se debe mencionar que la arquitectura del marco ha sido diseñada en función de satisfacer algunas de las necesidades que presenta la industria 4.0, en lo que respecta a programación de la producción.

Como trabajos futuros se propone aplicar el marco de trabajo al desarrollo de sistemas de scheduling en una problemática industria real y concreta. En donde se puedan diseñar e implementar todos los módulos de la arquitectura y realizar diversos experimentos que pongan a prueba las bondades que brinda la nube. Sin duda el conocimiento obtenido de tales experiencias ayudará a mejorar el marco presentado.

Referencias

1. Barker, K.R., *Elements of sequencing and scheduling*. 1974, New York: John Wiley and Sons.
2. Yen, B.P.-C. and M. Pinedo. *On the design and development of scheduling systems*. in *Fourth International Conference on Computer Integrated Manufacturing and Automation Technology*. 1994.
3. Guizzi, G., S. Vespoli, and S. Santini. *On The Architecture Scheduling Problem Of Industry 4.0*. in *CIISE*. 2017.
4. Rossit, D. and F. Tohmé, *Scheduling research contributions to Smart manufacturing*. *Manufacturing Letters*, 2018. **15**: p. 111-114.
5. MacCarthy, B.L. and J. Liu, *Addressing the gap in scheduling research: A review of optimization and heuristic methods in production scheduling*. *International Journal of Production Research*, 1993. **31**: p. 59-79.
6. Henning, G.P., *Towards a software engineering approach to the deployment and implementation of scheduling systems*, in *Computer Aided Chemical Engineering*. 2017, Elsevier. p. 1405-1410.
7. Fowler, M. and J. Lewis, *Microservices a definition of this new architectural term*. URL: <http://martinfowler.com/articles/microservices.html>, 2014: p. 22.
8. Smith, F.S., *OPIS: A Methodology and architecture for reactive scheduling*. 1994: Morgan Kaufmann.
9. Le Pape, C., *Implementation of Resource Constraints in ILOG SCHEDULE: A Library for the Development of Constraint-Based Scheduling Systems*. *Intelligent Systems Engineering* 3, 1994. **3**(2): p. 55-66.

10. Entrup, M.L., *Advanced Planning and Scheduling Systems*. 2005: Advanced Planning in Fresh Food Industries: Integrating Shelf Life into Production Planning. Physica-Verlag HD.
11. Baheti, R. and H. Gill, *Cyber-physical systems*. The impact of control technology, 2011. **12**(1): p. 161-166.
12. Monostori, L., *Cyber-physical production systems: Roots, expectations and R&D challenges*. Procedia Cirp, 2014. **17**: p. 9-13.
13. Hermann, M., T. Pentek, and B. Otto. *Design principles for industrie 4.0 scenarios*. in *2016 49th Hawaii international conference on system sciences (HICSS)*. 2016. IEEE.
14. Leusin, M., et al., *Solving the Job-Shop Scheduling Problem in the Industry 4.0 Era*. Technologies, 2018. **6**(4): p. 107.
15. Rossit, D.A., F. Tohmé, and M. Frutos, *Industry 4.0: Smart Scheduling*. International Journal of Production Research, 2018: p. 1-12.
16. Antonopoulos, N. and L. Gillam, *Cloud computing*. 2010: Springer.
17. Taylor, G., *Definición de Cloud Computing por el NIST*. blog Microsoft Technet, 2010. **accedido 05/08/2019:** p. <https://blogs.technet.microsoft.com/guillermotaylor/2010/08/25/definicion-de-cloud-computing-por-el-nist/>.
18. Castiglione, F.C., *Virtualización en Entornos Cloud*. Tesis de grado 2017. FCEFyN - UNSJ.
19. Dragoni, N., et al., *Microservices: yesterday, today, and tomorrow*, in *Present and ulterior software engineering*. 2017, Springer. p. 195-216.
20. Thönes, J., *Microservices*. IEEE software, 2015. **32**(1): p. 116-116.
21. Kovács, Á. *Comparison of different Linux containers*. in *2017 40th International Conference on Telecommunications and Signal Processing (TSP)*. 2017. IEEE.
22. Montesi, F. and J. Weber, *Circuit breakers, discovery, and API gateways in microservices*. arXiv preprint arXiv:1609.05830, 2016.
23. Meteor, *Meteor documentation*. www.meteor.com, 2019.
24. DOcplexcloud, *Decision Optimization on Cloud*. <https://developer.ibm.com/docloud/documentation/decision-optimization-on-cloud/>, 2019.
25. Auth0, *Auth0 Platform* <https://auth0.com/>, 2019.
26. MongoDB, *Motor de base de datos basados en documentos*. <https://www.mongodb.com/>, 2019.
27. Ambassador, *API Gateway* <https://www.getambassador.io/> 2019.
28. Docker, *Enterprise Container Platform*. <https://www.docker.com/>, 2019.
29. Kubernetes, *Gestor a escala de aplicaciones en contenedores*. <https://kubernetes.io/es/>, 2019.
30. DigitalOcean, *Proveedor de servidores virtuales para nube mediante el modelo de servicio IaaS*. <https://www.digitalocean.com/>, 2019.

HMSBeagleVR: Una experiencia inmersiva a bordo del HMS Beagle

Federico Cristina¹, Sebastián Dapoto¹, Pablo Thomas¹, Patricia Pesado¹,
Rebeca Kraselsky²

¹ Instituto de Investigación en Informática LIDI,
Universidad Nacional de La Plata – Argentina
Centro Asociado Comisión de Investigaciones Científicas de la Provincia de Buenos Aires

² Ministerio de Ciencia, Tecnología e Innovación de la Provincia de Buenos Aires

{fcristina, sdapoto, pthomas, ppesado}@lidi.info.unlp.edu.ar
cienciaenred.mcti@gba.gob.ar

Resumen. En términos generales, las aplicaciones tridimensionales (3D) con Realidad Virtual (RV) requieren hardware con gran capacidad de procesamiento. Debido al avance en el poder de cómputo de los dispositivos móviles, actualmente es posible la ejecución de aplicaciones cada vez más sofisticadas en lo que refiere a complejidad computacional. Esto permite, por ejemplo, el desarrollo de aplicaciones inmersivas con escenarios virtuales tridimensionales que el usuario puede recorrer libremente. Este trabajo presenta el diseño y desarrollo de una aplicación 3D con RV interactiva que permite recorrer el exterior e interior de la embarcación HMS Beagle utilizada por Charles Darwin en su visita a la Argentina, así como los obstáculos y desafíos que implica una implementación con estas características.

Palabras Clave: Realidad Virtual, Dispositivos Móviles, Aplicaciones 3D, Unity

1 Introducción

El Ministerio de Ciencia, Tecnología e Innovación de la provincia de Buenos Aires (MCTI) tiene entre sus funciones ser la entidad de aplicación y ejecución de temas y acciones relativas a las Ciencias y la Tecnología, articulando con los diferentes agentes de la Provincia y propiciando, así, el fortalecimiento de las redes dentro del Sistema Científico Tecnológico y de Innovación Provincial.

Entre sus objetivos está el impulso de programas que aseguren la transferencia de conocimiento orientada a reducir la brecha entre los sectores académicos, productivos y la comunidad. Con este fin, se han realizado diversas actividades a nivel territorial que buscan construir nuevas miradas y visiones innovadoras sobre temáticas relacionadas con la Ciencia y la Provincia.

Ruta Darwin es un proyecto expositivo e itinerante diseñado para ser exhibido en museos de Ciencia, tanto municipales como provinciales dentro del territorio bonaerense. Este proyecto fue inaugurado en julio de 2018 en la ciudad de Tandil como exposición temporal. Esta exposición aborda la figura de Charles Darwin desde una perspectiva histórica, cultural y científica. Tiene como objetivo reconocer su legado y el impacto en sus contemporáneos y destacar el paso del naturalista por la provincia de Buenos Aires.

Charles Darwin forma parte de la historia de la Ciencia a nivel internacional, pero además dejó una importante huella en la historia científica de la provincia de Buenos Aires. A los 22 años, llegó a la Argentina en el viaje del barco HMS Beagle comandado por Robert Fitz Roy, expedición que dio la vuelta al mundo y cuyo objetivo era hacer un relevamiento de costas. Visitó la provincia entre 1832 y 1833. Darwin recorrió el trayecto entre Carmen de Patagones, el río Colorado y Bahía Blanca, y luego el trayecto desde Bahía Blanca a Buenos Aires, para seguir más tarde hacia Santa Fe. En su “*Diario de un naturalista alrededor del mundo*”, escrito poco después de su regreso a Inglaterra, el científico menciona su paso por Sierra de la Ventana, describe la Ciudad de Buenos Aires, Bahía Blanca y Punta Alta, entre muchas otras localidades.

Ruta Darwin subraya cómo su estadía en la Provincia fue clave para la elaboración de la posterior teoría de la evolución. De este modo, la muestra inicia con el viaje, el mapa del recorrido, los instrumentos utilizados, los libros leídos y escritos por el científico a fin de “entrar en su mundo”. Continúa con las visitas que realizó en la Provincia, su actividad como geólogo, los hallazgos fósiles fundamentales en la elaboración de su pensamiento y sus apreciaciones sobre la fauna y la flora.

La propuesta *Ruta Darwin* se ambienta con imágenes impresas que ilustraron los libros editados luego de su expedición y exhibe una línea de tiempo que contextualiza su producción entre la de sus contemporáneos. El recorrido invita a descubrir los conceptos de Evolución y Selección Natural.

2 Motivación

En el afán de darle a la muestra *Ruta Darwin* una nueva forma de comprender los conceptos explorados en ella, surge la idea de incorporar alguna herramienta que permita aumentar la interacción de contenidos distantes en el tiempo de manera en que los visitantes, la mayoría niños y adolescentes en edad escolar, tuvieran un rol activo en la elaboración de preguntas, conclusiones parciales y reflexiones partiendo del recorrido y textos diversos de la muestra.

Así es que surge la propuesta de desarrollar una aplicación de software móvil inmersiva que permita conocer los espacios habitados por Darwin, sus objetos y los lugares cotidianos desde los cuales vivió su aventura.

En la actualidad los dispositivos móviles son equipos sofisticados que permiten ejecutar aplicaciones con exigentes requerimientos de hardware. Debido a esto, es posible el desarrollo de aplicaciones con entornos tridimensionales (3D) que presentan un costo computacional considerable.

En particular, cada vez existen más aplicaciones 3D que incorporan funcionalidad de Realidad Virtual (RV). La RV es la simulación de un entorno tridimensional real o imaginario que proporciona una experiencia visualmente interactiva en tiempo real y posibilita alguna forma de retroalimentación por parte del usuario. La RV también se puede definir como un entorno artificial creado mediante hardware y software de computadora y presentado al usuario de tal manera que se siente como un entorno real [1].

Otro recurso útil en las aplicaciones 3D con RV es la Virtualidad Aumentada (VA). La VA se puede definir como la combinación de un entorno virtual con elementos de la vida real. En las aplicaciones que incorporan la VA el concepto de inmersión es importante. El usuario navega un mundo sintético interactuando con objetos ficticios o reales [2] [3].

Existen diferentes dispositivos de hardware que permiten llevar a cabo una experiencia inmersiva mediante realidad virtual [1]. Las características de los dispositivos móviles, tales como su tamaño reducido, su poder de cómputo, su disponibilidad de sensores y su costo accesible, hacen de éstos una herramienta adecuada para este fin. El uso de gafas o lentes de RV junto a un dispositivo móvil son suficientes para lograr una experiencia inmersiva utilizando una aplicación 3D con funcionalidad de RV.

Este trabajo presenta el desarrollo de una aplicación móvil 3D inmersiva que permite recorrer la cubierta del barco HMS Beagle y la cabina en la que Darwin viajó durante cinco años. El resto del trabajo se organiza del siguiente modo: la sección 3 presenta en detalle la metodología de desarrollo; la sección 4 exhibe los resultados obtenidos; finalmente en la sección 5 se exponen las conclusiones y trabajos futuros.

3 Metodología de desarrollo

El framework de desarrollo utilizado para la creación de la aplicación interactiva fue Unity[4]. Unity es uno de los frameworks más populares y sencillos de utilizar. Además, cuenta con una documentación muy completa, una gran cantidad de tutoriales, un extenso repositorio de componentes, y una comunidad de más de dos millones de usuarios. Todo esto lo convierte en una de las mejores alternativas al momento de elegir un framework de desarrollo de software 3D.

Unity soporta la creación de objetos tridimensionales básicos a fin de construir un ambiente 3D. Sin embargo, la complejidad del modelo del barco requirió de una herramienta de modelado 3D específica, utilizándose en este caso Blender [5]. Blender es uno de los softwares de modelado y creación de gráficos tridimensionales más utilizados. Permite crear objetos y escenas configurando los materiales y la iluminación utilizados, entre otras características. Además, permite realizar simulaciones, renderizados y animaciones.

El diseño del modelo se basó en dibujos de la época, copias de los planos de la embarcación [6], así como de fotografías y videos obtenidos en documentales [7] y/o reconstrucciones actuales [8].

El modelado fue dividido en dos escenas diferentes, una exterior y otra interior. La escena exterior presentada en la figura 1 abarca toda la cubierta superior de la

embarcación, incluyendo mástiles, velas, cañones, botes de expedición entre otros. La escena interior, tal como se observa en la figura 2, contempla específicamente el camarote de Darwin, en donde tenía su herbario, biblioteca y realizaba además sus investigaciones.

La división del modelo en dos escenas distintas se fundamenta principalmente en la necesidad de poder ejecutar de manera fluida la aplicación. Esto se encuentra en gran medida restringido al número de polígonos del modelo que deben ser renderizados en tiempo real [9].

Para lograr reducir drásticamente el número de polígonos a procesar, se separó el modelo exterior e interior en dos escenas. Este aspecto incluso se vuelve más crítico cuando la aplicación debe ejecutarse en dispositivos móviles, dado que la capacidad de cómputo es menor que en las computadoras de escritorio.



Fig. 1. Modelo realizado en Blender de la embarcación HMS Beagle. Vista exterior y desde la cubierta.

Otro aspecto a considerar es la detección de colisiones, a fin de que el usuario realice el recorrido limitándose a los sectores habilitados de la embarcación y sin atravesar partes del modelo, tales como paredes.

La aproximación más sencilla es definir la colisión a nivel de modelo, esto es: cada polígono del modelo disparará una colisión. Sin embargo, el número de polígonos del

modelo generará una cantidad muy elevada de detectores, comprometiendo la fluidez de la aplicación.

La solución adoptada fue definir un conjunto de objetos de estructura sencilla para la definición de los límites de colisión, donde puede utilizarse un número reducido de hexaedros que comprenden - entre otros - el área del piso, las paredes y los objetos principales. Bajo esta implementación el número de polígonos involucrados en la detección de colisiones es considerablemente menor, logrando mantener así la fluidez deseada en la ejecución.

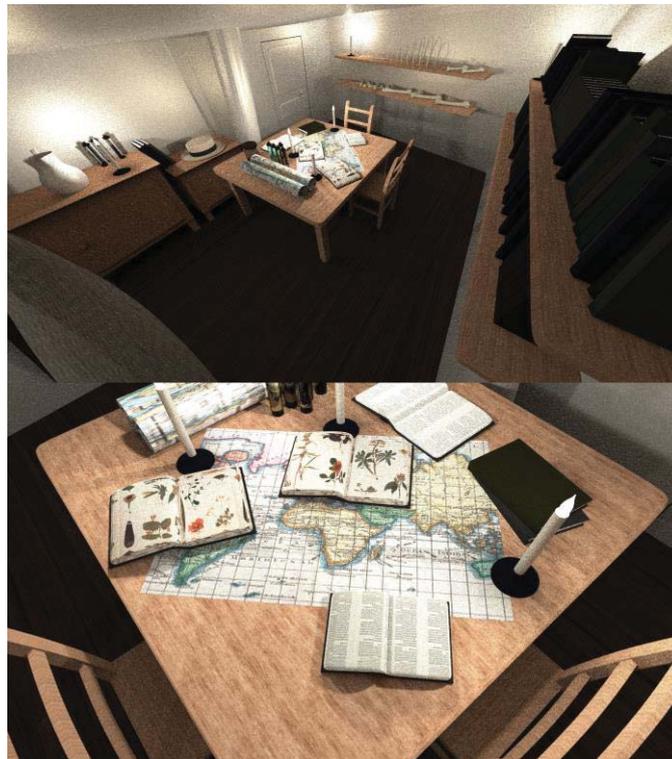


Fig. 2. Modelo realizado en Blender de la embarcación HMS Beagle. Camarote utilizado por Charles Darwin.

El proceso de importación en Unity del modelo realizado en Blender no presentó inconvenientes de consideración. Un punto a resolver fue la adecuación de ciertos objetos del modelo, en particular la necesidad de adecuar los vectores normales de estos objetos a fin de que puedan ser visualizados correctamente en Unity desde cualquier punto de vista.

4 Resultados obtenidos

Luego del modelado en Blender y desarrollo en Unity, se generaron dos aplicaciones 3D distintas, las cuales tienen usos diferentes.

La primera es una aplicación para equipos de escritorio o notebooks, la cual permite realizar de manera autónoma un recorrido virtual por la embarcación. No se requiere intervención alguna por parte del usuario. Dicha aplicación puede ser ejecutada de manera constante a fin de presentar un recorrido general de la embarcación HMS Beagle.

La segunda es la aplicación inmersiva con realidad virtual para dispositivos móviles, la cual requiere el uso de gafas de RV específicas para el caso. Las gafas a utilizar no son de alto costo dado que simplemente utilizan al dispositivo móvil como pantalla y sólo contienen las lentes necesarias para ajustar la visión estereoscópica. La figura 3 presenta el modelo de gafas de RV utilizado en el proyecto.



Fig. 3. Ejemplo de lentes para RV que utilizan un dispositivo móvil como pantalla.

A diferencia de la aplicación de escritorio, la aplicación móvil con RV es totalmente interactiva e inmersiva. Es posible desplazarse por la cubierta de la embarcación, así como recorrer el camarote usado por Charles Darwin. En todos los casos la visualización es de 360°, esto es: puede observarse el entorno desde cualquier punto de vista y hacia cualquier otra posición dentro de los límites de la escena. Esto puede observarse en la figura 4. Además, los objetos virtuales dentro del camarote son aumentados mediante carteles que presentan información relacionada con éstos. Por ejemplo, al observar directamente al herbario o a los restos fósiles, se visualiza información asociada perteneciente a las investigaciones realizadas por Charles Darwin sobre estos elementos.

Una experiencia piloto del uso de la aplicación fue llevada a cabo en el Museo de Charles Darwin. La experiencia abarcó tanto niños como adultos, y el impacto de su uso fue ampliamente favorable, destacándose la sensación de inmersión al utilizar la aplicación móvil. En la figura 5 se observa la experiencia piloto.

El proyecto *Ruta Darwin* cobró así una nueva dimensión, buscando que el público pudiera ser parte de la experiencia mediante la inmersión. El visitante ya no es sólo un espectador, puede dirigir su búsqueda de contenidos tomando sus propias decisiones.

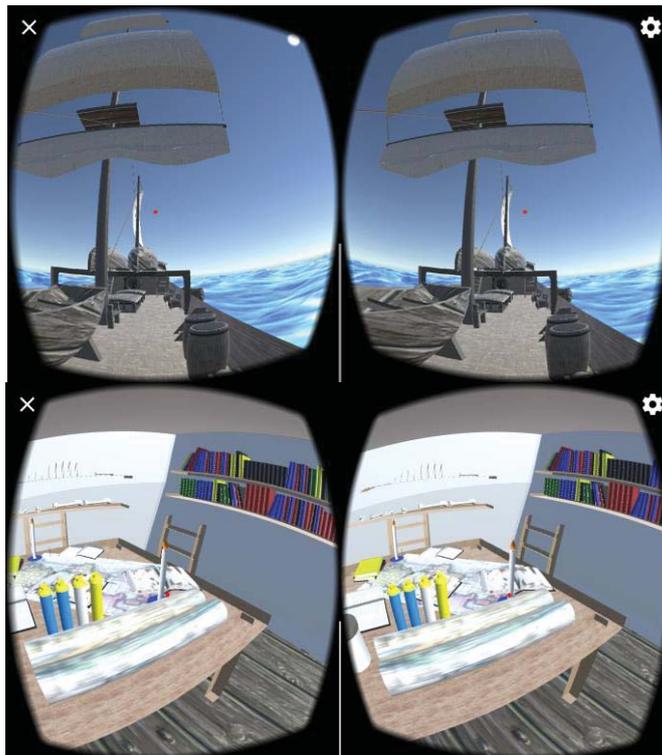


Fig. 4. La aplicación HMSBeagleVR en ejecución. Arriba: cubierta de la embarcación. Abajo: camarote usado por Charles Darwin.



Fig. 5. Experiencia piloto llevada a cabo en el Museo de Charles Darwin.

5 Conclusiones y trabajo a futuro

El presente trabajo brinda los detalles de diseño y desarrollo de dos aplicaciones 3D con realidad virtual. Ambas tienen como finalidad ayudar a conocer en detalle la embarcación HMS Beagle, la cual fue utilizada por Charles Darwin en su viaje a la Argentina entre los años 1832 y 1833.

La aplicación de escritorio permite visualizar un recorrido virtual general por la embarcación sin intervención del usuario.

La aplicación móvil brinda una experiencia inmersiva a lo largo de la embarcación y el camarote utilizado por Darwin, permitiendo su recorrido mediante gafas de realidad virtual e incluso pudiendo interactuar con ciertos objetos de la escena, a fin de obtener información detallada sobre estos elementos.

La nueva forma de interacción mediante la aplicación móvil 3D desarrollada permitió evolucionar al proyecto *Ruta Darwin*, dando un nuevo ritmo al recorrido de la visita en la que el conocimiento se convierte en vivencia personal y a la vez socialmente compartida.

A futuro se planea extender la interactividad de la escena a un número mayor de objetos, así como expandir la navegabilidad hacia otras zonas de la embarcación a fin de enriquecer la experiencia.

Referencias

1. Moses Okechukwu Onyesolu and Felista Udoka Eze. "Understanding Virtual Reality Technology: Advances and Applications". 2019.
2. Stefaan Ternier et al. "ARLearn: Augmented Reality Meets Augmented Virtuality". 2013.
3. Antonis Karakottas et al. "Augmented VR". Electronic ISBN: 978-1-5386-3365-6. 2018.
4. Unity. <https://unity3d.com>
5. Blender. <https://www.blender.org/>
6. Darwin Online. <http://darwin-online.org.uk/>
7. El joven Darwin: el viaje de un naturalista alrededor del mundo. <https://www.youtube.com/watch?v=SXOHVaq7Yj8&t=60s>
8. Building the HMS Replica. <http://construyendoelhmsbeagle.blogspot.com/>
9. Cristina, F.; Dapoto, S.; Thomas, P.; Pesado, P. "Performance evaluation of a 3D engine for mobile devices". Computer Science – CACIC 2017. Communications in Computer and Information Science, vol 790. A. De Giusti, Springer International Publishing. ISBN: 978-3-319-75213-6, 978-3-319-75214-3, pages 155-163. 2018.

Modelo de evaluación de Tecnologías de la Información y la Comunicación para la industria 4.0

Alicia Mon, Horacio René Del Giorgio

Universidad Nacional de La Matanza – Departamento de Ingeniería e Investigaciones Tecnológicas - Florencio Varela 1903 (B1754JEC) - San Justo, Buenos Aires, Argentina
alicialmon@gmail.com , horacio.delgiorgio@gmail.com

Abstract

El desarrollo de las TICs (Tecnologías de la Información y las Comunicaciones) permite vincular el mundo físico a través de dispositivos, materiales, productos, equipos, instalaciones y comunicaciones, con el mundo digital, expresado a través de sistemas colaborativos y productos software interconectados con infinidad de dispositivos para potenciar el desarrollo de la Industria 4.0.

El impacto generado por las TICs se manifiesta en los sistemas de producción, especialmente de la mano de la inteligencia artificial, la robótica y conectividad de los objetos a través de comunicaciones inalámbricas. Las diferentes partes del proceso productivo no sólo adoptan funciones inteligentes, sino que se comunican automáticamente y en forma autónoma entre ellas, donde la gestión del conocimiento forma parte de los sistemas de producción.

Para poder detectar cuáles son los aspectos centrales de la gestión del conocimiento, el presente artículo expone un modelo creado por el grupo de investigación GIS, que permite evaluar el nivel de avance tecnológico que la industria tiene implantado en la actualidad, distinguiendo por rama de actividad.

La evaluación de las tecnologías permitiría detectar las necesidades de desarrollo, implementación e innovación de software en las cadenas de valor.

Keywords: TICs, Índice de TICs, Desarrollo Tecnológico, Industria 4.0.

1 Introducción

En la actualidad, se está generando una revolución tecnológica en el campo de la industria, que se reconoce como la cuarta revolución industrial [1]. Una serie de avances en diversos campos como la robótica, la biotecnología, la genética, la nanotecnología, la internet de las cosas, el desarrollo de la inteligencia artificial, la realidad virtual, la realidad aumentada y la fabricación aditiva a través de impresiones 3D se suman a la revolución energética y el desarrollo de las TICs que caracterizaron a la tercera revolución industrial.

Este conjunto de avances científicos y tecnológicos toman la forma de innovación y se manifiestan no sólo en la vida cotidiana de las personas, sino también en los procesos que se desarrollan en la actividad económica, tanto en la producción industrial de bienes, así como en la prestación de servicios. Dentro de esos grandes campos, la digitalización ha cobrado un rol protagónico y en algunos sectores productivos y de servicios se ha tornado imprescindible. En este contexto cobra fuerza el término “Industria 4.0”, que refiere específicamente a la cuarta revolución industrial e implica un salto cualitativo significativo en la organización y gestión de las cadenas de valor.

El desarrollo de las TICs permite vincular el mundo físico y el digital a través de dispositivos, materiales, productos, equipos, instalaciones y comunicaciones, con el mundo digital, expresado a través de sistemas colaborativos y productos software interconectados con infinidad de dispositivos para potenciar el desarrollo de la Industria 4.0, también conocida como Industria Inteligente.

El impacto generado por las TICs se manifiesta principalmente en los sistemas de producción, especialmente de la mano de la inteligencia artificial, la robótica y las comunicaciones inalámbricas. Las diferentes partes del proceso productivo no sólo adoptan funciones inteligentes, sino que se comunican entre ellas mediante la Internet de las Cosas [2], donde la gestión del conocimiento forma parte de los sistemas de producción [3] [4]. Se espera que, en un futuro cercano, todos los sistemas de producción en la Industria 4.0 tengan interconectados los subsistemas constituyentes, los procesos, los objetos (tanto internos como externos) que intervengan, los proveedores, las redes de clientes y los canales de distribución. Todo será controlado en tiempo real.

Las plantas de las fábricas del futuro tendrán claramente definidos estos estándares y compartirán las interfaces establecidas. La conectividad colaborativa será la clave del éxito. El uso de estas tecnologías hará posible reemplazar de manera flexible las máquinas que se reparan o se mejoran de prestaciones a lo largo de la cadena de valor. La adaptación a los cambios del mercado y la productividad serán los grandes beneficiarios.

La Industria 4.0 representa la integración de extremo a extremo de la cadena de valor que va desde los cambios de demandas del gran público al logro de su satisfacción por parte de las fábricas inteligentes. Ya no tendrá sentido hablar de simples fábricas. Las fábricas serán inteligentes (Smart Factories) y llegará el día en que no tendrá sostenibilidad una fábrica que no se haya adaptado a la cuarta generación.

Frente a esta gran transformación, la Industria actual necesita cambios tecnológicos dado que la competitividad de las empresas pasa por la globalización, la productividad, la innovación y la incorporación de tecnología como pilar del desarrollo.

En este contexto se ha trabajado sobre un instrumento de medición que permita evaluar las características específicas de los productos software, hardware y de comunicaciones que definen con precisión los atributos de las Industrias 4.0. Si bien, las nuevas herramientas, tecnologías, materiales, metodologías, las fuentes de energía y todos los factores que se encuadran bajo el nombre de Industria 4.0 constituyen las palancas imprescindibles para alcanzarla, no se ha encontrado en la bibliografía actual, el análisis en que impactan directamente sobre los niveles de productividad cada uno de estos elementos o de la interacción entre los mismos.

Para ello, a continuación se expone un modelo de evaluación de TICs desarrollado por el grupo de investigación GIS que permite establecer con precisión las tecnologías presentes en las diferentes industrias.

2 Estructura del modelo

El modelo propuesto se estructura a partir de la detección de productos tecnológicos, diferenciados en los 3 componentes de TICs: Software, Hardware e Infraestructura.

TICS	SOFTWARE
	HARDWARE
	INFRAESTRUCTURA

TABLA 1 – Taxonomía para TICs

Fuente: Elaboración Propia.

Estos 3 componentes, se han identificado a su vez a través de una distinción que agrupa los productos según las especificidades de cada tipo de tecnología. A partir de tal agrupamiento, se identifican los productos específicos para cada categoría de Software, Hardware e Infraestructura, evaluando su aporte en base al mayor nivel de desarrollo tecnológico que contenga y al valor que agregue a la productividad.

Esta tipología define con precisión cada uno de los productos TICs que pueden encontrarse implementados en la industria, y a su vez, el modelo propone un cruzamiento con las áreas funcionales al interior de las industrias, donde estas tecnologías cumplen las diferentes funciones en las áreas donde se encuentren implementadas.

El Software (relativo a lo blando) es la parte intangible o lógica de la computadora. En general referencia a los programas, los sistemas de información, las aplicaciones, los simuladores y los sistemas operativos, entre otras opciones.

En cambio, el Hardware (relativo a lo duro, y opuesto al Software) referencia a la parte física de una computadora. Muchas veces se lo menciona como todo aquello que pueda ser tocado, como ser: teclado, mouse, monitor, impresora, cables, tarjetas electrónicas, disco duro, memorias, entre otras opciones.

Lo relativo a la Infraestructura, se define como el conjunto de hardware y software sobre el que se asientan los diferentes productos y servicios que el Hardware y Software necesita tener en funcionamiento, para poder llevar a cabo toda su actividad. La Infraestructura consta de elementos diversos como los sensores, las cámaras, los servidores de aplicaciones, los elementos de red, como Routers o Firewalls, entre otros.

Cada grupo de TICs se analiza a partir de una diferenciación de tipos y su aplicación concreta, ya sea para toda la organización o para alguna de las áreas en particular.

2.1 Software

Para el caso de Productos Software, el modelo identifica los elementos expuestos en la siguiente tabla:

SOFTWARE	TECNOLOGÍAS WEB	PÁGINA WEB (SITIO EXTERNO)
		INTRANET (SITIO INTERNO)
		EXTRANET (TRANSACCIONAL)
		PUBLICIDAD ON LINE
	SISTEMAS COLABORATIVOS	VIDEO CONFERENCIA
		TELEFONÍA IP
		MENSAJERÍA INSTANTÁNEA
		EMAIL
		REDES SOCIALES
		SINCRONIZACIÓN DE ARCHIVOS
	HERRAMIENTAS DE OFICINA	APLICACIONES MÓVILES
		PROCESADOR DE TEXTO
		HOJA DE CÁLCULO
		PRESENTACIONES
		GESTOR DE BASE DE DATOS
		GESTOR DE AGENDA Y CORREO ELECTRÓNICO
		GESTOR DE ARCHIVOS PDF
		LECTOR DE ARCHIVOS PDF
	SISTEMAS DE GESTIÓN	SISTEMA DE GESTIÓN INTEGRAL (ERP)
		SISTEMA DE RELACIÓN CON CLIENTES (CRM)
		ATENCIÓN DE RECLAMOS
		TABLERO DE CONTROL
		BUSINESS INTELLIGENCE
		HERRAMIENTAS DE BIG DATA
		MACHINE LEARNING
		SOFTWARE DE CONTROL ENERGÉTICO
		LOGÍSTICA / ABASTECIMIENTO
		SISTEMA DE GESTIÓN DE LA CALIDAD (SGC)
GESTIÓN DE RRHH		
SISTEMAS DE CONTROL DE LA PRODUCCIÓN	PROGRAMACIÓN DE PRODUCCIÓN (MRP)	
	INFORMACIÓN DE PRODUCTO (PDM)	
	GESTIÓN DE CALIDAD DE PRODUCTO	
	INGENIERÍA DE PLANTA / MANTENIMIENTO	
	SISTEMAS DE CONTROL DE AUTOMATIZACIÓN	
	SISTEMAS SCADA	
SISTEMAS DE DISEÑO DE PRODUCTO Y PROCESOS	SISTEMAS EMBEBIDOS	
	DISEÑO ASISTIDO POR COMPUTADORA	
	FABRICACIÓN ASISTIDA POR COMPUTADORA	
	INGENIERÍA ASISTIDA POR COMPUTADORA	
SISTEMAS DE GEOLOCALIZACIÓN	SOFTWARE DE REALIDAD AUMENTADA	
	SOFTWARE DE REALIDAD VIRTUAL	
	DISTRIBUCIÓN Y LOGÍSTICA	
SISTEMAS DE SEGURIDAD	PUBLICIDAD	
	SEGURIDAD DE INFRAESTRUCTURA CRÍTICA	
	SEGURIDAD DE INFORMACIÓN CRÍTICA	

TABLA 2 – Taxonomía para TICs: Software
Fuente: Elaboración Propia.

2.2 Hardware

Para el caso de Productos Hardware, el modelo identifica los elementos expuestos en la siguiente tabla:

HARDWARE	COMPUTADORAS	PCs DE ESCRITORIO
		ARQUITECTURAS RISC
		NOTEBOOKS
		TABLETS
	IMPRESORAS	IMPRESORAS LÁSER
		IMPRESORAS 3D
		SCANNERS
		PLOTTERS
	POS	
	DISCOS COMPARTIDOS	
	PLCs PARA CONTROL NUMÉRICO	
	EQUIPOS GPS	
	EQUIPOS DE RFID	
CENTRALES TELEFÓNICAS	CENTRALES TELEFÓNICAS TRADICIONALES	
	CENTRALES TELEFÓNICAS IP	
SENSORES		

TABLA 3 – Taxonomía para TICs: Hardware
Fuente: Elaboración Propia.

2.3 Infraestructura

Para el caso de la infraestructura, el modelo identifica los elementos expuestos en la siguiente tabla:

INFRAESTRUCTURA	REDES CONVERGENTES INALÁMBRICAS	TELEFONÍA CELULAR
		REDES WI-FI
		REDES BLUETOOTH
		REDES PARA INTERNET DE LAS COSAS (IoT)
	SERVIDORES LOCALES	
	CLOUD COMPUTING	
	REDES DE ÁREA LOCAL CABLEADAS	
	SEGURIDAD INFORMÁTICA	
	CONEXIÓN A INTERNET	
	CIRCUITO CERRADO DE TELEVISIÓN (CCTV)	
	EQUIPOS DE COMUNICACIONES PARA INTERNET DE LAS COSAS	

TABLA 4 – Taxonomía para TICs: Infraestructura
Fuente: Elaboración Propia.

El análisis detallado de cada una de las TICs se encuentra disponible en [5] y [6].

Por otra parte, se ha confeccionado una tipificación de los procesos industriales definiendo las funciones de una empresa tomando como base la Cadena de Valor de Michael Porter [7]. El modelo propuesto permite analizar el desempeño de una empresa organizando el análisis en relación con el conjunto de actividades primarias y de apoyo, siendo cada una de éstas fuente potencial de ventajas competitivas en costos o diferenciación, y cuyas interrelaciones permiten lograr un mayor valor diferencial emergente que pueda ser apreciado y reconocido por los compradores, en comparación con otras ofertas de la competencia.

De acuerdo a este modelo, el conjunto de funciones básicas se ha estructurado tal como se expone en la siguiente figura.



FIGURA 1 – Modelo simplificado para la Cadena de Valor
Fuente: Elaboración Propia.

El modelo desarrollado propone un cruce entre las dos tipologías identificadas (TICs y Procesos Productivos) a los fines de poder detectar con precisión los productos específicos que se utilizan para el correcto desempeño de las funciones en las diferentes áreas. Una vez identificados estos productos tecnológicos, el modelo propone una valoración por niveles según su grado de desarrollo en cuanto al tiempo que existen como herramientas utilizadas en el mercado, si el tipo de soporte que brindan aporta información sensible a las empresas, sobre cuál es la complejidad del problema que resuelven, si su utilización impacta en una mejora de los procesos o sobre el control de los procesos, si mejora la eficiencia en la utilización de recursos, si mejora la productividad en los procesos, si reduce costos operativos, así como el grado de innovación que genera su implementación y aplicación en el campo de la industria.

De este modo, se establecen 3 niveles de actualidad de los productos, según resulten de tecnología básica, tecnología de actualidad media o tecnología más avanzada, tendiente a la transformación de la industria 4.0. Esta calificación se ha instrumentado a partir de una tabla de doble entrada, donde se cruzan los productos específicos correspondientes a cada agrupamiento de TICs, en los tipos de Software, Hardware o Infraestructura con las áreas funcionales, y se le asigna una ponderación en cada cruce según el nivel de actualidad de cada producto identificado. Para ello, se utilizó una escala con los siguientes valores:

- Básica: con valor 1
- Media: con valor 2
- Avanzada: con valor 3

La siguiente tabla expone la Tabla Matriz del modelo, que define los cruces para determinar la relación entre las TICs y los Procesos Industriales e indica la valoración definida para cada producto específico. Las filas presentan a cada producto TICs agrupadas por tipo y coloreadas según la clasificación dentro de cada agrupamiento. Por ejemplo, para el caso de Software, serían “Tecnologías WEB”, “Sistemas Colaborativos”, “Herramientas de Oficina”, entre otros.

TICs↓		PROCESOS PRODUCTIVOS →						
		Dirección	Contabilidad y Finanzas	Ingeniería	Compras	Logística	Producción	Ventas
SOFTWARE	Tecnologías WEB - Página WEB (Sitio externo)	1						1
	Tecnologías WEB - Intranet (Sitio interno)	1	1	1	1	1	1	1
	Tecnologías WEB - Extranet (Transaccional)				2	2		2
	Tecnologías WEB - Publicidad online	2						2
	Sistemas Colaborativos - Video conferencia	3		3				3
	Sistemas Colaborativos - Telefonía IP	2	2	2	2	2	2	2
	Sistemas Colaborativos - Mensajería Instantánea	1	1	1	1	1	1	1
	Sistemas Colaborativos - Email	1	1	1	1	1	1	1
	Sistemas Colaborativos - Redes sociales	1						1
	Sistemas Colaborativos - Sincronización de archivos	2	2	2		2	2	2
	Sistemas Colaborativos - Aplicaciones móviles	2	2	2	2	2	2	2
	Herramientas de Oficina - Procesador de texto	1	1	1	1	1	1	1
	Herramientas de Oficina - Hoja de cálculo	1	1	1	1	1	1	1
	Herramientas de Oficina - Presentaciones	1	1	1				1
	Herramientas de Oficina - Gestor de Base de Datos		2	2	2	2	2	2
	Herramientas de Oficina - Gestor de Agenda y Correo Electrónico	1	1	1	1	1	1	1
	Herramientas de Oficina - Gestor de Archivos PDF	2	2	2				2
	Herramientas de Oficina - Lector de Archivos PDF	1	1	1	1	1	1	1
	Sistemas de Gestión - Sistema de Gestión Integral (ERP)	2	2	2	2	2	2	2
	Sistemas de Gestión - Sistema de Relación con Clientes (CRM)	2	2					2
	Sistemas de Gestión - Atención de Reclamos							2
	Sistemas de Gestión - Tablero de Control / Balanced Score Card	3						3
	Sistemas de Gestión - Business Intelligence (Cubos, Data Warehouse)	3	3	3	3	3	3	3
	Sistemas de Gestión - Herramientas de Big Data	3	3	3	3	3	3	3
	Sistemas de Gestión - Machine Learning	3	3					3
	Sistemas de Gestión - Software de Control Energético			3				3
	Sistemas de Gestión - Logística/Abastecimiento			2	2	2	2	2
	Sistema de Gestión - Sistema de Calidad (SGC)	2	2	2	2	2	2	2
	Sistemas de Gestión - Gestión de RRHH	2	2					
	Sistemas de Control de la Producción - Programación de Producción (MRP)			3				3
	Sistemas de Control de la Producción - Información de Producto (PDM)			3				3
	Sistemas de Control de la Producción - Gestión de Calidad de Producto							2
	Sistemas de Control de la Producción - Ingeniería de planta/mantenimiento			2				2
	Sistemas de Control de la Producción - Sistemas de Control de Automatización			3				3
	Sistemas de Control de la Producción - Sistemas SCADA			3				3
	Sistemas de Control de la Producción - Sistemas Embebidos			3				3
	Sistemas de Diseño de Producto y Procesos - Diseño Asistido por Computadora (CAD)			2				
	Sistemas de Diseño de Producto y Procesos - Fabricación Asistida por Computadora (CAM)							3
	Sistemas de Diseño de Producto y Procesos - Ingeniería Asistida por Computadora (CAE)			3				
	Sistemas de Diseño de Producto y Procesos - Software de Realidad Aumentada			3				3
	Sistemas de Diseño de Producto y Procesos - Software de Realidad Virtual			3				3
	Sistemas de Geolocalización - Distribución y Logística					2		
	Sistemas de Geolocalización - Publicidad							3
	Sistemas de Seguridad - Seguridad de Infraestructura Crítica	2	2	2	2	2	2	2
	Sistemas de Seguridad - Seguridad de Información Crítica	2	2	2	2	2	2	2
HARDWARE	Computadoras - PCs de escritorio	1	1	1	1	1	1	1
	Computadoras - Arquitecturas RISC			2				2
	Computadoras - Notebooks	1	1	1	1	1	1	1
	Computadoras - Tablets	1	1	1	1	1	1	1
	Impresoras - Impresoras láser	1	1	1	1	1	1	1
	Impresoras - Impresoras 3D			3				3
	Impresoras - Scanners	2	2	2	2	2	2	2
	Impresoras - Plotters			3				3
	POS							2
	Discos Compartidos	3	3	3	3	3	3	3
	PLCs para Control Numérico							2
	Equipos GPS					2		
	Equipos de RFID					2		
	Centrales Telefónicas - Centrales Telefónicas tradicionales	1	1	1	1	1	1	1
	Centrales Telefónicas - Centrales Telefónicas IP	3	3	3	3	3	3	3
Sensores			3				3	
INFRAESTRUCTURA	Redes Convergentes Inalámbricas - Telefonía Celular	1	1	1	1	1	1	1
	Redes Convergentes Inalámbricas - Redes Wi-Fi	1	1	1	1	1	1	1
	Redes Convergentes Inalámbricas - Redes Bluetooth	2	2	2	2	2	2	2
	Redes Convergentes Inalámbricas - Redes para Internet de las Cosas (IoT)			3		3		3
	Servidores Locales	2	2	2	2	2	2	2
	Cloud Computing	2	2	2	2	2	2	2
	Redes de Área Local cableadas	1	1	1	1	1	1	1
	Seguridad Informática	2	2	2	2	2	2	2
	Conexión a Internet	1	1	1	1	1	1	1
	Circuito Cerrado de Televisión (CCTV)							1
Equipos de Comunicaciones para Internet de las Cosas			3		3		3	

TABLA 5 – Tabla Matriz
Fuente: Elaboración Propia.

De acuerdo con esta asignación de valor a cada tecnología, la forma propuesta de evaluación es considerando que cada tecnología que se incorpora incluye a las anteriores, no reemplaza una por otra. Es decir que, cada vez que se incorpora tecnología nueva se agrega algo más avanzado; por lo tanto, el resultado que se alcanza en esta instancia responde a la sumatoria de contribuciones de los valores.

De la Tabla Matriz se derivan los resultados de la sumatoria de la totalidad de productos según su nivel, donde cada valoración (Básico, Medio, Avanzado) contiene una cantidad de casilleros que se representan en la columna de Cantidad total de valores. La sumatoria de esos valores indica el total posible incluyendo a todos los productos TICs contenidos en cada uno de los niveles, tal como se expresa en la columna de Sumatoria de los valores:

Valoración	Cantidad total de valores	Sumatoria de los valores
Básico (1)	120	120
Medio (2)	119	238
Avanzado (3)	67	201
Totales	306	559

TABLA 6 – Cantidad de Totales y Sumatorias de Valores de cada Valoración
Fuente: Elaboración Propia.

Para aplicar el modelo desarrollado, se han creado un conjunto de instrumentos de relevamiento, que permite identificar en cada industria en particular cuáles son los productos específicos que tiene implementados para cumplir las funciones específicas al interior de cada empresa.

Entre el conjunto de instrumentos, se ha diseñado una encuesta que contiene un formulario con los casilleros vacíos. Al realizar el relevamiento puntual en cada industria en particular se marca con una cruz los casilleros en blanco con cada producto que la misma utiliza en un determinado proceso industrial. En los casos en que no exista coincidencia, ese casillero quedará en blanco. A continuación, con ese registro se asignan las valoraciones para la empresa y se realizan los siguientes cálculos:

Se obtiene la sumatoria de todos los valores con calificación “1”

Se obtiene la sumatoria de todos los valores con calificación “2”

Se obtiene la sumatoria de todos los valores con calificación “3”

Finalmente, teniendo en cuenta el concepto de Indicador Cuantitativo Compuesto del tipo “lo parcial respecto de lo total”, se aplica la siguiente fórmula para obtener el Índice de Nivel de Adopción de TICs:

$$\text{Índice de nivel de Adopción de TICs} = \frac{\sum Val (1)}{120} + \frac{\sum Val (2)}{238} + \frac{\sum Val (3)}{201}$$

Esta fórmula representa las contribuciones que hacen cada una de las valoraciones por separado, y sin ningún tipo de ponderación para ellas. Obsérvese también que los denominadores coinciden con la sumatoria de los valores de cada valoración.

Así, en el caso de que la industria analizada posea todas las TICs en todos los procesos industriales, tal como se propone en la Tabla Matriz, el Índice de Nivel de Adopción de TICs dará como resultado el valor “3”, el cual será el valor máximo posible de esta fórmula. Por lo tanto, este Índice puede tener un valor entre 0 y 3, siendo “0” un valor que representa un Nivel de Adopción Insuficiente, y “3” un valor que representa un Nivel de Adopción Óptimo, pasando por valores intermedios que, debido al formato de la fórmula (que termina siendo una suma de 3 fracciones), los mismos podrían contener decimales. Sin embargo, esta fórmula representa las contribuciones que hacen cada una de las valoraciones por separado, y sin ningún tipo de ponderación para ellas.

Esto significa que, si una determinada industria posee solamente todas las TICs valoradas con “1”, la sumatoria será 120, con lo cual en la primera fracción se obtendrá “1” como resultado, y en las otras dos el resultado será “0”, ya que no habría TICs con esa valoración. En este caso, la fórmula indicará que el Nivel de Adopción de TICs es “1”, que coincide con la Valoración Básica propuesta en su momento.

Esta medición podría generar una situación particular, dado que podría existir una determinada empresa que, por ejemplo, posea todos los elementos de Valoración “3”, y ninguno de Valoración “1” ó “2”. Eso llevaría a que la sumatoria de las valoraciones dé 165 y que las primeras dos fracciones den “0” como resultado, mientras que la tercera daría “1”. Con lo cual, esto llevaría a que el Índice de Nivel de Adopción de esa empresa también sea “1”, como en el caso anterior, y que a esta Industria se la pueda categorizar injustamente como si su Nivel de Adopción de TICs fuera básico. Ahora, si bien esto es matemáticamente posible, es poco probable que una empresa que tenga todos los tipos de TICs de valoración “3” no posea TICs de valoración “1” o “2”.

3 Validación y revisión del índice

Se está trabajando en una mejora del cálculo del índice, dado que el valor máximo de cada una de las Valoraciones es “1”, se podrían expresar en formato de porcentaje, con lo cual se podría generar un Índice Global de Nivel de Adopción de TICs y además del Porcentaje de TICs para cada valoración. En este caso, dichos porcentajes se podrían expresar de la siguiente manera:

$$\text{Porcentaje de TICs de Nivel Básico [\%]} = \frac{\sum Val(1)}{120} * 100$$

$$\text{Porcentaje de TICs de Nivel Medio [\%]} = \frac{\sum Val(2)}{238} * 100$$

$$\text{Porcentaje de TICs de Nivel Avanzado [\%]} = \frac{\sum Val(3)}{201} * 100$$

Esta estructura permite, ir adecuando según los eventos que sucedan e impacten sobre la misma y la actualización de TICs a futuro. Por otra parte, se está revisando una modalidad de evaluación a partir de la asignación de ponderación de las TICs de modo tal que se pueden establecer rangos de valores entre los cuales se podrían discriminar los 3 niveles de manera diferenciada. Finalmente, se está realizando una actualización de las TICs según surgen nuevas soluciones, productos y servicios.

El presente índice sólo puede ser aplicado en la medida que permita definir los niveles de desarrollo que pueden ser actualizados hacia adentro en las tecnologías puntuales que lo componen. En este sentido, las nuevas tecnologías deberían ser evaluadas en el nivel más avanzado, del mismo modo en que las tecnologías que entran en desuso deben ser eliminadas del nivel básico.

4 Conclusiones y trabajo futuro

En el presente artículo se ha expuesto un modelo de evaluación de TICs para determinar el nivel de desarrollo tecnológico de implementación y uso en la Industria actualmente, y la misma se ha sometido a una validación a través de expertos.

Respecto de los resultados de dicha validación, y observando los porcentajes de los cuadros presentados que avalan el trabajo realizado en este documento, se dan por validadas las clasificaciones propuestas. A partir de la validación, se concluye que el modelo elaborado es adecuado para la medición de las diferentes TICs implementadas en la industria, independientemente que deban ser ampliados y mejorados algunos tipos específicos con los aportes de los expertos.

El nivel avanzado, constituye la base de las industrias que se encuentran en proceso de transformación hacia la industria 4.0. Sin embargo, resulta necesario diferenciar entre tecnologías existentes y tendencias del desarrollo tecnológico, para poder evaluar en el contexto real de uso cuáles son los niveles de desarrollo tecnológico por ramas de actividad según tengan implementadas tecnologías que llevan mucho tiempo de uso en el mercado, tecnologías que llevan un tiempo medio en el mercado o bien las últimas tecnologías que registra el mercado dentro de cada uno de sus tipos a los fines de detectar las necesidades de implementación y las capacidades para innovación.

5 Referencias

- [1] Ministerio de Ciencia y Tecnología (2009). Libro Blanco de la Prospectiva TIC - Proyecto 2020. Buenos Aires. Obtenido de <http://cdi.mecon.gov.ar/bases/docelec/va1028.pdf> el 23 de Julio de 2019.
- [2] Hewlett Packard (2017). The Internet of Things. Today and Tomorrow. Obtenido de http://chiefit.me/wp-content/uploads/2017/03/HPE-Aruba_IoT_Research_Report.pdf el 23 de Julio de 2019
- [3] ANETCOM. (2017). La TIC en la estrategia empresarial. Valencia. España.
- [4] Mazza, N. H. (2018). Gestión Estratégica de Recursos Informáticos. Buenos Aires: Sustentum. Obtenido de <http://www.sustentum.com/nTIC/nTIC2018.pdf> el 25 de Julio de 2019.
- [5] Del Giorgio, H. (2018). Exploración de la inserción de las Tecnologías de la Información y Comunicación en el desarrollo industrial. Universidad Nacional de La Matanza. Tesis Doctoral defendida el 26 de Julio de 2018.
- [6] Del Giorgio, H.; Mon, A. (2019). Las TICs en las Industrias. Libro actualmente en etapa de edición.
- [7] Mon, A.; Del Giorgio, H.; De María, E. (2017). La inserción de las TICs en el desarrollo industrial de La Matanza. 1^{er} Congreso Latinoamericano de Ingeniería (CLADI). Entre Ríos. Argentina.

Receptor Electrónico de Residuos (RER)

Una infraestructura conceptual de recompensas directa de hábitos ciudadanos.

Rodrigo Baccaro¹, Lucas Gonzalo Rau¹
Alejandro Sartorio²

¹Universidad Abierta Interamericana

²Centro de Altos Estudios en Tecnología Informática – Universidad Abierta Interamericana

¹baccarodrigo1@gmail.com, ¹lucasrau18@gmail.com
²alejandro.sartorio@uai.edu.ar

Resumen. Este trabajo propone un diseño tecnológico para el desarrollo de un artefacto de recolección de residuos urbanos. Su propósito es facilitar la instrumentación de políticas de conductas de responsabilidad ciudadana basadas en recompensas. Para este propósito se creó una prueba de concepto que involucra un diseño de software, una arquitectura, componentes electro-mecánicos embebidos y una aplicación informática de gestión integral y conexión digital con sistemas de gestión municipales. La articulación de estos componentes se denominará Receptor Electrónico de Residuos (RER), su diseño y construcción permiten diferentes tipos de adaptaciones para propósitos similares al cuidado del medio ambiente e innovaciones en tecnologías en ciudades digital públicas.

Keywords: Innovación ciudadana, Sistemas embebidos electrónicos, Ingeniería de Software, Ciudad digital.

1 Introducción

El producto final de este trabajo tiene que ver con la creación de un dispositivo de recolección de residuos con recompensas en áreas urbanas de la ciudad de Rosario. El ciclo de construcción fue desarrollado en el Centro de Altos Estudios en Tecnología Informática (CAETI) de la Universidad Abierta Interamericana (UAI), en el marco de un proyecto de investigación y desarrollo titulado "Receptor Electrónico de Residuos" (RER) perteneciente a la línea de investigación denominada Ingeniería de Software.

Partiendo del objetivo de construir un prototipo de sistema electrónico, mecánico y software con el propósito de disminuir la contaminación ambiental por residuos en las calles. Las componentes mecánicas serán instrumentadas a través de un sistema embebido electrónico [1] relacionado con un software, conformando una unidad de control. Esta unidad determina un subsistema electrónico de procesamiento, programado para realizar funciones de propósitos específicos. Esta configuración se

integra a un sistema general donde se incluyen partes mecánicas, eléctricas y/o electromecánicas.

En países desarrollados hay infraestructuras urbanas adecuadas y factores culturales para instrumentar sistemas del estilo RER, que alientan la instrumentación de estos sistemas de innovación formando un círculo virtuoso. En cambio, aquellos países en los cuales se inician estos desafíos, es imprescindible partir de buenos diseños, prácticas y metodologías de trabajos para que se pueda acortar la brecha y suplir limitaciones de base. Para este propósito, comenzaremos con una breve contextualización de los requerimientos definidos en promover la Innovación Ciudadana [2] a partir de recompensas, con el propósito de incentivar la participación activa de los ciudadanos en iniciativas innovadoras que buscan transformar la realidad social mediante el uso de las tecnologías digitales, con el fin de alcanzar una mayor inclusión social.

1.1 Contexto

Los desechos de colillas de cigarrillos, packaging de comida, bolsas de nylon, papeles, plásticos, aluminio, etc., arrojados a la vía pública, es una problemática de alto impacto social y que tiene grandes consecuencias en la sustentabilidad. Su recolección y la logística específica que necesita tratamiento, insumen un alto costo para los municipios.

Según un informe del Ministerio de Ambiente y Desarrollo Sustentable de la Nación [3], la generación de residuos en Argentina tiene un promedio de 1,02 kg per cápita, lo que representa unas 45.000 toneladas diarias para el total de la población; lo que equivale a la generación de una tonelada cada dos segundos, y alrededor de 16.500.000 toneladas anuales.

Teniendo en cuenta los efectos irreversibles y la importancia de esta problemática para la construcción de ciudadanía, en este trabajo se propone una infraestructura tecnológica extensible y adaptable basada en estándares tecnológicos, para su implementación en una ciudad como Rosario, Argentina, con la posibilidad de conexión con otros sistemas y políticas que se implementan en el municipio.

1.2 Problemática

Según estudios de P.N.L (Programación Neurolingüística) [4] el incentivo mediante premios se considera una de las principales estrategias para lograr cambios en comportamientos sociales. Este tipo de recursos de incentivos (bonificaciones, descuentos, etc.) es cada vez más usado en políticas públicas junto a los más habituales instrumentos de penalización (multas, recargos, etc.). Este trabajo propone un instrumento tecnológico para disminuir los residuos urbanos (basura) en la vía pública, diseñado específicamente para facilitar la implementación de políticas de conductas de responsabilidad ciudadana basadas en recompensas. Así, se implementará un sistema para que cualquier peatón pueda acercar su tarjeta MOVI (tarjeta de transporte público de la ciudad de Rosario) a un dispositivo que contiene un lector que disparará un evento que abrirá automáticamente la compuerta de recepción de residuo. Una vez que la materia depositada sea detectada, se le solicitará al peatón por medio de un mensaje audiovisual emitido en la pantalla y parlante del

dispositivo para que aproxime nuevamente su tarjeta al lector, esta acción registra unidades de créditos de recompensa debidamente calculado (ver Figura 1).

En Rosario en 2018, se comenzó a implementar terminales inteligentes de lectores de tarjeta en distintas paradas de colectivos en lugares estratégicos. Estos tipos de componentes se denominan Tótems [5], y son estaciones ideadas para poder ubicar en lugares estratégicos de la ciudad, poseen los recursos tecnológicos adecuados para capturar, procesar y transmitir información. Estos tipos de elementos se utilizarán como hardware del sistema de control de RER. Además, se incluirá una interfaz de seguimiento del proceso para observación del status de los puestos. También, el dispositivo ofrecerá un registro de uso por peatón, del cual podrán obtenerse resultados estadísticos históricos acerca de la utilización del receptor inteligente a partir del ID de tarjeta MOVI.



Fig. 1. Gráfico Conceptual

1.3 Raspberry PI

Raspberry PI es una placa simple, pudiendo afirmarse que es un ordenador de tamaño reducido, desarrollado en 2011 en el Reino Unido por la Fundación Raspberry PI (Universidad de Cambridge), con el objetivo de estimular la enseñanza de la informática en las escuelas. [6]

Para la construcción de las unidades de control, existen varios modelos de placas electrónicas de diferentes fabricantes que fueron analizadas. Raspberry PI presenta una gran ventaja en costo/calidad, permitiendo desarrollar un dispositivo con prestaciones muy superiores en comparación con las placas que se encuentran actualmente en el mercado. Otro fundamento para su elección se basa en que la misma es un estándar mundial, lo cual significa que existe en internet una gran comunidad que aportan variedades de recursos disponibles para reutilizar. Tanto el hardware como el software utilizado son libres de acceso u utilización.

2 Estado del Arte

A modo de comparación con el presente trabajo, se mencionan proyectos similares de recolección de residuos en áreas urbanas.

Para llevar a cabo el estudio de estos proyectos se desarrolló una tabla comparativa (ver Tabla 1) donde se analizó cómo responde cada uno a los criterios más relevantes que ofrece la solución del proyecto actual. Estos son:

- Uso de nuevas tecnologías.
- La incentivación mediante premios por el uso del receptor inteligente.
- La escalabilidad que otorga este proyecto ya que se podría incorporar la recepción de cualquier tipo de residuo, usando prácticamente la misma metodología.
- La personalización y el seguimiento de uso por peatón.

El trabajo número 1, denominado FUMO, es un receptor de residuos público desarrollado por la agencia holandesa Ioglo [7], su funcionamiento consiste en emitir un sonido totalmente aleatorio cuando los usuarios depositan una colilla de cigarrillo.

El trabajo número 2, creado por la empresa estadounidense TERRACYCLE [8], asegura la recolección de residuos a través de receptores ubicados sobre la vía pública.

El trabajo número 3, denominado PROTRASH [9] es un sistema de recolección de residuos de México creado por la empresa Protrash, basado en el intercambio de basura por comida, a partir del reciclaje de tres materiales: PET, aluminio y vidrio.

Nº	Trabajos	C1	C2	C3	C4	C5	C6	C7	C8
1	FUMO			X		X	X		
2	TERRACYCLE					X	X	X	X
3	PROTRASH	X			X		X	X	X
4	RER	X	X	X	X	X	X	X	X

Tabla 1. Tabla comparativa de soluciones existentes con el proyecto actual.

C1: Bajo Costo; **C2:** Nuevas Tecnologías ; **C3:** Personalización ; **C4:** Premiación por uso ; **C5:** Fácil Instalación ; **C6:** Simplicidad de uso ; **C7:** Escalabilidad e Integración ; **C8:** Reciclado del residuo.

De acuerdo a los resultados arrojados por la tabla anterior, se observa que el proyecto FUMO ofrece una personalización en el uso emitiendo un sonido amigable al usuario, pero no ofrece un seguimiento de uso. El proyecto TERRACYCLE ofrece escalabilidad e integración a nuevos residuos, pero no aplica la premiación por el uso del dispositivo. Por último, PROTRASH ofrece una premiación a través de un ticket para poder intercambiar por algún alimento de nutrición básica, pero no se enfoca en la premiación directa ni en la personalización de uso por peatón como el proyecto actual.

En respuesta a todos los criterios planteados anteriormente, se desarrolla RER enfocado en ser un dispositivo que utiliza nuevas tecnologías para la recolección de residuos en la vía pública, ofreciendo un seguimiento de uso por peatón y otorgando una premiación como punto de incentivación. Además, otorga una alta escalabilidad e integración en incorporar la recolección de cualquier tipo de residuo.

3 Solución

En esta sección se describen los elementos de hardware que fueron seleccionados en la construcción de los sistemas de control. Luego, se brindan detalles sobre las decisiones de diseños y arquitectura del software implementados.

Por último, se presenta un modelo de extensión de propiedades operacionales que permite adaptar la solución final a diferentes instrumentaciones de políticas de recolección inteligente de residuos con mecanismos de recompensas.

3.1 Componentes utilizados

Para llevar a cabo un prototipo se pensó en demostrar la funcionalidad del dispositivo siendo, en este caso, un receptor de colillas de cigarrillos (es importante destacar que el diseño es fácilmente escalable si se quiere agregar más sensores para ser receptor de múltiples residuos como lo pueden ser packaging de comida, bolsas de nylon, papeles, plásticos, aluminio, etc).

Para este ejemplo fueron utilizados los siguientes componentes principales:

RASPBERRY PI 3+B: Este proyecto, utiliza como núcleo del dispositivo un RASPBERRY PI [6] modelo 3+B (ver Figura 2) por su bajo costo, simplicidad y variedad de componentes disponibles en el mercado. Todos los componentes del proyecto se conectan a esta placa. Especificaciones técnicas más importantes: - Procesador: Broadcom BCM2837B0, Cortex-A53 (ARMv8) 64-bit SoC - Frec. reloj: 1,4 GHZ-Memoria:1GB LPDDR2 SDRAM - Conec. Inalámbrica: 2.4GHz / 5GHz - Red: Gigabit Ethernet



Fig. 2. Raspberry PI 3+B

MÓDULO RFID 13.56MHZ (LECTOR DE TARJETA): El módulo RFID [10] (ver figura 3) utiliza un sistema avanzado de modulación y demodulación para todo tipo de dispositivos pasivos de 13.56Mhz. Al ser ideal para dispositivos portátiles o tarjetas, con este módulo se hará el reconocimiento cuando el peatón aproxime la tarjeta MOVI al lector. Especificaciones técnicas más importantes:- Modelo: MF522-ED. - Corriente de operación: 13-26mA a 3.3V. - Isb de stand by: 10-13mA a 3.3V. - Ism de sleep-mode: <80uA. - Im máxima: 30mA.- Frecuencia de operación: 13.56Mhz. - Distancia de lectura: 0 a 60mm.- Protocolo de comunicación: SPI. - Velocidad de datos máxima: 10Mbit/s.- Máxima velocidad de SPI: 10Mbit/s.



Fig. 3. Módulo RFID 13.56MHZ (lector de tarjeta)

SENSOR DE TEMPERATURA DS18B20: El sensor DS18B20 [11] (ver Figura 4) permite medir temperaturas de hasta 125°C de forma fácil y además está sellado en un envoltorio estanco que permite sumergirlo en un líquido o protegerlo de la intemperie. Dado que es un sensor digital, la señal leída no se degrada debido a la distancia del cableado. Especificaciones técnicas más importantes: - Rango de temperatura: -55 a 125°C-Resolución: de 9 a 12 bits (configurable). - Múltiples sensores con mismo pin. - Precisión: $\pm 0.5^{\circ}\text{C}$ (de -10°C a $+85^{\circ}\text{C}$).- Tiempo de captura inferior a 750ms. - Alimentación: 3.0V a 5.5V



Fig. 4. Sensor de temperatura DS18B20

3.2 Esquema de conexión

La figura 5 muestra un diagrama de conexión simple, que permite entender la relación que existe entre todos los componentes antes mencionados. En el mismo podemos ver sobre la izquierda la placa Raspberry PI con sus pines conectados hacia los sensores correspondientes, el primero en entrar en acción es el sensor encargado de detectar la temperatura (que se encuentra en la parte superior). En la parte central se encuentra el sensor RFID cumpliendo la función de lecturas de tarjetas magnéticas. Luego por el sector derecho, se encuentra el driver encargado de realizar el manejo del motor el cual se ubica en la parte superior, el mismo es alimentado por una batería de 9v.

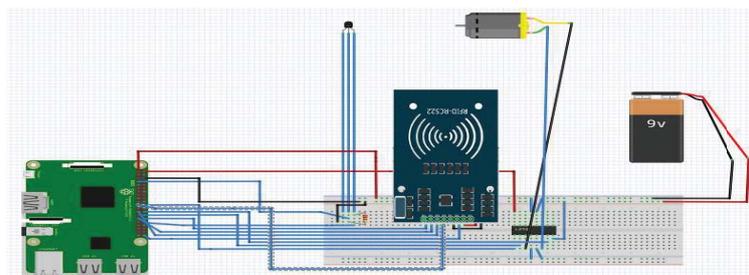


Fig. 5. Esquema de conexión general.

3.3 Diseño y Arquitectura

El diseño de RER está basado en una arquitectura de control, que puede verse en la Figura 6. Sus principales componentes son sensores, actuadores, interfaces y controladores electrónicos embebidos con conexión de pines (GPIO) que funcionan de manera entrada/salida en la placa Raspberry PI con su infraestructura propia de módulos para su configuración y programación. De esta manera, es posible

implementar las políticas de acciones y control sobre los sensores mediante un lenguaje de programación de alto nivel; para este trabajo se utilizó Python 3 en su versión estándar.

Para la recolección de información se utilizaron sensores y lector RFID que enviarán información al componente Controlador, encargado de procesar los datos recibidos en función de los algoritmos y procedimientos que configuran la lógica de negocio de todo el sistema. Además, se utiliza como mecanismo de comunicación, entre componentes del sistema, el protocolo HTTP sobre una red TCP/IP alimentado mediante un cable ethernet conectado en la placa del módulo Raspberry PI. De esta manera, se informará a un servidor HTTP cada evento que se esté realizando. A su vez, este servidor es el encargado de brindarle datos (en formato JSON) a una página web que mostrará el flujo de trabajo de los componentes en un monitor de control en formato para página Web con acceso para navegador Web.

El motor es un actuador conjunto con la interface (Pantalla), que conectados a un puerto GPIO son los únicos que generan eventos de salidas. En este caso, cada componente actuador cumple con su función correspondiente regulada por el controlador. Así, se proyectan los datos en una pantalla y se envían las instrucciones necesarias para la apertura y cierre de compuertas según corresponda.

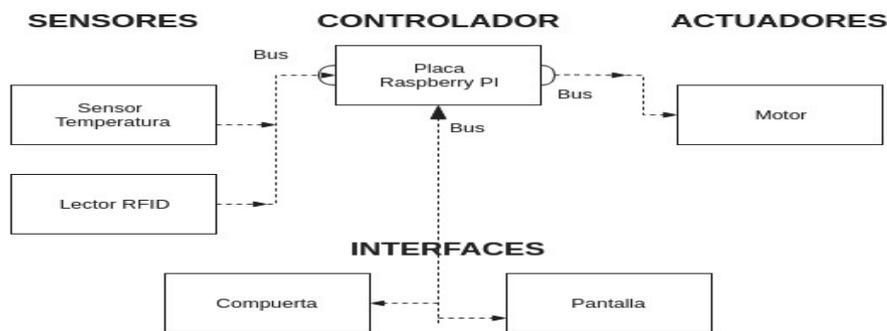


Fig. 6. Arquitectura

3.4 Adaptación a nuevas funcionalidades.

Desde la perspectiva del diseño de software se pensó en diseñar e implementar una aplicación capaz de adaptarse a nuevas funcionalidades y políticas de uso, o prueba de conceptos con el mínimo coste para concretarlo. Con este propósito, se utiliza un diseño de módulos funcional para la programación del controlador basado en la aplicación del patrón de diseño “command” [12]. En la figura 7 se muestra una representación conceptual de la utilización del patrón “command” con propósito de lograr una representación de todas las comunicaciones posibles entre el controlador y las demás componentes del sistema. A este tipo de comunicación la denominaremos operaciones de acción. Un posible ejemplo de una operación puede ser la instrucción que el controlador le indica al motor para que se prenda y abra la compuerta para la recepción de un residuo. De esta manera, se brinda un mecanismo para que las operaciones puedan verse y utilizarse como si fueran una componente acción más del sistema. En la figura 7, estos componentes son módulos

que se extienden de un módulo abstracto denominado Operador y que podrá ser utilizado a través de otro módulo invocador determinado; por ejemplo: el controlador. Luego, estas operaciones son recibidas por otros módulos denominados receptores, en este caso, formados por la extensión de un módulo abstracto denominado Receptor. Como ejemplos de receptores se pueden mencionar: pantalla o visor, luces, tarjeta magnética y componentes electrónicos embebidos.

Al tener esta representación de módulos implementados es posible construir, instanciar y utilizar operaciones de acciones para el RER. Esta propiedad se lleva a cabo desde el módulo Cliente de la figura 7, a través de la creación e instanciación de los módulos de operación y la visualización de los estados que pueden adquirir los receptores. Por ejemplo, un operador del sistema puede tener en su tablero de control una serie de comandos configurables para ir controlando y visualizando el funcionamiento integral de cada instancia del sistema.

Esto, a su vez, permite que, al realizar un cambio físico de sensores, las funcionalidades sigan ejecutándose de la misma manera.

La aplicación de este diseño le permite al proyecto definir una metodología técnica de extensión de las funcionalidades con las ventajas típicas de un estándar. Estas nuevas funcionalidades pueden ser agregadas como muestra el diagrama con entidades de líneas de punto.

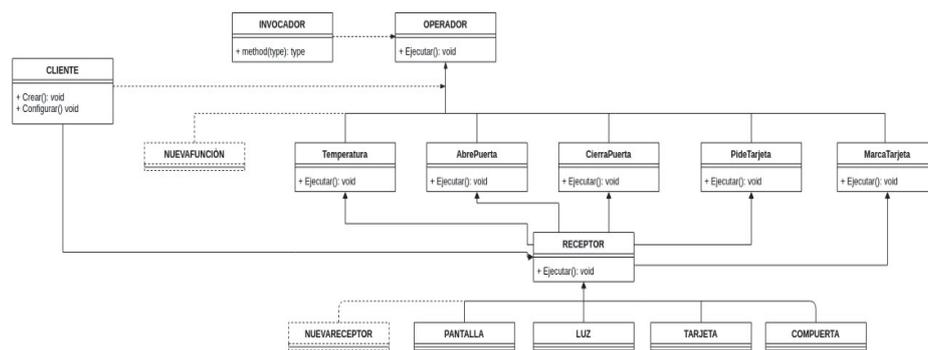


Fig. 7. Esquema del patrón comando.

Entidades

Operador: Declara una interfaz para ejecutar una operación.

Operador Concreto: Define un enlace entre un objeto “Receptor” y una acción.

Cliente: Crea un objeto por ej. “Temperatura” y establece su receptor.

Invocador: Le pide a “Operador” que ejecute la petición. (Sensores)

Receptor: Sabe cómo llevar a cabo las operaciones asociadas a una petición.

4 Caso Práctico

4.1 Caso de Uso

Siguiendo con el ejemplo del sensor de temperatura para la recepción de colillas de cigarrillos, en la Figura 8 se muestra la secuencia de uso del dispositivo:

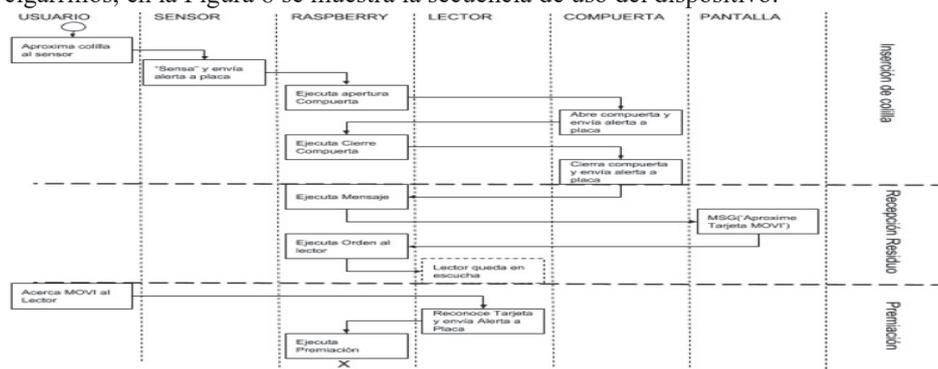


Fig. 8. Diagrama de Secuencia

- 1 Inserción de residuo:** El usuario del dispositivo acercará el residuo al sensor, donde emitirá una alerta al Raspberry PI, indicando la apertura de la compuerta.
- 2 Recepción Residuo:** Una vez que el residuo se depositó, se cerrará la compuerta. Posteriormente a esto se mostrará un mensaje en pantalla indicando al usuario que acerque su tarjeta MOVI al lector.
- 3 Premiación:** Luego de que el usuario acercó su tarjeta MOVI al lector, se premiará con una determinada cantidad de saldo en la tarjeta de acuerdo a un número de residuos depositados.

4.2 Interfaz gráfica del sistema

En la figura 9 se observa una captura de pantalla del sistema donde permite monitorear el proceso del dispositivo.

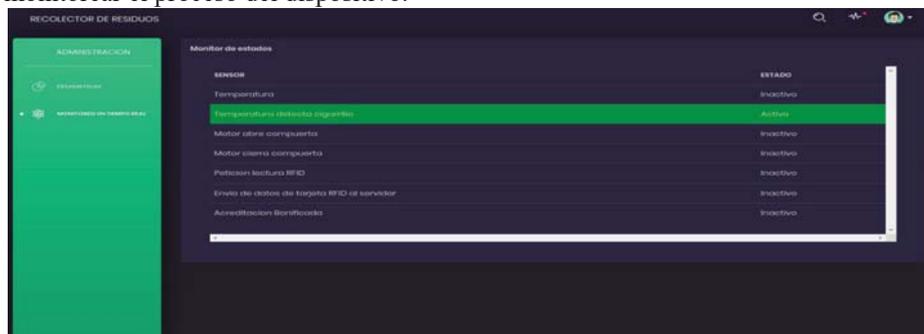


Fig. 9. Interfaz gráfica del sistema

4.3 Auditoría/histórico de uso del dispositivo

El sistema de RER registra todas las transacciones de uso realizadas. La aplicación cuenta con un tablero de control (ver Figura 10) que permite, filtrar a partir de una fecha desde y hasta el uso del dispositivo. Ofrece un reporte para el rango de fecha ingresado mostrando:

- La cantidad de bonificaciones acreditadas.
- La cantidad de residuos recolectados.
- El rango de edad de ciudadanos que utilizan el dispositivo. Esto se realiza obteniendo información a través del consumo de un servicio conectado a la base de datos de la municipalidad para aquellas tarjetas MOVI que hayan sido personalizadas.

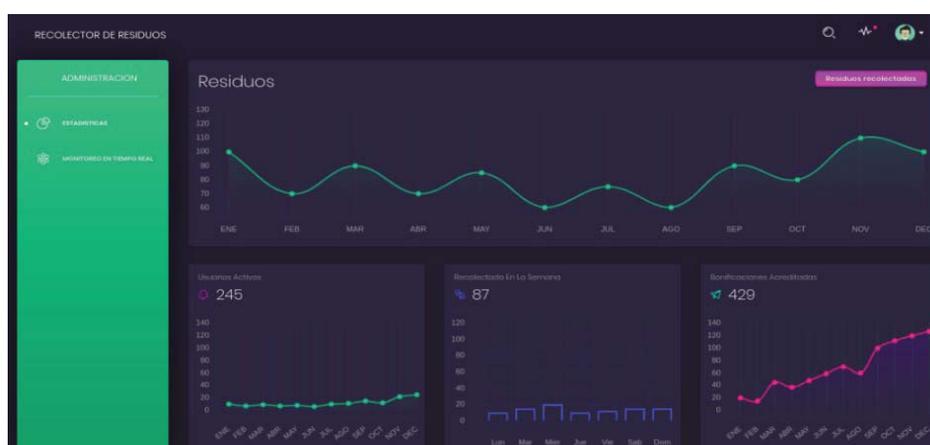


Fig. 10. Tablero de control

5 Conclusión

El presente trabajo se ha desarrollado con el objetivo de construir un prototipo de sistemas electrónico, mecánico y de software para la instrumentación de políticas de conductas de responsabilidad ciudadana basadas en recompensas. Además, empleado en buenas prácticas de diseño de software aplicadas a sistemas embebidos de bajo costo que se ofrecen en el mercado actual. Este prototipo tiene la capacidad de ser receptor de residuos para ser utilizado por cualquier ciudadano del microcentro de la ciudad de Rosario, utilizando los denominados Tótems, al emplear un sistema de premiación por uso que tiene como fin principal incentivar al usuario a modificar su comportamiento en arrojar residuos en las calles, y la contaminación que se genera a partir de esto. Las decisiones de diseño, construcción e implementación tuvieron el propósito de ser adaptados, extendidos y utilizados para pruebas de conceptos en políticas públicas de innovación en espacios urbanos basados en brindar servicios, fomentar buenas prácticas de convivencia y recompensas.

Referencias

- [1] Vega, J. I. H. (2010). El software embebido y los retos que implica su desarrollo.
- [2] Caamaño, H., & Pascale, P. (2014). Innovación Ciudadana en Iberoamérica: participación digital para la transformación social. In Congreso Iberoamericano de Ciencia, Tecnología, Innovación y Educación.
- [3] Ministerio de ambiente y desarrollo sustentable de la Nación (MAyDS). (2017). Recuperado de: <http://informe.ambiente.gob.ar/>
- [4] Dilts, R. (1998). Liderazgo Creativo. PNL Programación Neurolingüística. Para forjar un mundo al que las personas deseen pertenecer. Urano.
- [5] Tótems Inteligentes (2019). Recuperado de: <http://www.etr.gov.ar>
- [6] Sobota, J., PiŚl, R., Balda, P., & Schlegel, M. (2013). Raspberry Pi and Arduino boards in control education. IFAC Proceedings Volumes, 46(17), 7-12.
- [7] Agencia Holandesa IOGLO (2018). Recuperado de: <http://www.ioglo.com/>
- [8] Terracycle (2018). Recuperado de: <https://www.terracycle.com/es-ES/>
- [9] Protrash (2019). Recuperado de: <https://www.protrashco.com/single-post/2016/07/07/NUUESTRA-HISTORIA>
- [10] Sánchez, J. A. A. (2008). Sistema de Control de Acceso con RFID. México DF.
- [11] Xin-min, H. Z. J. L. (2003). Digital Temperature Sensor DS18B20 And Its Application [J]. Journal of Nanjing Institute of Technology (Natural Science Edition), 1, 001.
- [12] Larsen, G. (1999). Designing component-based frameworks using patterns in the UML. Communications of the ACM, 42(10), 38-45.

Sistema integrador de dispositivos de la salud con conexión BLE para envío de alarmas

Martín Destefano, Leandro Luque, José Luis Vázquez Noguera, and Cynthia Villalba

Universidad Nacional de Asunción, Facultad Politécnica
San Lorenzo, Paraguay

{destefanopy, lealuque09}@gmail.com
{jlvazquez, cvillalba}@pol.una.py

Resumen Actualmente los sistemas de monitoreo de la salud no tienen centralizada la información entregada por dispositivos que miden parámetros fisiológicos, a su vez no existe una manera fácil de integrar nuevos dispositivos al sistema, acarreado esto un alto costo de desarrollo.

Este trabajo plantea desarrollar un sistema, utilizando el protocolo BLE (Bluetooth Low Energy) y el estándar HDP (Health Device Profile) para solucionar el problema de integración de dispositivos, pudiendo aprovechar los datos en conjunto obtenidos de las mediciones. Las mediciones serán monitoreadas, y en caso de presentar riesgo el paciente, una alarma será enviada a personales de la salud o familiares.

Las pruebas muestran la efectividad del sistema, pudiéndose integrar los datos de varios dispositivos de medición y del envío de alarmas cuando el usuario se encuentre en una situación considerada de riesgo.

Keywords: monitoreo; sistemas; riesgo; alarma; BLE; salud; integración; dispositivos; medición

1. Introducción

La salud se pone en riesgo a consecuencia del carácter asintomático de afecciones tales como la presión arterial alta, la diabetes u otras con menos peligros para la salud. Estas afecciones pueden presentar consecuencias a futuro si no se los trata a tiempo [2].

Los controles y el monitoreo generalmente se realizan cuando el paciente tiene una consulta con el doctor [10]. La complicación ocurre cuando el paciente presenta una situación peligrosa para la salud. Esta circunstancia requiere actuar de manera rápida y precisa con el objetivo de determinar el procedimiento a seguir [3].

La gran cantidad de dispositivos que existen poseen protocolos propios diseñados por el fabricante, tales como tensiómetros [8], glucómetros [9], entre otros. Esta situación representa una dificultad en el campo de la medicina a nivel tecnológico tales como [1]: 1) problema en integrar los dispositivos a un

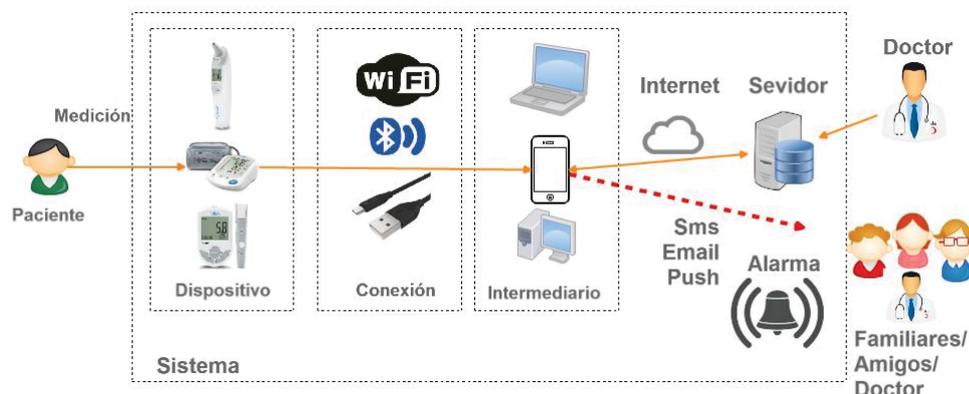


Figura 1. Componentes del sistema integrador de dispositivos de la salud

único sistema, debido a que algunos utilizan diferentes medios de conexión, 2) la información se encuentra aislada por falta de utilización de un estándar de intercambio de datos, no pudiendo aprovechar en conjunto con los datos de otras mediciones, 3) tiempo elevado de programación que se requiere para integrar dispositivos que utilizan diferentes tipos de medios de conexión y protocolos de traspaso de información.

Otra dificultad ocurre en el momento de las mediciones con estos dispositivos y afectan tanto a los profesionales de la salud como a los pacientes. La movilización al centro de salud por parte del paciente para la consulta médica resulta un problema de tiempo. La posible saturación de pacientes en el centro de salud y el tiempo de respuesta son factores que dificultan una atención eficiente [5].

El objetivo general de la investigación es desarrollar un sistema donde dispositivos de medición puedan ser integrados a un único sistema. Los dispositivos deberán soportar el protocolo Bluetooth Low Energy y seguir un estándar específico de traspaso de información, Health Device Profile. El doctor podrá configurar en el sistema reglas asociadas a los parámetros fisiológicos estables de salud. El personal de salud, familiares o amigos podrían ser advertidos en caso de que alguna regla disparada represente algún peligro para el paciente.

El documento está organizado como sigue: en la sección 2 se presentan conceptos claves que harán posible comprender la propuesta de solución. En la sección 3 se plantea una arquitectura y la implementación de este para la solución del problema. En la sección 4 se mostrarán los resultados de la evaluación del sistema. Al final, en la sección 6, se dará una conclusión al trabajo realizado.

2. Marco Conceptual

En esta sección se van a conceptualizar términos claves para comprender la propuesta de solución al problema planteado.

2.1. Componentes del sistema

Los componentes citados a continuación son de sistema similares que existen en el estado del arte, una representación de como se relacionan se puede ver en la Figura 1.

- **Dispositivos Medidores:** Los dispositivos móviles de salud son capaces de extraer los datos relacionados con la salud del paciente (tensiómetro, glucómetro, etc.) [11] [8] [9] . Estos dispositivos pueden asociarse con teléfonos móviles, notebooks o dispositivos . Las aplicaciones que ayudan a la conexión con el dispositivo son facilitadores de vigilancia.
- **Medio de Conexión:** Nexa por el cual los datos medidos por el dispositivo son enviados a un aparato intermediario con el propósito de ser procesados [11] [12].
- **Intermediarios:** Los gateways o intermediarios son los dispositivos cuya función es la de proporcionar conectividad a los hosts individuales a la red [12] [13]. Además, garantiza que los datos fluyan correctamente.
- **Servidor:** Es un ordenador especialmente diseñado con arquitectura de alto rendimiento, el cuál se encarga de proveer diferentes servicios a las redes de datos [15] . Algunos servicios tales como administración de bases de datos o poseer repositorios de archivos.
- **Alarma** La alarma es el componente clave para los sistemas de emergencias debido a que es la encargada de transmitir una señal. La misma informa a los doctores y/o familiares de la presencia real o inminente de una amenaza en la salud del paciente [16].

2.2. Perfil del dispositivo de salud - HDP

Las implementaciones en general se han basado en Bluetooth Serial Port Profile (SPP) y protocolos específicos del fabricante. Por lo tanto, diferentes implementaciones han tenido un bajo nivel de interoperabilidad entre sí. Esta es la razón por la cual el Special Interest Group (SIG) de Bluetooth formó el Medical Device Working Group. El mismo estableció un objetivo para desarrollar un perfil que introdujera la interoperabilidad entre diferentes sensores médicos y dispositivos de recolección de diferentes fabricantes. El trabajo dio como resultado el Multi-channel Adaptation Protocol (MCAP) y el Bluetooth Health Device Profile (HDP).

El perfil HDP proporciona sincronización de reloj opcional entre los dispositivos y la identificación del tipo de dispositivo.

Source: Actúa como fuente de los datos médicos.

Sink: Recibe los datos médicos a partir de fuentes individuales o múltiples

HDP está especializado en aplicaciones de salud y, por lo tanto, tiene las siguientes ventajas sobre otros perfiles más genéricos:

- Proporciona una fuerte interoperabilidad a nivel de aplicación al operar con el Protocolo de intercambio de datos de salud personal ISO/IEEE 11073-20601
- Las disposiciones para un método estandarizado mediante el cual el tipo de dispositivo y los tipos de datos de aplicación admitidos de un dispositivo se pueden determinar de forma inalámbrica, utilizando el Bluetooth Service Discovery Protocol (SDP).
- Orientado a la conexión para garantizar un comportamiento más confiable cuando el Source se mueve fuera del alcance o se desconecta (involuntariamente o involuntariamente), lo que permite que el dispositivo reconozca el estado y tome las medidas apropiadas.

3. Arquitectura Propuesta

La propuesta consiste en diseñar e implementar un sistema con una arquitectura capaz de dar solución al problema previamente descrito.

3.1. Diseño

La vista general de la arquitectura propuesta se puede observar en la Figura 2, donde se muestran grupos importantes de personas. El primero es el paciente, que mediante distintos dispositivos puede monitorear ciertos parámetros fisiológicos o signos vitales tales como la temperatura, pulso, nivel de glucosa, presión arterial, etc.

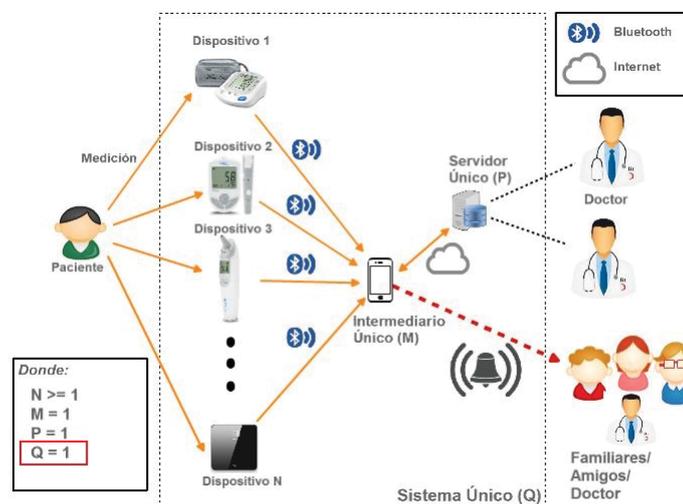


Figura 2. Arquitectura propuesta

Los dispositivos se conectaran a un intermediario cuya función principal será la de leer los datos de los dispositivos y recolectarlos para mantener un registro centralizados de las mediciones captadas. El segundo grupo corresponde a los médicos cabecera que deberán analizar los datos recolectados, y según estos establecer reglas de monitoreo. Las reglas serán analizadas desde el intermediario. El último grupo corresponde a los familiares, a quienes se les notificará, mediante SMS, que el paciente está pasando cierto umbral configurado por el médico.

A continuación se explica más detalladamente cada entorno de la arquitectura:

- **Entorno paciente:** corresponde a los dispositivos propicios para las mediciones de cada caso. Estos estarán configurados para lograr la comunicación entre el dispositivo de medición y el intermediario. El paciente al realizar una medición desde su dispositivo, y los datos no se encuentre dentro de los parámetros estables de salud, por lo cual se constituya en una situación riesgosa, el intermediario la captará y actuará en consecuencia.
- **Entorno servidor:** en este entorno se centralizará la información proveniente de cada escenario de monitorización almacenando la información asociada en base de datos [4]. Éstos serán enviados por el paciente, transmitidos por medio del intermediario, que se comunicarán, a través de internet con un servidor remoto que gestionará los diferentes intermediarios [6].
- **Entorno alarma:** se lanzará una alerta para las personas cercanas al paciente que en esos momentos se encuentra con niveles de salud diferentes a los establecidos en las reglas. Una vez detectado el peligro de salud por medio de los dispositivos, enviado el mensaje, esto generará una alerta también a los especialistas de la salud, para que tomen nota y sepan cuándo el paciente presenta signos anormales en sus controles de salud.
- **Entorno doctor:** una interfaz adecuada se desarrollará para que cada usuario y cada profesional médico puedan disponer de los elementos necesarios para trabajar. Este contará con una interfaz web en la cual el doctor podrá realizar la modificación y actualización de las reglas para cada paciente, que según su criterio, le parecen adecuados para evitar una situación de emergencia [4].
- **Entorno administrador:** este entorno contará una la interfaz web para el mantenimiento del sistema y se podrán realizar las gestiones de usuarios. Cuando un nuevo dispositivo deba ser usado para el control del paciente, como por ejemplo un termómetro, puede ser incluido dentro del sistema. En otras palabras, cada dispositivo de medición que soporte el protocolo BLE y el estandar HDP puede ser registrado al sistema. En la Figura 3 se observa un ejemplo de la configuración de un nuevo dispositivo al sistema.

3.2. Implementación de la arquitectura

En la Figura 4 se muestran todos los componentes implementados para dar soporte a la arquitectura planteada.

Datos Dispositivo			
Nombre	Codigo Data Type	Tipo de Atributo	
Blood Pressure	1007	a75	

Parametros Asociados			
Orden	Descripcion	Unidad de Medida	Eliminar
1	systolic	mmHg	
2	diastolic	mmHg	

Agregar Parámetro Guardar

Lista de Dispositivos

COD	Nombres	Tipo Atributo	Parametros	Opciones
100F	Body Weight Scale	a75	peso	
1004	Pulse Oximeter	a75	porcentaje oxigeno, frecuencia pulso	

Figura 3. Configuración de un nuevo dispositivo.

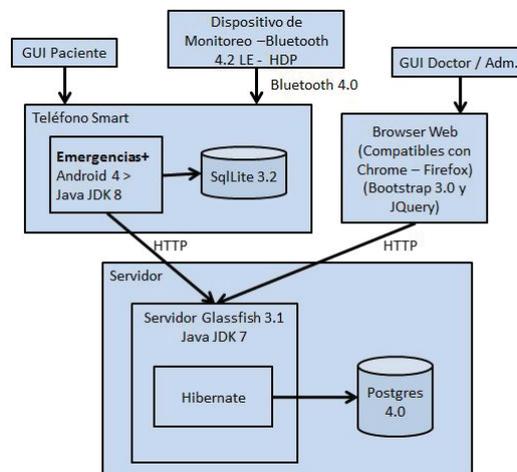


Figura 4. Diagrama de Componentes

A continuación se hace una breve descripción de los componentes del diagrama:

- **Servidor:** se implementó una aplicación Java, que disponibiliza una Api Rest con todos los servicios necesarios para la aplicación móvil y las interfaces web para los doctores y usuarios administradores.
- **Teléfono Smart (intermediario):** está compuesta de una aplicación móvil. Este posee una interfaz de Login para el ingreso a las demás opciones de la aplicación, se puede visualizar el historial de mediciones realizadas. Se posee un módulo de lectura de mediciones, donde se debe seleccionar la afección que se posee y luego establecer una conexión Bluetooth entre el intermedia-

rio y el dispositivo para capturar las mediciones. Las mediciones obtenidas del módulo anterior, pasan al módulo de reglas, donde se compara con las reglas obtenidas de la base de datos del teléfono smart, esta comparación se realiza con JEXL (Java Expression Language) [17] que es una librería que facilita las comparaciones de expresiones lógicas. En el punto que los datos recolectados por la medición del paciente es analizada por el módulo de reglas, y de ser necesario pasa al módulo SMS. Existe una módulo que permite configurar las personas a quienes les llegará el mensaje de alarma.

- **GUI Doctor/Administrador:** se disponibiliza una interfaz web para la gestión de reglas por parte del doctor para cada paciente, un ejemplo de esto es posible ver en la Figura 5. El doctor tiene la posibilidad de ver el historial de mediciones de cada paciente. El Administrador posee una interfaz web donde se encarga de la gestión de los doctores, pacientes y nuevos dispositivos que se ingresan al sistema Figura 3.

Regla Generada
(diastolica > 180 AND sistolica > 100 AND peso > 120 AND (diastolica < 70 OR sistolica < 60 OR peso > 120))

AND • Agregar Condición • Agregar Grupo • Remover Grupo

diastolica > 180

sistolica > 100

peso > 120

OR • Agregar Condición • Agregar Grupo • Remover Grupo

diastolica < 70

sistolica < 60

peso > 120

Agregar

Figura 5. Ejemplo de configuración de una regla

4. Evaluación y resultados

El objetivo principal es comprobar que la arquitectura propuesta es capaz de satisfacer las necesidades que competen a diferentes situaciones de emergencia, donde para este caso de estudio serán relacionadas a la presión arterial y a la glucosa en la sangre.

Las pruebas de campo se realizó en un entorno real, con pacientes que se encuentran en su hogar y pacientes del hospital Materno Infantil de San Lorenzo con un tensiómetro. Además se emuló otros dispositivos de medición con un smartphone que envía los mismo datos que envía un dispositivo con HDP, y este, simulara un termómetro, glucómetro, etc.

4.1. Evaluación de componentes

A continuación se citan las evaluaciones de los componentes que comprobarán el funcionamiento del sistema.

- Verificar la recepción correcta de los datos del dispositivo de medición al intermediario.
- Verificar la recepción/envío correctos de los datos del intermediario al servidor.
- Verificar la recepción/envío correctos de los datos enviados por el módulo de alarma.
- Verificar y evaluar el correcto funcionamiento de las reglas en un entorno simulado y uno real con la supervisión de un profesional de la salud.

4.2. Resultados

Los resultados se plasman en las siguientes tablas 1, 2 y 3, donde se valida si el caso de uso funciona o no como se espera. Las evaluaciones se realizan sobre tres grupos usuarios del sistema: doctores, administradores y pacientes.

Cuadro 1. Casos de uso del Doctor

Casos de Uso	Datos entrada	Resultado	Resultado Esperado	Tiempo transcurrido	Aprobado
Ingreso al sistema	Usuario y Contraseña	Ingreso al sistema	Ingreso al sistema	5 segundos	SI
Creación de reglas	Se configuró un regla que compara el parámetro sistólico y diastólico con valores que se consideran peligrosos para la salud del paciente	Se sobrepaso intencionalmente estos valores configurados y se activo la regla	Se activo la alarma	1 minuto	SI
Ver el historial de mediciones del Paciente	Se selecciona un paciente	Se pudo observar las nuevas mediciones tomadas en el caso de uso anterior	Se lista las nuevas mediciones	2 segundos	SI
Comparar parámetros fisiológicos en el tiempo	Se selecciona un paciente, uno o más parámetros fisiológicos que se desea comparar su variación en el tiempo y el rango de fecha que se desea utilizar para el filtro	Se ve la variación en el tiempo de los parámetros fisiológicos seleccionados		5 segundos	SI

5. Conclusiones

Se ha diseñado una arquitectura que hace posible que los pacientes puedan tener centralizada las mediciones entregadas por dispositivos que miden parámetros fisiológicos. Para solucionar el problema de integración se optó que los dispositivos se comuniquen mediante bluetooth con los intermediarios y se utilizó el estándar HDP para poder leer las mediciones.

Cuadro 2. Casos de uso de paciente

Casos de Uso	Datos entrada	Resultado	Resultado Esperado	Tiempo transcurrido	Revisado-Aprobado
Ingreso al sistema	Usuario y Contraseña	Ingreso al sistema	Ingreso al sistema	5 seg	SI
Agregar contacto de emergencia	Se agrega un Contacto o se selecciona de la lista de Contactos	Contacto agregado	Contacto agregado	15 seg	SI
Eliminar contacto de emergencia	Contacto existente	Contacto eliminado	Contacto eliminado	15 seg	SI
Escenario real: Medición de la presión arterial con tensiómetro	Datos de la presión sistólica y diastólica enviados por el Tensiómetro	Datos obtenidos y evaluados adecuadamente	Datos obtenidos y evaluados adecuadamente	30 seg	SI
Escenario simulado: Medición de la glucosa en la sangre	Datos de la cantidad de glucosa enviado por el Glucómetro	Datos obtenidos y evaluados adecuadamente	Datos obtenidos y evaluados adecuadamente	120 seg	SI
Ver el historial de mediciones	Mediciones nuevas	Se visualizan las últimas mediciones	Se visualizan las últimas mediciones	2 seg	SI

Cuadro 3. Casos de uso del Administrador

Casos de Uso	Datos entrada	Resultado	Resultado esperado	Tiempo transcurrido	Aprobado
Ingreso al sistema	Usuario y Contraseña	Ingreso al sistema	Ingreso al sistema	5 segundos	SI
Creación de pacientes	Se completó todo los datos solicitados en el formulario	Se ve en la base de datos que existe el nuevo paciente		1 minuto	SI
Eliminación de pacientes	Se selecciona el paciente a eliminar	Se elimina el paciente de la base de datos		5 segundos	SI
Modificación de los datos del pacientes	Se modifican algunos datos del paciente	Se verifica en la base de datos que los datos fueron actualizados		5 segundos	SI
Creación de doctores	Se completó todo los datos solicitados en el formulario	Se ve en la base de datos que existe el nuevo doctor		1 minuto	SI
Eliminación de doctores	Se selecciona el doctor a eliminar	Se elimina el doctor de la base de datos		5 segundos	SI
Modificación de los datos del doctores	Se modifican algunos datos del doctor	Se verifica en la base de datos que los datos fueron actualizados		5 segundos	SI
Asociación de pacientes a un doctor	Se asocia varios pacientes a un doctor	Se verifica en la base de datos que la relación se haya realizado	Se ve en la web los pacientes asociados al doctor		SI
Eliminar asociación entre paciente y doctor	Se selecciona y elimina un paciente asociado a un doctor	Se verifica en la base de datos que la relación se haya eliminado	Se ve en la web que ya no existe el paciente asociado al doctor	5 segundos	SI
Agregar nuevos dispositivos	Se completó todo los datos solicitados en el formulario	Se verifica en la base de datos que se haya insertado el nuevo dispositivo	Se ve en la web el nuevo dispositivo listado	45 segundos	SI
Asociar los parámetros fisiológicos al dispositivo	Se asociaron parámetros fisiológicos al tensiómetro	Se verifica en la base de datos que se haya realizado la asociación	Se ve en la web los parámetros fisiológicos asociados al dispositivo	30 segundos	SI

Se ha implementado un sistema para gestionar la integración de nuevos dispositivos de medición, para la gestión de pacientes y doctores, dotándoles de opciones muy útiles. El sistema implementado permite que el doctor pueda gestionar reglas para cada paciente y así se pueda monitorear el estado de salud, y en caso de que haya alguna parámetro fuera de lo normal avisar al centro médico y familiares.

Se ha testeado cada caso de uso citado, obteniendo muy buenos resultados. Entre los objetivos más importantes cumplidos se pueden citar, agregar nuevos dispositivos de medición, configurar reglas sobres estos y lanzar alarmas.

Con este trabajo se pretende alentar el uso de este tipo de sistemas, colaborando a la prevención de situaciones de riesgo en pacientes con afecciones diversas.

Referencias

1. Organización Panamericana de la Salud. “*La hipertensión arterial como problema de salud comunitario*”. En: (1990).
2. American Heart Association. “*Health Threats From High Blood Pressure*”. En: (2018).
3. Dr Nicolás Stavile, Dr Raúl F. Echeverría . “*Hipertensión arterial severa en salas de guardia (Crisis Hipertensivas)*”. En: (2007), Págs. 5-7.
4. Jad Noueihed, Robert Diemer, Samarjit Chakraborty, Stefanie Biala. Institute for Real-Time Computer Systems, TU, Munich . “*Comparing Bluetooth HDP and SSP for Mobile Health Devices*”. En: (2008).
5. Instituto de Bioética Facultad de Medicina Universidad de Buenos Aires. “*Impacto de la tecnología en la práctica de la medicina*”. (2013), Págs. 38-40.
6. Fareed ud din, Atta-ur-Rehman Shah, Muhammad Ibrahim. Superior University, Lahore, Pakistan. “*Mobile-Phone-Based Remote Patient’s Vital Signs Monitoring and Automated Alerts*”. En: (2013).
7. Elkstein M, ”Learn REST: A Tutorial”. blogger.com. Retrieved 16 April 2015.
8. Ben-Jeng Wang, Tzu-An Chiang, Fu-Chiang, Yung-Chieh Chien. “*Applying RFID Technology to develop a distant medical care service platform*”. En: (2010).
9. Giordano Lanzola, Davide Capozzi, Giuseppe D’Annunzio, Pietro Ferrari, Riccardo Bellazzi, Cristiana Larizza. “*Going Mobile with a Multiaccess Service for the Management of Diabetic Patients*”. En: (2007).
10. Registros Médicos Electrónicos de América Latina y el Caribe. “*Análisis sobre la situación actual y recomendaciones para la Región*”. En: (2016).
11. Jad Noueihed, Robert Diemer, Samarjit Chakraborty, Stefanie Biala. Institute for Real-Time Computer Systems, TU, Munich. “*Comparing Bluetooth HDP and SSP for Mobile Health Devices*”. En: (2008).
12. Osama Salameh, Faculty of Engineering and Information Technology, Arab American University, Palestine. “*A Mobile Phone SMS-Based System For Diabetes Self Management*”. En: (2011).
13. Victor Manuel Morales Rocha, Luis Felipe Fernández Martínez, Jorge Enrique Rodas Osollo, Karla Miroslava Olmos Sánchez. “*Monitoreo Remoto de Pacientes con Diabetes Utilizando Tecnologías Móviles Inalámbricas*”. En: (2012).
14. I. Martínez, J. Fernández, M. Galárraga, L. Serrano, P. de Toledo, J. García. “*Implementación Integrada de una Plataforma Telemática Basada en Estándares para Monitorización de Pacientes*”. En: (2007).
15. Paolo Giudici, Faculty of Economics, University of Pavia, Italy. “*Applied Data Mining, Statical Methods for Business and Industry*”. En: (2003).
16. Fareed ud din, Atta-ur-Rehman Shah, Muhammad Ibrahim. Superior University, Lahore, Pakistan. “*Mobile-Phone-Based Remote Patient’s Vital Signs Monitoring and Automated Alerts*”. En: (2013).
17. Apache commons, “*Commons JEXL*”, <http://commons.apache.org/proper/commons-jexl/>

SMARTWATER para la detección de fugas de agua

Cintia Rodríguez¹, Ismael Mamani¹, Analía Herrera Cognetta²

¹ Estudiante Ingeniería Informática, Facultad de Ingeniería, Universidad Nacional de Jujuy,
Av. Bolivia 1239, 4600, San Salvador de Jujuy, Jujuy, Argentina
{cinturodriguez19, ismamamani22}@gmail.com

² Profesor Sistemas de Información, Ingeniería Informática, Facultad de Ingeniería,
Universidad Nacional de Jujuy, Av. Bolivia 1239, 4600, San Salvador de Jujuy, Argentina
aniherrera012@gmail.com

Resumen. El trabajo que se presenta trata sobre la investigación de las causas que provocan fugas de agua potable en los sistemas de abastecimiento domiciliario. Se propone entonces una solución a partir de la implementación de sistemas embebidos, para detección de la fugas y control, de manera eficiente, eficaz y a bajo costo.

Palabras clave: sistema de abastecimiento de agua potable, control y detección de fugas, sistema embebido

1 Introducción

Una de las principales preocupaciones a nivel mundial, consiste en tomar medidas adecuadas, que ayuden a fomentar la preservación y control de los recursos naturales no renovables, entre los cuales se destaca el agua dulce.

Un gran porcentaje de agua en el mundo, lo constituye el agua salada, que proviene de océanos, mares y aguas subterráneas, y sólo alrededor de un 3% del total de agua en la Tierra es dulce, necesaria para la vida de la mayoría de los organismos, plantas y principalmente la vida humana. Este porcentaje de agua dulce, disminuye con el paso de los años por el aumento de la población mundial, por lo que es de fundamental importancia cuidar este recurso natural. [García, 2015]

El consumo doméstico es una fuente crítica del agotamiento de este recurso. Las conexiones de redes de agua, brindan a la sociedad un medio para satisfacer sus necesidades básicas, por lo que su diseño y mantenimiento es importante, no solo para garantizar la satisfacción de una necesidad, sino para prevención de la pérdida.

Actualmente en la República Argentina, las pérdidas en las etapas de producción y distribución de agua son del orden del 40% del total producido [Greenpeace, 2017] por lo que es de sumo interés implementar métodos y técnicas que resuelvan o al menos mitiguen el problema.

Una fuga se puede considerar como la salida de agua no controlada, en cualquiera de los componentes del sistema de distribución de agua potable; con mayor frecuencia ocurre en uniones de tuberías, codos, roturas de conductos y válvulas.

2 Propuesta

El presente trabajo propone una alternativa de solución para detectar en forma eficiente, las fugas en las redes de conexión de agua potable. Se desarrollará entonces, el prototipo de un sistema embebido, que incluye tecnología hardware de bajo costo, en relación a los productos disponibles actualmente en el mercado, con un software para control y manejo adecuado.

En el desarrollo del prototipo se usarán tecnologías propias de sistemas embebidos, que permitirán, entre otras cosas, integrar diferentes sensores y actuadores para medir y detectar el nivel del caudal de agua, implementando tecnología Arduino, para mostrar el estado y resultados a través de una aplicación de escritorio.

3 Alcance

El Sistema detector de fugas para redes de abastecimientos de agua potable, se limitará a las siguientes funciones:

Detección precisa y en el menor tiempo posible, de fugas de agua en redes de abastecimiento de agua potable, en la localidad Villa Jardín de Reyes de la provincia de Jujuy.

Registración de caudal y variación de presión dentro del conducto.

Monitoreo del control y funcionamiento de válvulas.

Envío de alarmas a los operadores en caso de detectar un punto de fuga.

4 Antecedentes

A nivel nacional, se han implementado costosas herramientas para detectar los posibles puntos de fugas de agua dentro de las redes. Entre ellas se pueden nombrar a IDROLOC, que utiliza helio para la detección; o detectores de fugas que emplean instrumentos que trabajan con ultrasonido tales como el Hydrolux HL 50. No obstante, aunque estas herramientas resultan ser precisas, el tiempo dedicado a la detección es considerable.

La localización de fugas, basadas en ultrasonido, sin embargo, necesita previamente del aviso o informe de una persona testigo de la fuga.

La fuga producida en las redes subterráneas es visible en la superficie luego de transcurridas entre 8 y 12 horas, dependiendo del caso. Tal transcurso, sumado al tiempo en que se demora el personal de mantenimiento capacitado, en llegar a la zona de pérdida, determina que la cantidad de agua que se desperdicia es importante.

Por ello, la implementación del sistema propuesto, logrará a una detección de la fuga de manera inmediata, evitando mayores pérdidas.

5 Relevamiento inicial

Para lograr datos específicos con respecto a las fugas y su tratamiento, se realizaron visitas a la empresa Agua Potable S. E. de la Provincia de Jujuy y se entrevistó a profesionales, y trabajadores.

En lo que respecta a la provincia de Jujuy, según datos aportados por la empresa AGUA POTABLE S.E. las pérdidas en el año 2010 fueron del 59% y actualmente son del 49,60%, según se refiera a la cantidad de volumen facturado o a la cantidad de volumen producido.¹ En este documento se advierte que el factor de agua no contabilizada para el año 2017 fue del 49,60%. Este factor se refiere al agua que es bombeada o producida, pero que posteriormente se pierde por factores como el robo, fugas en tuberías o incluso una incorrecta medición de los datos.

5.1 Sistema de abastecimiento de agua potable

Es un conjunto de instalaciones capaces de captar, conducir, almacenar y distribuir agua a una localidad, cumpliendo condiciones de cantidad y calidad. Para instalar estas redes de abastecimiento de agua, se deben realizar estudios previos para determinar la capacidad de consumo y su crecimiento poblacional.

5.2 Partes del sistema de abastecimiento de agua

Un sistema de abastecimiento de agua, está conformado generalmente por: fuentes de abastecimiento (río, laguna, pozo, agua de lluvia); obra de captación: presa, bocatoma, captaciones; línea de conducción; planta de tratamiento; línea de impulsión y línea de aducción; reservorio y red de distribución (red matriz y red secundaria)

5.3 Sistema basado en GIS y herramientas de detección manual de fuga

La Empresa Agua Potable cuenta con un sistema GIS, que proporciona información sobre el consumo de cada cliente, ubica de manera precisa a cada cliente en la red de distribución, además de las redes de distribución próximas y las mediciones de caudales, entre otras funciones.

En caso de detectar una fuga de agua en las redes de abastecimiento, un sector de la empresa, controla el estado y realiza el informe correspondiente. Este informe, es dirigido al personal responsable de la reparación, lo que significa ya una demora considerable.

¹ Documento de audiencia pública- REVISION TARIFARIA INTEGRAL DE AGUA POTABLE S.E- 10/02/2017

5.4 Diagnóstico de la situación actual (problemas y deficiencias)

A pesar de que el sistema cuenta con información valiosa, ésta se encuentra dispersa y variada, por lo que es necesaria la integración de toda la información obtenida para realizar un análisis más preciso, y presentar resultados esperados, tales como: reportes de balance de masa y estadísticas de lectura.

Integrar y analizar la información permitirá facilitar el proceso para la toma de decisiones sobre el tratamiento y control de fugas.

Resulta entonces imprescindible, contar con una herramienta que permita predecir las posibles fugas de agua en sectores críticos, teniendo en cuenta los factores de antigüedad, material empleado para las redes, circulación de vehículos, y otros.

5.5 Clasificación de fugas de agua

Las fugas de agua se pueden clasificar según el tipo de detección:

Observables: Son aquellas que pueden ser detectadas por el agua que emerge a la superficie.

No Observables: Son aquellas que debido a la poca presión que ejercen quedan subyugadas en el suelo.

Según el daño producido en el material. Por fisura: son causadas principalmente debido al paso de tiempo y a la resistencia del material, provocadas por el paso de vehículos, construcciones, reparación de asfalto, reparaciones consecutivas de cañerías. Y se clasifican en: longitudinal, transversal y combinada.

Por orificio: una de sus principales causas tiene relación con la presión interna a la que se somete o bien al material interno que transporta (piedras, granillas, etc.).

5.6 Causas principales de fugas de agua

Entre las principales causas de fugas de agua, se pueden mencionar las siguientes: Circulación de vehículos de gran porte. Material no apto o de baja calidad usado en la instalación de conductos de agua. Sismos o hundimientos del suelo. Trabajos de excavación y perforación cercanos a la red. Alta presión de caudal para diámetros de cañería inadecuados. Material rocoso que transporta el agua dentro del conducto. Mala instalación: conexiones, tuberías expuestas a la superficie. Falta de mantenimiento de las redes. Desgaste o fatiga en los materiales de los conductos.

5.7 Requisitos a satisfacer o Información esperada

Se espera que el sistema cumpla con los siguientes requisitos:

Observación y Control de zonas con antecedentes de fugas de agua o catástrofes naturales.

Según estadísticas y balances anuales, estimar pérdidas causadas por fuga y obtener información actualizada.

Por cada punto de fuga detectado, determinar causa, problema, tipo de solución y su tiempo estimado (corto o mediano plazo).

Control y seguimiento estadístico de los caudales en las tuberías.

5.8 Control de Fuga

El control de fuga se puede realizar en dos sectores: Planta de tratamiento y Redes de distribución. El sistema propuesto estará abocado a las redes de distribución.

A continuación, se definen ecuaciones con el fin de determinar indicadores necesarios para el control de las fugas de agua:

$$VD = V \text{ Prod} + V \text{ Exportado} - V \text{ Importado.} \quad (1)$$

$$VP = V \text{ Malgastado} + V \text{ desviado por vía fraudulenta} + V \text{ de fuga de agua} + V \text{ por defecto de medición.} \quad (2)$$

$$VD = V \text{ Contabilizado} + V \text{ sin medir} + V \text{ de servicio de red} + VP. \quad (3)$$

$$IP (\%) = ((V \text{ medido} - V \text{ Prod}) / V \text{ Prod}) * 100. \quad (4)$$

$$V \text{ consumido} = V \text{ contabilizado} + V \text{ de servicio.} \quad (5)$$

$$RR: (V \text{ consumido} / V \text{ Prod}) * 100. \quad (6)$$

$$RH: (V \text{ Consumido} + V \text{ desviado por fraude} + V \text{ por defecto de medida} / VD) * 100. \quad (7)$$

V: Volumen; VD: Volumen distribuido; V Prod: Volumen Producido, VP: Volumen de Pérdida; IP: Índice de Pérdida, V medido: Volumen estimado para facturación, RR: Rendimiento de la Red; RH: Rendimiento Hidráulico.

5.9 Identificación y descripción de las funcionalidades que son necesarias desarrollar para el nuevo sistema.

El sistema a desarrollar contemplará las siguientes funcionalidades principales:

Localización automática de fugas de agua en redes de abastecimiento de agua potable: El sistema permitirá localizar un punto de fuga de agua de manera automática.

Registración de caudal y variación de presión dentro del conducto: El sistema permitirá registrar en una base de datos el caudal y la variación de presión de agua en tiempo real.

Detección y registración de fugas: El sistema permitirá detectar y registrar una fuga de agua.

Monitoreo del control y funcionamiento de válvulas: El sistema permitirá el control de válvulas en cada momento, verificando su funcionamiento.

Envío de alarmas a los operadores en caso de la detección de un punto de fuga: El sistema enviará mensajes a los usuarios responsables en caso de registrar y detectar una fuga de agua, por medio de una aplicación.

5.10 Herramientas elegidas para el desarrollo - Justificación:

Para el desarrollo del proyecto se elige la versión de Laravel 5.6, ya que cuenta con documentación y tutoriales disponibles en la red que pueden ser consultados, además Laravel facilitará el mantenimiento del producto, la integridad y el tiempo de desarrollo. El framework cuenta con una curva de aprendizaje relativamente baja, entre otras ventajas, además de la facilidad de uso y la practicidad, brinda seguridad al respaldar las consultas SQL pasándolas al patrón Modelo Vista Controlador.

Este tipo de framework también se adapta con un desarrollo a medida ágil, compatible con la metodología con la que se pretende abordar el proyecto. Aportando funcionalidades y variedades de bibliotecas diseñadas para que sean utilizadas directamente por los desarrolladores.

ARDUINO IDE es una herramienta esencial para el desarrollo del prototipo, ya que consta de placas Arduino, que deben programarse y este entorno de desarrollo es ideal, brinda documentación extensa para ser consultada, así como foros donde varios usuarios resuelven problemas típicos.

6 Etapa de Diseño

6.1 Planificación

Para analizar concretamente la red de distribución, será necesario reducir el área de estudio a sectores más pequeños, con superficie y consumo inferior a la media normal, para tener mayor control de los factores involucrados y así detectar y localizar las fugas.

Se usará el concepto DMA, definido como un área de un sistema de distribución de agua creada por el cierre de válvulas, que aísla y desconecta a ésta del resto de la red, de forma que pueda ser medida la cantidad de agua que entra y sale de esa área.

El flujo de agua puede ser analizado a partir de ese aislamiento para medir el nivel de las pérdidas. La clave para la división de una red en áreas menores consiste en que el cierre de las válvulas, para medir el flujo de entrada de un área, no afecte a la provisión de agua de ninguna otra área. Para ello es necesario que ninguna arteria principal de la red de agua forme parte.

Es por ello que se presenta como solución realizar un análisis aplicando un modelo basado en DMA donde se instalen los medidores de flujo en sus llaves de entrada y salida. A fin de poder regular el nivel de pérdida en cada DMA, para que la actividad de localización concreta de la pérdida se dirija a las zonas más afectadas, o con mayor probabilidad o tendencia de presentar pérdidas.

6.2 Zona de estudio

Localidad Villa Jardín de Reyes. Área Comprendida en calle Jorge Cafrune segmento entre calles Daniel Andreoli- Chango Toconas.

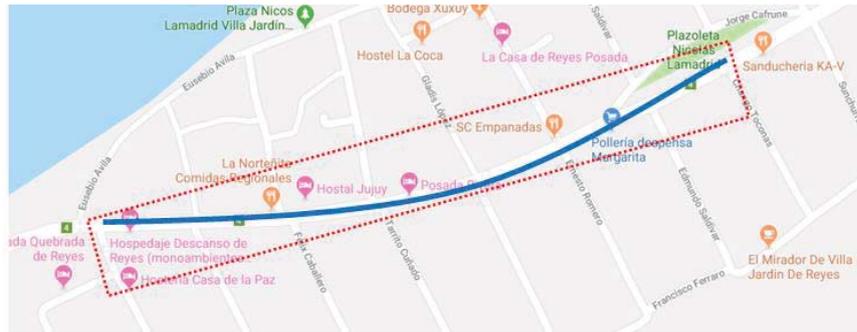


Fig. 1 Localización geográfica de la zona de estudio

En el área elegida se definen medidores, distinguiéndolos por las categorías: Doméstico, y No doméstico, dado que en el sector estudiado no existen clientes especiales (industrias, hospitales, entre otros).

En base a lo definido, se puede plantear un escenario con la siguiente estructura:

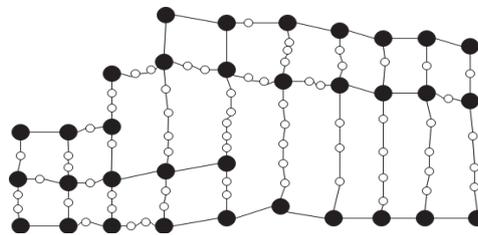


Fig. 2 Grafo G de medidores e intersecciones

Donde:

- Intersección del camino.
- Medidor.
- Longitud del camino.

Y cuya notación correspondiente es: $G(N,A)$, $N=\{Nd \cup Nnd\}$
 Nd: Nodo de consumo doméstico

Nnd: Nodo de consumo no domestico
 $A = \{(i,j)/i,j \in N\}$

6.3 Fase 1: Proceso de tratamiento de datos

A fin de no publicar datos privados, se realizará la simulación considerando dimensiones de red normalizada por estándares. Se utilizará el software libre EPANET versión 2.0, que permite dibujar la red y su comportamiento. Los datos serán preparados en un data set, con el cuidado suficiente para que el resultado sea compatible con la situación real. Se desarrollarán los procesos de ETL (*Extract, Transform and Load*) y se definirán todas las variables necesarias para la creación de los modelos.

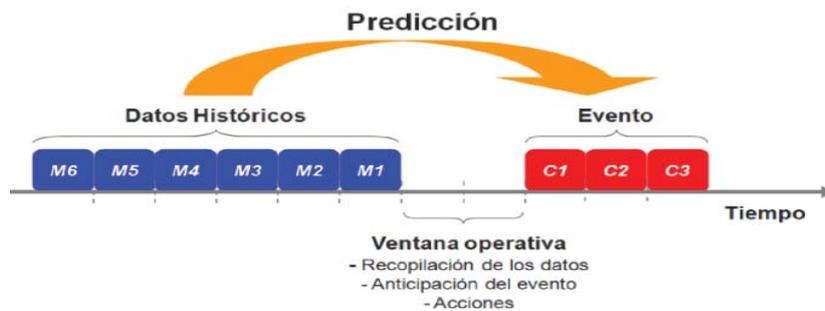


Fig. 3 Modelo de Predicción

6.4 Fase 2: Desarrollo del modelo de fuga

Para el desarrollo del modelo se considera como variable única la existencia de una fuga, se aplica entonces un modelo de regresión logística, que tomará el valor 1 si se produce la fuga en un periodo de tiempo, y el valor 0 en caso contrario. El modelo logístico dará en consecuencia, la siguiente representación de la probabilidad de fuga

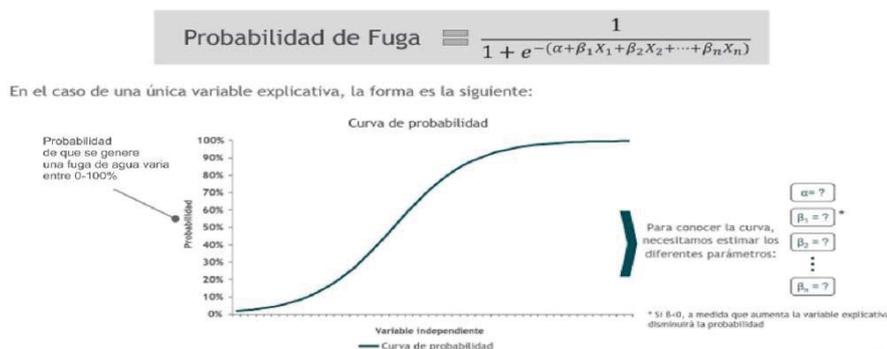


Fig. 4 Representación de la probabilidad de una fuga

En vista de la gran variedad de situaciones que pueden agregarse al análisis, tales como conexiones ilícitas y rotura de tuberías por actos delictivos, se agregarán variables relacionadas con el tipo de tubería (rigidez, diámetro, presión permitida), variables relacionadas con la temporada de consumo (tipo de consumo, nivel de presión, rendimiento de la red, etc.). Variables para representar el tipo de suelo, profundidad de cañería, ubicación de la red, entre otras.

Con este modelo, se busca determinar, en primer lugar qué variables afectan directamente para que se produzca la fuga, en segundo lugar su nivel de importancia y el grado de impacto.

Modelo 1: Control de las válvulas de presión. Gran parte de las fugas se debe a que estas generan una presión mayor a la soportada por los materiales, generando un desgaste y en consecuencia la rotura. Este caso se aplica a situaciones donde se mantiene tanto un sobreesfuerzo de presión al conducto como a cambios de presión.

Representación 1

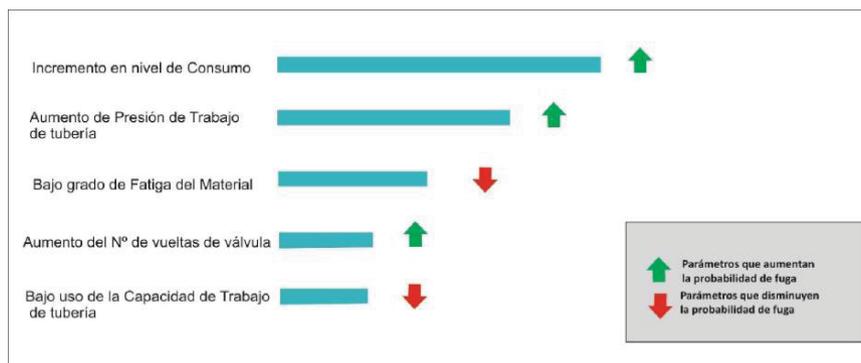


Fig. 5 Modelo de Predicción de fuga- Control de válvula

Modelo 2: Detectar los causantes de las fugas y mediante relaciones estimar el sector donde se halla la fuga. Para esto se estudiarán fenómenos naturales recientes o accidentes geográficos que influyan, en busca de determinar cuáles ocurren con mayor frecuencia, y cuales son responsables en la generación de la fuga.

Representación 2: Modelo de Detección de fuga

Se empleó el método de detección más simple que, consiste en comparar los caudales de entrada y salida obtenidos durante el muestreo, esto es:

$$Q_{\text{perdida}}(k) = Q_{\text{salida}}(k) - Q_{\text{entrada}}(k). \quad (2)$$

Existe pérdida si Q_{perdida} supera un valor umbral. Una mejora notable se obtiene si se emplean los datos de los caudales filtrados. En caso de pérdida, el caudal de entrada aumenta y el de salida disminuye, este comportamiento se conoce como firma de la falla.

Parámetros: Tipo de Consumo: Consumo doméstico-Material de tubería: policloruro de vinilo PVC-Circuito de red de distribución: abierto (tipo espina de pescado)-Nivel de suelo: uniforme.

7 Conclusión, situación actual y fases futuras

El sistema propuesto aun se encuentra en fase de desarrollo, sin embargo durante la investigación de los temas que son afines al mismo, fue posible adquirir y ampliar conocimientos, así mismo fue imprescindible realizar un análisis profundo para sintetizar los conceptos y proponer una solución eficaz en cuanto a tiempo de localización y detección de fugas, considerando el aspecto económico y utilizando nuevas tecnologías que se encuentran en el mercado.

En cuanto a los modelos que se presentaron en la sección de diseño:

El Modelo 1 presentado, fue aprobado por especialistas de la empresa, con experiencia en la temática.

El Modelo 2 está en estudio para su aprobación, también por especialistas de la empresa.

A futuro se espera cumplir con las siguientes fases de desarrollo:

Fase 3: Revisión del modelo. Será sometido a una serie de test estadísticos y validado sobre una muestra diferente a la que se utiliza inicialmente.

Fase 4: Implementación del modelo en los sistemas de la empresa.

Referencias

1. García, I El Mapa de Agua en la Argentina, <http://fabiangarcia.com.ar/el-mapa-del-agua-en-argentina/> (2015)
2. Greenpeace Cambio Climático <http://www.greenpeace.org/argentina/es/campanas/cambio-climatico/> (2017)
3. The Internet of Things. IoT Smart City – What is Smart Water?. The Internet of Things. <http://www.infiniteinformationtechnology.com/iot-smart-city-what-is-smart-water.> (2019)
4. Roger S. Pressman, Ingeniería del Software Un enfoque Práctico, Editorial McGraw Hill (2010)
5. Smart Water Management using Internet of Things Technologies. <https://iiot-world.com/connected-industry/smart-water-management-using-internet-of-things-technologies/> (2019).
6. Rosas Quiterio P. (2017) Desarrollo de un sistema para detección de fugas para tuberías horizontales en tiempo real por medio de gradientes de presión (Tesis doctoral). Instituto Politécnico Nacional, SEPI, ESIME Zacatenco, México.
7. Lloyd Owen, D. A. Smart Water Technologies and Techniques (1st ed.). Wiley-Blackwell (2018)
8. Arduino - Introduction <https://www.arduino.cc/en/Guide/Introduction> (2019)
9. Sanchez, J Laravel: ventajas del framework PHP de moda. <https://www.freelancer.com.ar/community/articles/ventajas-del-framework-moda-laravel> (2019)
10. Llamas, L. Medir Distancia con Arduino y Sensor de ultrasonidos Hc-Sr04 <https://www.luisllamas.es/medir-distancia-con-arduino-y-sensor-de-ultrasonidos-hc-sr04/>.(2015)
11. Llamas, L. Medir caudal y consumo de agua con arduino y caudalímetro. <https://www.luisllamas.es/caudal-consumo-de-agua-con-arduino-y-caudalimetro/>

**CACIC 2019**

Congreso Argentino de Ciencias de la Computación

X Workshop

Procesamiento de Señales y Sistemas de Tiempo

Coordinadores

Villagarcia Wanza, Horacio (UNLP)
Fрати, Emanuel (UNdeC)
Ierache, Jorge (UM)

Comité Científico

Acosta, Nelson (Argentina)
Bría, Oscar (Argentina)
Encinas, Diego (Argentina)
Lugo, Jorge (Argentina)

Maidana, Carlos (Argentina)
Ramón, Hugo (Argentina)
Riba, Alberto Eduardo (Argentina)
Romero, Fernando (Argentina)

Runco, Jorge (Argentina)
Suppi, Remo (España)

Temas

Señales y Arquitecturas orientadas al tratamiento de señales.
Sensores. Redes de Sensores.
Sistemas embebidos. Aplicaciones IoT.
Robótica Autónoma. Robótica situada. Drones.
Sistemas Industriales. Industrias 4.0. Manufactura Integrada por Computadora. Visión por computadora
Modelos de Sistemas de Tiempo Real.
Ingeniería de Software para STR.
Verificación y Validación de STR.
Redes, protocolos y Seguridad en STR.
Sistemas de Tiempo Real duro.

Cloud/Edge Robotics: Navegación autónoma de Auto-Robot y Cuadricoptero

Manuel Costanzo¹, Marcos Boggia¹, Ismael Rodriguez¹, and Armando De Giusti^{1,2}

¹Instituto de Investigación en Informática LIDI, Facultad de Informática, UNLP-CIC

²Consejo Nacional de Investigaciones Científicas y Técnicas (CONICET)

¹{mcostanzo, mboggia, ismael, degiusti}@lidi.info.unlp.edu.ar

Abstract: Este trabajo presenta el diseño, desarrollo y despliegue de un sistema Multi-Robots, compuesto por un robot que simula un vehículo tradicional, como así también, un Dron capaz de capturar imágenes aéreas; ambos conectados al Cloud de Amazon. Se detallan los prototipos desarrollados y el protocolo de comunicación utilizado; se mencionan los algoritmos implementados para el procesamiento y reconocimiento de objetos en imágenes, la simulación y planificación de caminos óptimos, junto con los métodos de transformación de los mismos y la determinación de los movimientos, con el fin de que el vehículo no tripulado pueda alcanzar un punto de destino, siguiendo las directivas desde el Cloud.

Keywords: Cloud Robotics, Edge Robotics, Robótica, Amazon Web Services, IoT, Procesamiento de imágenes, planificación de caminos, Multi-Robot, RRT

1. Introducción

Considerando la gran influencia de la robótica en la sociedad y la industria, y la diversidad de servicios ofrecidos por robots, nos encontramos que los mismos presentan grandes limitaciones en consumo de energía, poder de cómputo, capacidad de almacenamiento, toma de decisiones, tareas cognitivas, entre otras. Desde el año 2010, comenzaron a surgir proyectos de investigación (ej: RoboEarth [1]), que integran las tecnologías de Cloud con los sistemas de robots. Es así, que James Kuffner propone el concepto de Cloud Robotics, basado en combinar las tecnologías de robots con el paradigma de Cloud Computing [2].

La idea de Cloud Robotics, es permitir por medio de aplicaciones tratar los datos de los componentes de hardware del robot (sensores, actuadores, cámaras, memoria, etc.), sin importar las limitaciones de cómputo y almacenamiento de las placas de desarrollo de los robots [3]. En otras palabras, este concepto permite a los robots obtener resultados de tareas de cómputo intensivo, tales como: procesamiento de imágenes, reconocimiento de voz, determinación de rutas, confección de mapas, acciones cognitivas, etc., sin tratamiento local, sino en el Cloud [4].

Debido al alto incremento de los robots conectados y a la gran cantidad de información generada, se han presentado limitaciones con respecto a problemas de latencia, tiempos de respuestas variables, calidad de servicio, privacidad y seguridad, que han dado origen a la extensión del paradigma, realizando todo el cómputo que se requiera y sea factible de ejecutar, en el extremo de la red; a esto se lo denomina Edge Robotics [5] [6].

Estos paradigmas brindan la capacidad de establecer escenarios para Multi-Robots, donde cada robot se integra de un hardware mínimo con conectividad inalámbrica, y los datos de los sensores y la adquisición de imágenes se procesarán de forma local, intermedia o en el Cloud, según la capacidad y conveniencia del uso de los recursos [7].

2. Conceptos elementales

■ Amazon Web Services

Es una plataforma de Cloud Público, que provee “Infraestructura como servicio”; a través de la tecnología de virtualización. De sus servicios más destacados, utilizamos “Elastic Compute Cloud” (EC2) y “AWS Internet of Things” (AWS IoT) [8].

■ Internet of Things (IoT)

Es un nuevo paradigma que considera a los dispositivos como “objetos” conectados a Internet, utilizando protocolos de comunicación estándar [9].

■ Protocolo Message Queuing Telemetry Transport (MQTT)

Es un protocolo de comunicación ligero [10] [11], diseñado para tolerar conexiones intermitentes y reducir los requisitos de ancho de banda de red; soporta comunicación segura con TLS y calidad de servicio (QoS).

Está basado en el modelo publicación/suscripción, como muestra la figura siguiente:

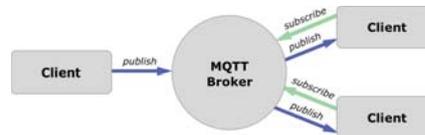


Figura 1: Comunicación MQTT

■ Amazon Web Services Internet of Things (AWS IoT)

Es un servicio que proporciona AWS con el fin de conectar, administrar y operar grandes conjuntos de dispositivos [12]. Para cada dispositivo, se emite un certificado y un par de llaves privada-pública, que junto al certificado de la Entidad Certificante (CA), permite conectar el dispositivo al servicio de AWS IoT, en forma segura.

■ Node-RED

Es una herramienta de programación visual WEB, orientada a flujos que son almacenados en formato JSON, y permite comunicar hardware y servicios de forma rápida y escalable, utilizando nodos MQTT sobre conexiones TLS [13]. En la figura 2 puede observarse un ejemplo de uno de los flujos Node-RED utilizado en el Auto-Robot.

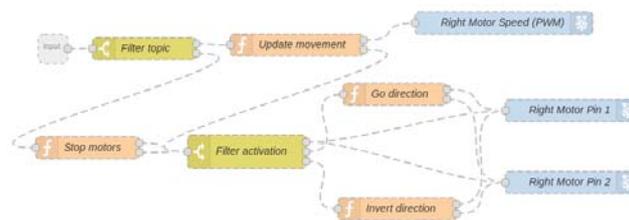


Figura 2: Ejemplo de flujo Node-RED del auto robot

3. Trabajo experimental

El problema a resolver en este trabajo consiste en guiar a un dispositivo de vuelo no tripulado (Dron) para sobrevolar un escenario compuesto con elementos que simulan objetos del mundo real. Cada objeto se identifica por medio de figuras geométricas; el Dron realizará capturas de imágenes del entorno mencionado, con el fin de detectar un vehículo de 4 ruedas no tripulado (Auto-Robot), un punto destino a alcanzar por este último y diversos obstáculos que se deberán sortear, para que el Auto-Robot llegue al destino. Las imágenes adquiridas requieren ser procesadas en tiempo real, con el objeto de reconocer cada uno de los elementos. Luego, se necesita planificar el camino óptimo que conecte el Auto-Robot con el punto de destino; y por último, se precisa determinar los desplazamientos que el Auto-Robot deberá efectuar para abordar el punto de destino final.

Además, se requiere diseñar una infraestructura que permita la interconexión de los diferentes dispositivos, en forma segura y con tiempos de respuesta aceptables.

Para el presente trabajo se utilizarán dispositivos de bajo recursos con limitaciones de cómputo. Tales limitaciones se subsanarán aprovechando los beneficios del Cloud.

Sin embargo, existen ocasiones en las cuales no será posible delegar la toma de decisión al Cloud, ya que existen acciones de máxima prioridad que el Auto-Robot debe ejecutar en forma inmediata; como son de común conocimiento las desventajas que proporciona la “dependencia de la red” (atenuación en la latencia y aún pérdida de conectividad), que puede provocar que el Auto-Robot se desvíe del camino y colisione con los obstáculos. Para evitar esta problemática es deseable aplicar los conceptos del paradigma de Edge Robotics, con el fin de procesar la información de los sensores de colisión en forma local y tomar la decisión de avanzar o frenar inmediatamente.

Asimismo, es deseable contar con un dispositivo intermedio que permita direccionar las imágenes que adquirirá el Dron hacia el Cloud, aprovechando los beneficios de la conectividad Ethernet con respecto a WiFi; como así también, controlar la navegación del Dron en forma segura.

Para alcanzar los requerimientos antes mencionados, se ha confeccionado la estructura del sistema con los siguientes cuatro componentes: el Auto-Robot, el Dron, el dispositivo intermedio (DI) y la instancia de cómputo EC2 en el Cloud de AWS (IEC2).

Sobre la componente IEC2 se ha desarrollado una interfaz WEB, que permite al usuario dar inicio al sistema y visualizar el comportamiento del mismo. Al inicio del sistema, la IEC2 notifica al componente DI, y este último es quien se encarga de controlar la navegación del Dron. Todas las imágenes capturadas, son comprimidas y posteriormente redireccionadas por el componente DI a la IEC2 vía Ethernet.

En la IEC2, cada imagen se descomprime y se procesa para detectar los diferentes objetos y características del escenario. En base a los datos obtenidos, se procede a planificar el camino óptimo, para posteriormente calcular y enviar el movimiento que el Auto-Robot deberá efectuar para desplazarse.

Por otro lado, el Auto-Robot aplica el movimiento recibido; cabe destacar aquí, que a lo largo de la vida del sistema, el Auto-Robot adquiere y procesa los datos del sensor de colisión de forma local, con el fin de evitar una incidencia.

Una vez que el Auto-Robot aborda el punto de destino, es la IEC2 quien se encarga de detectar dicho evento y notificar, tanto al Auto-Robot como al DI, que el sistema debe finalizar. Así, el Auto-Robot concluye su desplazamiento apagando los motores y el sensor de colisión; mientras que el Dron finaliza la captura de imágenes y retorna a su posición original para finalmente aterrizar.

Durante todo el tiempo de ejecución del sistema, el usuario puede visualizar en la interfaz WEB todos los eventos ocurridos en tiempo real, como así también la traza calculada junto

con las capturas de imágenes.

Las siguientes secciones detallan el desarrollo efectuado en cada uno de los componentes, como la integración de los mismos para lograr el trabajo propuesto.

3.1. Auto-Robot

Se ha confeccionado un prototipo que ejecuta los movimientos recibidos desde la IEC2, a través del protocolo MQTT. También, se ha desarrollado y provisto al Auto-Robot de toda la lógica para traccionar los motores ante las órdenes recibidas y procesar los datos extraídos del sensor de colisión, para evitar chocar con algún obstáculo.

El prototipo se abstrae de la capa lógica necesaria para la planificación del camino y el reconocimiento del escenario. El vehículo se restringe a seguir las órdenes provenientes de la IEC2; esto permite confeccionar Robots económicos, de bajo recursos y consumo energético.

3.1.1. Características y componentes

Se detallan las diferentes componentes y características pertinentes al Auto-Robot:

- **Estructura:** se ha adaptado un chasis de 4 ruedas, que simula un vehículo tradicional. Sus dimensiones son: 248mm x 146mm x 70mm. Proporciona tracción trasera (RWD) mediante un motor y dirección delantera a través de un servo que orienta las ruedas.
- **Raspberry Pi 3:** placa de bajo costo de desarrollo que brinda conexión WiFi, almacenamiento en memoria MicroSD y alimentación de energía de 5v.
- **RaspiRobot Board V3:** módulo que brinda la interfaz de conexión entre la RPi y los motores; suministra energía a estos componentes.
- **Fuente conmutada Step-Down:** regula la energía de salida de 1,25 a 36V.
- **Sensor Ultrasonido HC-SR04:** sensor de proximidad para evitar colisiones.

3.1.2. Funcionalidad

El vehículo es el actor que se encarga de cumplir con los movimientos enviados por la IEC2. Para esto, se diseñó un flujo de programación en Node-RED, el cual permite modificar el comportamiento del dispositivo, ya sea alterando la dirección, orientando sus ruedas delanteras, o el desplazamiento mediante la activación/desactivación del motor de tracción trasera. Asimismo, el flujo permite modificar la velocidad del motor.

Además, el Auto-Robot procesa los datos obtenidos de su sensor de colisión y toma la decisión de desactivar el motor cuando se detecta que el dispositivo se encuentra a menos de 5 centímetros de un obstáculo.



Figura 3: Sistema de direccionamiento del Auto-Robot



Figura 4: Auto-Robot de completo



Figura 5: Sistema de motor del Auto-Robot

3.2. Dron

Para integrar al sistema propuesto, se ha adquirido un Dron de marca Parrot, modelo Bebop 1 (figura 6), que se utiliza para capturar y enviar imágenes del escenario a la IEC2.



Figura 6: Parrot Bebop 1.

El Dron proporciona video en tiempo real comprimido en formato H.264, para posteriormente compartir las imágenes por medio de paquetes Real-Time Transfer Protocol (RTP). El DI se compone por una placa de desarrollo Raspberry Pi 3 con conexión a Internet vía Ethernet, lo que acelera la transmisión de los paquetes (figura 7).

Cuando el Dron recibe la notificación de inicio del sistema, se eleva a una altura preestablecida (1,5 metros) y se desplaza hacia el centro del escenario. El Dron provee de una cámara que tiene una visión de 90° orientada al suelo. Al recibir la notificación de finalización del sistema, el Dron retorna a su posición inicial y termina con la transmisión del video.

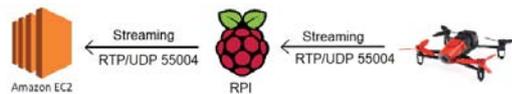


Figura 7: Camino que realiza el streaming de video.

Con el propósito de controlar los movimientos del Dron, se utilizó la librería desarrollada por la empresa Parrot, que permite ejecutar comandos en el dispositivo. Para ello, el DI recibe vía MQTT las directivas desde la nube y este los traduce en comandos que envía al Dron. En la figura 8 se puede observar la conexión del Dron con la IEC2.



Figura 8: Conexión del Dron, DI y la IEC2.

3.3. Instancia EC2 en AWS (IEC2)

Se utilizó una instancia del servicio EC2 sobre el Cloud de AWS, alojada en EE.UU. La instancia adquirida es del tipo T3.small, posee 2 CPU Core, 2GB de RAM y 8GB de almacenamiento SSD. Se desplegó la instalación del S.O. Linux Ubuntu 18.04 LTS.

Dicha instancia es utilizada para descomprimir y procesar las imágenes del entorno, como también para planificar y actualizar las rutas óptimas y calcular los movimientos a realizar por el vehículo. Además, brinda una interfaz WEB que permite gestionar y controlar el sistema. Asimismo, se desarrolló un simulador que permite generar diversos escenarios con el fin de testear la planificación.

Otro aspecto importante de la instancia, es que se encarga de la comunicación con los dispositivos mediante el protocolo de comunicación MQTT, que utiliza un canal de comunicación seguro sobre TLS, basado en certificados de seguridad.

3.3.1. Procesamiento de imágenes

El objetivo de este procesamiento es detectar los elementos del escenario, es decir, encontrar en la imagen tanto el Auto-Robot, como el punto de destino y los obstáculos; y obtener las propiedades de los mismos (coordenadas del centro, radio, orientación, etc).

Los elementos nombrados anteriormente son identificados con figuras geométricas para simplificar el proceso de detección. El punto de destino es representado con un círculo, los obstáculos con un triángulo y el Auto-Robot con un cuadrado o rectángulo, con un círculo interior, que denota su orientación. El procesamiento de las imágenes se realiza mediante el uso de la librería OpenCV v3.0.

En una primer instancia, se obtiene una imagen del streaming de video (figura 9), la misma se convierte a escala grises (figura 10) y se procede a aplicar una colección de filtros con tal de facilitar la detección de las figuras.

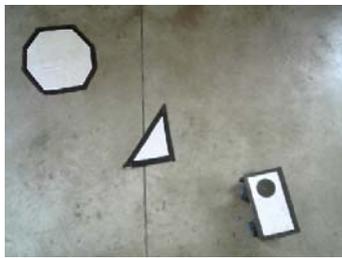


Figura 9: Imagen original

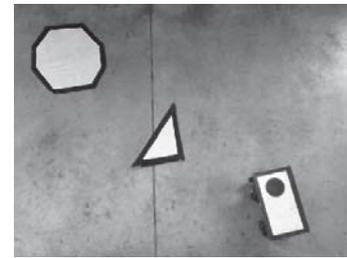


Figura 10: Imagen en escala de grises

Se aplica un primer filtro de suavizado (figura 11) y luego se procede a la aplicación de un segundo filtro que detecte bordes, como se puede observar en las figuras 12 y 13

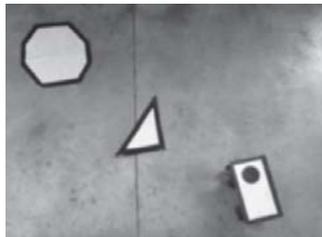


Figura 11: Imagen con suavizado Gaussiano.

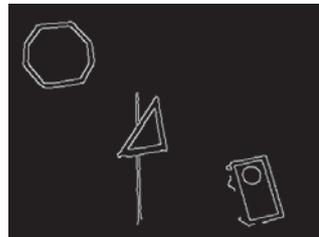


Figura 12: Imagen con filtro "Canny Edge"

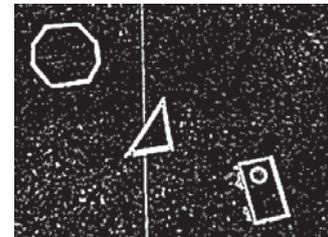


Figura 13: Imagen con filtro "Threshold"

Posteriormente, se obtienen los contornos y se itera por cada uno de ellos verificando la cantidad de vértices que posea. En caso de tener tres lados, se considera un triángulo, cuatro lados es un cuadrado o rectángulo y más de cinco es un círculo.

3.3.2. Planificación de caminos y cálculo de movimientos

La determinación de trayectorias consiste en encontrar aquella ruta que permita al Auto-Robot desplazarse desde su posición inicial hasta llegar al destino, sorteando los obstáculos presentes en el escenario. Para lograr este objetivo, es necesario utilizar un algoritmo que

detecte la ruta a realizar y un método de transformación y conversión que traduzca dicha ruta en movimientos entendibles por el Auto-Robot.

3.3.3. Planificación RRT

La técnica seleccionada es la denominada Rapidly Exploring Random Tree (RRT).

Un algoritmo RRT se basa en la construcción de un árbol de configuraciones que aumenta desde un punto origen hacia un destino preestablecido. Este método tiene como objetivo generar un árbol de exploración, que permita cubrir de manera uniforme todo el espacio de configuraciones libres de colisión.

El algoritmo recibe como entradas el punto de inicio y fin, el tiempo máximo, y la distancia entre los nodos (es decir, la longitud en píxeles de las ramas del árbol). Como resultado final, se obtendrá un árbol (conjunto de puntos), que representa el camino encontrado [14].

En las figuras 14 y 15, se puede observar el resultado de simular la ejecución del algoritmo RRT, tomando como punto de inicio la ubicación del vehículo (punto azul), el punto de destino (punto verde) y los obstáculos de por medio (puntos rojos). Como resultado de la planificación se obtiene una colección de nodos que representan una coordenada (x, y) .

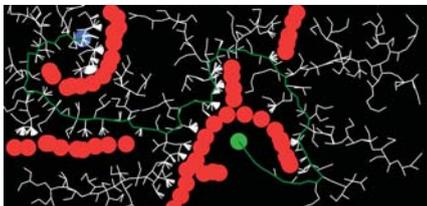


Figura 14: Ejemplo 1: de planificación RRT

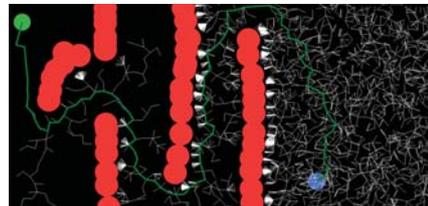


Figura 15: Ejemplo 2: de planificación RRT

3.3.4. Optimización del camino

El algoritmo RRT, como se mencionó, retorna un árbol que consiste en una lista de puntos (x, y) . Como se puede apreciar en las figuras 14 y 15, el camino resultante cuenta con una gran variedad de desvíos innecesarios, lo que repercutirá en un mayor esfuerzo por parte del Auto-Robot al momento de desplazarse.

Para solucionar este problema, se propone un método de optimización del camino que consiste en crear conexiones lo más longitudinales posibles entre dos nodos extremos, removiendo como consecuencia los puntos que no forman parte de esa conectividad. El nodo más lejano se obtiene generando una búsqueda dicotómica en la lista de coordenadas.

El resultado de esta operación consiste en una reducción de los nodos del árbol RRT, como muestran las figuras 16 y 17.

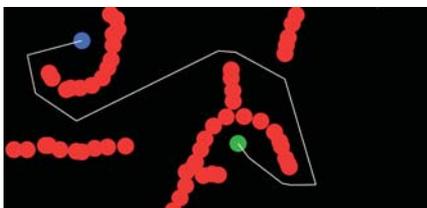


Figura 16: Ejemplo 1: camino optimizado

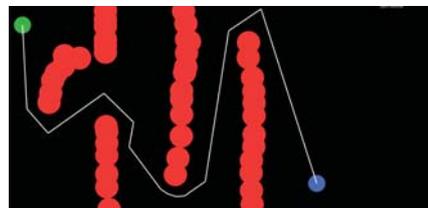


Figura 17: Ejemplo 2: camino optimizado

3.3.5. Suavizado del camino

Si bien la ruta obtenida en el paso anterior contiene menos inflexiones, es evidente que el Auto-Robot no podrá cumplir con exactitud la trayectoria, como consecuencia de los desvíos pronunciados que derivan de la unión de las diferentes partes del camino.

Para resolver este inconveniente, se aplica una función matemática de interpolación cúbica (o del inglés “cubic spline”), que permite suavizar el camino y posibilita que el vehículo se oriente en todo momento, para no tener que direccionar una gran cantidad de grados y evita realizar una curva pronunciada.

Esta función trata los puntos de la ruta anterior y retorna una nueva colección de puntos que conforman la curva suavizada. El algoritmo utilizado permite establecer la distancia en la que debe estar cada uno de los puntos de la curva, como se observa en la figura 18 y 19.

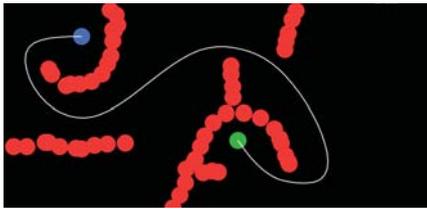


Figura 18: Ejemplo 1: trayectoria suavizada

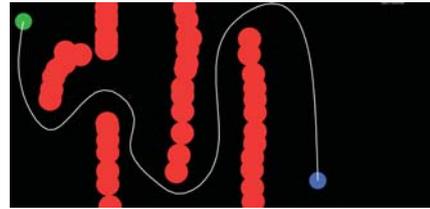


Figura 19: Ejemplo 2: trayectoria suavizada

3.3.6. Cálculo de la maniobra

Luego de obtener la ruta resultante, se procede a calcular el movimiento que el Auto-Robot deberá ejecutar. Dicho valor está conformado por el grado de giro ($[-180^\circ, 180^\circ]$) y el sentido de desplazamiento (1 hacia delante, -1 hacia atrás, 0 freno).

Se implementó una técnica que consiste en incorporar a nivel lógico un paralelogramo en los límites del Auto-Robot. Este polígono abarca el área de movimiento del vehículo, por lo que se requiere conocer el ángulo de giro máximo que posee el auto (este dato es enviado por el Auto-Robot en la interacción inicial previo al comienzo de la ejecución).

Como se puede observar en las figuras 20 y 21, si dentro del área de colisión se detecta que existe un obstáculo, se intenta revertir el sentido, siempre en función del punto de referencia calculado previamente.



Figura 20: Sin presencia de obstáculo en el camino

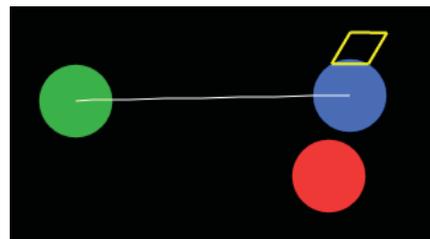


Figura 21: Con presencia de obstáculo en el camino

3.3.7. Plataforma WEB

Se desarrolló una interfaz WEB que permite al usuario interactuar con el sistema. Por un lado, se despliega una sección que posibilita simular la determinación de caminos, con

la propiedad de configurar diferentes aspectos, como por ejemplo el radio de los objetos, el grado al que está orientado el Auto-Robot, etc.

Por otro lado, se presenta una sección denominada tiempo real o “live”, utilizada para dar comienzo con la ejecución del sistema y visualizar en todo momento los eventos que ocurren en el transcurso del mismo, como también el streaming de video original y luego de procesado, junto con la traza resultante.

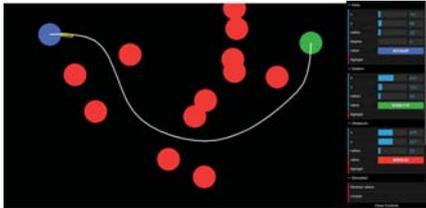


Figura 22: Interfaz de simulación

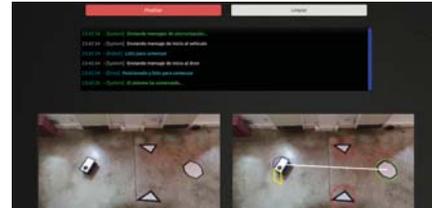


Figura 23: Interfaz live

4. Resultados obtenidos y líneas de trabajo a futuro

En este trabajo de investigación se analizó, diseñó, desarrolló e implementó un sistema de Multi-Robot conectados al Cloud público de Amazon Web Service. Para esto se aplicaron los conceptos de los paradigmas de Cloud y Edge Robotics.

Se confeccionó un Auto-Robot, un vehículo de 4 ruedas no tripulado, que simula un auto de tracción trasera tradicional. Se analizó, diseñó, desarrolló y se implementó la lógica que le otorga la capacidad de desplazarse para alcanzar un punto de destino, mediante las órdenes provenientes desde el Cloud; como así también, evitar toda posible colisión por medio del procesamiento de datos obtenidos por un sensor de colisión en forma local.

El sistema también lo integra un Dron no tripulado, que sobrevuela el escenario y envía imágenes al Cloud. Se adhirió un dispositivo intermedio, que permite gestionar y controlar el Dron bajo las decisiones y ordenes efectuadas en la instancia de cómputo del Cloud (IEC2). Con respecto al medio de conectividad a Internet, se optó por la comunicación vía Ethernet, pues permite reducir la latencia lo menor posible en el envío de las imágenes a la nube.

Se adquirió y desplegó una instancia de servidor en EC2 de AWS, con el fin de delegar la intensidad del cómputo del sistema a los recursos ofrecidos por la nube; esto ha posibilitado la confección de robots equipados con un hardware de requerimientos mínimos, lo cual permite soluciones más económicas y de menor consumo energético.

También, se desarrolló los algoritmos que permiten procesar imágenes para detectar las figuras geométricas que identifican a cada objeto del escenario. Juntamente, se ha probado y seleccionado diversos filtros de imágenes, aptos para escenarios con diferente luminosidad.

Así mismo, se diseñó y desarrolló un método de planificación de caminos que permite obtener la ruta óptima desde un punto de inicio hasta un punto de destino determinado. Para alcanzar tal objetivo, se utilizó el algoritmo de planificación RRT, el cual se adaptó a los requerimientos del sistema, y se desarrolló un método de optimización del camino, que simplifica la trayectoria a transitar. Además, se diseñó un algoritmo de suavización de ruta, que permite al Auto-Robot ejecutar un desplazamiento con la mayor precisión posible, a pesar de las limitaciones físicas que posee el vehículo.

Por otro lado, se confeccionó un algoritmo que permite calcular el movimiento a realizar por el Auto-Robot, contemplando un área de posible colisión con el fin de determinar la presencia de un obstáculo en su camino.

Por último, se implementó una interfaz WEB que brinda distintas funcionalidad, entre ellas, permite simular planificaciones de escenarios creados por el usuario o en base a imáge-

nes reales. También, permite que el usuario tenga el control de iniciar o finalizar ejecución del sistema, visualizando en tiempo real todos los eventos que se generan, como así también, las imágenes del escenario, que también incluyen la detección de los objetos junto con la traza a seguir por el Auto-Robot.

Es deseable destacar, que ante la necesidad del sistema de operar en tiempo real, se han implementado algoritmos lo más óptimos, eficientes y precisos posible.

Podemos concluir que la aplicación de los conceptos de Cloud y Edge Robotics, colaboraron a la confección y despliegue del sistema propuesto; en base a los mismos, se ha optado por realizar el procesamiento de los datos en forma local, intermedia y la nube, conforme a la necesidad, capacidad y la conveniencia del cómputo.

Nos hemos propuesto como trabajo a futuro, dotar al sistema con la capacidad de procesar y reconocer imágenes de objetos reales, con el fin de descartar la identificación de los mismos en base a figuras geométricas. Por otro lado, adaptar el sistema para integrar más de un Auto-Robot, y que los mismos trabajen colaborativamente; para esto, es deseable incorporar los conceptos del paradigma de Fog Robotics, añadiendo una cámara LIDAR que brinde la capacidad de mapear el escenario y compartirlo a otros Auto-Robots.

Referencias

- [1] *RoboEarth*. <http://www.roboearth.org>.
- [2] Kuffner J. «Cloud-enabled robots». En: *IEEE-RAS International Conference on Humanoid Robot* (2010).
- [3] Wang L.; Liu M.; Meng M.; Siegwart R. «Towards Real-Time Multi-Sensor Information Retrieval in Cloud Robotic System». En: *IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems (MFI)* (2012).
- [4] Costanzo M.; Boggia M.; Rodriguez I.; De Giusti A. «CLOUD ROBOTICS: Navegación de un vehículo autónomo en un entorno con obstáculos». En: *XII Workshop de Procesamiento Distribuido y Paralelo (WPDP) – XXII Congreso Argentino de Ciencias de la Computación (CACIC2018)* (2018).
- [5] Tanwani A.K.; Mor N.; Kubiawicz J.; Gonzalez J.E. «A Fog Robotics Approach to Deep Robot Learning: Application to Object Recognition and Grasp Planning in Surface Decluttering». En: *IEEE International Conference on Robotics and Automation, ICRA, 2019* (2019).
- [6] Antevski K.; Groshev M.; Cominardi L.; Bernardos C.; Mourad A.; Gazda R. «Enhancing Edge robotics through the use of context information». En: (2018).
- [7] Turnbull L. «Cloud Robotics: Formation Control of a Multi Robot System Utilizing Cloud Infrastructure». En: *Proceedings of IEEE – Southeastcon* (2013).
- [8] *Amazon Web Services*. <https://aws.amazon.com/es/>.
- [9] Atzori L.; Iera A. «The Internet of Things: A survey». En: *ELSEVIER Journal Computer Networks* (2010).
- [10] *Protocolo MQTT*. <https://www.oasis-open.org>.
- [11] *OASIS MQTT*. <http://mqtt.org>.
- [12] *Amazon Internet Of Things*. <https://aws.amazon.com/es/iot/>.
- [13] *Node-RED*. <https://nodered.org>. Último acceso 30 de Noviembre 2018.
- [14] Zulfiqar Habib Iram Noreen Amna Khan. «A Comparison of RRT, RRT* and RRT*-Smart Path Planning Algorithms». En: (2016).

Development and implementation of a smart greenhouse

Donadio, Maximiliano C.¹; Garcia-Martinez, Nicolas¹; Vivas, Luis¹; Britos, Paola V.¹

¹ Universidad Nacional de Río Negro – Sede Atlántica – Río Negro - Argentina
maxdonadio@hotmail.com; {ngarciam, lvivas, pbritos}@unrn.edu.ar

Abstract. A smart greenhouse was developed at Laboratorio de Informatica Aplicada with the department of Agronomic Engineering at the University of Río Negro, it must provide real-time measurements of parameters such as humidity, temperature and luminosity; it also must allow manual and automatic control for actuators such as heaters, sprinklers and fans based on user input. In order to fulfill the aforementioned requirements the following actions were performed: (a) Design and implementation of a webpage to communicate with and control the greenhouse and, (b) Development: at first a model using Object Oriented Programming was implemented in an Arduino Mega board equipped with an ethernet shield; posteriorly, given that Arduino could not fulfill the necessary tasks, it was decided to develop a second prototype using a Raspberry Pi 2 Model B+ board along with completely new software programmed in Python 3.

Keywords: Python, Actuators, Sensors, Robotics, Greenhouse.

1. Introduction

The need for a smart greenhouse arose from the difficulties concerning research that were presented by the department of Agronomical Engineering at the university, monitoring a greenhouse in real time can be a difficult task which may complicate research and crop production; which is the reason an effective and efficient tool was required.

At Laboratorio de Informática Aplicada (LIA) a project to develop a smart greenhouse was started, several requirements were established, among them are temperature, humidity and luminosity control in real time, collecting and storing data for later analysis; as well as activating or deactivating different actuators manually and automatically, changes in actuators such as the aforementioned must be documented as well to provide more information for future research, it is in this way that the user could observe not only the greenhouse's status at any moment in time but also know which factors were present at said moment. [1] [2] [3].

In order to build the first functional prototype, it was decided to utilize an Arduino MEGA board equipped with an ethernet shield, DHT22 sensor, LDR sensor modules with GL55 photoresistors, two 5 relay modules, an electric heater, a water pump and diverse electronic components to provide the greenhouse with energy. The software was designed using Object Oriented Programming (POO), which results efficient to perform the measurements and changes needed, as well as acting as an ethernet client

and server simultaneously in order to send and receive GET and POST requests to communicate with a webpage from which the user can manually control the greenhouse.

The present article is structured in the following way: Section 2 explains how the software was developed, followed by verification and validation of said software in Section 3, finally, Section 4 presents conclusions and future research.

2. Software development

Initially, the software was developed in an Arduino board using C as programming language. However, due to the need of measuring temperature, humidity and luminosity in real time and communicating with the webserver at the same time, it was imperative to utilize parallelism which Arduino cannot support due to architecture limitations. It was for this that for the second prototype, the board was replaced by a Raspberry Pi 2 Model B+ using Python 3 as programming language along with Raspbian as operative system since it is the manufacturer's own OS and is much more efficient in the interpretation and execution of the board's ARM processor. The sensor utilized in both prototypes were DHT22 for temperature and humidity and LDR modules with GL55 photoresistors for luminosity, the actuators did not change between the two prototypes. Details on the implementation of the aforementioned can be found in the following sub-sections.

2.1 Modules

The software design contains 7 modules along with the main module, a bash script, and a module called which was designed and implemented by the author "APIWebServer", details on their implementation and functionality are featured below:

- a) Actuator: Contains methods to activate, deactivate, check the actuator's parameters and status, just receives order and does not do anything else on its own than initializing
- b) Sensor: Checks and measures values for LDR y DHT22 sensors, as well as confirming if the established limits have been breached, the user can modify the sensor's properties using the webpage. The measurements from the DHT22 sensors are obtained using the module "MyPyDHT" [4], whereas measurements for the LDR sensors are collected in an analog manner since no working libraries to obtain digital measurements for LDR sensors were found.
- c) Estación: Defines methods to initialize all of the greenhouse's data and obtains data for all of the sensors, actuators and the greenhouse's own data from the webserver which then utilizes to initialize every sensor and actuator. Methods to search for sensors and actuators using their respective ID were implemented as well.

- d) Control: The most important module in the greenhouse, contains an instance of “WebController” which collects measurements periodically, as well as receiving commands to send them to the server and update the parameters for the actuators, sensors and for the greenhouse itself, it is also responsible of receiving and interpreting the user’s commands. Contains a subclass called “SchedPeriodico” which utilizes the “sched” module [5] in a recursive manner so as to queue functions and procedures to be executed every certain interval which at the time is set by the server. This allows automatic sensor and actuator control, which can be close to real-time control since the interval for measurement recollection can be smaller than 1 second.
- e) WebServer: Implements the webserver for the greenhouse, it was made using a framework called “Flask” [6]. This class handles all of the POST and GET requests that the server sends. Code example can be found below:

```
1.self.app.add_url_rule ('/<path: path>', 'manejarReq',
    view_func=self.manejarReq, methods= ['POST', 'GET'])
```

The sentence indicated that the class will handle POST and GET requests via a method called “manejarReq”. It will listen for requests in every path and every request will be handled by sending a message with certain parameters to an instance of “APIWebServer”.

- f) WebController: This class is responsible for the communication between the greenhouse and the webserver and acts as the web client for the greenhouse , it contains an instance of “Control”, as well as methods to obtain the greenhouse’s self IP for which a bash script is utilized, the script is executed using a python module called “subprocess” [7]. In order to perform all web-related activities it utilizes http.client [8] and Requests [9].
- g) ProcessController: Defines an object known as “processController” which has only one method that utilizes a module known as “threading” [10], in order to start 2 threads, one that is in charge of all activities regarding the greenhouse’s measurement recollection, actuator control and web client, and another one that allows the greenhouse to act as a webserver in of itself and receives and executes all the commands the user sends from the webpage. Most of the contents of this module can be seen below:

```
1. import threading
2.
3. def inicializarProcesos (self, estacion, servidor):
4.     estacion.inicializar()
5.     threadControl=threading.Thread(target=estacion.controlar)
6.     threadServer=threading.Thread(target=servidor.runServer)
7.     threadControl.start()
8.     threadServer.start()
```

- h) Bash script: Returns the IP for the greenhouse using 4 commands and pipes

```
1.#!/bin/bash
2.ifconfig eth0 | grep "inet " | cut -c 13-28 | tr -d ' \n'
```

At the moment of writing this paper, it will only succeed in the task if the greenhouse is connected to internet via Ethernet, for future research the script will be improved in order to make it more versatile and integral.

- i) APIWebServer: Contains an instance of Control and possesses only 1 method which is in charge of receiving and interpreting orders from the webservice.
- j) Main: The main module, declares one instance of WebController, Control, APIWebServer, Estacion and WebServer, followed by the execution of the only method in ProcessController to initiate functions.

2.2 Webservice and webpage

The webservice as well as the webpage are currently hosted at the University's servers, which counts with remote access. The webpage is made in HTML5 and PHP, which allows it to send and receive requests of the types POST (Requests that send information) and GET (Requests that solicitate information) both using JSON format. Said page, has the following functions implemented:

- Displaying sensor data, actuator status, pause the automatic control in the greenhouse, as well as changing parameters in both actuators and sensors and creating new ones as well.
- Information Storage: Data collected by the sensors, as well as any changes in actuators, are both saved in database for every greenhouse in a network. This allows the administration of several greenhouses at once and documents information for multiple greenhouses simultaneously, therefore being able to function no matter the amount or scale of the greenhouses. The information is displayed in the "Reportes" section, which shows a graph with a progression of the temperature of the greenhouse along time, in the "Mediciones" section, the raw data for every measurement can be seen in any period of time. This is achieved by extracting and processing data from the database at the time the user enters the webpage.

3. Verification and validation

Verification and validation for each of the features that the webpage currently has are detailed below:

- a) Addition of new sensors and actuators: In order to validate this feature, 3 sensors and 3 actuators were created in the webpage, it was consecutively verified that by connecting the sensors and actuators to their respective pins, activity remained normal with the newly added actuators and sensors. This test validated emission, reception and storage of data as well as adding new

sensors, actuators, the only limitation being the number of pins available on the board (Fig. 1 y 2).

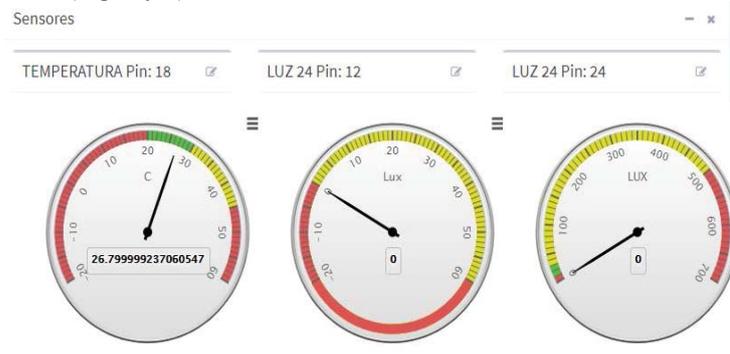


Fig.1. Luminosity and temperature sensors

```
ACCION:CONSULTAR_SENORES
ARGUMENTOS: {'estacion_id': '2'}
*****
*****
{'19': 0, '16': 0, '20': 0, '18': 26.799999237060547}
*****
```

Fig. 2. Reaction to the request of sensor data

b) Ping from the webpage to the greenhouse in order to verify connection status was successful (Fig. 3 y 4).

```
*****
ACCION:PING
ARGUMENTOS: {'estacion_id': '2'}
*****
OK
10.114.80.51 - - [12/Apr/2019 14:53:52] "GET /{"accion":"PING","parametros":{"estacion_id":"2"}} HTTP/1.0" 200 -
#####LIMITE#####
```

Fig. 3. The greenhouse's reaction to the connection test

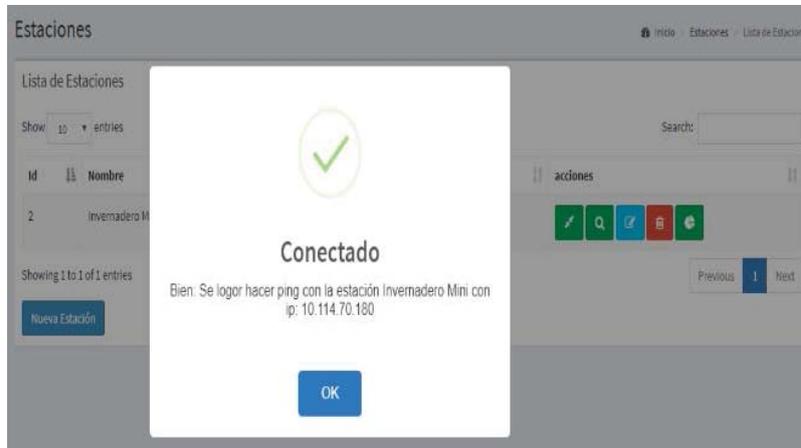


Fig. 4. Webpage receiving confirmation from the greenhouse, thus displaying a success message.

- c) Updating parameters for sensors and actuators: By modifying the parameters in the previously added, the commands were executed and were successful, only having to reconnect said components to their new respective pins to remain normal activity. This did not affect performance in any way. (Fig. 5 y 6).

```
10.114.80.51 - - [12/Apr/2019 17:13:06] "GET /{"accion":"ACTUALIZAR_SENSOR","parametros":{"pos":null,"pin":"18","limite_superior":"25","actuador_maximo_pos":"2","limite_inferior":"20","actuador_minimo_pos":"2","alerta_maximo":"30","alerta_minimo":"20","tipo":"T","lib":"DHT22","id":"18","activo":1,"nuevo":"0"}} HTTP/1.0" 200 -
```

Fig. 5. Sensor update received by the greenhouse

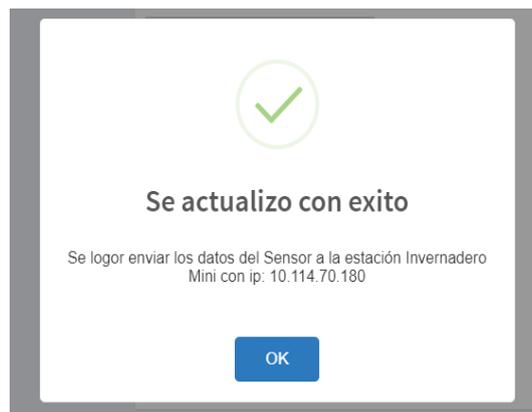


Fig. 6. Success message from the webpage

- d) Manual actuator control and automatic control stoppage: This was validated by activating and deactivating each actuator 3 times which was successful. Posteriorly, automatic control was paused which resulted in the greenhouse only taking commands from the user while still performing the expected sensor measurements, therefore being successful (Fig. 7, 8 y 9).

```
{'accion': 'CAMBIAR_ESTADO_CONTROLADOR', 'parametros': {'estado': 0, 'pos': '3'}}
*****
ACCION:CAMBIAR_ESTADO_CONTROLADOR
ARGUMENTOS: {'estado': 0, 'pos': '3'}
*****
2
Se activo: 6 Descripcion: Salida 51
10.114.80.51 - - [12/Apr/2019 17:03:35] "GET /{"accion":"CAMBIAR_ESTADO_CONTROLADOR","parametros":{"pos":"3","estado":0}} HTTP/1.0" 200 -
```

Fig. 7. Reaction to actuator activation and deactivation received

```
ACCION:PAUSAR_CONTROL
ARGUMENTOS: {'estacion_id': '2'}
*****
10.114.80.51 - - [12/Apr/2019 17:09:20] "GET /{"accion":"PAUSAR_CONTROL","parametros":{"estacion_id":"2"}} HTTP/1.0" 200 -
```

Fig. 8. Automatic function stoppage received



Fig. 9. Actuator control from the webpage

- e) Automatic control in case of temperature/luminosity limit breaches: This test was performed by covering the LDR sensors and exhaling near the DHT sensors, as well as positioning a heat source near them. It was verified that in cases where opposite limits were breached in a rapid and consecutive manner (Less than 2 seconds in between) actuators will not cease function, however in case the intervals are greater or equal to 2 seconds, it could be observed that the corresponding actuators for each sensor were activated even in cases where opposite limits were breached consecutively, in this last case doing so by deactivating the currently functioning actuator and activating the correct actuator.

Concerning performance, communication was considered acceptable based on the expert's opinion and response and reaction times were not greater than the intervals defined in the greenhouse's programming.

4. Conclusions and future research

In order to validate the results several hardware and software test were performed over several prototypes. As a result, it was proven that the greenhouse will turn useful for future research in the field of Agronomics.

It is believed that research in domotics in cooperation with agronomics could result in better ways to administrate crops, optimize their growth and development and improving the quality of future research in the field of botany as well. Smart greenhouses such as the one detailed in this paper can be escalated and improved to be later commercialized and possess an impact commercially as well as scientifically.

As future research, it is expected to utilize PID algorithms in order to optimize actuator control for the greenhouse, as well as using MQTT for all real-time communication-related activities. Furthermore, various types of sensors and actuators will be added, such as current sensors, humidity sensors for ground usage, sprinklers and several improvements in terms of hardware used for influencing the environment inside the greenhouse. Concerning the webpage, Angular will be utilized as the main framework due to the efficiency and wide variety of functions that it provides [11] as well as a custom architecture designed around the established requirements(Fig. 10), which will be adapted to any necessary changes.

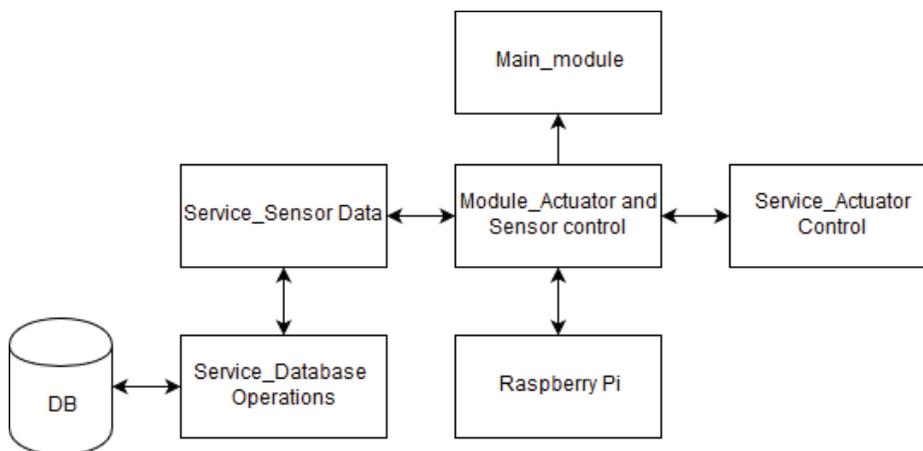


Fig.10. Architecture for the newly designed webpage.

Posteriorly, Data Science algorithms and processes will be utilized [11] for smart data analysis.

Bibliografía

- [1] K. Meah, J. Forsyth y J. Moscola, «A Smart Sensor Network for an Automated Urban Greenhouse,» 2019.
- [2] R. B. Salikhov y A. A. Zainitdinova, «System of Monitoring and Remote Control of Microclimate in Greenhouses,» 2019.
- [3] A. Potapovs, A. Avotins, P. Apse-Apsitis, M. Gorobetz y P. Ceirs, «Continuous Crop Weight Measurement Sensor Calibration Algorithm for Industrial Greenhouse,» 2019.
- [4] S. Vettor, «freedom27/MyPyDHT/A Python 3 library for Raspberry Pi to interact with the Humidity & Temperature sensors DHT11, DHT22 and AM2302,» Abril 2019. [En línea]. Available: <https://github.com/freedom27/MyPyDHT>.
- [5] G. van Rossum, «The Python Standard Library,» 2019. [En línea]. Available: <https://docs.python.org/3/library/sched.html>.
- [6] A. Ronacher, «Flask-web development, one drop at a time,» 2019. [En línea]. Available: <http://flask.pocoo.org/>.
- [7] G. van Rossum, «The Python Standard Library,» 2019. [En línea]. Available: <https://docs.python.org/3/library/subprocess.html>.
- [8] G. van Rossum, «The Python Standard Library,» 2019. [En línea]. Available: <https://docs.python.org/3/library/http.client.html>.
- [9] K. Reitz, «Requests: HTTP for Humans,» 2019. [En línea]. Available: <http://docs.python-requests.org/en/master/>.
- [10] G. van Rossum, «The Python Standard Library,» 2019. [En línea]. Available: <https://docs.python.org/3/library/threading.html>.
- [11] P. Britos, *Tesis Doctoral Procesos de Explotación Basados en Sistemas Inteligentes*, La Plata, Buenos Aires: UNLP, 2008.
- [12] G. van Rossum, «Python 3.0,» 2019. [En línea]. Available: <https://www.python.org/download/releases/3.0/>.
- [13] E. Upton, «Raspberry Pi 2 Model B+,» 2019. [En línea]. Available: <https://www.raspberrypi.org/blog/price-cut-raspberry-pi-model-b-now-only-25/>.
- [14] M. Banzi, «Arduino,» 2019. [En línea]. Available: <https://store.arduino.cc/usa/mega-2560-r3>.

Sistema de localización para robots móviles de bajo costo utilizando marcas de referencia artificiales en ambientes de interiores

Rafael Ignacio Zurita¹, Alejandro Mora¹, Candelaria Alvarez¹, y
Miriam Lechner¹

¹ Departamento de Ingeniería de Computadoras,
Facultad de Informática, Universidad Nacional del Comahue
{rafa, alejandro.mora, candelaria.alvarez, mtl}@fi.uncoma.edu.ar

Resumen. La localización es un requisito de la navegación con mapas de robots móviles. Usualmente se implementa con múltiples sensores y gran poder de cómputo, por lo que no suele ser una capacidad de los robots móviles de bajo costo. Se propone el diseño e implementación de un sistema prototipo que reporta su localización utilizando un hardware de mínimas prestaciones y marcas artificiales en el terreno. Se realizaron pruebas en un ambiente de interior, obteniendo desde el prototipo repetidas localizaciones, para evaluar la precisión y estimar la exactitud. En base a los resultados se concluye que el prototipo propuesto es capaz de determinar ubicaciones con una precisión menor a 10cm, y a una frecuencia de 2Hz; lo que permite plantear la posibilidad, en trabajos futuros, de incluir navegación con mapas a una clase de robots que hasta hoy no suele utilizar esta característica.

Palabras claves: Robot móvil, localización, navegación, sistema embebido, AprilTag.

1 Introducción

La localización es uno de los componentes requeridos para la navegación con mapas de los robots móviles, y requiere que el robot sea capaz de determinar su ubicación y orientación exacta en el ambiente en el cual se encuentre. Con esta información y un marco de referencia virtual que represente el ambiente real, el robot puede planificar una ruta y desplazarse para cumplir con sus objetivos [1].

En la actualidad, los robots móviles capaces de realizar navegación en interiores son diseñados y construidos para tareas complejas. Suelen trabajar en ambientes industriales, para desplazar grandes cargas dentro de galpones o edificaciones. También en ambientes públicos, por ejemplo en hoteles, donde los robots suelen navegar desde la recepción a diferentes habitaciones, realizando la entrega de suministros solicitados por huéspedes [2,3]. En estas clases de robots el costo y/o consumo energético usualmente no es una restricción primordial, por lo que suelen contar con sensores láser (LIDAR), visuales (cámaras) y de orientación y velocidad (unidades de medición inercial) para recolectar datos del ambiente; y con

computadoras capaces de procesar grandes cantidades de información en tiempo real, lo que en conjunto permite conocer la ubicación del robot y navegar en sus ambientes de trabajo [4].

Por el contrario, los robots móviles de bajo costo utilizados en interiores poseen un microcontrolador o microprocesador de bajas prestaciones como sistema central de control. Estos robots realizan tareas sencillas, tales como aspirar el polvo en un living (domésticos), o realizar movimientos simples (adelante, izquierda, etc) al ejecutar programas desarrollados por estudiantes (robótica educativa). Para ello utilizan sensores que permiten una interacción básica con el entorno, por ejemplo para evitar el choque con obstáculos o moverse dentro de un cerco invisible. Los sensores en esta clase de robots pueden detectar el objeto a evitar cuando se está a una distancia muy corta, devolviendo información básica que puede ser discretizada para indicar si existe o no un objeto frente al sensor. Por tal motivo estos robots realizan únicamente navegación reactiva (sin utilización de mapas), ya que no son capaces de determinar su ubicación [5].

En este artículo se presenta la arquitectura de un sistema embebido prototipo, que reporta su localización utilizando un microprocesador de bajas prestaciones y marcas artificiales en el terreno. Está diseñado para ser integrado fácilmente a robots móviles de bajo costo, con el fin de que esta categoría de robots pueda determinar su ubicación en ambientes de interiores. El sistema está basado en la adquisición de señales a través de una cámara de video, y la detección de marcas artificiales a través del software AprilTag. El dispositivo cuenta con una computadora de arquitectura MIPS de bajo consumo que controla el sensor de captura, procesa la imágenes con AprilTag, y calcula la localización. Por lo que las contribuciones principales de este trabajo son dos:

- Una aplicación para sistemas embebidos portable (desarrollada en C sin ninguna dependencia de bibliotecas externas) que captura imágenes de una cámara, y reporta la localización utilizando AprilTag.
- La validación de la precisión del sistema a través de la experimentación del mismo en un hardware de mínimas prestaciones.

El resto de este trabajo está estructurado de la siguiente manera: en la sección 2 se presentan trabajos relacionados; en la sección 3 se presenta la arquitectura del sistema propuesto, con énfasis en el algoritmo de localización; en la sección 4 se exponen los resultados obtenidos de la evaluación del sistema, al experimentar con un robot de bajo costo para la obtención de mediciones de localización; finalmente, en la sección 5 se detallan las conclusiones y trabajos futuros.

2 Trabajos Relacionados

Una solución a la localización es a través de marcas artificiales en el terreno (del inglés artificial landmark), que son colocadas para que puedan ser unívocamente identificadas de manera fácil y precisa. Si bien con este método se puede evitar el uso de varios de los sensores que habitualmente se utilizan en localización, aún requiere

visión por computadora para reconocer la ubicación y orientación de la marca artificial. Los algoritmos de visión por computadora para la detección de estas marcas fiduciales suelen implementarse y verificarse en computadoras con recursos suficientes para procesar un gran número de imágenes por segundo, y permitir el uso de un amplio rango de bibliotecas y aplicaciones al mismo tiempo. Por ejemplo, opencv para la captura y filtrado de imágenes, y el entorno matlab para programación. Esto posibilita el prototipado rápido y la validación de los algoritmos de visión en la misma computadora. En [6] se presenta el diseño de AprilTag para su uso en robótica y realidad aumentada. La validación se realizó en una computadora con una CPU Intel Xeon E5-2640 de 2.5GHz. Se menciona que es posible lograr un buen rendimiento del sistema en un equipo iPhone, sin especificar un modelo particular. En [7] se implementa un aterrizaje autónomo de un dron a una estación en movimiento, utilizando una computadora con un Nvidia Tegra K1 SoC, CPU ARM A-15 de 2.3GHz, y 8GB de DDR3 RAM para la detección de marcas de referencias con AprilTag. En [8] se presenta el diseño de un sistema para localización en interiores, utilizando también AprilTag. La experimentación se llevó a cabo mediante un robot móvil Pioneer 3-DX, que contó con una notebook empujada con procesador Intel Core 2 Duo 2.0 GHz y 2G de RAM. En [9] se presenta un sistema capaz de ser utilizado en el control de acercamiento de dos barcos para el traspaso de cargas entre embarcaciones. Se utilizó para el procesamiento de las marcas de referencia con AprilTag una notebook Lenovo W540 con un procesador Intel Quad-Core i7-4900MQ de 2.80GHz CPU y una NVIDIA Quadro K2100M para la rectificación de las imágenes. En [10] se analizan diferentes sistemas de detección de marcas de referencia y se propone uno mixto. Los experimentos se realizaron utilizando una computadora con procesador i7-7600U a 2.80GHz para el procesamiento de las imágenes. En [11] se controlan brazos manipuladores industriales basando sus movimientos a la posición de los objetos a manipular. A estos últimos se les añadieron marcas de referencia de AprilTag para estimar su posición. La ejecución de este sistema en los experimentos se realizó en una PC embebida con un procesador -i7 quad core, y 8GB de RAM.

A diferencia de los casos expuestos nuestro trabajo se focaliza en la implementación y validación de un sistema embebido que pueda reportar la localización de robots móviles de bajo costo, utilizando un hardware de mínimas prestaciones y el software AprilTag 3 para la detección de marcas de referencia. No hemos encontrado otros trabajos que evalúen el rendimiento o la precisión de AprilTag 3 en sistemas mínimos.

3 Arquitectura

3.1 Arquitectura de Hardware

La arquitectura de hardware del prototipo presenta dos componentes principales:

- El dispositivo de cámara rotativa;
- El hardware de procesamiento embebido

El dispositivo de cámara rotativa consta de una cámara USB genérica montada sobre un servo motor, el cual permite rotar a la cámara de 0 a 360°, con una resolución de giro de un grado. La señal de control para girar la cámara es una señal individual de modulación por ancho de pulsos (PWM), con una frecuencia estándar de 50hz para este dispositivo. La cámara tiene una interfaz USB 2.0 y es un dispositivo de clase UVC. Ambas interfaces están conectadas al hardware embebido.

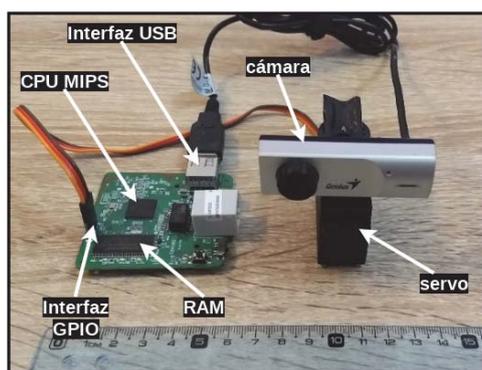


Fig. 1. Dispositivo de cámara rotativa y hardware de procesamiento embebido.

El hardware embebido es la computadora en donde se ejecuta el sistema de localización. Contiene un System On Chip (SOC) Atheros AR9331, con una CPU MIPS de 400Mhz. La memoria principal (RAM) es de 32MB. También cuenta con puertos de E/S (GPIO), un puerto USB 2.0, una interfaz UART y una interfaz Wireless WiFi 802.11n/g/b. El tamaño del módulo, que puede observarse en la Fig. 1, es de aproximadamente 6x6x9cm, y tiene un peso total de 120g. Los costos de los componentes se enumeran a continuación:

1. Computadora MIPS: USD 32.85
2. Servo motor DS04-NFC: USD 4.54
3. Cámara USB genérica: USD 5

3.2 Arquitectura de Software

El hardware está controlado por un sistema operativo Linux construido con buildroot¹, destinado a sistemas embebidos. Los controladores de hardware que se utilizan en el dispositivo presentado son el universal video class (UVC) para la captura de imágenes desde la cámara, y el de un puerto de E/S general (GPIO). Adicionalmente se integró también un driver en espacio de usuario de modulación por ancho de pulsos (PWM) (que utiliza a bajo nivel el driver GPIO). Este driver controla un servo motor para rotar la cámara. Cuando se requiere una localización la aplicación busca primero la marca de referencia. Para esto va rotando la cámara 45 grados y capturando imágenes, analizando cada una para detectar la marca de referencia. Cuando se detecta la marca

¹ <https://buildroot.org/>

de referencia en una imagen, el sistema detiene la rotación de la cámara, y se procede a realizar el proceso de localización, el cual consta de tres etapas, presentadas en la Fig. 2.

En la primera etapa del proceso la aplicación embebida captura una imagen desde la cámara, utilizando la interfaz en espacio de usuario del driver UVC del sistema Linux. El formato de la imagen digital recibida es JPEG y la resolución es de 320x240 píxeles.



Fig. 2. Etapas del proceso de localización.

En la segunda etapa el sistema procesa la señal obtenida (imagen JPEG) con el software AprilTag. Este software es un sistema visual fiducial, útil para una amplia variedad de tareas, incluida la realidad aumentada, la robótica y la calibración de la cámara [10]. Es capaz de reconocer marcas de referencia artificiales llamadas etiquetas (tags), las cuales son similares a códigos QR pero con menor cantidad de bits, lo que permite una detección más rápida que los sistemas QR tradicionales. Otra ventaja de AprilTag es que presenta un bajo porcentaje de falsas detecciones (falsos positivos) aún con cambios en la iluminación del ambiente, lo cual es una característica deseable en los sistemas de visión por computadora, que suelen presentar inconvenientes ante dicho factor.

En esta etapa AprilTag analiza la imagen, y reporta la posición y orientación de las etiquetas identificadas. Como en este trabajo se experimenta con una única marca de referencia en el ambiente, se obtiene como salida una única matriz de rotación y un vector de traslación de la marca detectada, ubicados en el sistema de coordenadas de la cámara, con origen el centro de la lente focal.

La última etapa, Algoritmo de localización, realiza el cálculo de la posición de la cámara en el sistema de coordenadas del mundo real (sistema de coordenadas de la marca de referencia) y consta de tres sub-etapas:

1. Se obtiene el ángulo euler pitch a partir de la matriz de rotación obtenida desde AprilTag. Para su cálculo se utilizan funciones de la biblioteca `eigen`², las cuales se incorporaron como parte del código de la aplicación. En la Fig. 3 (a) 1 se puede observar que el ángulo euler pitch obtenido α (alfa) es el ángulo formado por el eje X_{Cam} del sistema de coordenadas de la cámara y el eje X_{tag} del sistema de coordenadas del tag.
2. Luego, a partir del punto (X_c, Y_c) provisto por AprilTag (Fig. 3 (a)) se obtiene el punto X_c' e Y_c' en el sistema de coordenadas del mundo real (Fig. 3 (b) 2), donde $X_c' = X_c$ y $Y_c' = Y_c$ (sus valores algebraicos son equivalentes).
3. Finalmente, se realiza una rotación del punto (X_c', Y_c') , la cantidad de grados α obtenido en el paso 1. Para esto se define un vector columna a partir de X_c' e Y_c' , y se multiplica por una matriz de rotación calculada a partir del ángulo alfa α :

² <http://eigen.tuxfamily.org>

$$\begin{bmatrix} X_t \\ Y_t \end{bmatrix} = \begin{bmatrix} \cos(\alpha) & -\text{sen}(\alpha) \\ \text{sen}(\alpha) & \cos(\alpha) \end{bmatrix} \begin{bmatrix} X_{c'} \\ Y_{c'} \end{bmatrix}$$

Se obtiene entonces el punto (X_t, Y_t) (Fig. 3 (b) 3), a partir de la multiplicación recién descrita:

$$\begin{aligned} X_t &= \cos(\alpha).X_{c'} - \text{sen}(\alpha).Y_{c'} \\ Y_t &= \text{sen}(\alpha).X_{c'} + \cos(\alpha).Y_{c'} \end{aligned}$$

El punto (X_t, Y_t) (Fig. 3 (b) 3) indica la posición de la cámara en el sistema de coordenadas de la marca de referencia, es decir, la posición del dispositivo portátil en el escenario o mundo real.

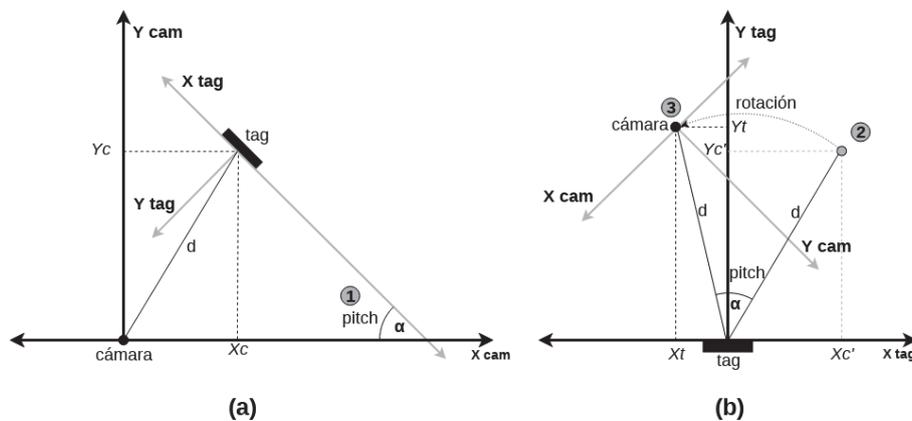


Fig. 3. (a) posición de la marca de referencia en el sistema de coordenadas de la cámara. (b) posición de la cámara en el sistema de coordenadas del mundo real (la marca de referencia).

4 Resultados

Para validar el prototipo se evaluó el tiempo, y precisión de las localizaciones reportadas por el sistema propuesto.

4.1 Tiempo de ejecución

En la etapa de obtención de requerimientos de este trabajo se tomó como referencia los robots móviles de bajo costo utilizados por la Facultad de Informática en proyectos de robótica educativa [12]. El robot Frankestito se desplaza a una velocidad máxima de 12cm/seg, y se estimó que, para futuras funcionalidades de navegación el robot educativo, debería indicar su ubicación aproximadamente cada 20 centímetros. Esto requiere una localización cada 1,6 segundos en este robot, por lo que una frecuencia de 1Hz sería suficiente si el error de la localización es +/- 12cm.

Se utilizó la función de la biblioteca `libc gettimeofday()` para evaluar el tiempo transcurrido de cada etapa del sistema propuesto (tiempo de E/S y tiempo de ejecución) y corroborar que el tiempo de localización está dentro de los requerimientos. En una prueba de 415 mediciones utilizando el hardware propuesto se obtuvo (como se observa en la Tabla 1) una frecuencia general de localizaciones de 2Hz, y una frecuencia para el caso más desfavorable de 1Hz.

Tabla 1. Tiempos promedios de E/S y CPU en el dispositivo prototipo en 415 localizaciones.

Etapa	Tipo	Tiempo (media aritmética)	Tiempo máx. (peor caso)
Captura	Tiempo de E/S	74 (ms)	95 (ms)
Detección con AprilTag	Tiempo de ejecución	292 (ms)	750 (ms)
Algoritmo de localización	Tiempo de ejecución	41 (ms)	46 (ms)
TOTAL	Tiempo de E/S + CPU	407 (ms)	891 (ms)

4.2 Precisión

Para las pruebas se delimitó un escenario real de navegación de 2mts x 2mts, con una cuadrícula en el terreno compuesta de mosaicos de 33cm x 33cm. Se colocó una marca de referencia (tag en la terminología de AprilTag) en este ambiente. El dispositivo prototipo fue entonces montado sobre un robot móvil de bajo costo, y se calibró la altura de la lente de la cámara para que se encuentre a la misma altura que el centro del tag, como se observa en la Fig. 4. Para la comunicación entre la CPU del prototipo y el microcontrolador del robot móvil se interconectó la interfaz serial UART de ambos.

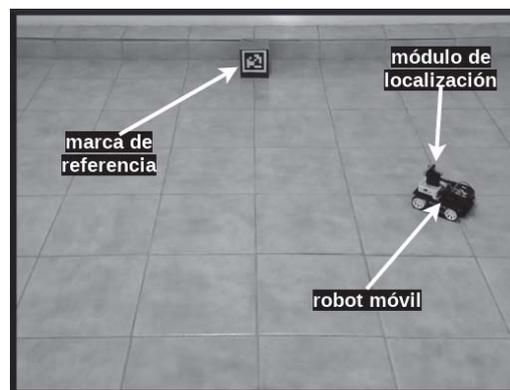


Fig. 4. Escenario real en donde se realizaron las mediciones con el dispositivo propuesto.

Durante el experimento se utilizó una marca de referencia (tag) de 20cm x 20cm, y se fue situando al robot móvil en cada esquina de la cuadrícula del ambiente, cuyas localizaciones exactas son conocidas a priori. En cada ubicación se realizaron aproximadamente 10 mediciones desde el dispositivo prototipo y se almacenaron los resultados para su análisis. El conjunto fue graficado en una cuadrícula que representa

el escenario real (Fig. 5). La marca de referencia (tag) se encontraba en $X=99$ e $Y=0$ de esta representación. Cada punto representa una localización reportada por el sistema.

Se observa que hasta una distancia en Y de 125cm el robot reporta localizaciones muy cercanas al valor real de la ubicación. El valor exacto en cada medición (la ubicación real) está representada en la Figura por cada intersección de la cuadrícula. Las mediciones son también muy precisas (las 10 mediciones en cada ubicación están concentradas). Luego, a más de 175cm de distancia ($Y=175$) la precisión decae en el eje x mucho más que en el eje y (puede notarse en la figura que los puntos de cada medición tienen una dispersión mayormente horizontal). En esta zona del escenario la distancia máxima (diferencia) entre mediciones de una misma ubicación es de hasta aproximadamente 20cm en el eje X, y de 10cm en el eje Y. Finalmente, a 2mts de distancia, las mediciones ya no se concentran alrededor de la ubicación real, por lo que estas mediciones sólo podrían utilizarse como un cálculo estimativo de la zona donde se encuentre el robot, y ya no como una localización exacta.

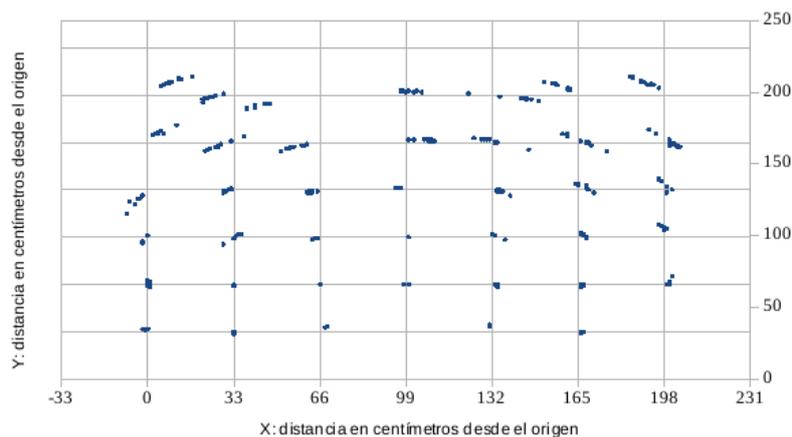


Fig. 5. Relación entre las localizaciones reportadas por el dispositivo prototipo (puntos) y la ubicación real (intersecciones de la cuadrícula).

Cada localización es una ubicación en un plano 2D con ejes (X, Y). En la Figura 6 se presentan dos histogramas (uno para el eje X y otro para Y) de las distancias de las localizaciones con respecto al valor de la ubicación real.

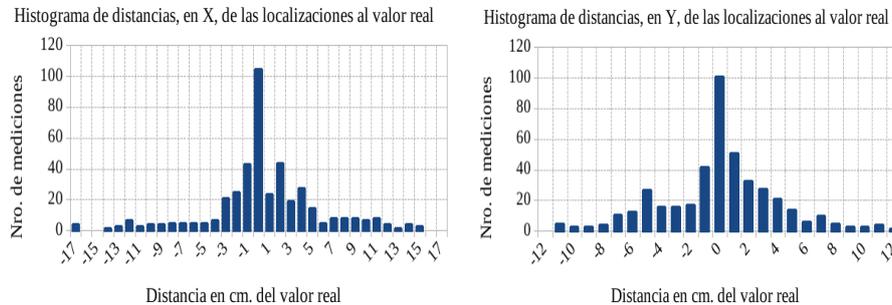


Fig. 6. Histograma de las distancias en centímetros de las localizaciones con respecto al valor real.

El valor $x=0$ en ambos es la distancia mínima con respecto valor estimativo real (ubicaciones conocidas previo al experimento), y representa las mediciones más exactas. Se observa que al menos unas 100 mediciones coincidieron con la ubicación real (de un total de 415), y también, que un gran número de mediciones están comprendidas en el rango $[-5,5]$ cm. Su justificación puede apreciarse en las estadísticas de la Tabla 2.

Tabla 2. Estadística de las distancias de localizaciones con respecto a su valor estimativo real.

Estadística	Valor
Cantidad de mediciones	415
Media del error (distancia media de las localizaciones al valor real de la ubicación)	4,971cm
Varianza	28,985
Desvío estándar	5,383
Error estándar del promedio	0,264
Resultado numérico de las mediciones (distancia al valor estimativo real)	(4,971 +/- 0,264) cm
Porcentaje de mediciones del experimento dentro del desvío estándar	65%

5 Conclusiones y Trabajo Futuro

Se presentó en este trabajo la arquitectura de un dispositivo prototipo para reportar localizaciones de robots móviles de bajo costo en ambientes interiores, y utilizando marcas de referencia en el terreno. Se experimentos para medir la frecuencia, precisión y exactitud, cuando se lo utiliza en un hardware de mínimas prestaciones.

En base a los resultados se concluye que el hardware y software propuesto es capaz de localizar un robot de bajo costo a 2hz, y con una precisión de (4,971 +/-

0,264) cm el 65% de los casos, si el robot navega en los límites impuestos por: tamaño de la marca de referencia en el terreno y la resolución de la cámara utilizada. Los robots de bajo costo suelen trabajar en ambientes pequeños (por ej. una habitación), y si bien se experimentó con pequeñas marcas de referencia se estima (y se plantea para validación futura) que con un tamaño de tag de 40cm x 40cm sería suficiente en la mayoría de estos casos (por ej. para ambientes de hasta 16m²).

Como trabajo futuro se espera continuar, también, con la implementación y validación de los componentes restantes de la navegación con mapas en robots móviles de bajo costo. Algunos ejemplos de estos son: representación de mapas, planificación de rutas, historial de trayectorias, y estimación de los errores de localización. También se pretende incrementar el rendimiento del módulo de localización, realizando principalmente mejoras en el software (por ejemplo, solapando la E/S con el tiempo de CPU).

Referencias

1. Roland Siegwart, Illah R. Nourbakhsh.: Introduction to Autonomous Mobile Robots. Massachusetts Institute of Technology. ISBN: 9780262195027. MIT Press (2004)
2. Melonee Wise, Michael Ferguson, Derek King, Eric Diehr and David Dymesich.: Fetch & Freight: Standard Platforms for Service Robot Applications. Fetch Robotics Inc. press (2018)
3. Michael Jae-Yoon Chung, Justin Huang, Leila Takayama, Tessa Lau, Maya Cakmak.: Iterative Design of a System for Programming Socially Interactive Service Robots. International Conference on Social Robotics. (2016)
4. Luc Jaulin.: Mobile Robotics. Book ENSTA-Bretagne, France. ISBN: 9781785480485. Elsevier (2015)
5. Peter Corke.: Robotics, Vision and Control. Fundamental Algorithms in MATLAB. Book. ISBN: 9783642201448. Springer (2011)
6. John Wang, Edwin Olson.: AprilTag 2: Efficient and robust fiducial detection. Proceedings of the International Conference on Intelligent Robots and Systems (2016)
7. Alexandre Borowczyk, Duc-Tien Nguyen, André Phu-Van Nguyen, Dang Quang Nguyen, David Saussié, Jerome Le Ny.: -Autonomous Landing of a Multirotor Micro Air Vehicle on a High Velocity Ground Vehicle. IFAC-PapersOnLine, Volume 50, Issue 1, pp 10488--10494. 2017
8. Xu Zhong, Yu Zhou, Hanyu Liu.: Design and recognition of artificial landmarks for reliable indoor self-localization of mobile robots. International Journal of Advanced Robotic Systems. pp 1--13. (2017)
9. Joshua G. Mangelson, Ryan W. Wolcott, Paul Ozog, Ryan M. Eustice.: Robust Visual Fiducials for Skin-to-Skin Relative Ship Pose Estimation. OCEANS 2016 MTS/IEEE Monterey (2016)
10. Maximilian Krogus, Acshi Haggemiller, Edwin Olson.: Flexible Layouts for Fiducial Tags. Proceedings of the {IEEE/RSJ} International Conference on Intelligent Robots and Systems (2019)
11. Christian Nissler, Stefan Büttner, Zoltan-Csaba Marton, Laura Beckmann, Ulrike Thomas.: Evaluation and Improvement of Global Pose Estimation with Multiple AprilTags for Industrial Manipulators. IEEE 21st International Conference on ETFA. (2016)
12. J. Rodriguez, G. Grosso, R. Zurita, L. Cecchi.: Intervención de la Facultad de Informática en la enseñanza de Ciencias de la Computación en la escuela media basada en robótica educativa. XI Congreso de TE&ET pp221--231 (2016)



CACIC 2019

Congreso Argentino de Ciencias de la Computación

VIII Workshop Procesamiento de Señales y Sistemas de Tiempo

Coordinadores

Sanz, Cecilia (UNLP)
Depetris, Beatriz (UNDTF)
De Vincenzi, Marcelo (UAI)

Comité Científico

Baldassarri, Sandra (España)	Manresa-Yee, Cristina (España)	Roig Vila, Rosabel (España)
Cukierman, Diana (Canadá)	Marco, Javier (España)	Simari, Guillermo (Argentina)
De Giusti, Armando (Argentina)	Motz, Regina (Uruguay)	
Feierherd, Guillermo (Argentina)	Pesado, Patricia (Argentina)	

Temas

Estrategias y metodologías innovadoras de Enseñanza
 Descripción y evaluación de experiencias innovadoras de enseñanza
 Herramientas estratégicas en el desarrollo de procesos educativos en la formación en Informática
 Procesos colaborativos mediados por TIC
 Competencias del profesional docente del área Informática /Ciencias de la Computación
 Diseños curriculares innovadores
 Ubicuidad en los proceso de enseñar y aprender
 Incidencias de las actuales tecnologías móviles
 Aprendizaje de la Informática centrado en el alumno

Análisis de una experiencia de la evaluación de la adquisición de la competencia “trabajo en equipo”

Verónica Aubin¹, Renata Guatelli¹, Luca Videla¹, José Cabrera¹,
Carolina Sanchez¹, Carolina Ravinale¹

¹ Departamento de Ingeniería e Investigaciones Tecnológicas.
Universidad Nacional de La Matanza.
Florencio Varela 1903, San Justo, Buenos Aires, Argentina
{vaubin, rguatelli, lvidela, jcabrera}@unlam.edu.ar

Resumen. Las necesidades de la generación Z conducen a nuevos modelos de enseñanza aprendizaje, orientados al desarrollo de competencias. Por lo tanto, es necesario considerar nuevas formas de evaluar, ya que la evaluación no es un fin, sino una instancia más de aprendizaje. Este trabajo expone la experiencia de la puesta en práctica del proceso de evaluación del desarrollo de las competencias del trabajo en equipo, por medio de la autoevaluación, la coevaluación y la heteroevaluación. Se utilizaron rúbricas, diseñadas especialmente para apoyar estas prácticas. Los resultados obtenidos a partir del análisis de los datos recopilados en dichas rúbricas, han permitido obtener las tendencias generales orientativas de los modos de evaluación del grado de adquisición de la competencia de trabajo en equipo. Así como detectar los problemas en el desarrollo de la experiencia, lo que ha llevado a plantear nuevas estrategias de trabajo para los próximos cursos.

Palabras clave: autoevaluación, coevaluación, heteroevaluación, competencias transversales.

1 Introducción

El contexto actual caracterizado por la inmediatez, la velocidad en las comunicaciones, y el aumento de la complejidad científica y tecnológica de los procesos productivos, llevan a que el mercado laboral exija competencias transversales que hasta hace unos años no habían sido contempladas. En este entorno, es donde nacieron y se desenvuelven los llamados Millenials. Ellos forman la mayor parte de la población de las universidades de hoy en día.

Entre los Millenials la Generación Z se refiere a los que han nacido entre los años 1995 y 2012. Tienen una serie de características que se resumen a continuación: nativos digitales, multitarea, inmediatez, impaciencia, interactivos, y resilientes. [1]

Es casi inevitable que una generación con características tan propias vaya acompañada de un cambio en las estrategias de enseñanza aprendizaje. Donde, el estudiante se convierte en el centro del proceso de aprendizaje y participa de manera activa en la construcción de sus conocimientos. En consecuencia, las universidades como instituciones responsables de la formación de profesionales, tienen que dar respuesta a las necesidades del momento. Deben considerar la relación existente entre el ámbito académico, el mundo laboral y las características de la Generación Z.

El documento «Declaración de Valparaíso - ASIBEI» [2] refiere que existe un consenso en cuanto a que el ingeniero no sólo debe saber, sino también saber hacer. El saber hacer no surge de la mera adquisición de conocimientos, sino que es el resultado de la puesta en funciones de una compleja estructura de conocimientos, habilidades, destrezas, etc. que requiere ser reconocida expresamente en el proceso de aprendizaje para que la propuesta pedagógica incluya las actividades que permitan su desarrollo.

Entendemos por competencias, el conjunto de habilidades que permiten hacer algo con un cierto nivel de excelencia o destreza. Las competencias presuponen ciertos saberes teóricos, prácticos y actitudinales. Un estudiante alcanza una competencia cuando adquiere esos saberes y actitudes [3].

El estudiante universitario debe formarse tanto en competencias específicas propias de su profesión, como en competencias transversales. Entre las múltiples competencias transversales hay una muy demandada por el mercado de laboral, la competencia de trabajo en equipo [4], [5], [6], [7].

Según Torrelles [5] la competencia del trabajo en equipo es el “*Conjunto de conocimientos, habilidades y actitudes que permiten colaborar con otras personas en la realización de actividades, para lograr objetivos comunes, intercambiando información, distribuyendo tareas, asumiendo responsabilidades, resolviendo las dificultades que se presentan y contribuyendo a la mejora y desarrollo colectivo.*” Tiene como objetivo principal desarrollar el trabajo colaborativo entre personas, para lograr metas establecidas [8]. Presenta un carácter multidimensional que Bunk [9] lo caracteriza proponiendo cuatro clases de competencias: técnicas (el saber), metodológicas (el saber hacer), personales (saber ser), y sociales (saber participar).

Las ventajas del trabajo en equipo van desde disminuir brechas relativas a la adquisición de conocimientos conceptuales o prácticos, hasta mejorar las dificultades para argumentar o para hablar en público [10]. A pesar de que, durante su recorrido en la universidad, los estudiantes realizan diferentes actividades en equipo, estas por sí solas no garantizan la adquisición de las habilidades necesarias para trabajar de este modo en forma eficiente [11], [12].

La adquisición de las competencias deber ser continua y progresiva por parte de los estudiantes y la evaluación debe acompañar el proceso de aprendizaje y no constituir la finalización del mismo.

En este artículo, se expone una experiencia realizada en los cursos de la cátedra de Programación Avanzada de la UNLaM. Se presentan distintos modos de evaluar, el grado de adquisición de las competencias de trabajo en equipo. Los resultados no se muestran con un análisis estadístico, sino con tendencias generales orientativas a seleccionar las metodologías más adecuadas para trabajos futuros.

2 Metodologías y métodos

2.1. El trabajo en equipo

Pichon Riviere, médico psiquiatra argentino define grupo como un “Conjunto de personas que, ligadas por constantes de tiempo y espacio y articuladas por su mutua representación interna, se proponen en forma explícita o implícita llevar a cabo una tarea que constituye su finalidad, interactuando a través de complejos mecanismos de adjudicación y asunción de roles” [13].

Un equipo no es solo una agrupación de personas, hay algo que los une que es una finalidad en común la cual será parte de la motivación, motor del trabajo en equipo, y acompañado de la gamificación con su alto poder motivacional y capacidad socializadora pueden ser una interesante herramienta no solo para conseguir mayor compromiso del alumno sino un aprendizaje más significativo, donde se convierte en protagonista. El trabajo en equipo refuerza las competencias de toma de decisiones, análisis y síntesis y de crítica y autocrítica [14].

2.2 Rúbricas

Una rúbrica o matriz de valoración se define como “un descriptor cualitativo que establece la naturaleza de un desempeño” [15]. Son instrumentos de medición en los cuales se definen escalas de estándares, para los diferentes criterios a evaluar, de forma que permitan determinar la calidad con la cual los estudiantes resuelven una amplia variedad de tareas específicas [16].

La rúbrica es considerada una herramienta de evaluación formativa, implicando a los alumnos en el proceso de evaluación de su propio trabajo. El alumno puede observar sus avances en términos de competencias, saber en cualquier momento qué le queda por superar y qué ha superado y cómo”, por lo cual, son una herramienta de gran utilidad para dar soporte al proceso de enseñanza-aprendizaje [17].

Estas se pueden crear para dar soporte al proceso de autoevaluación, coevaluación o heteroevaluación. La Autoevaluación es cuando el alumno es quien se evalúa a sí mismo. Propicia el desarrollo de la autocrítica, la autoestima y el autorreconocimiento de las cualidades. La coevaluación es cuando se evalúan entre pares. La heteroevaluación, es esencialmente una evaluación externa. En nuestra experiencia la realizan los docentes sobre el trabajo y el rendimiento de los estudiantes. Se diferencia de la coevaluación porque el evaluado y los evaluadores corresponden a diferentes niveles jerárquicos [18].

La utilización combinada de la heteroevaluación con la autoevaluación y la coevaluación, posibilita una mejor apropiación de los conocimientos y desarrollo de competencias.

3 Experiencias y Resultados

La cátedra de Programación Avanzada de la UNLAM se ha propuesto desde hace tiempo cambiar la forma tradicional de enseñanza centrada en el docente, hacia un contexto activo, centrado en el estudiante y sus necesidades, para acercarlo en forma temprana a su actividad profesional. Para cumplir con este objetivo se han diseñado escenarios de aprendizaje integrado con actividades que promueven la resolución de tareas de forma innovadora y colaborativa, incorporado gamificación, considerando una evaluación continua a través de radiadores de información.

En nuestra experiencia, se aplicó una evaluación multiactores a través de rúbricas. La evaluación multiactores permite evaluar la competencia de trabajo en equipo de un estudiante con la mirada de sus compañeros, la de sus docentes y su propia mirada.

Se consensuaron criterios de evaluación que contemplan diferentes aspectos de esta modalidad de trabajo. Entre los aspectos considerados podemos mencionar la implicación de los alumnos en las dinámicas de trabajo en equipo, su responsabilidad ante el aprendizaje y el trabajo final. De cada uno de estos aspectos se desarrollaron distintas herramientas o rúbricas que propician la autoevaluación, coevaluación y heteroevaluación.

3.1 Descripción de la actividad

Actividad: Investigación y exposición oral Esta actividad se desarrolló en la cátedra de Programación Avanzada de la UNLaM con un total de 80 alumnos divididos en dos cursos. Consistió en un trabajo en equipo de investigación y exposición oral. Los alumnos se agruparon en libremente en equipos de 4 a 6 integrantes. La cátedra presentó una lista de temas posibles a investigar. Cada grupo eligió libremente el tema con el cual tuviera mayor afinidad. Luego de asignados los temas, los grupos tenían un plazo de una semana para tener un feedback con el docente asignado, en el cual podían plantear todos los aspectos que el grupo considerase necesario, por ejemplo: definir forma de encarar el tema, límites de la investigación, nivel de profundidad, etc. A la siguiente semana se realizó la exposición oral. La forma de exponer el tema era libre, los grupos tenían a su disposición un cañón conectado a un pc, pizarra interactiva, pizarra tradicional, podían traer todos los materiales que considerasen convenientes. Cada grupo disponía de un plazo de 30 minutos de exposición total (armado, presentación, preguntas). La valoración realizada por cada actor se recogió en documentos preparados para tal fin (plantillas elaboradas en Excel), para obtener los resultados individuales y grupales.

3.2 Rúbricas de evaluación

Se desarrollaron dos rúbricas que estaban a destinadas a los alumnos y una tercera rúbrica destinada a los docentes. Además de las rúbricas se realizó una devolución oral del docente sobre todo el desarrollo de la actividad.

Se establecieron indicadores, relacionados con la competencia del trabajo en equipo, para evaluar el grado de dominio adquirido por los estudiantes, considerados desde distintas perspectivas: autoevaluación, coevaluación y heteroevaluación.

La primera rúbrica permitió indagar sobre el proceso del trabajo en equipo. Se aplicó para realizar una autoevaluación y una coevaluación entre los integrantes del mismo equipo. En esta rúbrica se utilizó una puntuación de 1 al 10 para recoger la valoración de los indicadores propuestos (1 malo - 10 excelente). Los criterios considerados fueron: compromiso, resolución de conflictos, participación en la preparación, dinámica de trabajo. Así mismo se preguntó si realizaron división de tareas y asignación de roles.

Con el criterio compromiso se valoró el grado de cumplimiento en la elaboración de las tareas asignadas. La resolución de conflictos, mostró la habilidad para manejar situaciones de desacuerdo o conflicto, así como la capacidad para proponer alternativas para el consenso o solución. El criterio participación en la preparación recogió la capacidad de aporte de materiales para la investigación, ofrecer ideas para la realización del trabajo o para su mejora. El último criterio de esta grilla, dinámica de trabajo consideró la capacidad de colaborar con el equipo, escuchar, compartir e integrar sus ideas. Así como también su actitud proactiva y esfuerzo para alcanzar los objetivos del equipo, manteniéndolo unido e integrado.

La segunda rúbrica recogió la evaluación de la presentación oral propiamente dicha con la mirada propia y la de los pares. Los indicadores propuestos fueron: contenido, exposición, innovación y creatividad, duración, aplicabilidad.

Con el criterio contenido se mostró su dominio del tema, calidad de la información y la capacidad para responder preguntas. Exposición, con este indicador se recogió el nivel de expresión oral, vocabulario, uso de términos específicos, uso del espacio, postura corporal, contacto visual, interacción con el público. Con el criterio dinámica de la exposición se valoró la coordinación entre el equipo, así como también que el formato de la exposición contemplara las secciones de apertura, desarrollo, cierre. La innovación y creatividad se refiere a los medios que utilizó para exponer, cómo los utilizó, en qué grado contribuyeron a la comprensión del tema, y su nivel de creatividad. La duración registró si se ajustaron al tiempo total asignado y si el tiempo de participación de cada integrante fue el adecuado. El criterio aplicabilidad hizo referencia a si utilizaron ejemplos basados de casos reales o mostraron su aplicabilidad en la vida real.

La tercera rúbrica recogió la mirada de los docentes sobre el desarrollo y exposición de la actividad propuesta. Está compartió algunos indicadores de las rúbricas anteriores: contenido, exposición, innovación y creatividad, duración y aplicabilidad. Se han considerado otros indicadores propios como el uso de las instancias de consulta, bibliografía y la organización de los contenidos. El criterio "uso de la instancia de consulta" permitió, en la etapa de feedback, valorar la calidad de las preguntas, nivel de avance, utilización de la instancia en sí misma. El indicador bibliografía consideró la calidad de las fuentes consultadas. Por último, organización de los contenidos valoró la relación de las ideas para facilitar la comprensión del tema.

3.3 Resultados de la actividad

Nuestra intención en este apartado no es hacer el análisis estadístico de los resultados obtenidos de la evaluación aplicada en esta actividad, sino mostrar las tendencias generales de las formas de evaluación utilizadas para registrar la adquisición de competencias de trabajo en equipo.

A partir de las distintas rúbricas podemos destacar que existieron resultados bastantes cercanos entre la coevaluación y la heteroevaluación para la mayoría de los ítems, cuando el objeto de evaluación era externo al grupo. Es decir que se encontró que los estudiantes tienen la suficiente madurez para evaluar el trabajo de otros equipos. Esto se debe a que ellos transitaron el mismo proceso de realización de las competencias, que estaban evaluando.

No sucedió lo mismo con la coevaluación del resto de los integrantes de su propio equipo, donde se observó que la nota se repartía en más del 90% de los casos en forma uniforme, con un pequeño rango de variación de notas, todas entre 8 y 9. Además, consultaron y consensuaron con sus compañeros las valoraciones que se otorgaron dentro del equipo, lo que denotó cierta inmadurez. Por consiguiente, cuando la coevaluación mide el rendimiento de los integrantes de su propio equipo, se asemeja a la autoevaluación, donde se dan una nota, en general, sobrevaluada.

La autoevaluación y la coevaluación intra-grupo, enfrentan al alumno a una toma de conciencia de su participación y la del resto de los integrantes dentro del equipo. Los estudiantes no lo reflejaron a la hora de plasmarlo en las rúbricas. Actuaron protegiendo al equipo, de forma de no exponer a nadie ante los docentes. Lo mismo ocurrió con el indicador “conflictos” ya que todos los alumnos manifestaron que no existieron conflictos dentro del grupo. Aunque en algunos casos fue notoria su existencia, denotado por el compartimiento de los integrantes durante la exposición o por comentarios informales.

Se registró que los estudiantes no hicieron asignación de roles, pero sí en todos los casos, se dividieron las tareas.

Los indicadores: contenido y bibliografía, tuvieron una valoración más alta que los de exposición, innovación y organización de contenidos. Se observó la poca experiencia de los equipos, en la exposición oral, falta de vocabulario, mal uso del espacio por parte sus integrantes. Se destacó la buena participación de los estudiantes al responder con fundamentos teóricos las preguntas de sus pares. Las bajas valoraciones en creatividad se debieron al pobre aprovechamiento de los materiales y la tecnología que tenían a su disposición. Se registró también en casi todos los equipos altas valoraciones en el indicador aplicabilidad debido a los buenos ejemplos presentados en relación con el mundo real.

4 Conclusiones

Los resultados obtenidos a partir del análisis realizado, nos ha permitido alcanzar los objetivos que nos habíamos propuesto. Obtener tendencias generales orientativas de los modos de evaluación del grado de adquisición de la competencia de trabajo en

equipo. Por otro lado, detectar los problemas en el desarrollo de la experiencia, lo que nos llevó a plantear nuevas estrategias de trabajo para los próximos cursos.

En esta ocasión se utilizó una evaluación multiactores, autoevaluación, coevaluación y heteroevaluación. Con una doble finalidad, evaluar el desarrollo de la competencia de trabajo en equipo y el trabajo final. Utilizando como instrumento para la recolección de datos la rúbrica.

La autoevaluación y la coevaluación dentro de los miembros del mismo equipo se utilizaron para dotar a los alumnos de mayor autonomía en su proceso de aprendizaje. Potenciando la responsabilidad individual para que no se diluya en el equipo. En este punto encontramos la dificultad de que la información brindada por los alumnos no reflejaba fielmente la realidad, debido a un consenso y unificación en la valoración asignada. Para subsanar esta situación proponemos las siguientes estrategias a considerar en futuras actividades: 1) incorporar a los alumnos en la formulación de la rúbrica para que aumente su compromiso y la implicación en la autoevaluación. 2) revisar el rango de cada indicador, disminuyendo la escala a 4 o 5 valores según se quiera permitir o no, un valor neutral. 3) realizar un cuestionario después de cada actividad en la que se aplique esta metodología de evaluación, para constatar el grado de metacognición sobre la adquisición de las competencias de trabajo en equipo. Esto le permitirá al estudiante analizar su historial de aprendizaje para monitorear su mejora.

Para el desarrollo de competencias de trabajo en equipo, al igual que en otras prácticas, es indispensable el feedback con el docente. La mirada del docente actúa como elemento motivador, ayuda a detectar y resolver errores en forma temprana y promueve la continuidad del trabajo. La coordinación del docente en relación al uso de la autoevaluación y la coevaluación consideramos que acompañaría a los estudiantes a habituarse a estos métodos de evaluación.

Referencias

1. Fernández Cruz, F. J., & Fernández Díaz, M. Los docentes de la Generación Z y sus competencias digitales. (2016).
2. Asamblea General de la Asociación Iberoamericana de Entidades de Enseñanza de la Ingeniería (ASIBEI). Declaración de Valparaíso. (2013)
3. Barriga Hernández, C. En torno al concepto de competencia. Educación, año I, núm. 1, mayo. (2004).
4. Viles, E., Jaca, C., Campos, J., Serrano, N., Santos, J. Evaluación de la competencia de trabajo en equipo en los grados de ingeniería. Dirección y Organización, 46, pp. 67-75. (2012)
5. Torrelles, C., Coiduras, J.L., Isus, S., Carrera, X., París, G., Cela, J.M. Competencia de trabajo en equipo: definición y categorización. Revista de currículum y formación del profesorado, 15(3), pp. 329-344. (2011)
6. Torrelles, C., París, G., Sabrià, B., Alsinet, C. Assessing teamwork competence. Psicothema, 27(4), pp.354-361. (2015)
7. Ruíz-Corbella, M., de Rivas Manzano, R. Los valores vinculados a la competencia del trabajo en equipo en entornos virtuales de aprendizaje. Un estudio en la Universidad Técnica Particular de Loja (Ecuador). Revista Complutense de Educación, 26(3), pp.759-780. (2015)

8. Masmitjà, J. A., Irurita, A. A., Trenchs, M. A., Miró, M. B., Marín, A. C., Busquets, M. C., ... & Ruiz, L. M. Rúbricas para la evaluación de competencias. *Cuadernos de docencia universitaria*, 26. (2013).
9. Bunk, G. La transmisión de las competencias en la formación y perfeccionamiento profesionales de la RFA. *Revista europea de formación profesional*, (1), 8-14. (1994).
<https://dialnet.unirioja.es/descarga/articulo/131116.pdf>
10. Cuadrado, C., Fernández, F.J., Fernández, M., Fernández-Pacheco, C., González, D. Lifante, I. y Moya, J. Técnicas de trabajo en equipo para estudiantes universitarios. En X Jornadas Redes de Investigación en Docencia Universitaria, Alicante (2012)
11. Rodríguez-Sandoval, E., Vargas-Solano, É. M., & Luna-Cortés, J. Evaluación de la estrategia " aprendizaje basado en proyectos". *Educación y educadores*, 13(1), 13-25. (2010).
12. Seat, E., & Lord, S. M. Enabling effective engineering teams: A program for teaching interaction skills. *Journal of Engineering Education*, 88(4), 385-390. (1999).
13. Pichon-Rivière, E. *El proceso creador: del psicoanálisis a la psicología social (III)*. Nueva Visión. (1977).
14. Videla L., Blautzik L., Gasior, F., Aubin V., Guatelli R., Cabrera J., Sanche C., Giulianelli D. Técnicas de Gamificación aplicadas a una experiencia práctica como factor de fortalecimiento del proceso de enseñanza aprendizaje de programación. VI Congreso Nacional de Ingeniería Informática – Sistemas de Información (CoNaIISI 2018, Universidad Atlántida Argentina, Universidad FASTA, Universidad Nacional de Mar del Plata y Universidad CAECE.). (2018)
15. Simon, M., & Forgette-Giroux, R. A rubric for scoring postsecondary academic skills. *Practical Assessment, Research & Evaluation*, 7(18), 103-121. (2001).
16. Vélez, Lamberto V.; Palomares, A. Medición, assessment y evaluación del aprendizaje. Publicaciones Puertorriqueñas. (2008).
17. Aubin, V., Blautzik L., Dejean G., Grinsztajn F., Imperiale M. Mejoras en el proceso de enseñanza y de aprendizaje de programación utilizando metodologías de la industria del software como caso particular de las metodologías activas. Hacia un Modelo Pedagógico Transformador. San Justo: UNLaM. 2018. p127 - 136. isbn 978-987-3806-71-12017 (2017)
18. Sotelo, A. F., & Arévalo, M. G. V. Proceso de autoevaluación, coevaluación y heteroevaluación para caracterizar el comportamiento estudiantil y mejorar su desempeño. *Revista San Gregorio*, 1(9), 6-15. (2015).

Enseñando Python en una propuesta de formación docente en enseñanza de la programación

Araceli Acosta¹, Cristián Rojo, M. Cecilia Martínez²,

¹ FAMAF - Universidad Nacional de Córdoba,

² FFyH - Universidad Nacional de Córdoba. CONICET

aacosta@famaf.unc.edu.ar, crirojo@gmail.com, cecimart@gmail.com

Resumen. Un debate actual que se presenta al diseñar propuestas de formación docente para la enseñanza de la programación, o de las ciencias de la computación en las escuelas, es si la formación debe incluir una introducción a la programación en algún lenguaje de alto nivel, o si se debe enfocar en conceptos de ciencias de la computación dado que la programación de alto nivel no es un objetivo que se pretende en la escuela, o simplemente porque una especialización docente no es apropiada para abordar tal desafío. En este trabajo analizamos una experiencia que se propone como objetivo que los docentes que cursan el trayecto formativo sean capaces de escribir programas simples en un lenguaje de alto nivel como Python.

Keywords: Enseñanza de la programación, Diseño curricular, Enseñanza de las ciencias de la computación

1 Introducción

La escuela tiene como misión la transmisión de los saberes que son válidos y relevantes en un momento histórico para permitir el ejercicio pleno de los derechos como ciudadanos y para promover el desarrollo cognitivo de los alumnos. En ese sentido, la programación es la técnica cultural de nuestro tiempo (Busaniche). Estar alfabetizado en esta técnica de la cual somos consumidores y usuarios intensos es necesario para ejercer nuestra ciudadanía. Cerrar la brecha digital, achicar la brecha de género entre quienes se dedican a las carreras relacionadas con la tecnología, socializar a los estudiantes en el oficio de los programadores para que puedan elegir sus futuros profesionales desde el conocimiento de la disciplina, son todos objetivos que se busca lograr con la introducción de la enseñanza de la programación en las escuelas.

La enseñanza de la programación se plantea como una estrategia clave en la construcción de una ciudadanía alfabetizada en los lenguajes de nuestro tiempo. La enseñanza de la programación en la escuela promueve herramientas para comprender las lógicas de funcionamiento de la tecnología digital. Es decir, se trata de reconocer

el conjunto de decisiones que toman y los saberes que dominan quienes desarrollan el software que controla dichas tecnologías. El conocimiento de estos saberes alienta un modo de pensamiento afín a los desafíos de una sociedad atravesada por las nuevas tecnologías. Al decir de Palamidessi (2014, p. 3), “. .. la tecnología nunca es reductible a un problema técnico (...) El problema se complica cuando el establishment educativo tiende a considerar a la tecnología como un objeto que no es digno de reflexión; un simple medio. Pero se trata de un error fatal”.

En consonancia con la Res. 263/15 del Consejo Federal de Educación, que destaca la relevancia que reviste en la actualidad la enseñanza y el aprendizaje significativo de la programación (...) “la Programación es de importancia estratégica en el Sistema Educativo Nacional durante la escolaridad obligatoria, para fortalecer el desarrollo económico y social de la Nación, conforme lo establecido por el artículo 3o de la Ley de Educación Nacional.” (art. 1), el gobierno de Córdoba, la Universidad Nacional de Córdoba y la Fundación Sadosky han realizado individualmente y articulada acciones en estos sentidos. La creación de las escuelas PROA por parte de la provincia, la realización de cursos de capacitación docente por parte de la Universidad y los diversos programas de la Fundación son algunos de los antecedentes.

En este sentido, la formación docente debe cumplir un rol protagónico, si entendemos, como lo establece la resolución del Consejo Federal de Educación 343/18, que la enseñanza de la programación y la robótica debe abordarse en todas las escuelas.

En el año 2015 el Gobierno de Córdoba convoca al equipo de la Universidad para trabajar juntos en el desarrollo de una Especialización Docente de Nivel Superior en Enseñanza de la Programación. Se elabora la propuesta y se dicta entre fines de 2016 y fines de 2018. En el año 2017 la Fundación Sadosky convoca a las universidades a proponer planes de formación para especializaciones docentes en el área, por lo que se presenta la posibilidad de revisar la propuesta pedagógica y comenzar con una segunda cohorte.

Un debate actual que se presenta a la hora de diseñar propuestas de formación docente para la enseñanza de la programación o de las ciencias de la computación en las escuelas es si la formación debe incluir una introducción a la programación en algún lenguaje de alto nivel, o si se debe enfocar en conceptos de ciencias de la computación. Algunos argumentos para optar por lo segundo son, por ejemplo, que la programación de alto nivel no es un objetivo que se persigue en la escuela o, simplemente, que una especialización docente no es apropiada para abordar tal desafío, dado su envergadura.

En este trabajo nos proponemos analizar el objetivo “*que los docentes sean capaces de escribir programas simples en un lenguaje de alto nivel como Python*”, y su cumplimiento en la segunda cohorte de la Especialización Docente de Nivel Superior que se está dictando en la Ciudad de Córdoba en el IES Simón Bolívar en el marco de un acuerdo entre la Facultad de Matemática, Astronomía, Física y Computación de la Universidad Nacional de Córdoba, el Ministerio de Educación de la Provincia de Córdoba y la Fundación Sadosky.

2 Propuesta Formativa

La Especialización tiene como objetivo formar a los docentes en las bases de la programación y su didáctica. El plan de estudios está organizado en proyectos acotados de programación siguiendo los principios del aprendizaje por descubrimiento, enfoque pertinente y privilegiado para su didáctica.

Los proyectos seleccionados tienen la intencionalidad de posibilitar la integración de contenidos fundamentales de la programación con los saberes pedagógicos y tecnológicos necesarios para su enseñanza. En ese sentido, se ofrece una formación que integra contenidos de la disciplina con saberes de la didáctica y pedagogía.

Los contenidos se presentan en un currículum de formato espiralado al modo de nociones que van ampliando su dificultad en tres niveles (desde la familiaridad al uso y, luego, al dominio) a lo largo de la especialidad. De este modo, se espera que los docentes, en primera instancia, se familiaricen con algunos conceptos; luego, los usen en un programa o análisis y, finalmente, produzcan nuevos saberes a partir del dominio conceptual.

En síntesis, la propuesta tiene el objetivo de formar a los docentes en la enseñanza de una área relativamente nueva para las instituciones educativas, recuperando “buenas prácticas” de formación docente: currículum por proyectos, saberes disciplinares y didácticos integrados, y contenidos espiralados.

2.1 Perfil de los cursantes

Este postítulo está dirigido a docentes en actividad dentro del sistema educativo que se desempeñen en el último ciclo de primaria y primero del secundario para promover la calidad de los aprendizajes en estos niveles que presentan los datos más críticos de retención y rendimiento. Si bien, al ser una especialización docente, los cursantes deben contar con formación docente, no se restringen los perfiles disciplinares o área en la que se desempeñan; por lo que entre los cursantes se encuentran docentes de diversa formación y que se desempeñan en diferentes espacios curriculares.

En relación a la formación en informática, el 45% de los casi 200 docentes que comenzaron 1° año cuentan con esa formación. El porcentaje se mantiene aproximadamente en un 48% al finalizar la primera mitad del 2° año (luego de los módulos 6 y 7).

2.2 Estructura de la carrera

La propuesta consta de cuatrocientas horas de cursado distribuidas en dos años. El plan de estudios está conformado por diez módulos, ocho de ellos con formato de proyecto y dos seminarios de trabajo final. Cada módulo-proyecto cuenta con dos clases virtuales y cuatro presenciales. El diseño de cada módulo y la coordinación

académica del mismo está a cargo de uno o dos docentes responsables. Los cursantes se distribuyen en cursos (aulas) de aproximadamente 25 personas y tienen un docente asignado (tutor) encargado del dictado de las clases presenciales por aula, tutorías, el seguimiento de las clases y la evaluación.

Los seminarios de trabajo final se estructuran a lo largo de la carrera con encuentros presenciales y/o tutorías para acompañar el análisis, diseño, implementación en el aula y evaluación de las propuestas de enseñanza de la programación que definen los cursantes en el marco de lo trabajado en el resto de los módulos.

Objetivos Generales

La carrera tiene como principales objetivos que los docentes, al finalizar, sean capaces de:

- Desarrollar programas de computación sencillos a partir del dominio de un lenguaje de programación.
- Comprender, seleccionar y aplicar los conceptos fundamentales de la programación en el desarrollo de proyectos educativos en los diferentes niveles educativos.
- Diseñar experiencias para la enseñanza los conceptos fundamentales de la programación.
- Diseñar experiencias de enseñanza que requieran recuperar los modos de pensamiento propios de la programación.
- Diseñar experiencias de enseñanza que permitan un abordaje crítico de las tecnologías informáticas y su relación con la sociedad.
- Promover trabajos colaborativos entre equipos docentes a partir de las posibilidades que ofrecen los proyectos vinculados a la programación informática.

3. La programación en el marco de la especialización

En el primer año de la carrera los docentes cursaron los módulos de: Lógica, programación y su enseñanza, ensamble y programación de un robot, Administración y configuración de herramientas informáticas, Introducción a lenguajes de programación a través de animaciones y videojuegos, además del primer Seminario de trabajo final. En estos módulos los cursantes se familiarizaron con conceptos básicos de programación y con algunas herramientas didácticas para enseñar estos conceptos. También, en el módulo Herramientas informáticas, se trabajaron conceptos de sistemas operativos, y hardware y experimentaron la instalación de un sistema operativo Linux.

Para el cumplimiento del primer objetivo de la carrera “*Desarrollar programas de computación sencillos a partir del dominio de un lenguaje de programación*” la estrategia perseguida es introducir conceptos de programación a lo largo del primer año de la carrera utilizando herramientas didácticas utilizadas actualmente en las

escuelas como juegos lógicos, programación por bloques, robótica, aunque dejando entrever algunas líneas de código en el último módulo del 1° año.

Al comenzar el 2° año los módulos 6 “Resolución de problemas de programación a partir del diseño de juegos interactivos” y el Módulo 7 “Introducción a la programación orientada a objetos a partir del desarrollo de un sistema de votación” presentan conceptos básicos de programación y estructuras de datos en un lenguaje de alto nivel, en particular Python. En el módulo 7 se introduce de manera muy general definición de abstracciones de datos, reuso, modularización y descomposición de programas, específicamente vinculados a la programación orientada a objetos.

Como objetivos de ambos módulos encontramos el afianzar conceptos básicos de programación. En el Módulo 6 dedicados a la programación de juegos como un “Preguntados” versión texto, mientras que en el Módulo 7 se optó por el desarrollo de un sistema de votación simple y de baja escala, destacando la importancia de propiedades no funcionales de sistemas de software, como la confidencialidad y seguridad, fomentando una mirada crítica de la tecnología.

3.1 Metodología y evaluación de los módulos 6 y 7

Los encuentros presenciales se basaron en una combinación de presentaciones expositivas por parte de los docentes y seguimiento y realización de ejercicios o desafíos que resolvían los cursantes durante la clase. Como espacio de seguimiento y consulta permanente los y las tutores/as guiaron y acompañaron el desarrollo de los ejercicios prácticos mediante la plataforma virtual de la especialización. Asimismo, se utilizó un foro en la misma plataforma donde se intercambiaron opiniones, consejos e informaciones pertinentes al desarrollo de los módulos, privilegiando en ese lugar el fortalecimiento del aprendizaje colaborativo entre pares.

Ante la necesidad puntual de docentes que requerían un acompañamiento orientado a la resolución de los ejercicios de cada módulo se realizaron dos tutorías presenciales para responder a las dudas o inquietudes respectivas las cuales no fueron de carácter obligatorio.

Con el objetivo de evaluar los aprendizajes propios de cada módulo, se realizaron entregas grupales e individuales de ejercicios prácticos de programación, en el caso del módulo 6 utilizando el programa Thonny y en el módulo 7 mediante la plataforma en línea Codeboard. Resulta relevante mencionarlas, porque para la mayoría de los/las cursantes resultó novedoso utilizarlas.

En términos generales se evaluó en primer término la capacidad de escribir programas alineados con los contenidos de cada módulo, acompañado por la reflexión individual que involucró en el módulo 6 explicar todas las líneas de código redactadas y en el módulo 7 reflexionar a partir del código escrito sobre potenciales problemas de seguridad y confidencialidad de su programa si es que éste se utilizara en una elección real.

Cada tutor o tutora recibió los trabajos realizados, y en el caso de no cumplir con los objetivos de aprendizaje, realizó devoluciones con el fin de alentar a la optimización del programa y sus comentarios. Dicha dinámica podía realizarse todas las veces que

fuese necesario. Al terminar el ciclo se evaluó numéricamente de 1 a 10, complementando con comentarios y sugerencias individualizadas para cerrar cada proceso formativo.

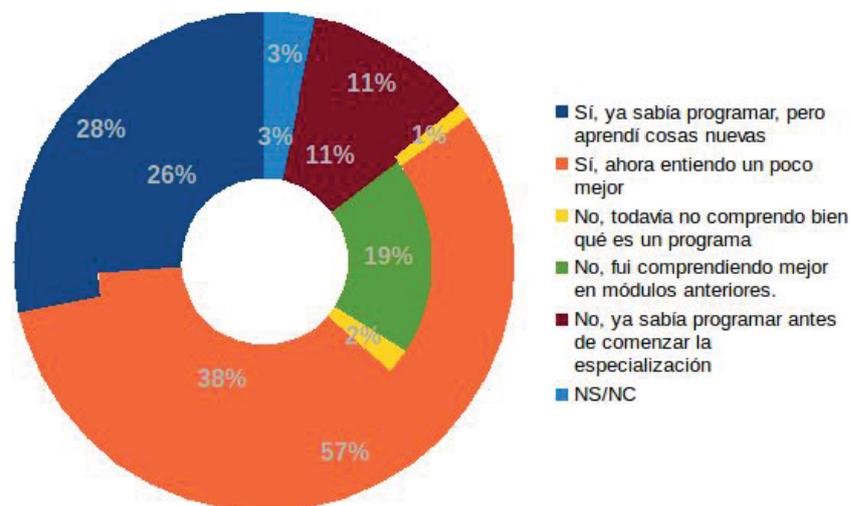
4. Análisis

Al finalizar los módulos 6 y 7 realizamos un análisis preliminar sobre el avance en los objetivos de la carrera a partir de la realización de una encuesta a los cursantes, una evaluación por parte de los docentes y un análisis de la matrícula y las evaluaciones. En este trabajo compartimos algunos elementos que surgieron de este análisis que pueden aportar al debate planteado.

4.1 Análisis según formación previa

Dadas las diversas trayectorias docentes de quienes participan de la carrera y especialmente las diversas experiencias previas relacionadas con sus conocimientos en informática, resulta pertinente preguntarse si dicha variable tiene incidencia en la experiencia de cursado y la apropiación de los conceptos propuestos. En este sentido se analiza, la respuesta a la pregunta “¿Considerás que el cursado de este módulo cambió tu idea de lo que es un programa?”.

¿Considerás que el cursado de este módulo cambió tu idea de lo que es un programa?



Consulta a cursantes luego de finalizar los módulos 6 (externo) y 7 (interno)

Fig. 1. Respuestas a la consulta “¿Considerás que el cursado de este módulo cambió tu idea de lo que es un programa?” al finalizar el 6° y 7° módulo. Consulta a 92 y 88 cursantes, respectivamente, de la Especialización Docente de Nivel Superior en Enseñanza de la Programación del IES Simón Bolívar cohorte 2018-2019.

Es importante destacar, que en esta consulta sólo el 2% manifiesta que todavía no comprende qué es un programa.

La mayoría de quienes declararon una formación formal previa (38 casos), indicaron mejoras en el aprendizaje. Un total de 18 docentes dijeron haber aprendido cosas nuevas, mientras que 12 afirmaron que entendían un poco mejor tras pasar por el módulo. Mientras que los 8 restantes indicaron que no cambió su idea de lo que es un programa.

En el grupo de 9 docentes que no contaba con formación alguna respecto a informática, todos salvo uno respondieron que entendían un poco mejor tras el módulo. Entre quienes señalaron tener una formación autodidacta, quince respondieron que entendían un poco mejor, mientras que uno dijo que no, porque ya sabía desde antes. Para quienes tenían una aproximación informal a los conocimientos en informática 3 aprendieron cosas nuevas, 9 entienden un poco mejor y 1 no comprende bien qué es un programa. En dos casos no hubo respuesta a la interrogante.

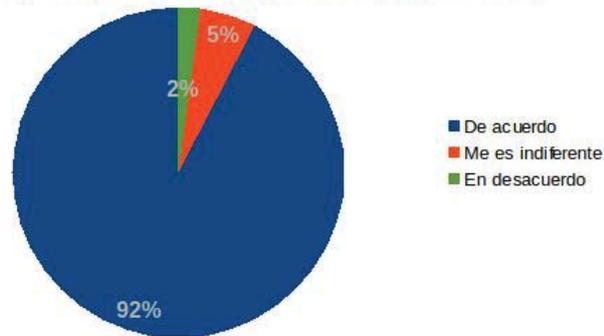
Observando la pregunta siguiente *¿Cuán confiada te sentís para escribir un programa simple en Python?* encontramos una tendencia parecida. Esta vez utilizando una escala de 5 puntos para contestar, donde 1 significa “No podría escribir ni una línea / No sabría cómo comenzar” y 5 significa “Muy seguro/a de poder escribir algo simple”. En general, a esta consulta de 92 cursantes 58% respondió 4 y 5, 32% respondió 3, y sólo un 10% 1 y 2. En particular, de quienes tienen formación formal en informática (45 de las consultadas), el 64% respondió 4 y 5, el 33% respondió 3 y sólo una persona respondió 2. De quienes no contaban con formación previa en el ámbito formal, el 51% respondió 4 y 5, el 32% respondió 3, y el 17% respondió 1 y 2.

4.2 Autopercepción sobre la utilidad

Tras las evidencias anteriores que muestran una adquisición o profundización de conceptos de programación, podemos detenernos a mirar lo que los/las docentes creen sobre el grado de aceptación y relevancia de aprender a programar como parte de su formación.

En ambos módulos se formularon ocho preguntas relacionadas con el aprendizaje de un lenguaje como Python. En ambos casos fueron mayoritarias las respuestas que apuntan a la pertinencia y aplicabilidad de estos conocimientos en sus prácticas docentes.

Aprender programación me permite pensar mejor las herramientas didácticas que aprendimos en módulos anteriores para enseñar en clases.



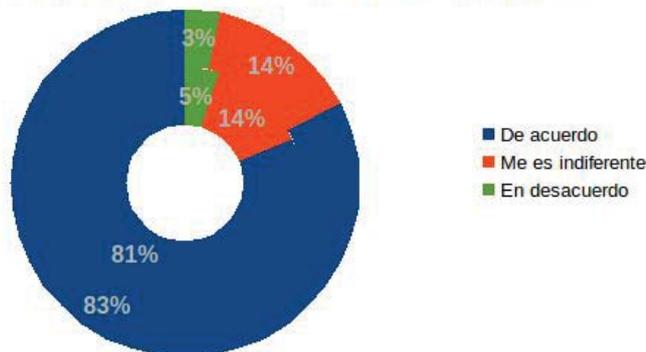
Consulta a los cursantes sobre nivel de acuerdo con la expresión al finalizar el Módulo 7

Fig. 2. Nivel de acuerdo con la frase “Aprender programación me permite pensar mejor las herramientas didácticas que aprendimos en módulos anteriores para enseñar en clases.” al finalizar el 7° módulo. Consulta 89 cursantes de la Especialización Docente de Nivel Superior en Enseñanza de la Programación del IES Simón Bolívar cohorte 2018-2019.

Si pasamos a enfrentar esos resultados con el nivel previo de enseñanza en estas materias tenemos una tendencia que indica que en su mayoría tanto quienes tenían algún tipo de experiencia previa, como los que no, consideran pertinente el aprendizaje de este tipo de lenguaje de alto nivel, el cual creen que podrían utilizar con sus estudiantes. Ambos módulos presentaron cifras parecidas.

Para graficar lo señalado anteriormente se presenta la cantidad de respuestas para ambos módulos ante consulta sobre cuán de acuerdo está con la siguiente afirmación: “Conocer de manera más conceptual qué es un programa, me permite planificar mejor qué temas enseñar y cómo abordarlos”:

Conocer de manera más conceptual qué es un programa, me permite planificar mejor qué temas enseñar y cómo abordarlos



Consulta a los cursantes sobre nivel de acuerdo con la expresión al finalizar el Módulo 6 (externo) y 7 (interno)

Fig. 3. Nivel de acuerdo con la frase “Conocer de manera más conceptual qué es un programa, me permite planificar mejor qué temas enseñar y cómo abordarlos” al finalizar el 6° y 7° módulo. Consulta a 92 y 88 cursantes, respectivamente, de la Especialización Docente de Nivel Superior en Enseñanza de la Programación del IES Simón Bolívar cohorte 2018-2019.

En la evidencia que muestra el gráfico (fig. 3) correspondiente al Módulo 6, participaron 93 docentes, de los/las cuales 84 estuvieron de acuerdo en la posible mejora de sus planificaciones y abordajes respecto a la enseñanza de la programación. Si consideramos la condición previa a la entrada a la especialización el 95,5% (43 de 45 docentes) con aprendizajes formales se manifestaron de acuerdo con la expresión. La misma respuesta se dio en el 85,1% (40 de 47) de quienes no tenían conocimientos formales en el área. A lo anterior sumamos un caso que no declaró sus conocimientos anteriores. En el otro extremo, nadie consideró estar en desacuerdo.

En la evidencia que muestra el gráfico (fig. 3) correspondiente al Módulo 7, participaron 88 docentes, de los/las cuales 79 estuvieron de acuerdo en la posible mejora de sus planificaciones y abordajes respecto a la enseñanza de la programación. Si consideramos la condición previa a la entrada a la especialización el 93% (40 de 42) docentes con aprendizajes formales manifestaron de acuerdo con la expresión. Desde el grupo sin conocimientos formales fueron 36 (83,7%) de un total de 43. Dando una lectura a estos indicadores podemos señalar que tanto quienes tenían como quienes no tenían formación formal previa encontraron en el Módulo de “Introducción a la programación orientada a objetos a partir del desarrollo de un sistema de votación” una vía para potenciales mejoras en sus prácticas de enseñanza.

4.3 Seguimiento en las aulas

Otro indicador que puede dar cuenta del nivel de apropiación y dominio de los contenidos son las dinámicas que se dan en el trabajo diario al interior de las comisiones. Es en este espacio presencial y virtual donde los/las docentes van avanzando en los desafíos presentados. El aula virtual es un canal de comunicación abierto que sirve para consultar, despejar dudas, pedir ayuda y finalmente entregar los trabajos prácticos y el proyecto integrador.

Tomando como referencia al Aula D, una de las cinco comisiones que se encuentran trabajando actualmente, existió una preocupación manifiesta de docentes que no cuentan con conocimientos formales de programación al sentirse en algún punto perdidos/as y con pocas expectativas de tener éxito en el aprendizaje. Dichas preocupaciones se sucedieron al iniciar los módulos 6 y 7, sin embargo, la mayoría demostró asimilación y aplicación de conocimientos volcados a proyectos de programación. Además de haber realizado observaciones críticas a partir de las implicancias sociales que hoy tienen la automatizaciones que median en nuestras vidas.

Ante el desafío de atender por igual las necesidades de quienes provienen y quienes no provienen desde el saber disciplinar de la programación y generar instancias críticas de aprendizaje para introducir estos conocimientos al aula escolar, se puede delinear la idea de que el acompañamiento permanente y la ejecución de ejercicios

concretos de programación aplicados a la resolución de problemas influyen positivamente en el aprendizaje general del grupo. Desde la mirada del tutor del Aula D parte de este buen desempeño se basa en la posibilidad que tuvieron los y las docentes de realizar múltiples entregas de sus trabajos con el fin de ir en un constante proceso de ida y vuelta tratando de optimizar sus propuestas hasta lograr los objetivos de cada módulo.

5 Conclusión

En este trabajo presentamos elementos para el debate sobre la introducción de lenguajes de alto nivel en las propuestas de formación de docentes para enseñar informática o ciencias de la computación en la escuela a partir de la experiencia formativa en la cohorte 2018-2019 del IES Simón Bolívar en la ciudad de Córdoba. Nos guiaron en esta selección preguntas como: ¿es posible introducir la programación en el marco de una especialización de estas características y con docentes que no tienen formación previa en informática? ¿Es pertinente? ¿En cuánto aporta a la formación de la enseñanza de la programación en la escuela asumiendo que la programación en un lenguaje de alto nivel no es algo que se pretenda enseñar en la escuela?.

En el punto 4 presentamos algunos resultados a partir de la evaluación docente y la consulta a los cursantes al finalizar el cursado de los módulos 6 y 7 que ponen el foco en la enseñanza de Python como caso concreto de lenguaje de programación de alto nivel. Esperamos que este aporte pueda servir para alentar la discusión y generar potenciales preguntas de investigación y más experiencias en un campo que cuenta con pocos antecedentes formales y cuya relevancia se refleja en la creciente necesidad de llevar estos conocimientos al aula primaria y secundaria.

Referencias

1. Busaniche, Beatriz (2006) "Alfabetización digital: las fronteras del aprendizaje y el control de la Información". Disponible en <http://www.bea.org.ar/wp-content/uploads/2006/10/alfabetizaciondigital.html>
2. Palamidessi, Mariano (2014) "De la Educación a Distancia a la Educación en Línea, Aportes a un campo en construcción". Disponible en: www.pent.flacso.org.ar/extras/micrositios/libro/prologo.pdf
3. Resolución CFE N° 263/15. Fecha de promulgación Agosto 12/2015.
4. Ley N° 26.206, "Ley de Educación Nacional". Fecha de promulgación Diciembre 14/2006.
5. Resolución CFE N° 343/18. Fecha de promulgación Septiembre 12/2018.

Experiencias en el uso de TIC y rampas digitales en la enseñanza de informática a alumnos de educación terciaria con discapacidad visual o auditiva

Adriana S. Fachal¹, María José Abásolo^{1,2}, Cecilia V. Sanz¹

¹ Instituto Investigación en Informática LIDI, Facultad de Informática – Universidad Nacional de La Plata, La Plata, Argentina

² Comisión de Investigaciones Científicas de la Provincia de Buenos Aires
{adrianafachal@gmail.com} {csanz, mjabasolo}@lidi.info.unlp.edu.ar

RESUMEN En el contexto actual en el que se encuentra nuestra sociedad altamente impactada por la incorporación de las TIC resulta de fundamental importancia resaltar la e-accesibilidad, es decir, las posibilidades de acceso a las TIC que se brindan a las personas con discapacidad. En el presente artículo se presenta el uso de las TIC y su accesibilidad en la enseñanza de informática a alumnos de educación terciaria con discapacidad. Se utiliza el modelo didáctico o de intermediación M-Free y su variante B-Free diseñados para educación inclusiva. Se muestran cuatro casos de estudio con el fin de utilizar y mejorar las competencias de alumnos con discapacidad visual o discapacidad auditiva, de la carrera de nivel terciario Tecnicatura en Análisis de Sistemas de la Escuela Metropolitana de Altos Estudios (EMAE), para lograr la implementación de diversos desarrollos de autoría propia en el contexto de su ámbito personal, educativo o laboral.

Palabras Claves: educación, inclusión, TIC, tecnología, e-accesibilidad, discapacidad, rampas digitales, modelo de intermediación, M-Free

1. Introducción

La incorporación de las TIC en el aula no genera en sí misma cambios en las prácticas educativas, supone un proceso de apropiación de herramientas y conocimientos, y la construcción de concepciones tendientes a incorporar los recursos y materiales digitales como contenidos flexibles, adaptables y transversales. Contextualizar las estrategias y propuestas didácticas para la incorporación de TIC en el trabajo del aula dentro del concepto de educación inclusiva supone la definición de objetivos y la búsqueda de recursos que promuevan aportes significativos para cada tipo de discapacidad e introduzcan una mejora en la calidad educativa.

La accesibilidad consiste en la exigencia de ajustes razonables o adecuación del entorno físico o social a las necesidades específicas de las personas con discapacidad. Se considera que esta adaptación es razonable cuando, sin que suponga una carga desproporcionada, facilita el acceso, utilización y participación de las personas con discapacidad en igualdad de condiciones al resto de los ciudadanos [8]. Desde la perspectiva digital el concepto de e-accesibilidad o accesibilidad Web se define como la posibilidad de acceso con una interacción exitosa entre las personas y los ordenadores a nivel de la interfaz Web [6].

Las teorías de inteligencias múltiples proporcionan un marco donde los usuarios pueden aprender a manejar sus dificultades. Se parte de lo que se denomina nivel inicial de competencia, es decir, los puntos fuertes del estilo de aprendizaje de la personas y se seleccionan estrategias didácticas que puedan ayudar a estimular sus inteligencias logrando de esta manera desarrollar las más eficaces a niveles aún superiores y trazando puentes cognitivos en los que se aprovechan las más desarrolladas para ayudar a mejorar las que presentan graves dificultades [13].

El concepto de adaptación tiene un carácter amplio y resulta necesario realizarlas para compensar las restricciones a las que el alumno está sometido como consecuencia de sus necesidades especiales. Para llevarlas a cabo se debe contar con personal especialmente preparado que diseñe, si fuera necesario, un plan para diferentes necesidades tales como pueden ser, modificar los espacios físicos y eliminar las barreras arquitectónicas hasta facilitar la movilidad en silla de ruedas o con bastón de ciego, adaptar el equipamiento con un mobiliario apropiado a las características físicas y sensoriales de los alumnos desde el que resulte fácil acceder al equipo informático, adaptar el tiempo a los ritmos del alumno, incluir ayudas pedagógicas (actividades complementarias para el aprendizaje del braille, lengua de señas y recursos informáticos entre otros) y seleccionar una metodología que incluya, modifique o excluya los contenidos que se consideren convenientes. Como ejemplo adicional es posible mencionar el bucle magnético [3] que es otro dispositivo que puede facilitar el uso de las instalaciones y los servicios a los usuarios que tengan implante coclear, esta tecnología adaptativa permite que la señal de audio llegue al audifono o al implante.

En cuanto a competencias digitales específicamente cuando los medios estándares no son accesibles se pueden utilizar Rampas Digitales o Tecnología de Apoyo (*Assistive Technology*), que son programas o utilidades cuyo propósito es facilitar el uso de aplicaciones informáticas de tipo general a los usuarios con discapacidad que, de otra forma, se verían obligados a usar sólo programas específicamente diseñados para ellos. En algunas ocasiones con el objeto de lograr la implementación adecuada y completa de determinadas rampas digitales se hace necesario algún tipo de producto de apoyo o ayudas técnicas para para que el usuario pueda interactuar con la computadora, tablet o celular.

Para poder llevar a cabo una adecuada incorporación de las TIC dentro del trabajo de educación inclusiva, enriqueciendo procesos de orientación para diferentes tipos de discapacidades, resulta de fundamental importancia comenzar por aplicar un modelo didáctico. En la sección 2 se presentan el modelo didáctico M-Free y el modelo didáctico simplificado B-Free. Se ha dejado de lado la idea de que existen alumnos estándar con una necesidad específica, un cociente intelectual determinado o una conducta adaptativa de tal o cual rango, pasando a considerar las individualidades de cada uno, con su propia personalidad y un potencial que podrá desarrollar más plenamente cuanto más adecuadamente sea estimulado. En la sección 3 se describen la aplicación de estos modelos didácticos en cuatro casos prácticos concretos de alumnos discapacitados de una carrera en el área informática de nivel terciario que pretenden desarrollar proyectos basados en TIC, dos de los cuales corresponden a alumnos con discapacidad visual y otros dos a alumnos con discapacidad auditiva. Para cada caso se analizan las competencias, los diferentes softwares utilizados y las rampas digitales de las que fue posible valerse para cada uno, así como los resultados obtenidos. Por último en la sección 4 se presentan las conclusiones y trabajos futuros.

2. Modelos didácticos M-Free y B-Free

Un modelo didáctico es un instrumento para abordar la enseñanza de diferentes niveles educativos, en tanto contribuye a establecer los vínculos entre el análisis teórico y la intervención práctica docente [9]. En este caso, se tiene como objetivo desarrollar estrategias didácticas con enfoque en las inteligencias múltiples dando la oportunidad de conocer de manera muy clara cuáles son los puntos fuertes del estudiante y sus motivaciones, y que, a partir de eso, se lo forme en sus competencias individuales detectadas [4].

2.1. Modelo M-Free

El modelo M-Free [13] es un modelo didáctico o de intermediación que tiene como finalidad planificar una actividad educativa dirigida a alumnos con necesidades específicas, potenciando la creatividad del profesorado y cómo esta circunstancia puede mejorar también el rendimiento escolar. Resulta de fundamental importancia que el alumno sea capaz de resolver, de producir o de actuar sobre su realidad y para esto se debe comenzar por pensar en las personas como sus competencias, es decir, comenzar a trabajar desde sus capacidades y no desde su discapacidad. El modelo M-Free original consta de las cinco fases siguientes (Figura 1):



Figura 1. Modelo didáctico M-Free

En la primera fase se averiguan las capacidades iniciales del alumno. Esta evaluación inicial no debe conformarse sólo con determinar las necesidades específicas y los déficits, sino que se debe indicar cuál es la mejor forma de superarlas. Creatica Fundación FREE cuenta con una base de conocimiento en Wikinclusion [15] que ofrece un listado específico y clasificado de competencias a evaluar [5]. La misma se encuentra clasificada en 6 categorías, a saber: Autonomía, Sensomotricidad, Habilidades Sociales; Comunicación y Lenguaje; Matemáticas; Medio Natural y Social; Competencia Digital; Conocimiento Artístico. En la segunda fase se organiza la propuesta curricular bajo la teoría denominada de Inteligencias Múltiples. Se utilizan las competencias de desarrollo detectadas en la fase 1 y partiendo de ahí se debe centrar la selección de los objetivos que pueden ayudar al alumno a conseguir las

competencias contenidas en esta propuesta de trabajo. En la tercera fase se busca correlacionar los objetivos de la propuesta curricular del alumno con los recursos informáticos disponibles. En otras palabras se trata de tejer una malla de objetivos y software basada en las propuestas realizadas por los docentes de educación especial. Es importante remarcar que algunas preguntas que deben plantearse en esta fase son: ¿Para qué necesita un alumno ayuda de las TIC? ¿Por qué usa una ayuda tecnológica y no otra? ¿Cómo integrarla en sus tareas? ¿Cómo redefinir los objetivos para adaptarlos a las necesidades del alumno en cada momento según la propuesta curricular? ¿Qué criterios sugiere el cambio de un hardware o software por otro? En la cuarta fase se debe observar si el alumno necesita alguna adaptación o rampa digital. En la quinta fase se realiza la puesta en marcha y posterior evaluación de las acciones realizadas y puede surgir la propuesta de mejoras que den continuidad al proceso. Lo que resulta relevante en esta fase es conocer si las acciones realizadas en las anteriores ha sido la adecuada y para esto se suele dividir el análisis en competencias conseguidas, competencias en proceso y competencias no conseguidas.

Para finalizar la descripción del modelo es importante destacar que sus 5 fases no en todas las ocasiones se producen en forma secuencial ya que como resultado de la evaluación es posible distinguir que es necesario rectificar la propuesta curricular, que necesita algún tipo de adaptación o software diferente porque no cumple las expectativas deseadas o incluso llegar hasta descubrirse nuevas competencias iniciales por el mismo proceso educativo.

El modelo de intermediación M-Free ha sido aplicado en diversas experiencias educativas que tratan de dar una nueva visión sobre cómo abordar la innovación educativa con TIC en la educación especial. En la Tabla 1 se resumen diversas experiencias recientes clasificándolas de acuerdo al nivel educativo y origen de la población destinataria, el objetivo, el área de trabajo y la discapacidad.

Nivel Educativo / Población	Objetivo	Área/s	Discapacidad	Ref.
Universitario/ Universidad de Sevilla. (Sevilla, España) Asignatura: Nuevas tecnologías aplicadas a la educación en Magisterio de Educación Especial	Mejorar adquisición de competencias educativas TIC con alumnos de distintos niveles socio-culturales y de desarrollo cognitivo	Autonomía, sensomotricidad y habilidades sociales. Competencia digital	Cognitiva - Necesidades Específicas de Apoyo Educativo (NEAE)	[10]
Inicial: niños y niñas del currículo de primer grado/ 8 escuelas del área metropolitana de Panamá	Crear un portal de objetos digitales de aprendizaje para la atención de un grupo de la población que se torna vulnerable por su discapacidad para potenciar su rendimiento escolar favoreciendo su integración en actividades académicas y en la vida diaria.	Autonomía, sensomotricidad y habilidades sociales Competencias de comunicación	Cognitiva - alumnos con necesidades educativas especiales (NEE)	[16]
Educación Especial (primaria): 3 estudiantes varones y 2 mujeres Escuela Normal Víctor Mercante (Argentina, Córdoba)	Desarrollar las competencias necesarias para mejorar problemas de conducta, niveles atencionales, inclusión social y autoestima incluyendo las TIC de forma significativa y gradual. Se dictó un taller bajo el diseño de proyectos transversalizados por diversos recursos TIC donde las actividades propuestas en los diversos espacios invitaron a los alumnos a apropiarse significativamente de los contenidos propuestos en diferentes áreas .	Comunicación y Lenguaje Matemáticas Medio Natural y Social Competencia Digital	Discapacidad Intelectual Síndrome de West	[11]
Adultos Mayores, Adultos y Jóvenes / Centros Culturales y en la Escuela Taller El Vado (Cuenca, Ecuador)	Realizar análisis estadístico de uso de teléfonos inteligentes Aminorar brechas digitales entre sectores de la sociedad logrando alfabetización digital en sectores vulnerables dictando talleres digitales a fin de potencializar destrezas y habilidades en el uso productivo de las TIC y dispositivos inteligentes en función de mejorar la calidad de vida de las persona. Determinar y comparar funciones de accesibilidad que mejoren las competencias de uso y manejo de las Tecnologías Inteligentes.	Competencia Digital	Personas con discapacidad auditiva, física, intelectual, lenguaje, psicosocial (mental) y visual Adultos Mayores que pueden tener alguna discapacidad relacionada con el envejecimiento (visual, auditiva, física)	[17]

Tabla 1. Experiencias Educativas que utilizan modelo de intermediación M-Free

Nivel Educativo / Población	Objetivo	Área/s	Discapacidad	Ref.
Educación Inicial/ Niños de 1 a 4 años (niñez temprana) / (Cuenca, Ecuador)	Proponer un sistema de acciones que propicien la estimulación sensorial de los niños con alteraciones en el neurodesarrollo	Autonomía, sensomotricidad y habilidades sociales Comunicación y Lenguaje	Síndrome de Down Síndrome de Sekeel Autismo Parálisis Cerebral Infantil (PCI)	[7]
Universitario/Universidad Nacional de General Sarmiento (UNGS) (Buenos Aires, Argentina)	Adecuar textos de estudio para estudiantes con discapacidad que no pueden acceder a los textos en forma tradicional. La adaptación de los materiales de estudio con formatos puestos a disposición de esta población de alumnos sirven de apoyo para su inclusión a la vida universitaria.	Autonomía, sensomotricidad y habilidades sociales Accesibilidad (Uso de Rampas Digitales, NVDA, JAWS)	Visual u otras discapacidades perceptivas	[2]
Educación Especial/ Escuela Especial Beatriz Angélica Martínez Allio (Córdoba, Capital [Argentina]) 32 alumnos entre 6 y 13 años	Verificar si el uso de la tecnología instalada en la sala multisensorial instalada en la escuela favorece el lenguaje, la comunicación y el aprendizaje de los alumnos	Comunicación y Lenguaje Autonomía, sensomotricidad y habilidades sociales (subárea: estimulación)	Trastornos psicomotores e intelectuales	[1]
Educación Especial/ 50 personas: jóvenes, adultos y adultos mayores (edad cronológica entre 16 y 55 años) (Quito, Ecuador)	Capacitar a personas con multidiscapacidad que forman parte de la organización FINE incorporando herramientas TICs para lograr una inserción laboral y social efectiva permitiendo a esta población tener una mejor calidad de vida tanto a nivel personal como familiar	Autonomía, sensomotricidad y habilidades sociales (subárea: hábitos de trabajo y estudio) Comunicación y Lenguaje Transición al mundo laboral	Discapacidad intelectual, física y sensorial Multidiscapacidad	[12]

Tabla 1. Experiencias Educativas que utilizan modelo de intermediación M-Free (continuación)

2.2. Modelo didáctico simplificado B-Free

El modelo B-Free completa el modelo de intermediación M-Free, proporcionando una estructura visual que muestra los diferentes elementos clave que participan en la intermediación con TIC desde el punto de vista de las múltiples interacciones que los caracterizan [14]. El modelo B-Free muestra los vínculos entre las capacidades múltiples actuales de la persona, las competencias que se desean lograr o propuesta curricular, el software y el hardware estándar que se ha seleccionado, y las rampas digitales y adaptaciones que le van a permitir interactuar con la computadora y la discapacidad que le dificulta su participación en la actividad. Como se muestra en la Figura 2, la dinámica consiste en el diseño de una plantilla en forma de barco, diseñada con el software Inspiration.¹

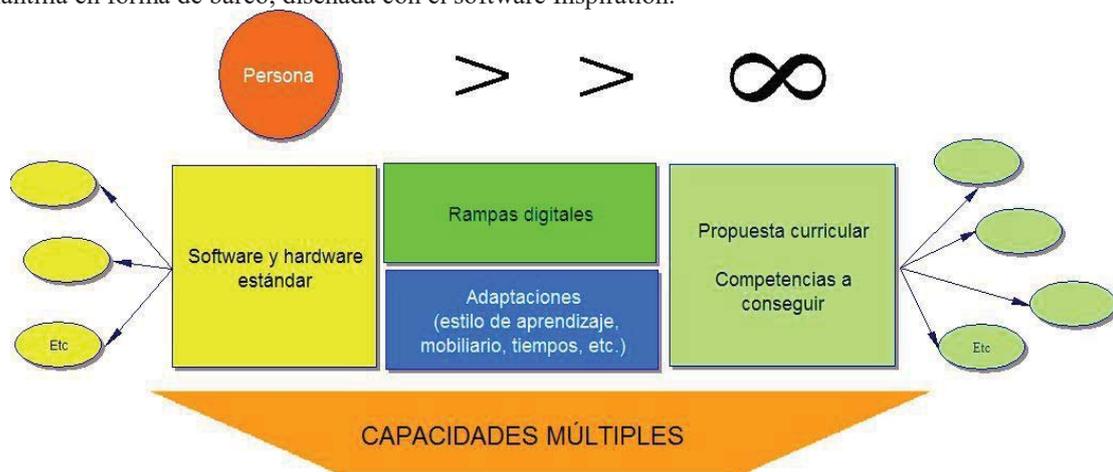


Figura 2. Modelo didáctico B-Free

¹ Inspiration <http://www.inspiration.com/>

3. Casos de estudio

3.1. Metodología

Se tiene por objeto ayudar a estudiantes informáticos de nivel terciario que poseen alguna discapacidad a lograr la implementación de desarrollos informáticos. Los casos de estudio se trabajan con dos objetivos particulares. El primero es el de reforzar y desarrollar niveles superiores de las competencias detectadas en cada uno de los estudiantes con discapacidad. El segundo consiste en distinguir cómo la informática favorece tanto la interdisciplinariedad entre algunas de las materias de la carrera de estudio como la accesibilidad a todos los usuarios con independencia de sus limitaciones.

Se trabaja con alumnos de EMAE². Dicha institución fue el primer instituto superior de educación terciaria de Latinoamérica en brindar a personas con discapacidad motora, visual y auditiva un espacio donde adquirir la formación técnica requerida para ser productivos en un mercado laboral dinámico y competitivo. Se aborda la discapacidad desde el foco de la igualdad de oportunidades poniendo énfasis especial en el diseño universal, el software libre y la eliminación de los obstáculos tecnológicos y metodológicos que dificultan la inserción, integración, inclusión y normalización. En la materia Práctica Profesional dictada en tercer año de la carrera Análisis de Sistemas de Información se presentan diferentes propuestas cuya motivación transcurre a través de necesidades personales o laborales para luego planificar una currícula basada en proyectos educativos que permitan a los alumnos reforzar competencias existentes o desarrollar nuevas. Se presentan cuatro casos, dos de los cuales son estudiantes con discapacidad visual y los otros dos corresponden a estudiantes con discapacidad auditiva (Tabla 2).

Caso de estudio	Discapacidad	Objetivos del proyecto	Implementación informática
Caso 1	Visual	Desarrollar un sitio web para su espacio de trabajo y así lograr un mejor desempeño laboral.	Sitio web dinámico
Caso 2	Visual	Desarrollar una aplicación de escritorio con base de datos para uso personal	Aplicación de escritorio
Caso 3	Auditiva	Desarrollar un sitio web para proyectarlo a un cliente en particular.	Sitio web dinámico
Caso 4	Auditiva	Desarrollar una aplicación web que permita un nuevo aporte a su comunidad	Aplicación web

Tabla 2. Casos de estudio

Los proyectos se llevaron a cabo durante 12 meses. Todos los casos, a excepción de uno que se realizó de forma extracurricular exclusivamente, se llevaron adelante durante la cursada, horas extras-curriculares y orientación según la necesidad de implementación de cada caso en particular.

Se aplicaron los modelos didácticos M-Free y B-Free con el objeto de presentar y llevar adelante una propuesta didáctica que parta de la valoración de las capacidades de cada alumno. Considerando que el perfil de los alumnos es informático es necesario dentro del análisis de sus capacidades iniciales focalizarse en competencias digitales y no capacidades en general.

3.2. Competencias digitales

El modelo M-Free se basa en el aspecto de capacidades iniciales comenzando por definir el nivel de competencia de cada persona poniendo énfasis en las destrezas y habilidades que puede desarrollar siempre recordando que el modelo se apoya en el paradigma de crecimiento y no en los aspectos relativos a la incapacidad de la persona. Como parte de este trabajo, se han añadido 29 nuevas competencias digitales no existentes en el listado Wikinclusion [15], clasificadas dentro de las siguientes categorías HTML, CSS, uso de Servidores Locales, Programación PHP, Motor de Base de datos (MySQL) y Programación PHP para acceder a base de datos (Anexo 1). Esta decisión es justificada en que los alumnos son estudiantes de la carrera de análisis de sistemas y por consiguiente las competencias digitales a desarrollar son de un nivel más elevado a las existentes.

3.3. Adaptaciones y rampas digitales utilizadas

La rampa digital utilizada con los estudiantes con discapacidad visual es el software JAWS (*Job Access With Speech*) que es un producto de la compañía Freedom Scientific³ y se trata de un lector de pantalla desarrollado para usuarios de PC cuya pérdida de visión les impide ver el contenido de la pantalla o navegar con un mouse. Este tipo de accesibilidad se denomina “leer con los oídos” ya que el software

² Escuela Metropolitana de Altos Estudios (EMAE) <http://alpi.org.ar/es/emaec/>

³ JAWS Freedom Scientific <https://support.freedomscientific.com/JAWSHQ/JAWSHeadquarters01>

lector de pantalla incluye un sistema de voz integrado que le permite al estudiante su accesibilidad a la PC contando con la posibilidad de interactuar auditivamente con cualquier software instalado. Este tipo de usuarios no cuentan con la posibilidad de utilizar el mouse y a través del uso de esta rampa digital consiguen acceder tanto a los softwares como a la información de su PC mediante atajos de teclado especiales que no serían posibles de acceder sin ella. Otras opciones posibles de acceso libre son el NVDA, lector de pantalla gratuito y de código abierto para Microsoft Windows y el Orca, lector de pantalla para Linux.

En relación a los estudiantes con discapacidad auditiva como ayuda pedagógica se utilizó el recurso del uso de Lengua de Señas Argentina (LSA). En diferentes clases hubo intérprete de LSA, material con videos en LSA y subtítulos y el docente a cargo de la experiencia con formación previa en LSA ha interpretado en forma presencial temas específicos y consultas de práctica.

3.4. Caso 1: Discapacitado visual - Sitio Web

Se propone el desarrollo de un sitio web dinámico a utilizar en un comercio de ventas de insumos de informática. La aplicación cuenta con un catálogo en el que el usuario encontrará todos los productos disponibles. Además dispondrá de un buscador que permite al usuario seleccionar diferentes categorías y de esta forma tener mayor precisión para poder encontrar el producto. La tabla 3 presenta el objetivo de su trabajo propuesto, las competencias digitales iniciales así como las competencias deseadas relacionadas con el software y las actividades a desarrollar. Debido a que el estudiante no tiene posibilidad de acceder al mouse se presenta el uso del lector de pantalla JAWS como rampa digital. El alumno es muy hábil tanto en el manejo del teclado como en el dominio del uso del JAWS. La figura 3 muestra el diagrama B-Free de este caso. Como resultado el estudiante pudo lograr la implementación de su sitio web respetando el diseño corporativo solicitado. Pese a su discapacidad visual, pudo incluir imágenes y manejo de colores a través del uso de código de colores. Con toda evidencia expresa que ha podido llevar adelante el proyecto gracias a las posibilidades ofrecidas por el JAWS y en este sentido expone que ha logrado reafirmar sus competencias informáticas y laborales. El estudiante hizo fuerte hincapié acerca del esfuerzo extra que debió llevar a cabo para lograr mejorar sus competencias de programación pero en contrapartida afirmó que la dinámica adicional aportada por su web al usuario le ha permitido alcanzar mayor número de consumidores además de agilizar a él mismo la consulta de información.

Datos personales Sexo: masculino Edad: 50 Discapacidad: No vidente. Desprendimiento de retina bilateral. Escolaridad: nivel terciario Trabajo: comercio propio de ventas de insumos de informática	
Objetivos Desarrollo de un sitio web dinámico a utilizar en un comercio de ventas de insumos de informática de su entorno laboral. La aplicación cuenta con un catálogo en el que el usuario encontrará todos los productos disponibles. Además dispondrá de un buscador que permite al usuario seleccionar diferentes categorías y de esta forma tener mayor precisión para poder encontrar el producto. Motivación: uso de la web dentro de su ámbito laboral	
Competencias digitales iniciales 6.1 Ergonomía y Reconocimiento de Dispositivos: 9001 9002 9003 9052 9055 9058 9061 6.2 Procesador de Texto: 9100 9102 9104 9110 9111 9112 9113 9115 9122 9129	Competencias digitales de la propuesta curricular HTML: A001 A002 A003 A004 A005 A006 CSS: B001 B002 B003 SERVIDOR LOCAL: C001 C002 C003 C004 C005 C006 PROGRAMACIÓN PHP: D001 D002 D003 D004 D005 D006 D007 D008 D009 D010 PROGRAMACIÓN PHP BASE DATOS: E001 E002 E003 6.3 BASE DE DATOS: 9200 9202 9213 9215 9216 9219 9228 PROGRAMACIÓN PHP BASE DATOS: E001 E002 E003
Software Bloc de notas Navegador web Easy PHP PHP MyAdmin Adaptaciones y Rampas Digitales Grabadora/Reproductor de audio. JAWS (Job Access With Speech)	Actividad a realizar Armar la estructura de la página web Visualizar la página HTML Agregar diseño y color a la página Armar la estructura completa del sitio web Habilitar el servidor y cargar el sitio web en éste Armar formulario de mails Implementar BD y tablas para el catálogo de productos Crear buscador de productos con filtros adecuados

Tabla 3. M-free caso 1

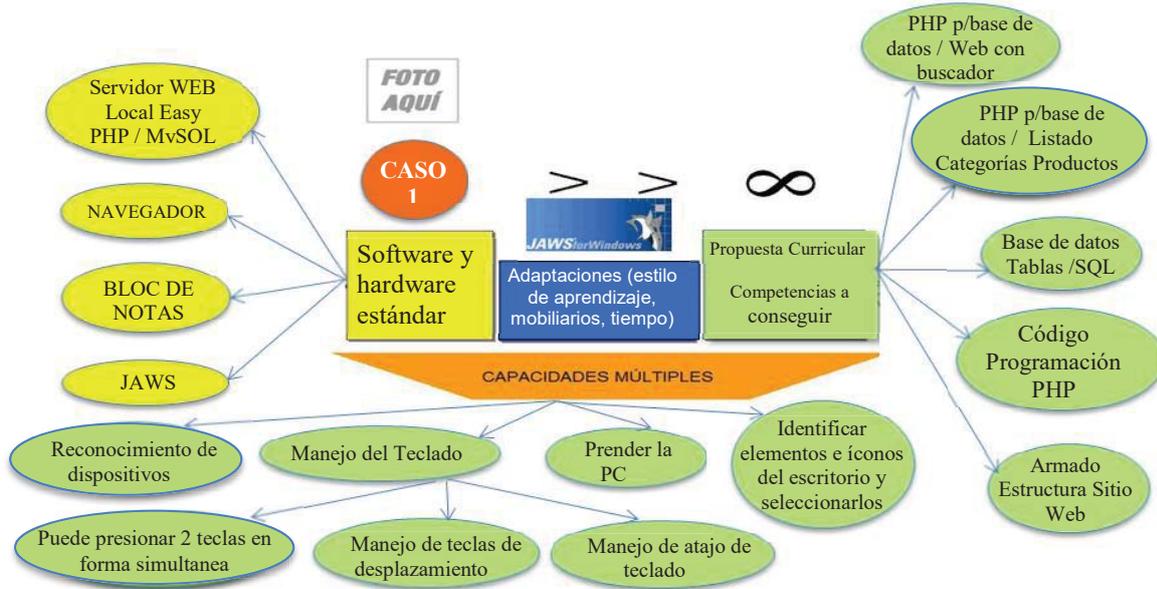


Figura 3. Diagrama B-Free correspondiente al caso 1

3.5. Caso 2: No vidente - Aplicación de escritorio en ACCESS

Se propone el desarrollo de una aplicación de escritorio para el control de objetos personales. La Tabla 4 presenta el objetivo de su trabajo propuesto, las competencias digitales iniciales así como las competencias deseadas relacionadas con el software y las actividades a desarrollar. Debido a que el estudiante no tiene posibilidad de acceder al mouse se presenta el uso del lector de pantalla JAWS como rampa digital. Si bien el estudiante maneja atajos mínimos de JAWS, no tiene gran habilidad para valerse de la herramienta. No conoce los atajos de teclado ni de las operaciones básicas del explorador de windows ni de las funciones de access. La figura 4 muestra el diagrama B-Free de este caso.

Como resultado el estudiante pudo lograr construir su propia aplicación de escritorio utilizando Access mediante el uso de estilos de diseños disponibles en el software. Más aún, se han creado consultas de acción con el objeto de desarrollar reportes con información de mayor complejidad. El estudiante puso un muy fuerte acento en el esfuerzo extra que debió llevar a cabo para conseguir todos los atajos de teclado específicos para poder acceder a las opciones de access, no obstante afirma que esto ha sido parte de lo que le ha permitido adquirir todas sus nuevas competencias digitales referidas específicamente al manejo de base de datos. Evidentemente el logro del proyecto ha sido gracias a la posibilidad de accesibilidad ofrecida por el JAWS.

Datos personales Sexo: masculino Edad: 60 Discapacidad: No vidente. Retinita Picmentaria Observaciones: no tiene ninguna percepción a la luz Escolaridad: nivel terciario Trabajo: no trabaja	
Objetivos Desarrollo de una aplicación de escritorio que permita el ingreso de datos personales, datos de CDS y préstamos de CDS realizados. Además deberá contar con la posibilidad de generar informes de préstamos históricos y de CDS que no estén disponibles por estar prestados actualmente.	
Motivación: uso de la aplicación para su control personal de CD's para evitar pérdidas de los mismos	
Competencias digitales iniciales 6.1 Ergonomía y Reconocimiento de Dispositivos: 9001 9002 9003 9052 9058 9061 6.2 Procesador de Texto: 9100 9102 9104 9110 9111 9112 9113 9127 9129 9143	Competencias digitales de la propuesta curricular 6.3 Base de Datos 9200 9202 9204 9209 9212 9214 9215 9218 9219 9221 9224
Software Explorador Windows ACCESS Adaptaciones y Rampas Digitales JAWS (Job Access With Speech)	Actividad a realizar Armar una aplicación de escritorio que permita el ingreso de datos personales, datos de CDS y préstamos de CDS realizados Generar informes de préstamos históricos y de CDS prestados en el momento.

Tabla 4. M-Free caso 2

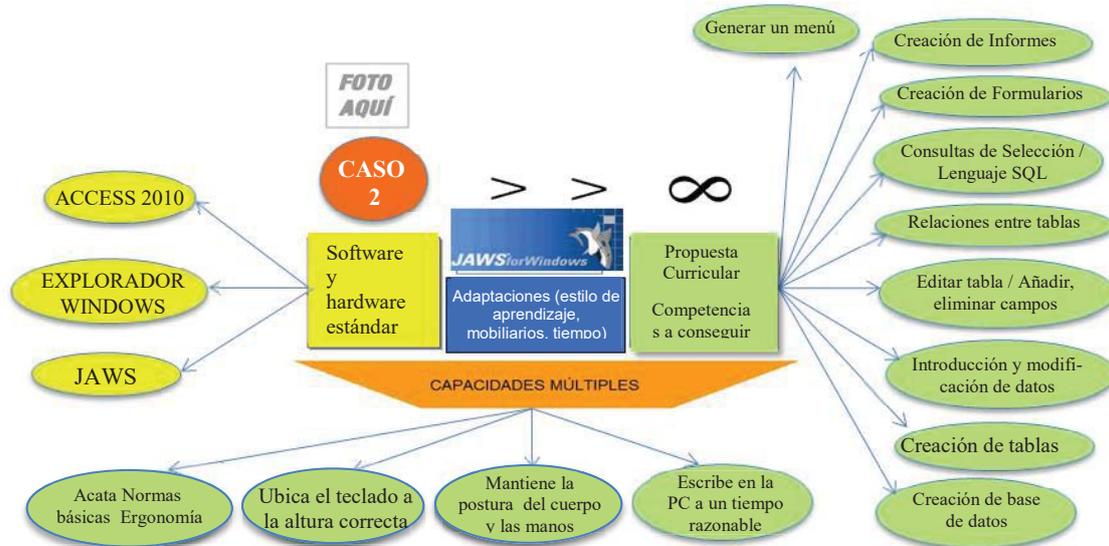


Figura 4. Diagrama B-Free correspondiente al caso 2

3.6. Caso 3: Discapitado auditivo - Sitio Web

Se propone el desarrollo de un sitio web dinámico para el mercado inmobiliario que permita de forma simple y dinámica cargar las propiedades en venta o alquiler con fotos, características técnicas y ubicación del inmueble. Por otra parte cuenta con un motor de búsqueda dentro de la página web con el objeto de que el usuario aplicando diferentes filtros pueda encontrar fácilmente el inmueble deseado de acuerdo a sus necesidades. La tabla 5 presenta el objetivo de su trabajo propuesto, las competencias digitales iniciales así como las competencias deseadas relacionadas con el software y las actividades a desarrollar. La figura muestra el diagrama B-Free de este caso. Debido a la dificultad que presenta el estudiante tanto para la lectura y comprensión de textos técnicos como ayuda pedagógica se ha incluido el uso de material en formato digital en LSA e interpretación en LSA por parte del docente durante las clases presenciales. Si bien el alumno es muy hábil en lectura labial siempre se presenta dificultad frente a términos con fonemas de igual vocalización.

Datos personales Sexo: masculino Edad: 22 Discapacidad: Hipoacusia neurosensorial bilateral Escolaridad: nivel terciario Trabajo: no trabaja	
Objetivos Desarrollo de un sitio web dinámico para el mercado inmobiliario que permite de forma simple y dinámica cargar las propiedades en venta o alquiler con fotos, características técnicas y ubicación del inmueble. Por otra parte cuenta con un motor de búsqueda dentro de la página web con el objeto de que el usuario aplicando diferentes filtros pueda encontrar fácilmente el inmueble deseado de acuerdo a sus necesidades. Con respecto a la implementación se proponen 2 potenciales clientes. Motivación: uso de la aplicación con clientes potenciales reales además de aprender un nuevo lenguaje de programación incluyendo manejo de base de datos.	
Competencias digitales iniciales 6.1 Ergonomía y Reconocimiento de Dispositivos 9001 9002 9003 9052 9053 9055 9056 9057 9058 9059 9060 9061 9063 9066 9067 6.2 Procesador de Texto 9100 9102 9104 9106 9108 9109 9110 9111 9112 9113 9114 9115 9117 9120 9121 9122 9129 9131 9132 9134	Competencias digitales de la propuesta curricular HTML: A001 A002 A003 A004 A005 A006 CSS: B001 B002 B003 SERVIDOR LOCAL: C001 C002 C003 C004 C005 C006 PROGRAMACIÓN PHP: D001 D002 D003 D004 D005 D006 D007 D008 D009 D010 6.3 BASE DE DATOS: 9200 9202 9213 9215 9216 9219 9228 PROGRAMACIÓN PHP BASE DATOS: E001 E002 E003 E004 E005 E006
Software Bloc de notas: HTML - CSS - Código PHP Navegador web: uso de servidor local, prueba del sitio web Easy PHP: Servidor Local PHP MyAdmin: Base de datos Adaptaciones y Rampas Digitales LENGUA DE SEÑAS (LSA)	Actividad a realizar Armar la estructura de la página web Agregar diseño corporativo del cliente Habilitar el servidor y cargar el sitio web Armar formulario de envíos de mails Implementar base de datos y tablas necesarias para la carga y consulta de inmuebles Crear buscador de inmuebles con filtros adecuados Desarrollar un módulo de ABM de inmuebles

Tabla 5: M-Free caso 3

Como resultado el alumno pudo lograr la implementación de su sitio web implementando correctamente su base de datos y controlando de forma adecuada los filtros de búsqueda necesarios. Además resulta de importancia resaltar que otra de las ventajas en particular de este desarrollo es que el mismo usuario cuenta con la posibilidad de actualizar la información de su negocio de forma independiente dentro de la misma aplicación. Evidentemente manifiesta que ha logrado participar de forma activa en el desarrollo de su proyecto gracias al uso de lengua de señas considerándose integrado en cuanto a accesibilidad a la información y comunicación. Por último, en lo que respecta al diseño no solo puntualizar que ha resultado formal y profesional en su trabajo, sino que además el alumno ha sido capaz de adecuar el diseño a la imagen corporativa de otra organización.

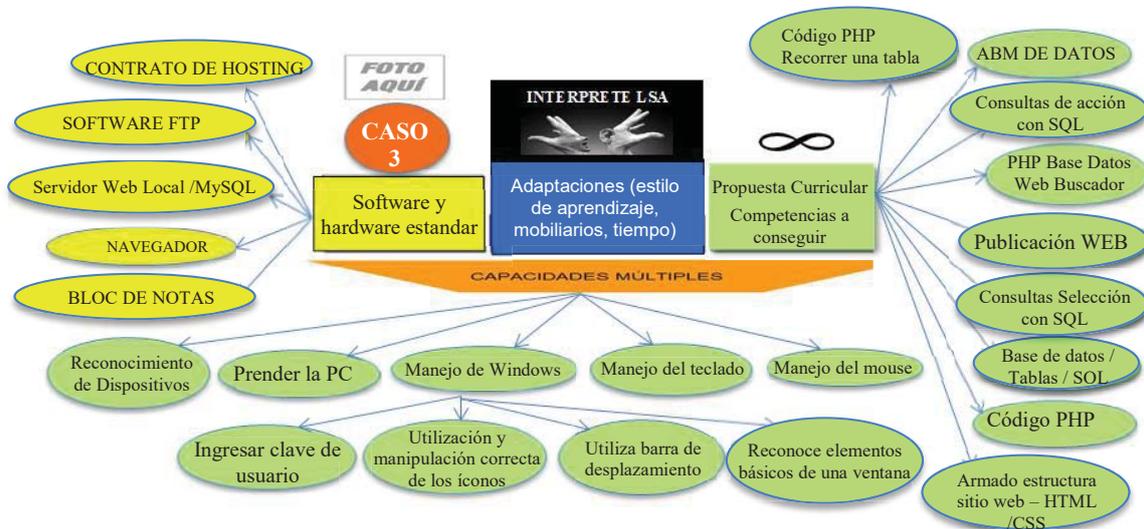


Figura 5. Diagrama B-Free correspondiente al caso 3

3.7. Caso 4: Discapitado auditivo - Aplicación Web

Se propone el desarrollo de una aplicación web que contenga vocabulario técnico correspondiente al área de informática cuyo objetivo sea aportar señas operativas básicas de la especialidad para poder ser utilizadas por la comunidad sorda dentro de un ámbito determinado. La aplicación presenta el vocabulario a través de 9 categorías, cada una de ellas con sus términos y correspondientes señas en LSA además de una frase complementaria explicativa del término tanto en LSA como en español. La tabla 6 presenta el objetivo de su trabajo propuesto, las competencias digitales iniciales así como las competencias deseadas relacionadas con el software y las actividades a desarrollar. Si bien el alumno es hábil en lectura labial se presentan dificultades de comprensión frente a términos con fonemas de igual vocalización y además no ha sorteado la dificultad de lectura y comprensión de textos técnicos. Debido a estas condiciones y sumando que el alumno tiene un excelente dominio de LSA se ha incluido como ayuda pedagógica interpretación en LSA a cargo de una intérprete profesional. La figura 6 muestra el diagrama B-Free de este caso.

Como resultado el estudiante pudo lograr satisfactoriamente la implementación de su aplicación web con vocabulario de informática en LSA. Pese a su discapacidad auditiva ha podido adquirir nuevas competencias informáticas complejas en las áreas específicas de programación e implementación de base de datos. Evidentemente afirma que ha podido lograr con éxito su proyecto gracias a la interacción existente en LSA e hizo fuerte hincapié en que se le ha facilitado su tarea de adquirir nuevos conocimientos de alta complejidad en las áreas planteadas debido a que no ha tenido que hacer un esfuerzo extra adicional para sortear barreras de comunicación. A esto el estudiante agrega que se ha visto estimulado específicamente en su aprendizaje de programación logrando objetivos más elevados de los que consideraba en un comienzo debido a su fuerte intención de implementar un nuevo aporte positivo dentro del ámbito educativo del grupo social al que pertenece.

En cuanto a adaptaciones y ayudas tecnológicas ofrecidas como rampas digitales tales como el JAWS y el uso de lengua de señas han permitido a las personas con discapacidad utilizar los mismos programas que el resto de las personas (no etiquetados como discapacidad) además de haber podido recibir el mismo entrenamiento obteniendo una igualdad de oportunidades.

Datos personales Sexo: masculino Edad: 34 Discapacidad: Auditiva Escolaridad: cursando licenciatura en sistemas actualmente Trabajo: Agente de Servicio Penitenciario Bonaerense	
Objetivos Desarrollo de una aplicación web con vocabulario técnico correspondiente al área de informática. La aplicación presentará clasificados en categorías diferentes términos escritos en español, su seña en LSA y además una explicación del término también en ambos idiomas (español - LSA). Motivación: Dar un nuevo aporte para su comunidad. Se trata del agregado de vocabulario técnico que no existe estandarizado y posibilita el aprendizaje dentro de una nueva área a las personas con discapacidad auditiva.	
Competencias digitales iniciales I – Desarrollo de contenidos: I-1 I-2 I-3 I-4 I-5 I-6 I-7 I-8 I-9 II – Integración y re-estructuración: II-1 II-2 II-3 II-4 II-5 II-6 III – Identificación de Necesidades y Respuestas Tecnológicas: III-1 III-2 III-3 III-4	Competencias digitales de la propuesta curricular HTML: A001 A002 A003 A004 A005 A006 CSS: B001 B002 B003 SERVIDOR LOCAL: C001 C002 C003 C004 PROGRAMACIÓN PHP: D001 D002 D003 D004 D005 D006 D007 D008 D009 PROGRAMACIÓN PHP BASE DATOS: E001 E002 E003 6.3 BASE DE DATOS: 9200 9202 9213 9215 9216 9219 9228 PROGRAMACIÓN PHP BASE DATOS: E001 E002 E003
Software NotePad++ Navegador Web XAMPP PHP MyAdmin Adaptaciones y Rampas Digitales LENGUA DE SEÑAS (LSA)	Actividad a realizar Crear la estructura de la página web con HTML y CSS Habilitar el servidor y cargar el sitio web Implementar la base de datos y tablas necesarias para la carga del vocabulario Escribir código PHP que permita acceder al vocabulario

Tabla 6. M-Free caso 4

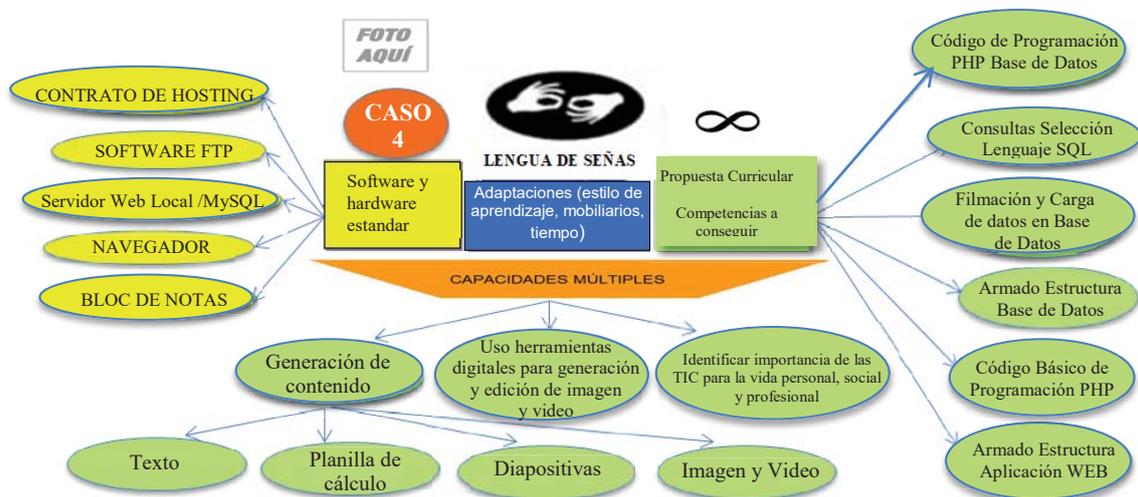


Figura 6. Diagrama B-Free correspondiente al caso 4

4. Conclusiones y Trabajo Futuro

En este artículo se presenta la aplicación del modelo de intermediación M-Free y B-Free para planificar el desarrollo de un proyecto que permita fortalecer competencias existentes y aplicar una planificación transversal integrando conocimientos de otras materias. Se revisan experiencias encontradas en la bibliografía y se describen cuatro experiencias educativas llevadas a cabo con estudiantes de nivel terciario en el área de informática, dos con discapacidad auditiva y 2 personas con discapacidad visual.

Los resultados alcanzados ponen de manifiesto que la utilización de software y hardware apropiados junto a una metodología basada en el uso de las capacidades múltiples de la persona, han posibilitado el logro del cumplimiento de los objetivos propuestos por los alumnos pudiendo trabajar de esta manera por una igualdad de oportunidades de los estudiantes. El uso de herramientas TIC que se requerían para el desarrollo de su proyecto y además su producto final ha resultado un beneficio dentro de su ámbito laboral contando de esta manera con logros viables en igualdad de condiciones que sus pares.

El trabajo futuro incluye el uso de métricas para evaluar accesibilidad y usabilidad. Se están evaluando técnicas tradicionales tales como encuestas, observación y entrevistas, modificados para satisfacer necesidades de comunicación sorda por medio de ayudas visuales, subtítulos y adaptación de redacciones de preguntas de acuerdo al nivel de alfabetización

5. Referencias

- Ballarino V.; Rivarola M.; Beltramone D. (2018). *Aplicación de Interfaces Naturales y Sala Multisensorial en Escuela Especial para Comunicación Aumentativa, Alternativa y Aprendizaje*. Recuperado de: <https://revistas.unc.edu.ar/index.php/med/article/view/21275>
- Biset, Daniel H.; Goyochea, Gustavo L. (2015). *Tecnología y procesos aplicados a la inclusión de estudiantes universitarios con discapacidad*. Recuperado de: http://bib.fcien.edu.uy/jbdu2015/wp-content/uploads/2015/05/Biset_Tecnolog%C3%ADa-y-procesos-aplicados-a-la-inclusi%C3%B3n-de-estudiantes.pdf
- Chamorro M.F. (2015). *Adaptive technology and access to information in university libraries*. ACADEMO Revista de Investigación en Ciencias Sociales y Humanidades, vol 2, núm 2. Recuperado de: <https://revistacientifica.uamericana.edu.py/index.php/academo/article/view/25/23>
- Chrobak, R.; Leiva Benegas, M. (2006). *Concept Maps: Theory, Methodology, Technology*. San José, Costa Rica.
- Creatica Fundación FREE Iberoamericana para la Cooperación (s/f). Base de Conocimiento Wikinclusión. Disponible on line <http://capacidad.es/Anexo.pdf>
- Fernández-Aquino, L. C. (2009). E-accesibilidad y usabilidad de contenidos digitales. Por una sociedad de la información y el conocimiento no excluyente (Tesis doctoral). Valencia: Universitat Politècnica de València. Recuperado de <https://riunet.upv.es/handle/10251/4330?show=full&locale-attribute=en>
- Figuerola Cruz, M.; Milton A.; Campoverde Molina; Calle Vintimilla S.G. (2015). *Intervención Temprana en niños con alteraciones en el neurodesarrollo desde la Sala Multisensorial. Un reto en la Academia Ecuatoriana*. Latin American Journal of Computing, Vol II, num 3, pp. 55-62.
- García Martín, J. M. (2005). *La discapacidad hoy. Psychosocial Intervention*, vol. 14, núm. 3, pp. 245-253. Colegio Oficial de Psicólogos de Madrid. Madrid, España. Recuperado de: <http://www.redalyc.org/articulo.oa?id=179817547001>
- García Pérez (2000). Los modelos didácticos como instrumento de análisis y de intervención en la realidad educativa. Biblio 3W. Revista Bibliográfica de Geografía y Cs. Sociales. Universidad de Barcelona ISSN 1138-9796. Nº 207
- Gonzalez Perez, A. (2012). El uso de las TIC a través del desarrollo de microproyectos con alumnos de Educación Especial. Universidad de Sevilla.
- Lamberti, E. M. (2018). ¿Estudiantes aburridos o propuesta pedagógica poco motivadora? – Experiencia de inclusión significativa de las TIC en los procesos de enseñanza y aprendizaje. En Gallegos Navas M., La inclusión de las TIC en la educación de personas con discapacidad. Relatos de experiencias (pp. 85)
- Maldonado Garcés, V.; Ortiz Carranco, N.Y. (2017). TIC en inclusión y discapacidad. En V.Robles, R. Sánchez, P. Ingavélez y F. Pesántez, Enfoque práctico desde las Tecnologías Emergentes" (pp. 419).
- Montoya, R.; Castellano, R. (2011). *Laptop, Andamiaje para la Educación Especial*. Günther Cyranek, Montevideo, Uruguay.
- Robles Bykbaev, V.; Sanchez Montoya, R.; Ingavelez Guerra, P.; Pesántez Avilés, F. (2017). *Inclusión, Discapacidad y Educación. Enfoque práctico desde las tecnologías emergentes*. Quito, Ecuador: Editorial Universitaria Abya-Yala, Universidad Politécnica Salesiana.
- Wikinclusión (2018). Recuperado de: <http://wikinclusion.org/index.php>
- Sugeys I.Castillo G.; Montoya R.; Yaizet Griffin; Aguirre L; Zenith Hernández (2013). Portal de objetos digitales de aprendizaje para niños y niñas con necesidades educativas especiales. Recuperado de: http://www.ciditic.utp.ac.pa/sites/ciditic.utp.ac.pa/files/pdf/articulo_poda-4utp.pdf
- Vizñay Durán Jenny Karina; Milton Alfredo Campoverde Molina; Poma Japón Diana Ximena (2017). Funciones de accesibilidad que logran y mejoran las competencias de uso y manejo de los teléfonos inteligentes. En Pesántez Avilés, F.; Sánchez, R.; Vladimir Robles Bykbaev; Ingavélez Guerra, P.. *Inclusión, discapacidad y educación. Enfoque práctico desde las Tecnologías Emergentes*. (pp. 97). Quito, Ecuador: Editorial Universitaria Abya-Yala, Universidad Politécnica Salesiana.

Anexo 1. Códigos de Competencias

Los códigos existentes se pueden consultar en la Base de Conocimientos Wikinclusion [15]. A continuación se listan los códigos de competencias agregados:

A - HTML: A001 Crear una Página Web | A002 Estructura de una página | A003 Usar etiqueta DIV | A004 Usar etiqueta de imagen | A005 Usar etiqueta hipervínculos | A006 Crear Formularios Web

B- CSS: B001 Propiedades de las fuentes | B002 Propiedades fondo de página | B003 Hoja de Estilo

C- SERVIDOR LOCAL: C001 Cargar servidor local | C002 Descargar servidor local | C003 Iniciar servidor local | C004 Detener servidor local | C005 Iniciar PHP MyAdmin | C006 Detener PHP MyAdmin

D- PROGRAMACIÓN PHP: D001 Uso de variables y tipos de datos | D002 Uso de Constantes | D003 Uso de Operadores Aritméticos | D004 Uso de Operadores de Comparación | D005 Uso de Operadores Lógicos | D006 Toma de decisiones condicionales | D007 Decisiones múltiples – switch | D008 Iteraciones – Bucle FOR | D009 Iteraciones – Bucle While | D010 Manejar instrucciones PHP para envío de mail

E- PROGRAMACIÓN PHP BASE DATOS: E001 Recorrer una base de datos | E002 Consultar una base de datos | E003 Crear Informe Web | E004 Dar de alta un registro | E005 Modificar un registro | E006 Eliminar un registro

Formación docente en Ciencias de la Computación: experiencias de la primera Especialización para el nivel Primario de la Provincia de Santa Fe.

Ana Casali¹, Dante Zanarini¹,
Natalia Monjelat², Patricia San Martín²

¹ Facultad de Ciencias Exactas, Ingeniería y Agrimensura - UNR y
Centro Internacional de Ciencias de la Información y de Sistemas (CIFASIS),

² Instituto Rosario de Investigaciones en Ciencias de la Educación (IRICE. CONICET_UNR)

{acasali, dante}@fceia.unr.edu.ar
{monjelat, sanmartin}@irice-conicet.gov.ar

Abstract. En este trabajo se presenta el diseño y desarrollo de la primera especialización para el nivel primario en Ciencias de la Computación aprobada por el Ministerio de Educación de la provincia de Santa Fe. La “Especialización Docente de Nivel Superior en Didáctica de las Ciencias de la Computación” plantea la formación de docentes del nivel primario en el Pensamiento Computacional y la Programación. El objetivo de la misma es que los cursantes puedan apropiarse de esta forma de pensamiento y del potencial de la programación para incluirlas en prácticas docentes situadas, interdisciplinarias y no excluyentes. Luego de la implementación de la primera cohorte, se presenta el diseño de la misma así como resultados preliminares que permiten dar cuenta de las opiniones de los participantes en relación al dictado. De esta forma se aportan datos que posibilitan revisar futuros diseños e implementaciones en un campo innovador de la formación docente.

Keywords: formación docente, educación primaria, ciencias de la computación, programación, pensamiento computacional

1 Introducción

En la última década, la introducción de nociones relacionadas con las Ciencias de la Computación (CC) en el currículum escolar, especialmente en cuanto al desarrollo del pensamiento computacional (PC) y la apropiación de conceptos de programación ha adquirido relevancia en el campo educativo internacional [1, 2, 3, 4, 5]. Siguiendo a Wing [6] el aprendizaje de CC beneficia al conjunto de la sociedad, por lo cual es recomendable su enseñanza en todos los niveles educativos. Argentina lanzó en el año 2013, el proyecto Program.ar¹ que a partir de múltiples iniciativas, acerca a niños y jóvenes al aprendizaje de las CC. Asimismo, el Consejo Federal de Educación señaló al aprendizaje de la programación de importancia estratégica para el Sistema Educativo Nacional durante la escolaridad obligatoria (Resolución CFE N° 263/15, 2015). Como un avance en esta dirección, en setiembre de 2018, en dicho Consejo se

¹ <http://program.ar/de-que-se-trata/>

aprobaron los “Núcleos de Aprendizajes Prioritarios de Educación Digital, Programación y Robótica” (NAP EDPR) para los diferentes niveles de la educación obligatoria (Resolución CFE N° 343/18, 2018). En el Anexo II de la citada resolución se establece un plazo de dos años para la adecuación curricular donde se incluyan los contenidos fijados en estos núcleos, el desarrollo de un plan de formación docente continuo orientado a la sensibilización, difusión e integración de los NAP EDPR, y la integración de estos contenidos a la formación docente inicial. En este sentido, desde el marco institucional hay dos aspectos clave e interrelacionados que condicionan fuertemente la introducción de nociones de las CC en el sistema educativo argentino. Por un lado, la necesidad de realizar modificaciones en los planes de estudio a nivel primario, secundario y en la formación docente, tarea que se está comenzando a llevar a cabo en las diferentes provincias, con distinto nivel de desarrollo. Por otro, implica la inmediata puesta en obra de propuestas de formación docente en el aprendizaje y enseñanza del pensamiento computacional y la programación, que habiliten paulatinas transformaciones tanto de lo curricular como de la práctica educativa escolar.

Particularmente el contexto de la provincia de Santa Fe, se observa que casi todas las escuelas públicas e institutos de formación docente fueron dotados de infraestructura tecnológica digital a través de programas nacionales y provinciales. En referencia a la formación y capacitación docente, la misma se ha enfocado principalmente hacia un primer nivel de apropiación instrumental de aplicaciones, herramientas y recursos digitales de las actuales Tecnologías de la Información y Comunicación (TIC) vinculadas a la práctica educativa [7].

Siguiendo estos lineamientos, en el marco de un Convenio de Cooperación Académica suscripto entre la Fundación Sadosky, la Universidad Nacional de Rosario y el Ministerio de Educación de la provincia de Santa Fe se diseñó la “Especialización Docente de Nivel Superior en Didáctica de las Ciencias de la Computación: Aprendizaje y Enseñanza del Pensamiento Computacional y la Programación en el Nivel Primario” aprobada por Resolución N° 1565/17 del Ministerio de Educación de la Provincia de Santa Fe. La primera cohorte de esta formación se dictó en las instalaciones del Instituto Superior de Formación Docente N° 36 "Mariano Moreno" de la ciudad de Rosario y se desarrolló en el período comprendido entre agosto de 2017 y julio de 2019 con un carga horaria total de 400 horas reloj. Dada la naturaleza de esta experiencia piloto, única en cuanto a la formación de docentes del nivel primario en CC en la provincia y una de las tres que se encuentran en desarrollo para este nivel educativo en el país, es objeto de esta publicación compartir los puntos relevantes de su diseño y desarrollo, tomando como fuentes datos obtenidos a lo largo de la cursada.

2 Diseño de la Especialización

Este trayecto formativo se propuso como objetivo general formar docentes capaces de experimentar y reflexionar críticamente acerca de los procesos de desarrollo del pensamiento computacional y la programación, a los fines de construir las

competencias adecuadas al nivel primario que posibiliten una práctica educativa innovadora con énfasis en la resolución de problemas mediante la producción colaborativa e interdisciplinaria de Tecnologías para la Inclusión Social (TIS). En referencia a los objetivos específicos, se plantearon los siguientes:

- Promover el estudio de los principios del pensamiento computacional y la programación para su adecuación didáctica al nivel primario de escolaridad.
- Activar un posicionamiento crítico y ético acerca del uso, impacto y potencial de las CC en el contexto socio-cultural actual.
- Brindar herramientas teóricas, metodológicas y técnicas para la selección y aplicación de conceptos del pensamiento computacional y la programación en función del diseño y desarrollo de proyectos educativos enfocados hacia la producción de TIS.
- Desarrollar el interés y compromiso responsable hacia la participación en la producción colaborativa de programas sencillos integrados tanto a temáticas de la educación primaria como a proyectos institucionales utilizando diversas herramientas bajo metodologías de trabajo interdisciplinario.
- Favorecer en el marco del sistema educativo provincial, el desarrollo de procesos institucionales de adecuación curricular y transformación de las prácticas educativas de nivel primario aportando fundamentos teórico-metodológicos y técnicos sobre contenidos relacionados a las CC.

La Especialización se organizó en cuatro semestres (Tabla 1), presentando en cada uno de ellos tres módulos (12 módulos en total), en los que se articulan los contenidos de manera espiralada trabajando en formato de Taller. La modalidad de dictado fue presencial (80% de la carga horaria y 20% virtual). Se diseñó sobre los ejes de la enseñanza del Pensamiento Computacional (PC) de forma desconectada (enfoque *Unplugged*) y mediante el trabajo en programación usando distintas herramientas y plataformas (enfoque *Plugged*). En esta dirección, en cinco módulos se trabajó en forma totalmente *unplugged*, mientras que otros cuatro módulos se centraron en el mencionado enfoque *plugged*. El diseño curricular se completó con tres Proyectos Integradores, incluyendo al Proyecto final. En estos proyectos los cursantes de forma gradual diseñan un proyecto auténtico acorde al contexto de su trabajo docente, que integre los conceptos aprendidos, combinando estrategias desconectadas y conectadas.

Cabe señalar que, el trabajo bajo un enfoque *Unplugged*, se ha considerado de suma importancia ya que puede aplicarse en cualquier escuela sin necesidad de equipamiento informático y además, muchas de las actividades son familiares a los docentes, pudiendo visitar aprendizajes previos.

Tabla 1. Módulos y carga horaria del diseño curricular de la Especialización

<i>Semestre</i>	<i>Módulo</i>	<i>Carga Horaria</i>
Sem 1	Introducción: CC en la Escuela Primaria	20
	Pensamiento Computacional I	40
	Taller de Programación I	40
Sem 2	Proyecto Integrador I	20
	Pensamiento Computacional II	40
	Taller de Programación II	40
Sem 3	Proyecto Integrador II	20
	Introducción a la Organización de Computadoras	40
	Taller de Robótica	40
Sem 4	Ciencia de datos, privacidad y seguridad de la información	35
	Desarrollo de aplicaciones móviles	40
	Proyecto Final	25
4 sem	12 Módulos	400

En esta dirección, trabajos realizados en distintos contextos de educación primaria, han demostrado que el desarrollo de procesos de enseñanza y aprendizaje a través de este enfoque, resulta efectivo para el desarrollo de habilidades cognitivas referidas al PC [2,5,8]. La distribución de horas según la preponderancia del enfoque desconectado (*Unplugged*), conectado (*Plugged*) o mixto, en el primer y segundo año, pueden verse en la Figura 1. En el primer año hay una carga algo mayor en el enfoque *unplugged*, mientras que en el segundo año los enfoques están equilibrados, sumando una mayor carga horaria los módulos mixtos.

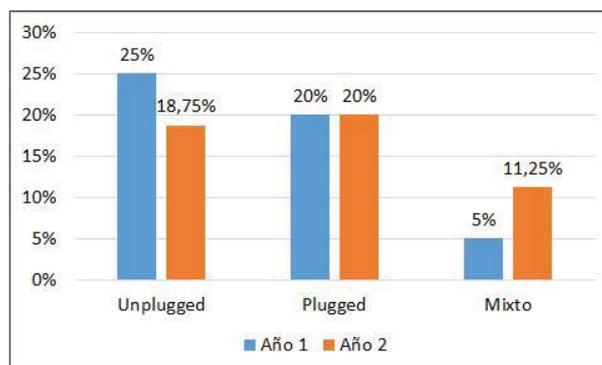


Fig. 1. Distribución de carga horaria según el enfoque predominante de los distintos módulos

Por otra parte, el aprendizaje de nociones de programación en el nivel primario de escolaridad posibilita también el desarrollo del PC, habilitando a los destinatarios ser protagonistas activos y creativos de las CC [9,10]. La recursividad de los procesos del *aprender a programar - programar para aprender* permite que no sólo se desarrollen distintas habilidades sino que se generen nuevas oportunidades de aprendizaje. En la actualidad la comunidad educativa dispone de una variedad importante de herramientas adecuadas para enseñar programación en el nivel primario como Lightbot², Pilas Bloques³ y Scratch⁴, entre otras, las cuales se integran a los módulos taller de programación de la mencionada especialización docente. Además, es necesario brindar en el trayecto de formación docente nociones básicas de organización de las computadoras, sistemas operativos, robótica y programación para dispositivos móviles, con el objetivo de poder utilizar efectivamente distintos soportes y desarrollar así diferentes tareas de programación. También se considera importante incluir contenidos y formación respecto a privacidad y seguridad de la información, con el fin de fomentar en los docentes un análisis crítico del uso de la tecnología. Estos conceptos forman parte de los contenidos recomendados para la enseñanza de las ciencias de la computación [11].

Resulta relevante señalar que considerando las particularidades del contexto y en atención a los lineamientos del Ministerio de Educación de Santa Fe, la especialización se enmarcó pedagógicamente desde una doble articulación que considera tanto los procesos como los productos implicados en el desarrollo del PC y la programación, como posibles Tecnologías para la Inclusión Social [12, 13]. Con ello se pretende que tanto los procesos implicados en las prácticas docentes que incluyan contenidos relacionados con las CC, como los productos generados (programas, videojuegos, historias interactivas, etc.) puedan abordar problemáticas socioeducativas y regionales desde su complejidad, promoviendo un uso crítico de las

² <http://lightbot.com/>

³ <http://pilasbloques.program.ar/>

⁴ <https://scratch.mit.edu/>

herramientas en contextos proyectuales, a partir de poner en obra perspectivas pedagógicas activas, críticas e interdisciplinarias [14,15].

3 Desarrollo de la Formación

A partir de la firma de un convenio tripartito entre la Fundación Sadosky, el Ministerio de Educación de la Provincia de Santa Fe y la Universidad Nacional de Rosario, celebrado en julio de 2017, la Especialización comenzó a dictarse en el ISFD N° 36 de la ciudad de Rosario, en agosto del mismo año, con 78 ingresantes que asistieron al primer encuentro.

Al inicio de la formación, se administró un cuestionario para identificar perfiles y prácticas de los participantes, lo que permitió una mayor adecuación de las propuestas didácticas. Los resultados de este primer cuestionario (n=78) mostraron que del total de participantes, 46% se encuentran en el rango etario 41-50 y un 41% entre 31-40, 85% son mujeres, 38% tiene entre 10 y 20 años de experiencia docente y la mayoría son docentes titulares. El 68% ha realizado trayectos formativos en educación y TICs aunque las experiencias mencionadas son dispares, donde un 45% son autodidactas y un 21% tiene alguna experiencia en programación. Por otra parte, el 90% tiene laboratorio o aula digital (aunque no siempre funcionando) y el 61% reporta mala conexión a internet en su escuela. Destacamos además la heterogeneidad del grupo, dado que los docentes trabajan en distintos niveles y áreas de la educación primaria (Matemática, Informática, Lengua, Cs. Sociales, Naturales, Música, Educación Especial, etc.).

En cuanto a las características del dictado se planteó un abordaje espiralado desde diferentes módulos con propuestas que permitieran abordar los conceptos específicos de CC y PC cada vez con mayor complejidad y profundidad. Para ello, en los diferentes módulos se propusieron actividades grupales a resolver en el espacio de clase presencial, articulando con momentos de reflexión, diálogo y puesta en común, con el objetivo de revisar los contenidos abordados desde revisiones colaborativas.

En la misma línea, los trabajos prácticos realizados dentro de los módulos se abordaron como instancias de conexión con la práctica diaria de los maestros y las maestras, vinculando contenidos curriculares propios del nivel primario y su contextualización en la práctica situada de cada docente. Para la evaluación de los trabajos prácticos se emplearon criterios cualitativos, aportando reflexiones motivadoras hacia la revisión de lo realizado desde una perspectiva constructivista. En los diferentes módulos se ofrecieron múltiples herramientas, opciones de abordaje y materiales, que permitieran a los participantes desplegar un amplio abanico de posibilidades de acción a la hora de desarrollar sus propias prácticas.

4 Resultados

En junio del corriente año se finalizó el dictado de la Especialización, actualmente los y las cursantes se encuentran elaborando el Informe del Proyecto Final, dentro del cual se incluye la observación en aula de una instancia del mismo. En el cursado del último semestre contamos con 31 maestros. Cabe destacar que además de la formación de este grupo, hubo instancias de socialización de esta formación en distintas escuelas donde trabajan y un taller para sus directivos. Lo cual representan acciones multiplicadoras a partir de estos docentes formados.

Para conocer las opiniones y percepciones del profesorado participante de este trayecto formativo novedoso, se administraron encuestas al finalizar cada semestre de cursado, donde se solicitaba a los participantes que dieran sus impresiones en relación con cada uno de los módulos. Un reporte de las opiniones relevantes encontradas en el primer año pueden verse en [16]. Considerando la organización de la propuesta formativa en contenidos conectados y desconectados, se realizó un análisis de las opiniones de los docentes en relación a la didáctica de los módulos más representativos de cada enfoque (n=24), presentes en el primer año de cursado (desconectados: Pensamiento Computacional (PC) 1 y 2; conectados: Taller de Programación (Prog) 1 y 2). Como muestra la Figura 2, las opiniones favorables son mayoría y aumentaron en el segundo año. Esto puede deberse a modificaciones introducidas en el dictado, ya que en el primer semestre se trabajaron en paralelo los módulos de PC y de Programación, pero los resultados de las encuestas administradas indicaron que esta metodología resultaba compleja para los cursantes, por lo que se optó por continuar con un esquema más secuencial, dejando en paralelo el recorrido por los proyectos integradores.

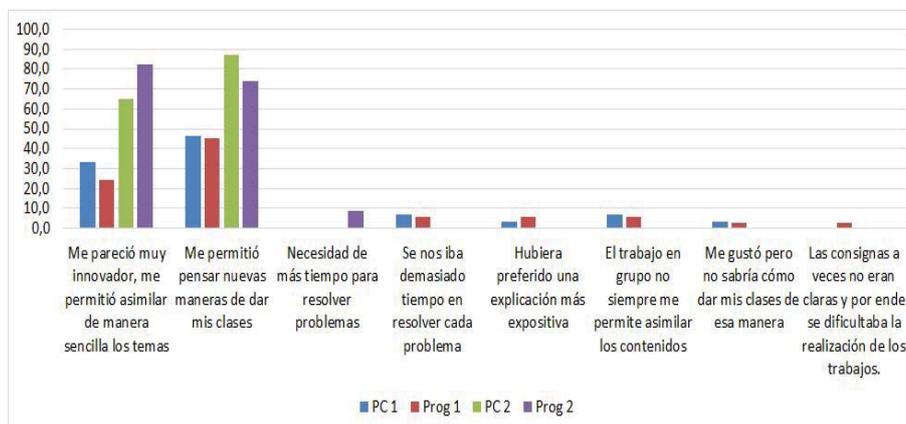


Fig. 2. Propuesta didáctica: opinión docente

Por otro lado y atendiendo también al binomio conectado-desconectado, se analizaron las opiniones de los docentes en relación a los contenidos abordados durante el primer

año. Como se observa en la Figura 3, las valoraciones son positivas y las negativas disminuyeron en el segundo cuatrimestre.

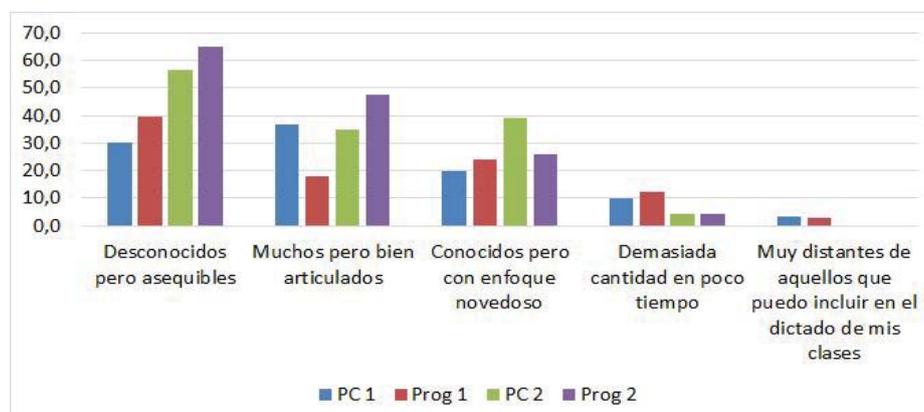


Fig. 3. Contenidos: opinión docente

Finalizado el dictado, se realizó una encuesta integradora sobre la percepción de los y las cursantes sobre toda la especialización. Si bien la misma se encuentra en la etapa de procesamiento, se presentan a continuación algunos resultados preliminares. Todos los encuestados ($n=18$) manifestaron que *se sienten preparados para incluir los contenidos vistos en la especialización en el espacio de enseñanza* que tienen a su cargo. Al ser consultados por qué se sentían preparados, podemos destacar las siguientes respuestas: “La especialización me brindó herramientas para poder desempeñarme correctamente”, “Se ha recibido una formación muy completa desarrollando las competencias necesarias”, “Porque me brindaron las herramientas necesarias para que pueda relacionar los diversos contenidos de escuela primaria con el pensamiento computacional y la programación” y “Considero que los contenidos son totalmente adaptables a las áreas curriculares tradicionales”.

En cuanto a cómo se sienten preparados para implementar los contenidos de la especialización en el aula, la Figura 4 muestra que la mayoría podrían realizar desde objetos de aprendizaje digital para usar en otros espacios curriculares (94%) hasta tomar a cargo una asignatura de programación/informática (72%). Asimismo, el 89% manifestó que ya realizó experiencias de aplicación en el aula de los contenidos trabajados en la especialización. La Figura 5 presenta qué tipo de actividad fue incorporada en el aula. Se observa que la mayoría (56%) realizó tanto actividades conectadas como desconectadas, mientras que un 38% implementó actividades desconectadas y sólo un 6% actividades exclusivamente conectadas. Predomina entonces el desarrollo de actividades que combinen los dos enfoques ejes del trayecto formativo, por encima de la preferencia de uno sobre el otro.



Fig. 4. Percepción de los cursantes sobre el tipo de actividades a implementar

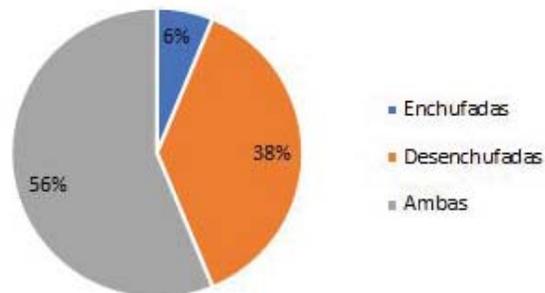


Fig. 5. Tipo de actividad implementada por los y las cursantes

5 Conclusiones

Se ha logrado diseñar e implementar una formación superior para docentes del nivel primario en Ciencias de la Computación, la primera en la provincia de Santa Fe y una de las tres primeras en el país, ya que se encuentran en curso formaciones similares en Río Cuarto, provincia de Córdoba y en Tandil, provincia de Bs As. Los resultados son sumamente alentadores, tanto desde la apropiación de los maestros de los conocimientos de CC, como desde su percepción de estar capacitados para implementar los contenidos del PC y la Programación en el aula. Las observaciones realizadas hasta el momento, de instancias áulicas del Proyecto Final, también han sido evaluadas positivamente. Además, en este proceso se ha conformado un equipo docente interdisciplinario, competente en la formación docente en CC para el nivel primario y contextualizado a la realidad regional. Lo cual, hará posible replicar y expandir esta experiencia en la provincia. Se espera que la presente especialización se

constituya como una propuesta de referencia para el campo de la enseñanza de las ciencias de la computación en Argentina, cuyo carácter innovador posibilite un trayecto transformador de las currículas y prácticas docentes actuales [12,13 y 16].

Agradecimientos. A la Fundación Sadosky, a la Dirección General de Educación Superior del Ministerio de Educación, a todo el equipo docente de la Especialización y a los maestros que con gran esfuerzo fueron los protagonistas de esta formación.

Referencias

1. A. Balanskat and K. Engelhardt, "Computing our future. Computer programming and coding. Priorities, school curricula and initiatives across Europe," Brussels, Belgium, 2015.
2. C. Brackmann, M. Román-González, J. Moreno-León, G. Robles, A. Casali & D. Barone, Computational Thinking Unplugged: Teaching and Student Evaluation in Primary Schools. In Proceedings WIPSCENijmegen, The Netherlands, ACM. November, 2017.
3. P. Hubwieser, M. Armoni, M. Giannakis, and R. T. Mittermeir, "Perspectives and Visions of Computer Science Education in Primary and Secondary (K-12) Schools," *Trans. Comput. Educ.*, vol. 14, no. 7, 2014.
4. S. Peyton Jones, *Computing at School: International comparisons*, Microsoft Research, 2011.
5. A. Yadav, H. Hong, and C. Stephenson, "Computational Thinking for All: Pedagogical Approaches to Embedding 21st Century Problem Solving in K-12 Classrooms," *TechTrends*, vol. 60, no. 6, pp. 565–568, 2016.
6. J. M. Wing, *Computational Thinking Benefits Society*. Social Issues in Computing, 2014.
7. J. C. Tedesco, C. Steinberg, and A. Tófaló, *La integración de TIC en la educación básica en Argentina*. Buenos Aires, Argentina: Unicef, 2015.
8. T. Bell, I. H. Witten & M. Fellows. *CS Unplugged: An enrichment and extension programme for primary-aged students*. 2015.
9. S. Y. Lye and J. H. L. Koh, "Review on teaching and learning of computational thinking through programming: What is next for K-12?," *Comput. Human Behav.*, vol. 41, pp. 51–61 (2014).
10. S. Grover and R. Pea, "Computational Thinking in K–12: A Review of the State of the Field," *Educ. Res.*, vol. 42, no. 1, pp. 38–43 (2013).
11. K-12 Computer Science Framework Steering Committee. *K-12 Computer Science Framework*. Technical Report. ACM, New York, NY, USA, 2016.
12. N. Monjelat y P. San Martín, "Programar con Scratch en contextos educativos: ¿Asimilar directrices o co-construir Tecnologías para la Inclusión Social?," *Prax. Educ.*, vol. 20, no. 1, pp. 61–71, 2016.
13. H. Thomas, P. Juárez y F. Picabea, *¿Qué son las tecnologías para la inclusión social? 1° Edición*. Bernal: Universidad de Quilmes, 2015.
14. N. Monjelat, "Programming Technologies for Social Inclusion With Scratch: Computational Practices in a Teacher's Professional Development Course," *Educare*, vol. 23, no. 3, pp. 1–25, 2019.
15. San Martín, P. S. *Abrir la memoria habitando Creativa Monumento*. Propuesta educativa, vol. 2, n° 50, pp. 92-104 (2018).
16. Casali, D. Zanarini, N. Monjelat & P. San Martín, *Teaching and Learning Computer Science for Primary School Teachers: an Argentine Experience*, LACLO 2018, San Pablo, Brasil, IEEE Xplore (2018).

Habilidades de Pensamiento Computacional en docentes de primaria: evaluación usando Bebras

Francisco Bavera¹, Marcela Daniele¹, Teresa Quintero², Flavia Buffarini³

¹Departamento de Computación, ²Departamento de Física, ³Departamento de Matemática
Facultad de Ciencias Exactas, Físico-Químicas y Naturales
Universidad Nacional de Río Cuarto
{pancho,marcela}@dc.exa.unrc.edu.ar, {tquintero,fbuffarini}@exa.unrc.edu.ar

Abstract. Se presenta un análisis de las habilidades de Pensamiento Computacional que poseen docentes de nivel primario, que en la actualidad cursan la *Especialización Docente de Nivel Superior en Didáctica en Ciencias de la Computación*, en la ciudad de Río Cuarto. Como herramienta de recolección de datos se utilizaron los Problemas Bebras para medir habilidades de Pensamiento Computacional que poseen estos docentes, detectando que un alto porcentaje, pudo resolver problemas de complejidad media que implican habilidades, tales como, abstracción, reconocimiento de patrones, modelos y simulación, algoritmos y descomposición.

Keywords: Pensamiento Computacional, Habilidades, Evaluación, Formación Docente Continua.

1 Introducción

La construcción del Pensamiento Computacional (PC) se manifiesta como un importante desafío para los formadores actuales de todos los niveles educativos. Además, se sabe que el desarrollo de ciertas habilidades, tanto cognitivas como actitudinales, son claves en la primera infancia. En la actualidad, el uso de las tecnologías constituye un importante elemento que ayuda a plasmar la solución de diferentes tipos de problemas en los más diversos ámbitos y contextos. Construir Pensamiento Computacional permite aprender y comprender el desarrollo y diseño de las tecnologías, incluyendo estrategias de descomposición para la resolución de problemas, abstracción, generalización de las soluciones encontradas, la aplicación de procesos que definen pasos racionales y sistémicos hacia la búsqueda de soluciones, logrando la definición y el diseño de algoritmos que permiten encontrar las soluciones o resultados deseados [1,2,3]. La construcción del pensamiento computacional posibilita el desarrollo de ciertas habilidades de orden superior que aportan y enriquecen la manera de comprender y resolver problemas, y ser capaces de *crear* y *hacer* la automatización de dichas soluciones, a partir del conocimiento de principios y conceptos fundamentales de las Ciencias de la Computación, como lógica, abstracción, representación de datos y diseño de algoritmos.

Caracterizar la construcción del Pensamiento Computacional, en el marco de la formación docente continua y el impacto sobre sus propias prácticas, es el principal objetivo del proyecto de investigación y desarrollo¹ que da marco a este estudio. Entre los aspectos que se abordan, se encuentra el análisis de las prácticas de enseñanza de docentes de nivel primario que participan en procesos de formación en didáctica de las Ciencias de la Computación y el desarrollo del PC. Se vislumbra la necesidad de profundizar y consolidar investigaciones que aborden al estudio del impacto de la construcción del PC en los formadores, para lo que es necesario evaluar, por un lado, las actividades de formación propuestas, y por otro lado, las habilidades construidas por los educadores que transitan una formación continua en didáctica de las Ciencias de la Computación. En función de la complejidad, se hace ineludible el uso de diferentes enfoques, métodos y técnicas que permitan obtener evidencia e indicadores para analizar y evaluar el impacto de estas formaciones.

Los autores de este trabajo, forman parte del equipo que coordina y dicta una carrera de formación dirigida a profesores de educación primaria, denominada *Especialización Docente de Nivel Superior en Didáctica de las Ciencias de la Computación*. Dicha carrera surge como una iniciativa conjunta entre la Fundación Dr. Manuel Sadosky, el Instituto Superior de Formación Docente Ramón Menéndez Pidal, y el Departamento de Computación de la Facultad de Ciencias Exactas Físico-Químicas y Naturales, de la Universidad Nacional de Río Cuarto. En este marco, se llevan a cabo diferentes estudios con el objeto de acercarse a la complejidad inherente del proceso de construcción del PC, en este espacio de formación continua de docentes.

En una investigación inicial, se abordó el estudio de las producciones de integración, presentadas en formato póster, por los cursantes de la especialidad como actividad de cierre del primer año. Se analizó cada producción usando una metodología cualitativa, basada en la teoría fundamentada, los datos fueron triangulados entre sí por al menos dos investigadores considerando las categorías de la Taxonomía de Bloom adecuada por Churches [4]. Este análisis permitió determinar que todas las producciones y las propuestas de prácticas áulicas, reflejaron ciertas habilidades cognitivas relacionadas al mundo digital, mostrando diferentes niveles de desarrollo y profundidad [5].

Posteriormente, se realizó un análisis de las habilidades de PC, tales como, formular problemas donde se usen computadoras y otras herramientas para solucionarlos, organizar datos de manera lógica y analizarlos, representar datos mediante abstracciones, las que son mayormente abordadas en los módulos específicos de enseñanza de la programación. Este análisis brindó importantes evidencias de que la formación recibida por estos educadores, impacta positivamente en la transformación de sus prácticas áulicas [6].

La evaluación del pensamiento computacional está en desarrollo, por lo que los educadores encuentran considerables limitaciones, al momento de evaluar estas habilidades que son consideradas centrales para la formación de los estudiantes del

¹ **La construcción del pensamiento computacional: estudio del impacto desde la formación de formadores.** Proyecto de Investigación y Desarrollo (PID). financiado por el Ministerio de Ciencia y Tecnología de la Provincia de Córdoba. 2019-2020.

mundo actual [8]. En este contexto, se encuentra que los Problemas de Bebras [14] que están especialmente diseñados para evaluar habilidades de pensamiento computacional, resultan un instrumento adecuado para medir estas habilidades antes y después de una capacitación. Bebras es una competencia internacional que tiene como objetivo promover el pensamiento computacional, entre estudiantes de todas las edades, con actividades de tipo “desconectadas”. Se han realizado estudios que analizan la efectividad de los problemas de Bebras y también comparaciones con otras pruebas de pensamiento computacional [8, 9, 12, 13, 10, 11]. No obstante, no se encuentran registros de resultados obtenidos a partir de la utilización de Problemas Bebras en la evaluación de la formación docente continua en PC, por lo que resulta interesante indagar sobre los alcances de esta alternativa de evaluación.

En este trabajo se muestran los resultados obtenidos del análisis de una experiencia, basada en actividades seleccionadas de problemas de Bebras, donde se evalúan las habilidades de PC de los docentes de educación primaria que cursan la especialidad en didáctica de las ciencias de la computación, aportando otra forma de evaluación de las mismas.

Este trabajo está organizado de la siguiente forma: la sección 2 expone la experiencia de evaluación realizada a partir de problemas Bebras, se describe la población de individuos que se evalúan y el test realizado. En la sección 3 se presenta un análisis de los resultados obtenidos, y por último se exponen algunas conclusiones y los posibles trabajos futuros que dan continuidad a esta investigación.

2 La Experiencia

2.1 Población

En esta experiencia se realiza una evaluación de las habilidades relacionadas con el PC, que presentan los docentes de escuelas primarias que se encuentran cursando la *Especialización Docente de Nivel Superior en Didáctica de las Ciencias de la Computación*.

La primera cohorte comenzó en marzo de 2018 y cuenta con 47 docentes (44 maestras y 3 maestros) cursando el segundo año. Son todos docentes de nivel primario, el 98% maestras/os de grado y sólo una maestra del área Informática. La *media* de edad es 42 años, en un rango que va desde los 23 a los 54 años. Tenemos una *mediana* y una *moda* de 43 y 44 años, respectivamente. La *desviación media* es de 4,5. El 50% son docentes titulares y los restantes son interinos y suplentes, que se desempeñan en los diferentes grados de la escuela primaria. Además, actualmente dos docentes no están relacionados con institución alguna y un docente desempeña actividades en forma ad-honorem. Los docentes pertenecen a 28 escuelas, donde el 80% son instituciones públicas de gestión estatal y el restante 20% instituciones públicas de gestión privada. Un 75% de estas escuelas están localizadas en la ciudad de Río Cuarto, y el restante 25% en localidades de su región de influencia (en un radio de 30 a 150 km de distancia de Río Cuarto).

La Especialización tiene una duración de 400 horas, divididas en 8 módulos, a lo largo de dos años. Los distintos módulos introducen a los docentes en conceptos de ciencias de la computación, ciudadanía digital, pensamiento computacional, programación y robótica. A la fecha se dictaron 5 módulos: (1) Herramientas de comunicación y colaboración, (2) Introducción a la Resolución de problemas, (3) Introducción a los Lenguajes de Programación, (4) Administración y Configuración de Software y Hardware, y (5) La Programación como Recurso Educativo. La evaluación de cada módulo estuvo centrada en las actividades de práctica áulica, generadas e implementadas en forma grupal por los docentes, a modo de actividad integradora en cada uno de ellos.

2.2 Test

Este trabajo presenta un estudio en el que se utilizan técnicas cuantitativas de recolección de datos, donde los efectos en el aprendizaje que se van a observar y analizar se estudian una vez que ya han ocurrido [7]. El estudio realizado utiliza un *post-test* sin grupo de control, lo que limita la capacidad de analizar cambios producidos en las habilidades de los docentes a partir de la formación recibida, aunque posibilita analizar las habilidades del Pensamiento Computacional de los docentes de primaria.

El test utilizado está basado en los *Problemas Bebras*, específicamente se utilizaron problemas del *Bebras Australia Computational Thinking Challenge 2018* [15]. Como se mencionó anteriormente, Bebras es una competencia internacional que tiene como objetivo promover el pensamiento computacional, entre estudiantes de todas las edades, con actividades de tipo “desconectadas”. La iniciativa Bebras es parte de una propuesta internacional, abierta y basada en la colaboración entre docentes. El proyecto posee una importante popularidad, ya que, actualmente participan más de 50 países. Los participantes generalmente son supervisados por maestros y el desafío se realiza en las escuelas usando computadoras o dispositivos móviles. Los problemas o desafíos no requieren conocimientos previos de programación o de Ciencias de la Computación, pero todos abordan estas temáticas, por ejemplo patrones, codificación, criptografía, árboles, entre otros.

Para construir el instrumento de evaluación, se seleccionaron y tradujeron al español, ocho problemas de complejidad media del *Bebras Australia Computational Thinking Challenge 2018*. Los problemas seleccionados plantean actividades que incluyen encontrar caminos -un camino y el mejor camino-, reconocer patrones, identificar y aplicar algoritmos, ordenar secuencias y utilizar razonamiento lógico. En el siguiente vínculo se encuentra disponible el test completo para su consulta, <https://drive.google.com/file/d/1uP66o5LCUdEnekITGFDDkcb-E0I5G7ad/view>. En la Tabla 1 se listan los problemas seleccionados y las habilidades del PC involucradas en los mismos, las cuales son:

- Descomposición: dividir los problemas en partes (sub-problemas).
- Reconocimiento de patrones: analizar los datos y buscar patrones para que cobren sentido dichos datos.

- Abstracción: eliminar detalles innecesarios y concentrar la atención en los datos importantes.
- Modelado y simulación: crear modelos o simulaciones para representar procesos.
- Algoritmos: crear una serie de pasos ordenados para resolver un problema.

Tabla 1. Problemas seleccionados y habilidades PC asociadas

Habilidad PC	Descomposición	Reconocimiento Patrones	Abstracción	Modelado y Simulación	Algoritmos
Problema					
1 Camino a Casa	x	x	x		x
2 El Bailarín	x	x	x		x
3 Torneo de Castores	x		x		x
4 Espadas y Escudos	x		x	x	x
5 Ciudad Redonda	x		x	x	x
6 Pulseras	x	x	x		x
7 Ferrocarril	x		x	x	x
8 Caramelos	x		x		x

Las actividades en formato papel fueron entregadas a los docentes en el aula, para ser resueltas en 40 minutos. Se utilizó esta modalidad, a diferencia de los Bebras Contest que se realizan frente a computadoras.

Es importante resaltar que los docentes se enfrentaron por primera vez a un desafío con problemas en este formato. Es decir, las actividades de los distintos módulos de la especialización no utilizaron *Problemas Bebras*, pero si se desarrollaron otros tipos de actividades que implican las habilidades incluidas en los problemas seleccionados.

3 Resultados

Se encuentra que los docentes respondieron correctamente un 73,5% del total de los problemas del test. Con una *media* de 6 respuestas correctas por individuo. Cabe destacar que la *mediana* y la *moda* coinciden con el valor de la *media*. Aproximadamente, un tercio de los docentes está por encima de la media y una cantidad similar por debajo de la misma. En la Tabla 2 se puede apreciar el porcentaje de respuestas correctas discriminadas por problema.

Tabla 2. *Porcentaje Respuestas Correctas*

Problema	Correctas
1 - Camino a Casa	100,0%
2 - El Bailarín	29,0%
3 - Torneo de Castores	96,0%
4 - Espadas y Escudos	58,0%
5 - Ciudad Redonda	79,0%
6 - Pulseras	66,5%
7 - Ferrocarril	96,0%
8 - Caramelos	62,5%

Los mejores resultados se observaron en los problemas asociados a *encontrar un camino*. Aunque, se observa una leve disminución en la efectividad de resolución, en las actividades donde se debía encontrar *el mejor camino*. Es necesario indagar en las causas que motivan esta diferencia, que pueden ser numerosas y diversas, pero una de las posibilidades es que la mayoría de las respuestas incorrectas pueden deberse a una interpretación errónea de las consignas, a no identificar un dato relevante en el enunciado, o a dificultades relacionadas con las habilidades de abstracción.

Una posible explicación respecto de los resultados obtenidos en la resolución de los problemas 2 (El bailarín) y 4 (Espadas y Escudos), podría ser consecuencia de la realización de la actividad en papel. Esto generó una dificultad extra para los docentes evaluados en la resolución de este tipo de ejercicios *interactivos*, que no se vió en los desafíos de múltiple opción.

Se encontraron dos casos que presentan gran cantidad de respuestas incorrectas. No pudiendo, con la información disponible, dar cuenta de los factores que llevan a estos resultados. Considerando que las causas pueden ser múltiples, tales como, interpretación errónea de los enunciados, falta de comprensión de las consignas, o hasta desgano o falta de interés. Es necesario indagar con mayor profundidad para poder extraer conclusiones fundamentadas sobre estos casos.

Los resultados del test, mostrados en Tabla 3, arrojaron que los docentes, en promedio, resolvieron correctamente 6 de los 8 problemas que involucran habilidades relacionadas con *descomposición*, *abstracción* y *algoritmos*. En estas habilidades la *mediana* y la *moda* arrojan el mismo valor (6 respuestas correctas de un máximo posible de 8). Tomando el total de respuestas asociadas a estas tres habilidades, los docentes resolvieron correctamente el 73,5% de los problemas.

En cuanto a *abstracción*, se puede apreciar que la *media*, *mediana* y *moda* arrojan igual resultado: 2 respuestas correctas (de 3 posibles como máximo). Con un 65% de respuestas correctas en total. En el caso de *modelado* y *simulación*, se puede observar que los docentes respondieron en promedio 2 preguntas correctamente (de 3 posibles), pero la *media* y la *moda* es 3 (de 3 como máximo); y del total posible de respuestas respondieron correctamente el 73,5%

Tabla 3. *Respuestas Correctas por Habilidad del PC*

Respuestas correctas	Porcentaje Respuestas Correctas	Media (respuestas correctas p/docente)	Mediana (respuestas correctas p/docente)	Moda (respuestas correctas p/docente)
Habilidad PC				
Descomposición	73,5%	6	6	6
Reconocimiento Patrones	65%	2	2	2
Abstracción	73,5%	6	6	6
Modelado y Simulación	78%	2	3	3
Algoritmos	73,5%	6	6	6

4 Conclusiones y Trabajos Futuros

A partir del análisis de de las habilidades de Pensamiento Computacional en los docentes de primaria que actualmente cursan la *Especialización Docente de Nivel Superior en Didáctica de las Ciencias de la Computación*, se detecta que un alto porcentaje de los docentes, que realizaron las actividades de Bebras seleccionadas para el test, pudieron resolver problemas de complejidad media que implican el uso de habilidades del PC, tales como, abstracción, reconocimiento de patrones, modelos y simulación, algoritmos y descomposición.

Es necesario complementar estos resultados cuantitativos con otros métodos de tipo cualitativo para obtener una visión más amplia. En particular, se realizarán entrevistas a los docentes sobre los procesos involucrados en la resolución de los problemas. Como así también, se realizarán nuevas evaluaciones usando Problemas de Bebras de mayor complejidad, complementando y profundizando el seguimiento de las habilidades del PC a lo largo del dictado de la especialidad.

Los resultados de estas investigaciones intentan aportar al desarrollo de conocimientos sobre la formación continua de docentes de educación primara en las ciencias de la computación. Por otra parte, representan un avance de la investigación que se está desarrollando y se complementa con estudios de corte cualitativo, que incluye entrevistas a los docentes participantes, que se están llevando a cabo.

Asimismo, este trabajo aporta datos y conocimientos para el debate que se está dando en la actualidad en nuestro país, y en particular en nuestra provincia, en cuanto a la adecuación de las currículas para la inclusión de las ciencias de la computación y la programación en la escuela obligatoria.

Se pretende continuar evaluando las habilidades de Pensamiento Computacional, que poseen los docentes participantes en la especialización, con otras metodologías.

También se está trabajando en la evaluación de la inclusión del PC en las prácticas docentes y como las formaciones en el tema impactan en las mismas.

Por otra parte, se planifica incursionar en el análisis de las percepciones de los docentes de primaria con respecto a las Ciencias de la Computación, la programación y la robótica. Este análisis también pretende vislumbrar el aporte, en cuanto al cambio de las percepciones de los docentes, de distintas formaciones en Ciencias de la Computación, Pensamiento Computacional, Programación y Robótica que se están llevando a cabo.

Bibliografía

1. Aho, A. V. (2012). Computation and Computational Thinking. *The Computer Journal*, 55(7), 832-835. doi: <https://doi.org/10.1093/comjnl/bxs074>
2. Denning, P. J. (2017). Remaining trouble spots with computational thinking. *Communications of the ACM*, 60(6), 33-39. doi: <https://doi.org/10.1145/2998438>
3. Wing, J. M. (2006). Computational thinking. *Communications of the ACM*, 49(3), 33-35. doi: <https://doi.org/10.1145/1118178.1118215>
4. Churches, A. (2009) Taxonomía de Bloom para la era digital. *EduTEKA*. Consultado 10/01/19. Disponible en: <http://eduteka.icesi.edu.co/articulos/TaxonomiaBloomDigital>
5. M. Daniele, T. Quintero, F. Bavera, F. Buffarini, D. Solivellas, C. De Dominici (2019). Análisis de producciones de docentes de educación primaria con formación en didáctica de las ciencias de la computación. Aprobado y presentado en las Segundas Jornadas de Didáctica de la Programación, FAMAF, Universidad Nacional de Córdoba, 7 y 8 junio 2019.
6. F. Bavera, T. Quintero, M. Daniele, F. Buffarini (2019). Análisis de prácticas de docentes de educación primaria en el marco de una formación en pensamiento computacional. Aprobado para las 48° Jornadas Argentinas de Informática, SAEI-JAIIIO, Universidad Nacional de Salta, 16 al 20 de septiembre 2019.
7. Bernado, J. y Calderero, J.F. (2000). *Aprendo a investigar en educación*. Ediciones Rialp, Madrid, España.
8. Lockwood, James & Mooney, Aidan. (2018). Developing a Computational Thinking Test using Bebras problems. *Proceedings of the CC-TEL 2018 and TACKLE 2018*.
9. L. Gouws, K. Bradshaw, and P. Wentworth. (2013). In First year student performance in a test for computational thinking. In *Proceedings of the South African Institute for Computer Scientists and Information Technologists Conference*. ACM.
10. V. Dagiene and G. Stupuriene. (2016). In *Bebras-a sustainable community building model for the concept based learning of informatics and computational thinking*. *Informatics in education*.
11. P. Hubwieser and A. Mhling. (2014). In *Playing PISA with bebras*. 9th workshop in primary and secondary computing education (wipsce).
12. P. Hubwieser and A. Mhling. (2015). In *Investigating the psychometric structure of Bebras contest: towards measuring computational thinking skills*. *Learning and teaching in computing and engineering (latice)*.
13. Jiri Vaicek. (2014). *Bebras informatics contest: criteria for good tasks revised*. In *International Conference on Informatics in Schools: Situation, Evolution, and Perspectives*. Springer.
14. Iniciativa Bebras. <https://www.bebas.org/>
15. Bebras Australia Computational Thinking Challenge 2018. Disponible en <https://www.bebas.edu.au/wp-content/uploads/2019/02/Bebras-2018-Solution-Guide.pdf>

Introducing Agile Methods in Undergraduate Curricula, a Systematic Mapping Study

Nicolas Paez¹, Alejandro Oliveros¹, Diego Fontdevila¹, and Alejandra Zangara²

¹ Department of Science and Technology , Universidad Nacional de Tres de Febrero

² Faculty of Informatics, Universidad Nacional de La Plata

nicopaez@computer.org

{aoliveros,dfontdevila}@untref.edu.ar

azangara@info.unlp.edu.ar

Abstract. Agile approaches to Software Engineering are widely used nowadays in industry and have also reached academic environments, with universities all around the world including agile related content in their programs. There are no formal studies about the current situation of Agile Software Development in Argentinian Universities. A systematic mapping study was conducted to understand the state of agile in undergraduate curricula. Results show that Agile Software Development is part of the Information Technology and Computer Science Programs and that Scrum is the most popular agile method in that context. There is little information regarding the teaching strategy used but a learning-by-doing approach is used in many cases.

Keywords: Agile, Software Engineering, Education

1 Introduction

Agile Methods have their origin in the late 90' but their foundational year was 2001 when the Agile Manifesto [1] was published. Since then the popularity of Agile has been in constant growth. A study published by Forrester [2] in 2010 considered Agile Methods as mainstream. Companies, practitioners, professors and researchers have embrace Agile Software Development. Classical textbooks on Software Engineering like the books by Pressman [3] and Sommerville [4] have included Agile topics several editions ago; and the ACM has included agile in its curricula recommendations for Software Engineering and Information Technology [5].

In Argentina, many software companies have adopted Agile methods too [6], and there are reported experiences of teaching agile in university courses [8, 9]. There is also a proposal by Reingart and Dominguez [10] for a Free Software Diploma Program that explicitly includes Agile Methods. Still, for higher education, the regulations and guidance for the accreditation of Information Technology and Computer Science ³ programs does not mention Agile methods at all [7].

³ In the context of this article the term Information Technology and Computer Science programs is used to refer to the programs covered by Resolution 786/2009 [7]

When talking about Agile and Education we distinguish two clearly different topics: teaching Agile Software Development and using Agile techniques to teach any subject matter even beyond software development. This work is focused on the first topic: teaching Agile Software Development.

There are some studies reporting evolution and/or state of the art in Agile Software Development education in particular regions worldwide which is something similar to our goal. A study similar to ours was carried out in Thailand by Chookittikul et al [11]. The researchers performed interviews in several leading universities in Thailand to understand how agile methods were incorporated into computer science curricula. A broad study on Agile evolution in Brazil was conducted by Melo et. al. [12] which covered industry, education and research.

At the time of this writing, there are no formal studies of the current situation of Agile Software Development teaching in the Computer Science and Technology programs in Argentina. The goal of this study⁴ is to understand the state of Agile Software Development teaching, to identify further research opportunities.

Our research questions are:

- RQ1: Are Agile methods part of the Information Technology and Computer Science programs in Argentina?
- RQ2: What is the Agile method most commonly taught?
- RQ3: Is there any particular teaching strategy used to teach Agile methods?

The rest of this article is organized as follows: Section 2 explains the details of our review method, Section 3 presents our results and findings, Section 4 discusses specific issues of interest in our findings, Section 5 presents the threats to validity and finally Section 6 presents conclusions and future lines of work.

2 Review Method

This work was conducted following the guidelines provided by Kitchenham [18] and Kuhrmann [17] for Systematic Mappings Studies.

2.1 Data Sources and Search Strategy

In Argentina there are seven national conferences in the area of Information Technology and Computer Science:

- CACIC: Congreso Argentino de Ciencias de la Computación.
- JAIIO: Jornadas Argentinas de Informática e Investigación Operativa.
- TEyET: Congreso Nacional de Tecnología en Educación y Educación en Tecnología.
- WICC: Workshop de Investigadores en Ciencias de la Computación.

⁴ This study is part of the final specialization assignment in the context of the Specialization Degree in IT Applied to Education Program of the Faculty of Informatics at Universidad Nacional de La Plata

- CONAIIISI: Congreso Nacional de Ingeniería Informática - Sistemas de Información.
- CACIDI: Congreso Argentino de Ciencias de la Informática y Desarrollos de Investigación.
- ARGECON: Congreso Bienal de IEEE Argentina.

In addition to these conferences there are some relevant Argentinian journals in the same area:

- JCS&T: Journal of Computer Science & Technology
- Revista TEYET: Revista Iberoamericana de Tecnología en Educación y Educación en Tecnología
- EJS: Electronic Journal of SADIO

To search on all these sources (journals and conferences) different search engines were used because there was no unique search engine covering them all.

The search string was composed of two parts connected with AND:

1. agile OR ágil OR Scrum
2. educación OR education OR enseñanza OR teaching

In the first part of the search string we decided to explicitly include Scrum because we consider that it has such great popularity in industry that sometimes practitioners refer to Scrum as a synonym of Agile [13] even when this is not correct because Scrum is just one of the agile methods.

So the final search string was:

(1) AND (2)

A third part might be added to the search string to restrict articles to Argentina as that is the scope of this work, but we assumed that authors might not mention the country where the research work is performed but only their institution. So we decided to not include "Argentina" in the search string.

When trying to use this search string in the different search engines several restrictions were found. Because of that this "conceptual" search string had to be adjusted to the limitation of each search engine which in some cases implied the execution of several partial searches.

To search on the proceedings of CACIC, JAIIO, TEYET and WICC, we used SEDICI[14], the institutional repository of the Universidad Nacional de La Plata that beyond items authored by its members it also stores items generated by other institutions like Red UNCI (the institution that organizes CACIC, TEYET and WICC) and SADIO (the institution that organizes JAIIO). It is important to mention that SEDICI has JAIIO proceedings since 2013, so JAIIO publications in JAIIO proceedings prior to 2013 are not covered in this study. SEDICI does not support connectors in the search string but it offers a feature to filter search results by keywords and subject. So when using SEDICI we performed several partial searches and used the keywords and subject filters.

To search on CACIDI and ARGENCON proceedings IEEE Xplore [16] was used because these two conferences are IEEE Conferences. In these cases the

search string was extended to include the name of the corresponding conference (CACIDI or ARGENCON in each case).

To search for international publications we used Scopus [15] which covers most relevant conferences and journals in the area of education and software engineering like the International Conference in Software Engineering, Computer and Education Journal and Transactions on Education Journal, etc. This particular search engine allowed us to add a restriction to the search string to limit articles by affiliation country, so this way we limited the search to Argentina.

To search on CONAIIISI proceedings a manual procedure was used because these proceedings are not covered by any search engine. At the same time the different editions of this conference have taken different approaches to publishing their proceedings. Editions 2013, 2016 and 2018 published their proceedings in their websites offering a link for each published paper. At the same time editions 2014 and 2017 offered a unique link to a single PDF file containing all the articles. Finally, proceedings of edition 2015 are not available online so they were not covered in this study.

Finally we also performed a manual search on the available issues of the Electronic Journal of SADIO (EJS).

Table 1 summarizes the search engine used to search each source.

Table 1. Search engines and sources.

Search Engine	Source
SEDICI	CACIC, WICC, JAIIO, JJCS&T, TEYET (proceedings and journal)
IEEE Explore	CACIDI, ARGENCON
Scopus	International publications
Manual Search	CONAIIISI, EJS

2.2 Inclusion and exclusion criteria

Studies eligible for inclusion in the review were those that describe or mention the teaching or usage of agile software development within undergraduate programs in the area of Information Technology and Computer Science programs in Argentina.

Studies were excluded if their focus was the use of agile techniques to teach something other than software development. This exclusion criteria is important because there is an increasing trend regarding using agile techniques beyond software development. Other excluded studies were those describing a proposal to teach agile but without an associated validation experience. Finally we also discarded those studies that mentioned the use of agile to develop education software.

2.3 Review Process

The review process was composed of three stages. In the first stage of the process we collected articles by performing searches in the different sources using the search string. We managed the results provided by each search engine in separate spreadsheets, this was because each search engine had different export capabilities. The result of this first stage gave us 582 items.

In the second stage we filtered items by reading the title and abstract of each item, and applying the inclusion and exclusion criteria. We completed this stage with 40 items that we stored in an unified spreadsheet. Having all items in the same spreadsheet would allow us to eliminate duplicates in the next step.

In the third and final stage we reviewed the items in detail by reading the full articles and performing the data extraction. In this third stage we eliminated duplicated items. While working on this stage we found a special kind of publication in JAIIO proceedings called "Oral Communications". This kind of publication was not an original work but a summary of work that had been previously published in another conference. Based on this we considered oral communications as duplicates.

The whole review process is summarized in figure 1.

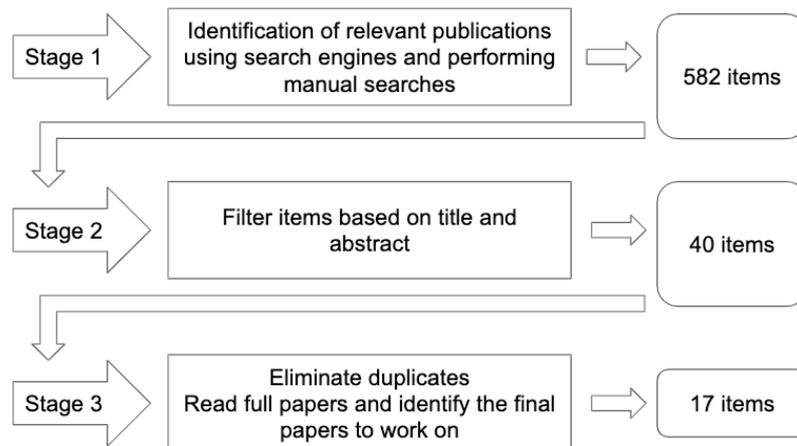


Fig. 1. Stages of the review process

2.4 Quality assessment

Each of the 40 studies that remained after stage 2 was assessed independently by two authors, according to the inclusion/exclusion criteria.

3 Results

We started to work on this study on May 2019 and the different searches where executed on June 2019. We identified 17 articles⁵ that met the inclusion/exclusion criteria about teaching agile software development. These articles are listed on table 3. The selected articles belong to nine different institutions as shown in table 2. Most of these institutions (eight out of nine) are public universities.

Table 2. Institutions with publications about teaching agile.

Institution	Count of publications
UNICEN	8
UNLAM	2
UTN Cordoba	1
U. de la Cuenca del Plata	1
UN de Salta	1
UN del Nordeste	1
UN de La Plata	1
UN de Rio Cuarto	1
UN de Tres de Febrero	1

In addition to selected articles we found another set of articles that did not match the selection criteria for different reasons but we think it is interesting to mention them: a) an article proposing a pre-degree program that included agile methods and b) two articles describing extra-curricular courses about Agile Software Development.

Regarding the specific programs mentioned in the articles we found 14 articles related to Engineering programs, two articles related to Bachelor programs and one article related to a Systems Analyst program.

Regarding the publication year, the first publication dates back to 2012. Table 4 shows the count of publications per year.

Regarding Agile Methods, most of the articles (11) mention Scrum while some others (3) mention Extreme Programming. There is one article that mentions Scrum and Extreme Programming. Two of the articles do not mention any agile method at all. This popularity of Scrum is aligned with what is observed in industry [20].

Most of the papers do not mention any agile practices beyond those that are part of Scrum. It is reasonable to think that those teaching Scrum are also teaching its practices. Beyond Scrum and its practices there are two articles that explicitly mention three practices each. Table 5 shows the details on practices mentioned in articles.

⁵ Full references of these articles are available in the appendix <https://doi.org/10.6084/m9.figshare.9178097.v1>

Table 3. List of selected papers

Id	Title	Source
1	La Evolucion de las Catedras para Influir en los Know-mads y Formar al Trabajador de la Industria 4.0	CONAIIISI 2017
2	Experiencia sobre desarrollo con Genexus siguiendo practicas agiles	TEYET 2016
3	Mejora continua aplicada en la ensenanza de la Ingenieria del Software	TEYET 2015
4	Implementacion practica del agilismo en proyecto de Ingenieria de Software	JAIIO 2016
5	Experiences in software engineering education: Using scrum, agile coaching, and virtual reality	Book Chapter
6	Propuesta para documentar trabajos finales utilizando metodologias agiles	TEYET 2014
7	Are learning styles useful indicators to discover how students use Scrum for the first time?	Computer in Human Behaviour Journal
8	El rol del estilo de aprendizaje en la ense´anza de practicas de Scrum: un enfoque estadistico	JAIIO 2013
9	Measuring the Impact of Agile Coaching on Students' Performance	IEEE Transactions on Education
10	Supporting assessment of practices in software engineering courses	IEEE Latin America Transactions
11	Teaching scrum to software engineering students with virtual reality support	ADNTIIC 2011
12	Towards better Scrum learning using learning styles	ADNTIIC 2011
13	Virtual Scrum: A teaching aid to introduce undergraduate software engineering students to Scrum	Computer Applications in Engineering Education
14	Mejoras en el proceso de ensenanza / aprendizaje de la programacion utilizando metodologias de la industria del software	CONAIIISI 2013
15	Metodologias activas y correccion por rubricas en el proceso de ensenanza / aprendizaje de programacion	CONAIIISI 2014
16	A flipped classroom experience teaching software engineering	SECM 2017
17	Como Pueden Scrum y las TICs Mejorar el Aprendizaje Colaborativo en el ambito Universitario?	CONAIIISI 2017

Regarding the teaching strategy about a third of the articles does not mention anything. Those that make an explicit mention regarding this concern are:

- Article 1 and 15 mention a problem-based learning approach.
- Article 16 mentions a flipped-classroom approach.
- Articles 5, 11 and 13 mention the use of virtual reality (these articles belong to the same group of authors).
- Articles 7, 8 and 12 mention the Felder-Silverman learning styles.
- Article 17 mentions a collaborative learning approach.

In many cases students are required to develop a project in the context of a software engineering course using an agile approach which may represent a "learning by doing" strategy.

Table 4. Publication about teaching agile by year.

Year	Count of publications
2012	1
2013	1
2014	5
2015	3
2016	5
2017	4
2018	1

Table 5. Practices mentioned in articles beyond Scrum.

Practice	Articles id
Self-Organization	2, 9
Definition of Done	2
Automated testing	9
Continuous Integration	9
Release Planning	12
Planning Poker	12
Pair-Programming	14, 15
Test-Driven Development	16
Story Mapping	16
Retrospectives	16

4 Discussion

We found 17 articles on teaching Agile Software Development in the context of Information Technology and Computer Science programs in Argentina. Those

articles belong to nine different institutions but according to Fundación Sadosky [19] there are 43 different university level institutions in Argentina offering degree programs in the Information Technology and Computer Science area. Beyond this, the authors personally know several professors (i.e. in UBA, UTB-FRBA and UCA) that are teaching Agile Software Development but that have never published any articles about this. So it might be assumed that Agile Software Development is present in the Argentinian universities to a higher degree than was found in this research study.

5 Threats to validity

The ability to draw general conclusions from this research is limited by the fact that only published experiences are being surveyed, as we have mentioned. This research carefully considered actual curricula, not planned approaches, to avoid misreading situations where the intent was not carried out.

The conclusions of this work might also be affected by the fact that some relevant articles might have been missed for 2 different reasons:

- Part of the search was performed manually, so like any manual task it is error prone.
- JAIIO publications prior to 2013 are not available online in a digital format, so it was not possible to cover them.

6 Conclusions

Evidence shows that Agile Software Development is part of the Information Technology and Computer Science Programs in nine Argentinean universities, this answers our first research question.

Regarding our second research question, Scrum is the most common method taught (it is mentioned in 12 out of 17 publications). Some reports do not mention any method but describe specific practices.

Regarding the third research question, there is little information about the teaching strategy and several papers do not mention it at all. Among the articles that do mention elements related to this there is no single preferred approach but a learning-by-doing strategy is used in many cases.

The results and findings presented in this article open several possible lines of work. In the context of the Specialization Degree assignment our next step will be performing interviews with professors of software development courses in order to have a deeper understanding of the teaching strategy. The goal will be to perform enough interviews to obtain representative results to describe the state of agile education in Argentina. Beyond our next steps, it might also be interesting to assess the impact that agile education experiences have on practitioners if and when they join the software development industry.

References

1. Manifiesto for Agile Software Development, <https://agilemanifesto.org/>. Last accessed 20 July 2019
2. West, D., and Grant, T.: Agile Development: Mainstream Adoption Has Changed Agility. Forrester, 2010.
3. Pressman, R. and Maxim, B.: Software Engineering: A Practitioner's Approach. McGraw-Hill Education; 8 edition. 2014
4. Sommerville, I.: Software Engineering. Pearson, 10 edition . 2015
5. ACM Curricula Recommendations, <https://www.acm.org/education/curricula-recommendations>. Last accessed 15 July 2019
6. Paez, N., Fontdevila, D. and Oliveros, A.: HELENA Study: Initial Observations of Software Development Practices in Argentina. 18th International Conference on Product-Focused Software Process Improvement, PROFES 2017; Innsbruck; Austria; 29 November 2017 through 1 December 2017
7. Ministerio de Educación, "Resolución 786/2009," InfoLEG Información Legislativa, <http://servicios.infoleg.gob.ar/infolegInternet/verNorma.do?id=154121>. Last accessed 15 July 2019
8. Paez, N.: A Flipped Classroom Experience Teaching Software Engineering. In: IEEE/ACM 1st International Workshop on Software Engineering Curricula for Millennials (SECM), Buenos Aires, 2017, pp. 16-20. doi: 10.1109/SECM.2017.6
9. Scott, W., Rodríguez, G., Soria, A. & Campo, M.: Experiences in software engineering education: Using scrum, agile coaching, and virtual reality. doi: 10.4018/978-1-5225-3923-0.ch050.
10. Reingart, M. and Dominguez, S.: Diplomatura abierta en Software Libre, Propuesta de cursos de formación profesional en Tecnologías de la Información y Comunicación libres y de fuente abierta. XLIII Jornadas Argentinas de Informática e Investigación Operativa, Simposio Argentino de Tecnología y Sociedad. Buenos Aires, 2014.
11. Chookittikul, W., Maher, P. E., & Kourik, J. L.: Agile methods in thai higher education and beyond. Paper presented at the 2011 24th IEEE-CS Conference on Software Engineering Education and Training, CSEE and T 2011 - Proceedings, 557. doi:10.1109/CSEET.2011.5876153
12. Melo, O., Santos, C.V., & Katayama, E.: The Evolution of Agile Software Development in Brazil. Journal of the Brazilian Computer Society 19, no. November 2013 (n.d.): 523552.
13. Tranter, L.: Are agile and Scrum the same? <https://www.extremeuncertainty.com/are-agile-and-scrum-the-same/> Last accessed 27 July 2019
14. Servicio de Difusión de la Creación Intelectual de la Universidad Nacional de La Plata, <http://sedici.unlp.edu.ar/>. Last accessed 20 July 2019
15. Scopus, <https://www.scopus.com/>. Last accessed 26 July 2019
16. IEEE Xplore Digital Library, <https://ieeexplore.ieee.org/>. Last accessed 30 July 2019
17. Kuhrmann, M., Mndez Fernndez, D., Daneva, M. (2017) On the pragmatic design of literature studies in software engineering: an experience-based guideline. Empirical Software Engineering 22(6), 2852-2891
18. Kitchenham, B. (2007) Guidelines for performing Systematic Literature Reviews in Software Engineering, Keele University, UK
19. Estudiar Computación Website, <http://www.estudiarcomputacion.gob.ar/>
20. 13th Annual State of Agile Report. Version One, 2019, <https://www.stateofagile.com/ufh-i-521251909-13th-annual-state-of-agile-report>. Last accessed 15 July 2019

ProgEnv: Un entorno para la enseñanza de la programación

Arroyo Marcelo, Bavera Francisco, Bengolea Valeria, Cerda Ernesto, Kilmurray Cecilia, Novaira Maria Marta, Permigiani Sonia, Ponzio Pablo, Scilingo Gastón, Tardivo María Laura, Uva Marcelo

Universidad Nacional de Río Cuarto,
Río Cuarto, Argentina

{marcelo.arroyo, pancho, vbengolea, ecerda, ckilmurray, mnovaira, spermigiani, gaston, lauratardivo, uva}@dc.exa.unrc.edu.ar

Abstract. El curso de Ingreso Universitario en carreras de computación, es para muchos estudiantes, el primer contacto con la disciplina y por ello es de gran importancia el desarrollo de actividades que permitan introducir los conceptos básicos, favoreciendo la participación de los estudiantes a través de consignas que estimulen la creatividad y el interés por las ciencias de la computación, en particular por la programación. La complejidad de las actividades abordadas es sin duda un factor determinante para promover el aprendizaje de conceptos asociados a la programación, siendo muchas veces motivo de frustración para alumnos que se inician en esta práctica. En este contexto, la elección de las herramientas y lenguajes de enseñanza adecuados juegan un rol fundamental, permitiendo desarrollar nociones complejas con un alto grado de abstracción de manera simple y clara. En este trabajo presentaremos un entorno de desarrollo visual para la enseñanza de la programación, al que llamamos *ProgEnv* por *Programming Environment*, su utilización en el dictado del módulo “Introducción a la Programación” que forma parte de las actividades de ingreso a las carreras Analista en Computación, Profesorado y Licenciatura en Ciencias de la Computación de la Facultad de Ciencias Exactas de la Universidad Nacional de Río Cuarto y los resultados obtenidos en dicha experiencia.

1 Introducción

La enseñanza de la programación tanto en niveles educativos iniciales como en cursos introductorios universitarios, es objeto constante de estudio en el mundo. Muchos lenguajes y entornos de programación han sido desarrollados con el propósito de introducir nociones asociadas a la programación [10, 5, 1, 2, 7], en los cuales se propone representar las instrucciones con bloques que se pueden encastrar y desencastrar para formar programas minimizando de esta manera, problemas de tipo sintáctico, los cuales suelen ser muy frustrantes para aquellos que se inician en la programación. De este modo, se simplifica la tarea de construcción de programas poniendo el foco en la creatividad y el ingenio para resolver problemas algorítmicos. Asociado a estos lenguajes, se han desarrollado y

están disponibles para su uso, muchas plataformas online [4, 9, 6], que proveen secuencias de actividades utilizando como base este tipo de lenguajes para enseñar nociones tales como repetición, condicional, abstracción procedural, etc.

Este tipo de lenguajes y plataformas están siendo utilizadas en muchas instituciones educativas, en particular el Departamento de Computación de la facultad de Ciencias Exactas Físico Químicas y Naturales de la Universidad Nacional de Río Cuarto ha utilizado durante los últimos años una variedad de estos lenguajes y plataformas en el dictado del módulo Introducción a la Programación que forma parte de la currícula de ingreso a las carreras: Analista en Computación, Profesorado y Licenciatura en Ciencias de la Computación. El programa curricular de este módulo consiste básicamente en análisis de problemas, diseño de soluciones algorítmicas, lenguajes y conceptos introductorios de programación; consta de 7 clases de 3 horas cada una y persigue como propósito principal brindar una primera mirada de la disciplina general, y en particular de la programación, pensada desde una perspectiva vocacional que despierte entusiasmo por las Ciencias de la Computación en los estudiantes. Por otra parte, también se utilizan en actividades orientadas a estudiantes de los últimos años del ciclo escolar anterior. Estas actividades tienen una clara intención vocacional.

Los lenguajes de programación visuales han resultado propicios para el dictado del módulo anteriormente descrito por varias razones: proveen una interfaz intuitiva para los alumnos permitiendo la construcción de programas simples en la primera clase, ayudan a estimular la creatividad y entrenar a los estudiantes en la resolución de problemas algorítmicos, sin dejar de mencionar la posibilidad de utilizar una innumerable cantidad de actividades que se encuentran disponibles como parte de las plataformas. En contraparte, motivaron el desarrollo de este trabajo algunas dificultades encontradas en el sucesivo dictado del módulo en años anteriores:

- Los alumnos presentan dificultad en trasladar los conceptos aprendidos utilizando los lenguajes visuales a un lenguaje textual. Creemos que esto se debe a que, debido al escaso tiempo de duración del módulo, no es posible poner énfasis en este punto.
- Los docentes del módulo pretenden realizar un pequeño taller que permita introducir los conceptos básicos de programación de manera incremental a través de una secuencia de ejercicios cuya composición culmine en el desarrollo de un juego simple. El principal objetivo de este taller es captar el interés de los alumnos, abordando consignas desafiantes que le permita a los alumnos percibir la aplicación de los nociones enseñadas. Muchas de las plataformas existentes no proveen actividades configurables sino una secuencia de ejercicios que poseen un hilo temático en común.

En este artículo presentamos *ProgEnv* [8], un ambiente de programación web para la enseñanza de la programación. El mismo consiste en un entorno gráfico mediante el cual es posible utilizar recursos funcionales, predefinidos por el docente, permitiendo la incorporación progresiva de elementos sintácticos y construcciones lógicas del lenguaje de programación, en este caso particular JavaScript [14], aunque podría extenderse para incorporar el uso de otros

lenguajes de programación. Además, *ProgEnv* le provee al docente una interfaz de programación flexible para la incorporación y modificación de actividades disponibles de manera sencilla e intuitiva. A través de la experiencia en el uso de *ProgEnv* en el dictado del módulo Introducción a la Programación del ingreso a las carreras de computación de la Universidad Nacional de Río Cuarto en febrero de 2019 pudimos observar una significativa mejora en los conceptos adquiridos por los estudiantes al finalizar el dictado del módulo, donde la mayoría de los estudiantes logró los objetivos planteados. En las secciones a continuación describiremos en detalle el ambiente desarrollado, las funcionalidades que posee, las ventajas observadas durante su uso, algunas actividades abordadas utilizando el entorno y los resultados obtenidos en tal experiencia.

2 *ProgEnv*: Entorno de Desarrollo para la enseñanza de la programación

ProgEnv es un ambiente de desarrollo web pensado para ser utilizado en cursos introductorios de programación, su nombre tiene origen en el término en Inglés: *Programming Environment*. La herramienta fue desarrollada en HTML y JavaScript utilizando *w2ui* [13] como librería para la creación de la interfaz de usuario y *codeMirror* [3], aportando una experiencia gráfica placentera a los alumnos y docentes. El entorno puede ser utilizado usando sólo un navegador web y está disponible en la web de ingresantes del Departamento de Computación [11]. Éste puede ser utilizado tanto dentro del aula como fuera. Cabe mencionar que utilizar un navegador web como ambiente de desarrollo trae consigo algunos beneficios:

- No requiere la instalación por parte de los usuarios de ningún software extra.
- El docente puede integrar, de forma natural, recursos multimedia como imágenes o videos, así como también vincular la aplicación con la API de otros sitios web
- Todos los actores, alumnos y docentes, están familiarizados con su uso.

Al igual que la mayoría de las plataformas de enseñanza basadas en lenguajes de programación con bloques, *ProgEnv* dispone de un área para escribir el programa y otra para visualizar los resultados de su ejecución. Esta última se compone de un lienzo para exhibir resultados gráficos y de una consola para mostrar resultados que comprenden texto. El entorno incluye un visualizador de la posición del mouse cuando éste es arrastrado por encima del lienzo. Esta funcionalidad, puede ser usada en la resolución de problemas que requieren conocer la ubicación exacta de un “click” sobre el lienzo, o utilizar el lienzo como un plano de coordenadas cartesiana, por ejemplo para dibujar figuras geométricas en posiciones o tamaños específicos del mismo.

El entorno cuenta con una barra lateral izquierda donde se muestran las actividades disponibles para el alumno. La selección de una actividad específica, permitirá, en el área inferior de la pantalla, visualizar las instrucciones disponibles

para lograr los objetivos planteados en dicha actividad, así como también el enunciado de la actividad misma. Las instrucciones disponibles están predefinidas por el docente como bloques modulares (no visibles al alumno) en el lenguaje JavaScript. Con el objetivo de esclarecer lo antedicho, supongamos una actividad sencilla:

Arrojar un dado, si el número obtenido es par entonces dibujar un triángulo, caso contrario, dibujar un círculo.

Asumimos que se cuenta con las siguientes instrucciones (que pre-define el docente y se despliegan al seleccionar la actividad):

- `tirarDado()`: Devuelve de manera aleatoria un número entre 1 y 6, y escribe el número aleatorio en la consola.
- `esPar(nro)`: Retorna `true` si `nro` es par, y `false` en caso contrario
- `dibujarCirculo(ox,oy, r)`: Dibuja un círculo con centro (ox,oy) y radio r .
- `dibujarTriangulo(ox,oy, l)`: Dibuja un triángulo equilátero con vértice izquierdo (ox,oy) y lado de longitud l .
- `cambiarAnchoLapiz(nro)`: Cambia el ancho del lápiz de dibujo al ancho dado por `nro`
- `cambiarColorLapiz(c)`: Cambia el color del lápiz de dibujo al color c . colores disponibles: *verde, rojo, blanco, negro, azul, amarillo y violeta*.

El conjunto de instrucciones predefinidas podrá ser utilizado en conjunto con las estructuras de control (`if then else` en este caso) para construir una solución al problema planteado. Mediante este conjunto de instrucciones predefinidas se define la capa de abstracción que el docente pretende, según los conceptos que desea exponer en la actividad. Por ejemplo, el docente podría no brindar la instrucción `esPar(nro)` como parte de la interfaz de programación, y proponer que el alumno implemente esta función. Como otro ejemplo, si se tratase de una actividad inicial, el docente podría proporcionar instrucciones `dibujarCirculo()` y `dibujarTriangulo()`, que dibujan respectivas figuras de algún tamaño y en una posición fija del lienzo, en lugar de proveer instrucciones genéricas que requieren argumentos, para luego, moverse a un conjunto de instrucciones más generales en etapas más avanzadas del dictado del módulo.

Es importante notar que, a diferencia de las plataformas basadas en bloques el alumno podrá utilizar, además de la API predefinida, las estructuras de control, abstracciones, definición de variables, etc del lenguaje JavaScript (Aunque si fuera de interés, el docente también podría definir abstracciones para tales construcciones).

En la figura 1, se muestra una solución del problema descripto.

El entorno resulta adecuado para definir una secuencia de actividades como la del ejemplo, que mantengan un hilo en común, para alcanzar de manera incremental, reusando creativamente lo realizado, algún objetivo general.

Algunos puntos que diferencian *ProgEnv* de otros entornos para la enseñanza de la programación:

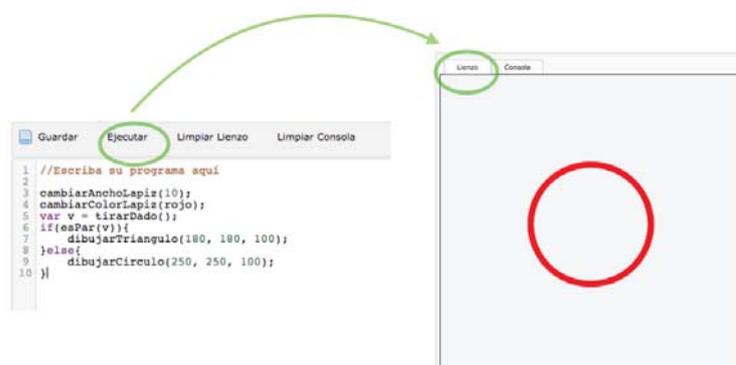


Fig. 1. Solución al problema enunciado en la sección 2

- Permite que un estudiante que recién se inicia en la programación pueda escribir un programa en un lenguaje textual (JavaScript en este caso) utilizando el conjunto de instrucciones propuesta por el docente.
- Posibilita el desarrollo de actividades de manera incremental y en diferentes niveles de abstracción, atendiendo la diversidad y heterogeneidad de los ingresantes. Esta característica posibilita su uso en otros niveles educativos.
- El docente es quien define, según los objetivos de la clase, la API de la que dispondrá el alumno para resolver las actividades de la clase, dando la posibilidad al docente de generar propuestas creativas que despierten el entusiasmo de los estudiantes.
- Es simple de utilizar dentro y fuera del aula ya que no requiere más que de un navegador web. Inclusive desde un celular el estudiante puede realizar las actividades propuestas.

2.1 JavaScript como lenguaje de programación inicial

Como se ha mencionado en las secciones anteriores de este trabajo, *ProgEnv* soporta JavaScript como lenguaje de programación.

Si bien la elección de JavaScript como lenguaje base tiene su fundamento principal en cuestiones técnicas: *ProgEnv* es una aplicación del lado del cliente, y en este contexto ejecutar un programa JavaScript es trivial; es importante mencionar algunas consideraciones que han hecho adecuada esta elección en el ámbito del módulo dictado:

- En la práctica sólo hemos usado un subconjunto del lenguaje, el cual incluye:
 - Composición secuencial
 - Declaración y uso de variables (**Var**)
 - Asignación simple: =

- Operadores de comparación: `==`, `!=`, `>`, `<`, etc
- Operadores lógicos : *and* (`&&`), *or* (`||`) y *not* (`!`)
- Operadores aritméticos estándar: suma (+), resta (−), multiplicación (*), división (/) y resto (%)
- Constantes String: ej: “hola mundo”
- Constantes booleanas: `true` y `false`
- Estructuras de control:
 - * `if then`
 - * `if then else`
 - * `for(inicialización; condición; incremento)`
 - * `while(condición)`
- Abstracción procedural : funciones con y sin parámetros. Funciones que retornan valores (`return`)

La selección de este subconjunto del lenguaje se basó en las necesidades específicas del curso y las actividades abordadas, de acuerdo al escaso período de duración del ingreso: 7 clases introductorias.

- El subconjunto de JavaScript mencionado resulta suficiente para abordar actividades cuyo objetivo principal es iniciar a los estudiantes en la resolución algorítmica de problemas captando el interés y despertando la creatividad y el ingenio de los mismos. Teniendo en cuenta este punto, no es necesario introducir conceptos más avanzados del lenguaje (objetos, eventos, etc).
- El docente puede utilizar otras características específicas del lenguaje ocultando detalles a los estudiantes.
- Si bien javascript es un lenguaje con tipado dinámico, la introducción de tipos de datos no fue un objetivo del curso de ingreso, por lo que esto no resultó, en nuestro caso, un inconveniente. Cabe destacar que la mayoría de las plataformas mencionadas en la introducción de este trabajo, no abordan en primera instancia tipo de datos. Por otro lado, es bien conocido el lenguaje TypeScript [12], un superconjunto de JavaScript que esencialmente añade tipado estático, por lo que si fuera necesario *ProgEnv* podría extenderse para soportar este lenguaje.

3 Ingreso Universitario 2019: Una experiencia de uso

La intención de esta sección es presentar alguna de las actividades llevadas a cabo utilizando *ProgEnv* durante el dictado del módulo “Introducción a la Programación” que forma parte del ingreso universitario 2019.

En esta experiencia, las actividades desarrolladas utilizando *ProgEnv* permitieron abordar, de manera incremental, conceptos básicos de programación estimulando razonamientos lógico-matemáticos necesarios para resolver situaciones problemáticas interesantes.

Las primeras actividades fueron diseñadas y realizadas con el objetivo de introducir el entorno de desarrollo y nociones iniciales de programación. Se plantearon actividades tales como dibujar figuras geométricas simples empleando las instrucciones predefinidas (presentadas como un ejemplo en la sección

2), donde además de ir conociendo el entorno de desarrollo, los estudiantes aplicaron conceptos de composición secuencial, condicional (ver figura 1); y operadores lógicos y matemáticos de manera muy natural. Estas actividades motivaron a aquellos estudiantes curiosos y creativos a escribir diferentes programas para realizar dibujos más complejos.

En una de las actividades propuestas se presentó el Lienzo con dos líneas perpendiculares ya dibujadas (horizontal y vertical) quedando éste dividido en cuatro cuadrantes iguales. La tarea consistió en escribir un programa que identifique e informe el cuadrante en el que se produjo un *click*. Para realizar esto se proporcionó el siguiente conjunto de instrucciones:

- `escribir(mensaje)`: Muestra el mensaje dado en la Consola
- `obtenerX()`: devuelve la coordenada X del punto (dentro del lienzo) en donde se hizo *click*
- `obtenerY()`: devuelve la coordenada Y del punto (dentro del lienzo) en donde se hizo *click*

A diferencia del ejemplo dado en la sección anterior, en este caso la ejecución del programa ocurre cuando se hace *click* en el Lienzo. Cabe destacar que, nuevamente las instrucciones predefinidas ocultan los detalles del manejo de eventos.

Esta como otras actividades abordadas, tuvieron como objetivo proporcionar conocimientos y herramientas que ayuden a dibujar y manipular un tablero de *TaTeTi*, actividad que se presentó llegando al final del módulo.

El desarrollo del *TaTeTi*, consistió en dibujar el tablero (conformado por una grilla de 3x3), identificar la posición dentro de la grilla en donde se produjo un *click* y, según el jugador (Rojo o Azul) de turno, dibujar en la posición de la grilla *clikeada* un cuadrado rojo o un cuadrado azul. Cabe destacar que, la posibilidad de predefinir el conjunto de instrucciones, y proporcionar variables y funciones ya definidas en el área de código (por ejemplo, asociadas al cambio de turno), ayudó significativamente a que los estudiantes pudieran completar esta actividad. La secuencia de pasos completa para llegar al desarrollo de este juego se puede ver en la página web de Ingresantes del Departamento de Computación [11].

Por último, el desarrollo de una función que pudiera identificar el jugador ganador también fue, de manera guiada por el grupo de docentes, implementada por los estudiantes.

Es importante destacar que la mayoría de los alumnos pudo llegar al final del desarrollo, los alumnos no solo se mostraron muy entusiasmados en realizar esta actividad sino también en compartir con el resto del grupo los logros alcanzados.

Una evaluación diagnóstico final permite valorar las actividades realizadas en el módulo. En la misma se pudo observar que un 70% de los estudiantes evaluados lograron adquirir los conceptos abordados, tales como, lenguaje, diseño de soluciones algorítmicas, conceptos básicos de programación, estructuras de control y abstracción, siendo éste un resultado altamente alentador para continuar con el camino emprendido en esta experiencia.

4 Trabajos a Futuro

Como hemos mencionado en diversas ocasiones en este artículo, *ProgEnv* fue utilizado en una única experiencia: el ingreso universitario 2019 a las carreras de computación de la Universidad Nacional de Río Cuarto. De dicha experiencia han surgido varias propuestas para darle continuidad a este proyecto. En lo que se refiere al entorno en sí mismo, se propone desarrollar un primer prototipo cliente-servidor que cuente con las siguientes características:

- Posibilidad de crear y administrar cuentas de usuarios con vista para docentes y alumnos, permitiendo de esta manera que los alumnos pueden crear soluciones a las actividades planteadas, guardarlas y modificarlas. La vista del docente, permitirá crear y manipular actividades con una interfaz de usuario integrada que permita de manera simple codificar instrucciones pre-definidas y ocultar detalles no relevantes para el alumno.
- Soporte a diferentes lenguajes de programación

Por último, desde el punto de vista de la enseñanza, se pretende estudiar otras experiencias similares de enseñanza para evaluar intensivamente tanto la elección del lenguaje de programación como las actividades propuestas.

5 Conclusiones

En este artículo hemos presentado *ProgEnv* un ambiente de programación web para la enseñanza de la programación. El desarrollo de este ambiente surge de la necesidad de repensar las actividades de ingreso a las carreras Analista en Computación, Profesorado y Licenciatura en Ciencias de la Computación de la Facultad de Ciencias Exactas de la Universidad Nacional de Río Cuarto. La utilización de *ProgEnv* durante el dictado del módulo Introducción a la Programación actuó como hilo conductor en las diferentes clases planificadas, logrando una incorporación progresiva de los conceptos abordados. Las actividades desarrolladas utilizando *ProgEnv* nos permitieron introducir de manera incremental los distintos conceptos, estructuras de control y abstracciones culminando en la implementación del juego TaTeTI.

Es importante remarcar que los estudiantes se mostraron motivados en cada una de las actividades propuestas y como se destacó en la sección 2, el examen diagnóstico final arrojó que un 70% de los estudiantes evaluados lograron adquirir los conceptos abordados.

Por último, el desarrollo y uso de *ProgEnv* brindó la oportunidad de incorporar las valiosas experiencias de docentes que han estado involucrados en numerosas oportunidades en las actividades de ingreso y pre-ingreso facilitando por otra parte, el inicio de los estudiantes a las primeras materias específicas de la disciplina.

References

1. Alice: <https://www.alice.org/>.
2. Appinventor: <https://appinventor.mit.edu/>.
3. Codemirror: <https://codemirror.net/>.
4. Code.org: <https://code.org/>.
5. Gobstones: <https://gobstones.github.io/>.
6. Mumuki: <https://mumuki.io/paths>.
7. Pilas bloques: <http://pilasbloques.program.ar/>.
8. Progenv: <https://github.com/marcelodarroyo/progenv>.
9. Program.ar: <http://program.ar/>.
10. Scratch: <https://scratch.mit.edu/>.
11. Sitio del Departamento de Computación, UNRC: <https://dc.exa.unrc.edu.ar>.
12. Typescript: <https://www.typescriptlang.org/>.
13. w2ui: <http://w2ui.com/web/>.
14. Axel Rauschmayer. *Speaking JavaScript: An In-Depth Guide for Programmers*. O'Reilly Media; 1 edition, 2014.

Revisión Sistemática: Elicitación de requerimientos educativos en serious games

Adolfo Spinelli¹, Stella Massa¹:

¹ Grupo de investigación en Tecnologías Interactivas (GTI), Facultad de Ingeniería de la Universidad Nacional de Mar del Plata, República Argentina, Provincia de Buenos Aires, ciudad de Mar del Plata. avenida Juan B. Justo y Ratero.
spinelliadolfo@gmail.com y smassa4@gmail.com.

Resumen. La introducción de los serious game en el ámbito de la educación formal es reciente y enfrenta importantes desafíos. Estas aplicaciones se concebían para el entrenamiento de profesionales en temas críticos, siendo su desarrollo caso por caso, lo que implicaba un alto costo. Reducir estos costos acercándose a los estándares de la industria de videojuegos, es una meta necesaria para que los docentes consideren el uso de esta herramienta de aprendizaje. Nuestro desafío es proveer buenas prácticas de diseño y desarrollo, específicas para serious game, con énfasis en la introducción de las secuencias de aprendizaje dentro del juego, para ello, es determinante la elicitación de requerimientos educativos. El presente estudio revisa las metodologías propuestas tanto en el ámbito académico como, privado, con el objeto de identificar las líneas de trabajo y detectar las oportunidades de mejora en este campo de investigación.

Palabras Clave: serious game, Herramientas de aprendizaje, Requerimientos educativos, Elicitación, Diseño.

1 Introducción

Ya en 1790 el P.D. Jaime Roig y Benet, vicario de la Real Cartuja de Porta proponía un método para enseñar números, letras, puntuación y sílabas mediante juegos de mesa.

En 1979 Abt [1] introduce el término Serious Game (SG) para identificar aquellos juegos cuyo objetivo era la capacitación. La extensión del concepto a los videojuegos era cuestión de tiempo.

Para [2] y [3] un videojuego (VJ) simula una realidad basada en ciertas reglas, donde los jugadores conviven con un conflicto artificial donde ponen sus emociones en juego. Estas emociones y sensaciones atraen al jugador [4] y componen el “estado flow” descrito por [5] el cual se manifiesta en el grado de inmersión definido por [6]: “nos sumergimos completamente en otra realidad [...] que acapara toda nuestra atención y aparato sensorial.”. El grado de inmersión que disfruta el jugador a través de estas sensaciones y emociones, configuran se la conoce como la experiencia del jugador que [7] denomina “jugabilidad”.

Según [8] y [9] los SG pretenden que el jugador incorpore conocimiento o modifique su conducta. Su uso en la educación formal es reciente, estudios como los propuestos por [10] avalan su eficacia. En tanto que [11] y [12] reportan la contribución de la actividad lúdica en el aprendizaje.

Si queremos un SG efectivo según [8] en el debe primar el entretenimiento, con un equilibrio entre aprendizaje y diversión sin perder el “estado flow”. En consonancia [13] y [14] afirman que deben aprovecharse la experiencia del jugador para propiciar y potenciar el aprendizaje.

Jenkins [11] identifica once habilidades sociales y competencias culturales que deben adquirir los jóvenes para participar en el nuevo ambiente de medios: Jugar, Actuar, Simular, Apropiar, Multitarea, Cognición distribuida, Inteligencia colectiva, Juicio, Navegación transmediática, Trabajo en la red y Negociación.

A través de ellas se experimenta con la realidad adoptando diversos puntos de vista, se interpreta la realidad, se interactúa con el medio y los pares, se intercambia información, se resuelven problemas y se aprende.

Por su parte Prensky [12] enumera cinco niveles de aprendizaje : Aprender como jugar, Aprender que hacer o no hacer en el juego, Aprender por que hacer algo en el juego, Aprender donde hacerlo, Aprender cuándo hacerlo y si conviene hacerlo.

Estas competencias y niveles están relacionados, pues acceder a un determinado nivel requiere dominar algunas de estas competencias. Podemos aprovechar esta relación para embeber las rutinas de aprendizaje, sin perder el “estado flow”.

Al diseñar [15] afirma que se debe asumir la necesidad de despertar interés para que el juego contribuya con el aprendizaje. Por su parte [16], [17], [18], [19] y [20] coinciden en afirmar que esto solo se logra al crear una motivación intrínseca de forma que el aprendizaje ocurra naturalmente.

Como señala [21] el éxito de un SG depende de una especificación que cubra los aspectos educativos, lúdicos y de contexto. Para ello se debe combinar la experiencia del juego y la motivación intrínseca, embebiendo los objetivos de aprendizaje.

En el 2008 Sawyer y Smith [9] proponen una taxonomía para SG, donde clasifican estos según el sector de la sociedad que los solicita (Gobierno/ONG, Defensa, Salud, Marketing, Educación, Empresas e Industrias) y el área del objetivo de capacitación buscado (Salud, Publicidad, Formación, Educación, Ciencia, Producción y Juego como Empleo).

Analizando dicha taxonomía se observa que salvo en el aprendizaje en el aula, las aplicaciones se caracterizan por contener un objetivo corporativo. Son desarrollos de alto costo, pensadas para un público reducido, donde la motivación intrínseca radica en la necesidad del jugador de capacitarse y entrenarse, en este contexto la inmersión aporta realismo a las situaciones de juego, pero no atrae al jugador.

Los SG acumulan casos de éxito, sin embargo estos no son tan evidentes en el aula, esta situación puede explicarse en la naturaleza de los SG tradicionales y las particularidades de las actividades dentro del aula:

- Los usuarios son alumnos y docentes, se necesitan SG que capaciten en los contenidos educativos de los docentes y estos evoluciona permanentemente, en contraposición con los otros SG, no tienen porque perdurar en el tiempo.

- Los docentes (que no son expertos operadores de una aplicación) deben ser capaces de entender la tecnología y aprovecharse de ella creando rutinas de aprendizaje acorde a sus necesidades.

- Los alumnos no están obligados a aprender mediante el juego, por lo cual es preciso aprovecharse del “estado flow”, para atraerlos.

- El desarrollo de los SG, debe lograr productos asequible a los docentes y alumnos. Y flexibles para adaptarse a los cambios de contenido.

Este listado de razones no es exhaustivo y aunque podemos encontrar otras causas, es útil obtener información sobre una causa posible: la elicitación de requerimientos educativos en el contexto del diseño de SG, pues ella es determinante a la hora del diseño.

2 Método

En esta revisión sistemática se utiliza el método propuesto por [22], el cual consiste en planificar la revisión, obtener los documentos, analizar los mismos y obtener conclusiones.

Para obtener información relevante sobre los mecanismos de elicitación en SG se formulan cuatro preguntas que se enumeran en la Tabla 1.

Tabla 1. Preguntas de investigación.

Q1	¿Cuáles son las metodologías o procesos utilizados en la obtención de requerimientos para la especificación de un SG?
Q2	¿Qué tipos de requerimientos se obtienen en una elicitación de SG, para la especificación del mismo?
Q3	¿Qué características o atributos relevantes se toman en cuenta en los requerimientos educativos para un SG?
Q4	¿Cuáles son las metodologías o procesos utilizados en la obtención de requerimientos educativos para la especificación de un SG?

Las bases de datos utilizadas son: *SpringerLink*, *ScienceDirect (Elsevier)*, *IEEEExplore*, *Taylor & Francis*, *Wiley*, *ACM*, *Sage Journal*, *DOAJ* e *Inderscience*. Las mismas fueron elegidas por ser las de mayor contenido sobre SG. Con el objeto de consultar dichas bases y procurando responder a las preguntas formuladas se diseñó la siguiente sentencia de consulta:

((serious game OR serious games) AND (Specification OR Elicitation OR Specifications OR Elicitations) AND (Requirement OR Requirements) AND (Pedagogical))

Las consultas se realizaron entre 28/02/2019 y el 1/3/2019, de la cuales se obtuvieron 112 documentos a los cuales se aplicaron los criterios de exclusión e inclusión (resumidos en la tabla 2). Luego de esta tarea quedaron 28 trabajos cuya distribución por año y fuente se refleja en la Figura 1. Por último se procedió a leer los artículos en forma completa, al final de lo cuál quedaron seleccionados 14 trabajos, cuya distribución por año y fuente refleja la Figura 2.

Tabla 2. Criterios de Inclusión y Exclusión

Inclusión	Exclusión
Artículos publicados entre 2007 y 2019	Artículos anteriores al 2007
Artículos de revistas, conferencias y congresos	Artículos no presentados en revistas, conferencias o congresos.
Contenido en el resumen relacionado con el tema	Contenido en el resumen no relacionado con el tema
Artículos accesibles	Artículos no accesibles

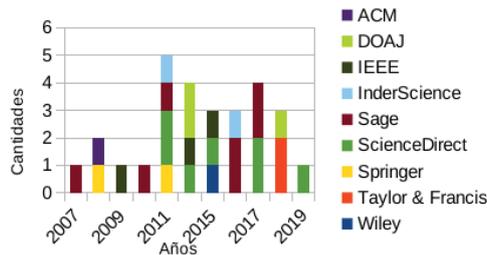


Figura 1: Distribución por Banco de Datos y año de publicación luego de aplicar los criterios de inclusión y exclusión.

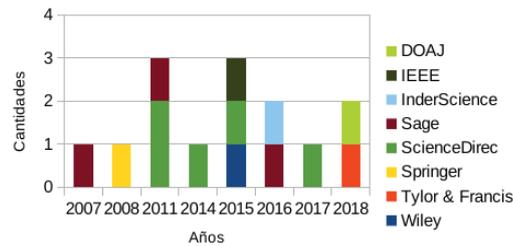


Figura 2: Distribución por Banco de Datos y año de los artículos utilizados en la revisión.

3 Resultados

En esta sección, se resumen los resultados de los trabajos seleccionados para esta revisión con respecto a las preguntas de investigación citadas anteriormente en este artículo.

Q1: ¿Cuáles son las metodologías o procesos utilizados en la obtención de requerimientos para la especificación de un SG?

De Lope [27] afirma que las metodologías y herramientas existentes para el desarrollo de SG son escasas y no son específicas. Como consecuencia de ello, [31] y [29] afirman que las mismas no garantizan su efectividad en el aprendizaje.

Los marcos de diseño aún están en desarrollo y se enfocan en el desarrollador, no hay diseño participativo, colaborativo y multidisciplinario [28], [27] y [25], a pesar que el producto es especialmente apto para estos enfoques [23] y [36].

Los procesos de diseño para VJ propuestos no son descriptos en detalle, un ejemplo es el propuesto por [27], que enumera nueve fases pero no menciona técnicas y métodos.

Sólo los modelos conocidos como MDA (Mecánica Diseño y Estética) y su extensión el DPE (Diseño Juego y Experiencia) describen el proceso de diseño del juego, la estrategia de aprendizaje y la experiencia del jugador, al tiempo que toman en cuenta la colaboración [35]. Algunos de los modelos propuestos para SG amplían el modelo DPE, dotando a este de mecanismos que promuevan la participación de los stakeholders [32] y [35].

Aunque toman ideas de modelos de diseño pensados para VJ, algunos modelos para SG, usan lenguajes de modelado para describir las narrativas, tales como DVSL (Lenguaje Visual de Dominio Específico) [25] y DSML (Lenguaje de Modelado de Dominio Específico) [30], al hacerlo apuestan a lograr un mecanismo descriptivo que involucre a todos los interesados en el diseño. Dado que los lenguajes de modelado no sirven para especificar la relación juego/aprendizaje [25] propone el uso del protocolo IMP (Protocolo e Mapeo de Intervenciones) junto a anotaciones y [30] hace uso del modelo LM-GM (Máquina de Aprendizaje – Máquina de Juego).

Otro enfoque es el de [23], donde el diseño se basa en el desarrollo de un prototipo evolutivo donde los interesados en el diseño puedan visualizar el resultado de sus propuestas, evitando confusiones y malos entendidos.

EMERGO es una metodología propuesta por [33] en línea con la propuesta anterior, que modela situaciones reales con secuencias de aprendizaje que implican toma de decisiones y resolución de problemas, inteligencia, razonamiento, y otras habilidades cognitivas. Representadas mediante componentes re-utilizables.

Avanzando en la misma idea los modelos descriptos por [24], [26], [29] y [31], proponen desde diferentes puntos de vista la construcción de tablas de criterios y componentes re-utilizables (plantillas de mecánicas, patrones por ejemplo). Con respecto a esto último [32] propone una arquitectura de diseño, dentro de la iniciativa RAGE (*Realising an Applied Gaming Eco-System*), basada en la construcción de componentes donde la especificación del dominio y los contenidos de aprendizaje, se diseñen en forma genérica, con el objeto de reutilizarlos y al mismo tiempo permitir que los expertos en dominio y educación trabajen en forma independiente del resto.

Q2: ¿Qué tipos de requerimientos se obtienen en una elicitación de SG, para la especificación del mismo?:-

Existe consenso en afirmar los requerimientos a especificar en un SG, cubren tres dimensiones, aquellas características de todo producto de software, aquellas que describen los mecanismos del juego que han de proveer jugabilidad e inmersión y aquellas que describen los requerimientos educativos relacionados con los contenidos y la evaluación de los mismos. La especificación de los dos últimos no sería completa si en cada uno no consta la interacción entre ellos con el objeto de proveer el balance juego/aprendizaje.

Para obtener un modelo de diseño eficaz es preciso determinar los aspectos relevantes en un SG, para lo cual debe analizarse su uso en diversos ambientes de aprendizaje, por ejemplo el aula. Según el análisis realizado por [30] existen dos facetas la diversión y la educación, esto concuerda con la opinión del resto de los

autores presentes en esta revisión.

Desde diferentes enfoques los autores expresan el consenso existente sobre la necesidad de lograr la coexistencia de ambas facetas dentro del SG y que solo su adecuada integración dará como resultado un producto que cumpla sus funciones educativas.

Sin embargo es necesario [26] y [24], la existencia de nuevos marcos teóricos que además de tratar la motivación o el aprendizaje, incluyan la integración de ambas facetas.

Aunque en línea con estos nuevos desarrollos, algunos proponen técnicas de mapeo, no existen líneas claras para lograr una integración efectiva entre diversión y aprendizaje [25], ni métodos formales para identificar la eficacia de la integración durante la fase de diseño [29]. Al mismo tiempo este autor destaca que es crucial una selección adecuada del proceso de aprendizaje, para que interacción con la mecánica de juego sea natural.

Según [27] estos nuevos marcos deben incorporar los procesos de aprendizaje mencionados por [29], al tiempo que deben definir la forma de describir e incluir los retos educativos. Y establecer técnicas de diseño como lenguajes para modelar o herramientas gráficas que soporten la descripción de la narrativa.

Los SG deben contar con una narrativa y estrategias de aprendizaje desafiante [25] sin embargo, los marcos actuales le dan escasa importancia a la capacidad que tiene la narrativa e historia de captar la atención y mantener la motivación [27].

En síntesis estos nuevos marcos deben conseguir un equilibrio entre diversión y aprendizaje al mezclar el contenido de instrucción, los mecánica de juego y los mecanismos de retro-alimentación, con el fin de mantener el compromiso y la motivación del jugador [32].

Estos nuevos marcos implican la propuesta de nuevas estrategias o modelos de diseño, salvo [23], [28] y [34] el resto de los autores proponen un modelo propio, desarrollado para casos específicos, los cuales no están generalizados.

Otro enfoque es el de [24], que parte de la existencia de las mecánicas de juego (GM) e incorpora el concepto de mecánica de aprendizaje (LM), que define como el conjunto de tareas, actividades, objetivos o relaciones que el alumno/jugador debe cumplimentar para adquirir un conocimiento o habilidad, sus instrucciones de uso y la relación con el contexto, las cuales en conjunto constituyen una estrategia de aprendizaje.

Las GM y las LM deben interactuar, para lo cual [32] y [36] introducen en su modelo LM-GM, el concepto de mapeo, aunque reconocen que no existe un marco o modelo para descubrir y entender cómo se relaciona la mecánica del juego con los objetivos educativos. El tríptico LM, GM y mapeo, constituyen la mecánica de SG o SGM, la cual refleja las relaciones complejas entre pedagogía, aprendizaje, entretenimiento y diversión. Esta visión es compartida por [26], [30] y [32], que incluyen el modelo de especificación LM-GM dentro de sus modelos de diseño.

Observando el conjunto de propuestas también se advierte la necesidad de determinar en una misma fase, los aspectos de diversión, las estrategias de aprendizaje y su interacción, este hecho sumado a la naturaleza iterativa de los modelos participativos, torna difusa la frontera entre especificación y diseño, al punto que se las puede considerar como una sola fase.

Q3: ¿Qué características o atributos relevantes se toman en cuenta en los requerimientos educativos para un SG?

Aunque no hay mención directa sobre los atributos de los requerimientos educativos, de su lectura se puede inferir que se focaliza en:

- a) Estrategias de aprendizaje: descripción de los contenidos y de la forma en que el alumno se capacitará mientras juega, su definición requiere el trabajo compartido de expertos docentes, expertos del dominio [24] y del juego, abarcando todas las teorías educativas [29].
- b) Feedback: necesario para que el alumno comprenda el concepto envuelto en el contenido y las razones del éxito ó el fracaso de las decisiones que adopta [29] [35].
- c) Repetición: posibilidad repetición ilimitada de las sesiones de juego, como soporte del aprendizaje [32] y [35].
- a) Evaluación: mecanismos que permitan evaluar el progreso de los alumnos mediante el uso de analíticas del aprendizaje y considerando los principios educativos [31].

Q4: ¿Cuáles son las metodologías o procesos utilizados en la obtención de requerimientos educativos para la especificación de un SG?

Dado que la tarea es multidisciplinaria [34] propone la identificación de tareas que impliquen adquirir un conocimiento o habilidad. En ese camino utiliza entrevistas, diagrama de tareas, auditoría del conocimiento y entrevistas de simulación. Con el objeto de obtener una descripción detallada de los componentes, los desafíos, el conocimiento subyacente, los procesos de pensamiento y la estructura de objetivos.

Por su parte [28] proponen un esquema de reuniones plenarias de stakeholders (incluidos los jugadores/alumnos), donde se debaten un conjunto de escenarios planteados mediante una herramienta participativa que aporta una lista de puntos a considerar, al tiempo que ayuda a encontrar respuestas y aporta trazabilidad de las decisiones adoptadas

Con respecto al balance juego/aprendizaje, se propone el uso de dos enfoques: el endógeno (contenido y gameplay juntos) y exógeno (primero un modelo de contenido y luego el gameplay). En el caso exógeno se construye un modelo cognitivo y sobre este un sistema experto para utilizarlo como base en la construcción de las mecánicas de juego. Mientras que en el enfoque exógeno se utiliza un motor de juego donde se va construyendo una simulación del dominio, sobre la cual se construye el modelo de juego [32].

Por último cabe recordar que los modelos de diseño participativos son iterativos y por ende el proceso de especificación y sus técnicas han de ser ejecutadas en repetidas ocasiones y sus resultados validados siguiendo criterios de eficacia [29].

4 Conclusiones

Los documentos centran su análisis en los procesos de desarrollo en general y

hacen referencias indirectas a la especificación. Por ello en respuesta a Q1 los autores destacan que la especificación debe ser iterativa como los procesos de desarrollo y al igual que estos colaborativas, multidisciplinarias y centradas en el usuario.

Todos en que esta etapa debe proveernos los requerimientos para lograr un SG entretenido y útil para el aprendizaje, garantizando un equilibrio entre estas dos facetas. En este sentido plantean diferentes métodos de modelado capaces de representar los requerimientos y sus interacciones. En la pregunta Q2 acuerdan que existen de dos tipos de requerimientos específicos; los relacionados con el juego (que aportan jugabilidad) y los relacionados con el aprendizaje.

En el contexto de Q2 y de acuerdo con Q1, manifiestan que es preciso determinar qué requerimientos del juego contribuyen a cumplir un requerimiento educativo, para lo cual proponen diferentes tipos de mapeo, tendientes a definir una mecánica de SG (SGM) que una los conceptos de mecánica de aprendizaje (LM) y mecánica de Juego (GM).

Para lograr una SGM es preciso determinar cómo, cuándo, dónde y porque el alumno a de adquirir una competencia a través del juego. La pregunta Q3 apunta a esto e indica que la estrategia de aprendizaje, el feedback, la repetición y la evaluación, son los aspectos relevantes a tomar en cuenta.

La pregunta Q4 apunta a detectar las metodologías específicas para la especificación de SG. Como se adelantó en Q1, los trabajos no hacen mención específica a esta fase. Solo algunos mencionan reuniones plenarias de stakeholders, entrevistas y cuestionarios. Lo más interesante es la propuesta para lograr el equilibrio entretenimiento/aprendizaje mediante dos enfoques uno endógeno y otro exógeno con el auxilio de prototipos.

Los documentos seleccionados a la luz de las preguntas propuestas, nos dicen que es preciso definir procesos de desarrollo específicos para SG, que sean de naturaleza iterativa, colaborativos, multidisciplinarios y centrados en el usuario.

Con una etapa de especificación que nos permita obtener los requerimientos del juego que describen la máquina del juego (GM) y los requerimientos educativos que describen la máquina de aprendizaje (LM).

Ambos tipos de requerimientos deben relacionarse entre sí para lograr el objetivo de aprendizaje, para ello es preciso mapear unos a otros, lo que constituye la mecánica del SG (SGM) ó LM-GM.

El mapeo o relación a la que hacemos referencia es aquella responsable del equilibrio entre entretenimiento y aprendizaje. Para debemos conocer los atributos de nuestros requerimientos entre los que se destacan en esta revisión; las estrategias de aprendizaje, el feedback, la repetición y la evaluación.

La información relevada no presenta un conjunto de metodologías consistente para lograr el mencionado equilibrio, más bien describe un proceso artesanal, donde lo más destacado es el uso de prototipos como un medio de llegar a dicho equilibrio mediante un proceso de simulación.

Para terminar, como toda revisión sistemática, se remite a los documentos seleccionados y sus conclusiones no pueden ir más allá de las mismas. Trabajos posteriores seguramente han de aportar mayor luz al tema que nos ocupa.

5 Referencias

1. Abt, C. (1970). *Serious games*. The Viking Press. New York, EEUU.
2. Salen, K. y Zimmerman, E. (2004). *Rules of play: Game design fundamentals*. The MIT Press.
3. Crawford, C. (2003). *Chris Crawford on game design*. New Riders, Indianapolis EEUU.
4. Connolly, T. M., Boyle, E. A. y Hainey, T. (2007). Motivations for playing computer games: a comparative analysis. In *Proceedings of the 1st European conference on games-based learning (ECGBL)*. pp. 71-78. Paisley, Escocia, Reino Unido.
5. Csikszentmihalyi, M. (1990). *Flow: The psychology of optimal experience*. New York: Harper and Row.
6. Murray, J. H. (1997). *Hamlet on the Holodeck: The Future of Narrative in Cyberspace*. Simon and Schuster.
7. González Sánchez, J. L. (2010). *Jugabilidad. Caracterización de la experiencia del jugador en videojuegos*. Tesis doctoral. Universidad de Granada, España.
8. Zyda, M. (2005). From visual simulation to virtual reality to games. *Computer*, 38(9), 25-32. IEEE.
9. Sawyer, B. y Smith, P. (2008). *serious games taxonomy*. serious game Summit 2008. San Francisco, USA.
10. Urquidi, M. & Tamarit Aznar, C. (2015). Juegos serios como instrumento facilitador del aprendizaje: evidencia empírica. *Revista Opción*, 31(3), 1201 - 1220. Universidad de Zulia, Venezuela.
11. Jenkins, H. (2009). *Confronting the challenges of participatory culture: Mediaeducation for the 21st century*. White Paper. MacArthur Foundation.
12. Prensky, M. (2001). *Digital game-based learning*. New York: McGraw-Hill.
13. González Tardón, C. (2006). *Emociones y Videojuegos*. En III Congreso Online-Observatorio para la cibernsiedad. Conocimiento Abierto. Sociedad libre.
14. González, C. y Blanco, F. (2008). Emociones con videojuegos: incrementando la motivación para el aprendizaje. *Revista electrónica teoría de la educación. Educación y cultura en la sociedad de la información*, 9, 69- 92.
15. Portnow, J. (2008). The power of tangential learning [Mensaje en un blog].. Recuperado de <http://www.edgeonline.com/blogs/the-power-tangential-learning>.
16. Hecker, C. (2010). *Achievements considered harmful?*. Conference presented at Game Developer's Conference. San Francisco, California.
17. Blair, L. (2012). *The use of video game achievements to enhance player performance, self-efficacy, and motivation*. Doctoral Dissertation. University of Central Florida.
18. Carcía Mundo, L., Vargas Enriquez, J. , Genero, M. y Piattini, M. (2014). ¿Contribuye el Uso de Juegos Serios a Mejorar el Aprendizaje en el Área de la Informática?. En *Actas de las Jornadas de Enseñanza Universitaria de la Informática (JENUI 2014)*. Oviedo.
19. Bossolasco, M., Enrico, R. Casanova, A. y Enrico, E. (2015). Kokori, un serious game. La perspectiva de los estudiantes ante una propuesta de aprendizaje innovadora. *Revista de Educación a Distancia*, 45.
20. Fanfarelli, J. & McDaniel, R. (2015) Individual Differences in Digital Badging: Do Learner Characteristics Matter?. *Journal of Educational Technology Systems*, 43(4), 403–428.
21. Rocha V.R., Isotani S. y Biitencourt I. (2015). Análise, Projeto, Desenvolvimento e Avaliação de Jogos Sérios e Afins: uma revisão de desafios e oportunidades. IV Simpósio Brasileiro de Informática na Educação e X Conferencia Latino Americana de Objetos e Tecnologias de Aprendizagem , 26(1), pp. 692. Maceió, Alagoas, Brasil.
22. Kitchenham, B. (2004). *Procedures for Performing Systematic Reviews*. Joint Technical Report, Keele University TR/SE-0401 and NICTA 0400011T.1

23. Ampatzidou, C., & Gugerell, K. (2018). Participatory game prototyping – balancing domain content and playability in a serious game design for the energy transition. *CoDesign*, 0(0), 1-16. <https://doi.org/10.1080/15710882.2018.1504084>
 24. Arnab, S., Lim, T., Carvalho, M. B., Bellotti, F., Freitas, S. de, Louchart, S., ... Gloria, A. D. (2015). Mapping learning and game mechanics for serious games analysis. *British Journal of Educational Technology*, 46(2), 391-411. <https://doi.org/10.1111/bjet.12113>
 25. Broeckhoven, F. V., Vlieghe, J., & Troyer, O. D. (2015). Mapping between Pedagogical Design Strategies and serious game Narratives. 2015 7th International Conference on Games and Virtual Worlds for Serious Applications (VS-GAMES), 1-8. <https://doi.org/10.1109/VS-GAMES.2015.7295780>
 26. Carvalho, M. B., Bellotti, F., Berta, R., De Gloria, A., Sedano, C. I., Hauge, J. B., ... Rauterberg, M. (2015). An activity theory-based model for serious games analysis and conceptual design. *Computers & Education*, 87, 166-181. <https://doi.org/10.1016/j.compedu.2015.03.023>
 27. De Lope, R. P., López Arcos, J. R., Medina-Medina, N., Paderewski, P., & Gutiérrez-Vela, F. L. (2017). Design methodology for educational games based on graphical notations: Designing Urano. *Entertainment Computing*, 18, 1-14. <https://doi.org/10.1016/j.entcom.2016.08.005>
 28. De Troyer, O., & Janssens, E. (2014). Supporting the requirement analysis phase for the development of serious games for children. *Special Issue: Learning from Failures in Game Design for Children*, 2(2), 76-84. <https://doi.org/10.1016/j.ijcci.2014.05.001>
 29. Gunter, G. A., Kenny, R. F., & Vick, E. H. (2008). Taking educational games seriously: using the RETAIN model to design endogenous fantasy into standalone educational games. *Educational Technology Research and Development*, 56(5), 511-537. <https://doi.org/10.1007/s11423-007-9073-2>
 30. Marchiori, E. J., del Blanco, Á., Torrente, J., Martínez-Ortiz, I., & Fernández-Manjón, B. (2011). A visual language for the creation of narrative educational games. *Journal of Visual Languages & Computing*, 22(6), 443-452. <https://doi.org/10.1016/j.jvlc.2011.09.001>
 31. Mariais, C., Michau, F., & Pernin, J.-P. (2011). A Description Grid to Support the Design of Learning Role-Play Games. *Simulation & Gaming*, 43(1), 23-33. <https://doi.org/10.1177/1046878110390764>
 32. Mestadi, W., Nafil, K., Touahni, R., & Messoussi, R. (2018). An Assessment of serious games Technology: Toward an Architecture for serious games Design. *International Journal of Computer Games Technology*. <https://doi.org/10.1155/2018/9834565>
 33. Nadolski, R. J., Hummel, H. G. K., van den Brink, H. J., Hoefakker, R. E., Slootmaker, A., Kurvers, H. J., & Storm, J. (2007). EMERGO: A methodology and toolkit for developing serious games in higher education. *Simulation & Gaming*, 39(3), 338-352. <https://doi.org/10.1177/1046878108319278>
 34. Seager, W., Ruskov, M., Sasse, M. A., & Oliveira, M. (2011). Eliciting and modelling expertise for serious games in project management. *serious games Development and Applications*, 2(2), 75-80. <https://doi.org/10.1016/j.entcom.2011.01.002>
 35. Slimani, A., Yedri, O. B., Elouaai, F., & Bouhorma, M. (2016). Towards a design approach for serious games. *International Journal of Knowledge and Learning*, 11(1), 58-81. <https://doi.org/10.1504/IJKL.2016.078649>
 36. Strzalkowski, T., & Symborski, C. (2016). Lessons Learned About serious game Design and Development. *Games and Culture*, 12(3), 292-298. <https://doi.org/10.1177/1555412016673524>
- T.F., Waterman, M.S.: Identification of Common Molecular Subsequences. *J. Mol. Biol.* 147, 195--197 (1981)

Scrum como Herramienta Metodológica para el Aprendizaje de la Programación

Nicolás Tymkiw, Juan Manuel Bournissen y Marisa Cecilia Tumino
Universidad Adventista del Plata, Libertador San Martín, Entre Ríos, Argentina
{Nicolas.tymkiw, juan.bournissen, marisa.tumino}@uap.edu.ar

Resumen. El objetivo de este estudio fue identificar el impacto de la utilización de la metodología de desarrollo de software Scrum, como una técnica que permite reforzar las estrategias de enseñanza y aprendizaje utilizadas en la enseñanza de nivel superior en la asignatura de Programación II, en el aprendizaje de los estudiantes de una universidad privada.

El proyecto ha proporcionado datos suficientes que permiten estimar que la aplicación de la metodología Scrum tiene un impacto positivo como refuerzo en el proceso de aprendizaje de los estudiantes, ofreciendo una alternativa innovadora a las estrategias tradicionales utilizadas por las cátedras donde se realizó el experimento.

Palabras clave: Scrum en el aula, Aprendizaje de programación, Metodología, Impacto

1. Introducción

En la actualidad, dentro de las carreras informáticas, la programación es una asignatura fundamental, debido a que es una de las materias base que determina el progreso del alumno en la carrera y luego, una vez recibido, determina su desempeño profesional.

Investigadores como Costelloe [1] demuestran que los estudiantes no logran conceptualizar todos los temas de programación enseñados, siendo este uno de los puntos principales por el cual dejan la carrera. Otro aspecto que enfatizan los autores es que si logran pasar la materia más adelante, el rendimiento en asignaturas correlativas es muy bajo.

Esta metodología requiere de un proceso evaluativo, en el que se reveen las practicas docentes, viendo los obstáculos y dificultades que se presentan en los estudiantes.

Barberis, y Del Moral Sachetti [2], señalaron que desde que se implementó esta metodología en el año 2014, la tasa de deserción bajó un 23% con respecto a los años anteriores, mientras que el rendimiento académico aumentó un 27%.

Perazo [3] afirma que el porcentaje de la deserción en las carreras informáticas se aproxima al 80%, teniendo en cuenta los datos suministrados por la Secretaria de Políticas Universitarias del Ministerio de Educación. Esta problemática puede deberse a que, en su mayoría, es una decisión que toman adolescentes entre 17 y 20 años que desconocen las incumbencias de la profesión elegida.

1.2. Beneficios de Scrum

Albaladejo [4] menciona que es necesario saber que SCRUM cuenta con beneficios, fundamentos y requisitos, entre los que cabe mencionar los que se detallan a continuación.

- Entrega de resultados que debe ser dentro de un plazo corto, mensual o quincenal, en el que los requisitos que son prioridad ya estén completados.
- Facilita la gestión de las expectativas del cliente mediante dos requerimientos, en el primero el cliente presenta una lista de requisitos, indicando su prioridad; en el segundo, al finalizar la iteración, el equipo le muestra al cliente los resultados de los requisitos que han completado.
- Utilización de las funciones más importantes antes de que el proyecto termine. El cliente puede empezar a recuperar su inversión con anterioridad, lo que se consigue mediante la priorización de requisitos por coste y valor. El Retorno de Inversión (ROI) es un beneficio que el cliente maximiza en el proyecto, si ve que le beneficio es menor al coste del desarrollo el cliente puede finalizar el proyecto.
- Proporciona flexibilidad y adaptación, donde el cliente va dirigiendo el proyecto en función de sus nuevas prioridades, lo que se consigue replanificando los requisitos y prioridades en el inicio de cada iteración.
- Desde la primera iteración el equipo debe definir los riesgos para poder mitigarlos, si hay que equivocarse, es mejor hacerlo lo antes posible. Para lidiar con este beneficio es necesario un desarrollo iterativo e incremental, de esta manera no se deja para el final ninguna actividad riesgosa vinculada con la entrega de requisitos.
- Productividad y calidad es otro de los beneficios que proporciona SCRUM, el equipo mejora y simplifica su manera de trabajar, para esto es necesario que en cada iteración el equipo realice una retrospectiva para analizar su método de trabajo, como así también se necesita una comunicación continua del equipo.
- Alimentación entre cliente y equipo, para lo que se necesita que ambos sean un equipo, que trabajen juntos en los requisitos, detalles y análisis de los resultados obtenidos.
- Motivación del equipo, para lo que es importante dejar que las personas puedan usar su creatividad para resolver problemas y decidir cómo organizar su trabajo, por lo que es necesario un equipo autogestionado que se comprometa a completar todos los requisitos determinados en una iteración.

1.3. Componentes de Scrum

La metodología Scrum se encuentra dividida en fases y roles, a las fases las podemos identificar como reuniones, también conocidas como Sprint.

Reuniones. Gallego [5] divide las reuniones en tres fases: (a) en la primera fase se desarrolla una planificación del Backlog, dentro del cual se definen las prioridades de los requisitos y la planificación del primer sprint 0, con los objetivos y el trabajo que se necesita completar en esa iteración. Durante esta fase se obtiene la lista de tareas a realizar; (b) para la segunda fase se genera un seguimiento del sprint. Es una fase de reuniones diarias en las que se evalúa el lugar del proyecto en el que se encuentra el equipo, los avances desde la última reunión, los trabajos que se realizarán hasta la siguiente reunión y se analiza la forma de solucionar los inconvenientes que surgieron durante el desarrollo y (c) en la tercera fase, cuando concluye el Sprint, se aplica una revisión del incremento que se ha generado, se presentan los resultados finales y una versión de ayuda para mejorar la realimentación del cliente..

Roles. Los roles se dividen en dos grupos: (a) los que están comprometidos con el proyecto y proceso de Scrum y (b) los que no son parte del proceso pero que se necesitan para la realimentación de la salida de los procesos y el planeamiento de los sprint. En el primer grupo se pueden identificar a los integrantes en tres áreas:

Product Owner: es el encargado de tomar las decisiones del proyecto y es el conocedor del negocio del cliente y su visión sobre el producto. Es responsable de recolectar los requisitos del cliente y ordenarlos por prioridad.

Scrum Master: comprueba que la metodología funciona, para lo que debe sobrepasar los inconvenientes que obstaculicen la fluidez del proceso e interactuar con el cliente y los gestores.

Equipo de Desarrollo: conformado por un grupo pequeño de personas y tienen la autoridad para organizar y tomar decisiones para conseguir su objetivo.

Dentro del segundo grupo se pueden encontrar los siguientes roles:

Usuarios: son los destinatarios finales del producto.

Stakeholders: son los que participan de las revisiones del sprint.

Managers: encargados de la toma de decisiones finales, participan en la selección de los objetivos y los requisitos del proyecto..

1.4. Scrum en el aula

Kuz, Falco y Giandini [6] señalan que, dentro del ámbito educativo, los estudiantes necesitan desarrollar capacidades y aptitudes que posteriormente le servirán en su carrera profesional. Las competencias generales requieren de métodos de aprendizaje activos que posibiliten el desarrollo de la capacidad de organización, planificación, liderazgo, evaluación, autoevaluación y trabajo en equipo, entre otros.

Teniendo en cuenta lo mencionado anteriormente, Scrum favorece la creación de un ambiente propicio para que los estudiantes puedan experimentar un aprendizaje creativo, enriquecedor y confiable.

Miller [7] explica que un “aula ágil” precisa de la integración de cinco elementos fundamentales para alcanzar objetivos específicos: (a) clase visible, (b) ritmo de

aprendizaje, (c) colaboración, (d) capacitación y (e) el camino para evolucionar un aula hacia la autoorganización.

1.5. Evaluación de Impacto

Tejada Fernández [8] explica que la evaluación del impacto intenta responder la siguiente pregunta. ¿Cuáles son los efectos que la formación produce en un determinado ámbito? Se intenta averiguar en qué medida los aprendizajes adquiridos en las acciones formativas son útiles para mejorar el desempeño.

El proceso de evaluación de acciones formativas cuenta con tres finalidades esenciales: (a) diagnóstica, que permite conocer el desarrollo del proceso de enseñanza y aprendizaje, (b) formativa, que permite valorar una acción educativa durante su desarrollo y (c) sumativa o de productos, que permite conocer si los objetivos se consiguieron o no, los cambios que se produjeron y la toma de decisiones con respecto a las actividades planteadas.

1.6. Scrum en programación

Varios autores muestran que la utilización de metodologías ágiles, como Scrum, pueden ser altamente eficientes al desarrollar un software. A continuación, se presentan investigaciones que avalan lo expuesto.

Chávez Andrade [9] indagó sobre la estandarización de los procesos de desarrollo utilizando buenas prácticas de programación y Scrum en departamentos de TI. Tuvo como objetivo incrementar el grado de funcionalidad de un producto de software. Afirma que la carente aplicación metodológica conlleva a obtener un producto de baja calidad. Explica que la aplicación de Scrum logra la integración del usuario al proceso de desarrollo, así como genera un ambiente de compromiso en el equipo de desarrollo con el fin de obtener un producto de software de calidad, que cumpla con las aspiraciones del usuario. Se identificó la mejora de funcionalidad como atributo de la calidad por parte de los implicados en el proyecto.

Hervás Lucas [10] sostiene que, para alcanzar un aprendizaje eficiente en un lenguaje de programación, se emplean técnicas de integración continua. La investigación del autor fue realizada con alumnos del tercer curso de ingeniería informática. El aprendizaje con este método se basa en el desarrollo guiado y supervisado de prototipos de complejidad incremental. La conclusión a la que llega el autor es que el método prototipado ágil fomenta el aprendizaje y el desarrollo de competencias clave como el trabajo en equipo, la autodidáctica y el análisis crítico.

2. Metodología

El presente estudio es de tipo explicativo y cuasiexperimental. Explicativo porque busca determinar el nivel de impacto en dos grupos que utilizaron una metodología de estudio diferente, y cuasiexperimental porque para la experiencia se tomaron los grupos tal como estaban conformados. El diseño metodológico estuvo basado en un estudio realizado en la Universidad Nacional de Salta por Barberis y otros, [2].

La variable independiente en esta investigación fue la metodología utilizada y las variables dependientes: (a) el nivel de impacto en el aprendizaje de los estudiantes y (b) las calificaciones promedio cuatrimestrales.

2.1. Implementación

Durante los últimos años se ha identificado que en la asignatura de Programación II existía un alto porcentaje de alumnos que perdían la materia y otro porcentaje, menor pero significativo, de alumnos que decidían abandonarla.

En base a esta problemática se decidió desarrollar un plan de trabajo con el fin de que los alumnos puedan obtener todos los conocimientos que estipula la asignatura y que la implementación de estos se vea reflejado en las calificaciones.

Este plan de trabajo consistió en aplicar la metodología Scrum para el aprendizaje de la programación. Se dividió al curso en grupos de tres, y se les explico a los estudiantes la modalidad de trabajo del año. Al dividir al curso en grupos lo que se buscó fue que a través del trabajo en grupo se pueda potenciar las habilidades de cada uno de los estudiantes.

Durante el cursado de la asignatura, a medida que se incrementaban los conceptos, se les asignaron trabajos prácticos. Los estudiantes debían cumplimentarlos en un plazo de tres semanas, atendiendo a una serie de requisitos:

- a. La composición del grupo durante un trabajo practico debería contar con un líder de Scrum, mientras que los demás estudiantes asumirían el rol de equipo de desarrollo.
- b. Cada uno de los estudiantes tendría que pasar por el rol de líder de Scrum.
- c. El equipo debería realizar al menos un sprint (reunión de equipo) por semana, donde mediante un brainstorming podrían elegir como desarrollar los requisitos propuestos o solucionar los problemas que fueron apareciendo durante el transcurso del desarrollo.
- d. Cada semana el equipo se reuniría con el profesor de la cátedra, quien asumiría el rol de cliente e iría revisando si el sistema cumplía con los requisitos que se habían planteado e incrementando de acuerdo con los nuevos conceptos vistos en clase.
- e. En la tercera semana los equipos deberían exponer a toda la clase el práctico que se les había asignado. En esta exposición debían describir la modalidad que adoptaron para desarrollarlo, los problemas con que se encontraron y las soluciones encontraron para avanzar. Por su parte el líder Scrum debía relatar la forma en que se realizaron los sprints y como fue el trabajo del grupo con el cliente. Por último,

cada uno de los estudiantes del grupo debía mostrar la parte del proyecto que había desarrollado y explicar porque eligió desarrollarlo de esa manera. Esto último serviría para observar si el alumno realmente había podido comprender los conceptos aprendidos en clase.

Además de los trabajos prácticos, las calificaciones se tomaron en base a dos evaluaciones. La primera constó de un examen individual en el que los estudiantes desarrollaron un sistema con los conceptos aprendidos hasta el momento. La segunda evaluación tuvo una temática distinta a la primera en el que el profesor le asignó a cada grupo una problemática distinta y les concedió tres horas para que en equipo elaboraran un sistema que solucionara dicha problemática.

La cursada finalizó con un trabajo final, con un grado de dificultad mayor al resto de los prácticos realizados, en el que se integraban todos los conceptos vistos durante el cuatrimestre. Para calificarlo se les pidió a los equipos que, además de cumplir con cada uno de los requerimientos, le agregaran funcionalidades extras y que se le dedique tiempo al diseño del software.

2.2. Sujetos

La muestra estuvo compuesta por sujetos que cursan su carrera de sistemas de información en una universidad de la provincia de Entre Ríos. Se evaluaron dos grupos de estudiantes:

El primer grupo estuvo conformado por estudiantes de ambos géneros, entre 18 y 25 años, que implementaron la metodología Scrum, en la asignatura de programación II de la carrera de Ingeniería en Sistemas de la Información, durante el segundo cuatrimestre del año 2018.

El segundo grupo se conformó por estudiantes, de género masculino, entre 22 y 26 años, que habían cursado en años anteriores al 2018 la asignatura de programación II bajo la implementación de metodologías distintas a la metodología del primer grupo.

El profesor titular y el profesor adjunto de la cátedra de programación II fue el mismo en ambos grupos. Los docentes, además de enseñar los conceptos estipulados, asumieron el rol de clientes para cada uno de los trabajos prácticos asignados.

2.3. Instrumentos

Se midió el impacto mediante el test de Nivel de Impacto de Tumino y Bournissen [11]. El test presenta una estructura compuesta por 8 variables vinculadas al aprendizaje y 10 variables relacionadas con estrategias de aprendizaje. La escala utilizada en las dos dimensiones es de intervalo desde 1 = muy en desacuerdo, 2 = en desacuerdo, 3 = neutro, 4 = de acuerdo y 5 = muy de acuerdo.

Esta encuesta fue aplicada a los estudiantes que cursaron la asignatura de programación II con la metodología Scrum como estrategia de aprendizaje y a estudiantes que cursaron con metodologías previas.

2.4. Procedimientos para recolección de datos

En primer lugar, se realizó una reunión con la dirección de la carrera y el profesor titular de la cátedra para informar sobre el propósito y las características de la investigación. En dicha reunión se ofreció un modelo a seguir que consta de un trabajo experimental en el que se evaluaría el aprendizaje de los estudiantes, con la finalidad de disminuir la tasa de deserción o pérdida de la asignatura respecto de años anteriores.

El docente sugirió el modelo de armado de equipos, a partir de una evaluación inicial niveladora, a fin de asegurar el beneficio de todos los estudiantes.

Durante los meses de cursado se asistió constantemente a las clases prácticas para poder compartir observaciones acerca del trabajo realizado por los estudiantes y constatarlo en una bitácora de anotaciones.

Finalizada la cursada se les administró a los estudiantes los test para recolectar los datos de nivel de impacto en su aprendizaje. También se le aplicó el test a estudiantes que años anteriores habían cursado la asignatura con otras metodologías.

Por último, se realizó una reunión con los profesores para conocer sus percepciones en cuanto a la metodología aplicada y se obtuvieron las calificaciones de quienes respondieron el test de nivel de impacto. Asimismo, se obtuvieron los datos de cursado de las cohortes de los últimos diez años a fin de analizar comparativamente los resultados.

2.5. Procedimientos para el análisis estadístico de los datos

Una vez recolectados los datos, se utilizó el programa IBM SPSS Statistics versión 23.0 para el análisis descriptivo y comparativo de los datos.

Se trabajó con dos pruebas no paramétricas mediante las que se compararon el nivel de impacto en el aprendizaje y la calificación promedio entre los estudiantes que utilizaron la metodología SCRUM, como estrategia de enseñanza y aprendizaje, y los que aprendieron mediante otras estrategias anteriormente.

3. Resultados

A continuación, se muestran los resultados obtenidos a partir de las variables presentadas en la investigación.

El objetivo principal fue identificar si los estudiantes que trabajaron con la metodología Scrum, podrían mostrar un mejor nivel de impacto en el aprendizaje percibido de la

asignatura, lo que se reflejaría tanto en las calificaciones promedio, como en el impacto de la implementación de la metodología en su aprendizaje.

3.1. Datos de cursado

Mediante el análisis realizado con los datos de cursado de la asignatura obtenidos de las últimas diez cohortes, se pudo observar que en el año en que se realizó la experiencia, el 66,67% de los estudiantes promocionó la asignatura mientras que el 33,3% restante la regularizó. De esta manera ningún estudiante perdió o abandonó la asignatura.

De los últimos diez años de cursado de la asignatura, los nueve años anteriores se utilizó una misma metodología y en el último año se implementó la metodología Scrum. Se observa que tan solo en este último año ningún alumno perdió o abandonó la cursada, tal como se muestra en la Tabla 1.

Tabla 1: Datos de las calificaciones de los últimos diez años.

Año	Promoción Directa	Promoción Indirecta	Pierde la Materia	Abandona
2009	30,0%	30,0%	30,0%	10,0%
2010	11,8%	41,2%	35,3%	11,8%
2011	33,3%	40,0%	26,7%	0%
2012	20,8%	37,5%	33,3%	8,3%
2013	10,0%	50,0%	40,0%	0%
2014	29,4%	23,5%	41,2%	5,9%
2015	28,6%	28,6%	42,9%	0,0%
2016	9%	18,2%	54,6%	18,2%
2017	28,5%	52,4%	19,1%	0%
2018	66,7%	33,3%	0%	0%

3.2. Desempeño de los equipos

Los resultados acerca del desempeño de los equipos se pudieron observar mediante la bitácora de observaciones. En esta bitácora se plasmó el trabajo de cada uno de los equipos durante la cursada de la asignatura de programación II, utilizando la metodología SCRUM como estrategia de enseñanza y aprendizaje. Se pudo observar que los equipos mantuvieron un alto desempeño, buena organización, y concluyeron con los proyectos en el tiempo estipulado, en la medida en que cumplieron con las pautas de trabajo de la metodología. También se observó que desarrollar proyectos, dentro de un ámbito de equipo como exige la metodología Scrum, generó un ambiente de compromiso y potenció la competitividad de los estudiantes en cuanto a los requisitos que debían cumplimentar en cada uno de los proyectos. Otro resultado positivo que arrojó el trabajo con la metodología Scrum fue que les permitió a los estudiantes de segundo año de la carrera de Ingeniería en Sistemas, defender los proyectos oralmente y adquirir un lenguaje técnico apropiado.

3.3. Pruebas de hipótesis

A continuación, se presentan los resultados concernientes a las hipótesis planteadas durante la investigación.

3.3.1. Primera hipótesis

H1: Existe diferencia estadística significativa de rangos promedios de rendimiento académico, en términos de calificación promedio, obtenido en programación II entre los estudiantes que utilizan la estrategia de enseñanza Scrum y los que cursaron con las metodologías utilizadas previamente.

Esta hipótesis se probó mediante las pruebas no paramétricas para dos muestras independientes tomando a la calificación final como variable de prueba y a la metodología como variable de agrupación. En base a esos datos se observó que existe una diferencia estadísticamente significativa de rango promedio de calificación promedio entre los dos grupos ($p < 0,05$). Se encontró un mayor rango promedio de calificación promedio en los estudiantes que trabajaron con la metodología Scrum (RP = 9.93), m. Mientras que los estudiantes que trabajaron con otras metodologías obtuvieron un menor rango promedio (RP = 5.07).

3.3.2. Segunda hipótesis

H2: Existe diferencia estadística significativa en el nivel de impacto, en el aprendizaje percibido, obtenido en programación II entre los estudiantes que utilizan la metodología SCRUM como estrategia de enseñanza y aprendizaje y los que cursaron con las estrategias utilizadas previamente.

En esta hipótesis se analizaron los datos obtenidos de las muestras mediante las pruebas no paramétricas para dos muestras independientes y se encontró un rango promedio en la categoría aprendizaje más elevado (ver Tabla 2) que demostró una diferencia estadísticamente significativa del rango promedio. Se encontró un mejor nivel de impacto en el aprendizaje percibido ($p < 0,05$) y un rango promedio más elevado (RP = 10.07) para los estudiantes que utilizaron la metodología Scrum, mientras que los estudiantes que usaron otras metodologías obtuvieron un rango promedio de 4,93.

En cuanto a la categoría estrategia de aprendizaje percibido del test del Nivel del Impacto, no se obtuvo una diferencia estadísticamente significativa, pero si se obtuvo una diferencia en el rango promedio entre las dos metodologías. Con la metodología Scrum el rango promedio fue mayor (RP = 9.5) que con las otras metodologías (RP = 5.5) como se observa en la Tabla 2.

Tabla 2: Diferencia entre rango promedio de aprendizaje y estrategia-aprendizaje.

Categoría	Metodología	Rango promedio	Suma de rangos
Aprendizaje	Scrum	10,07	70,5
	Otras	4,93	34,5
Estrategia-Aprendizaje	Scrum	9,5	66,5
	Otras	5,5	38,5

4. Discusión y Conclusión

En el presente apartado se exponen la discusión y conclusión de la investigación, como así también se presentan recomendaciones para futuras investigaciones.

4.1. Discusión

Mediante el análisis de los datos obtenidos, se pudo observar que existe una diferencia estadísticamente significativa de rango promedio en el aprendizaje percibido entre los dos grupos estudiados. También existió una diferencia significativa de rango promedio de la calificación final de los alumnos que utilizaron la metodología Scrum, respecto de quienes trabajaron con otras metodologías.

Otro aspecto que se observó fue que en el transcurso de los años 2009 al 2017, existía un predominio en la promoción indirecta, e incluso en la pérdida de la materia, mientras que, en el año 2018, con la implementación de la metodología Scrum, se denotó el predominio de la promoción directa de los estudiantes.

Los estudiantes que tuvieron la posibilidad de trabajar con Scrum mostraron un mayor impacto mediante sus respuestas. Los docentes, que se habían mostrado abiertos ante la propuesta de implementar la nueva metodología, expresaron entusiasmo con los resultados obtenidos y decidieron mantenerla durante los próximos años.

Por otra parte, en las encuestas realizadas a los estudiantes que cursaron con metodologías anteriores se observó, a través de sus respuestas, que no se encontraban muy convencidos con la metodología utilizada. Se estimó que este resultado pudo darse por distintos factores, algunos de ellos pueden haber sido por la falta de comprensión sobre los temas dados en clase, la falta de compromiso con la asignatura, o la falta de comprensión de la metodología aplicada.

Es por esto que el objetivo principal fue medir y mostrar el nivel de impacto que causó la utilización de la metodología Scrum en el aprendizaje percibido y en la estrategia de enseñanza dentro del ámbito académico.

4.2. Conclusión

La utilización de la metodología Scrum en el ámbito académico, y específicamente dentro de las asignaturas de programación, donde los estudiantes pueden aplicarla e integrarla a los proyectos que les sean asignados, resulta una práctica muy satisfactoria ya que causa un impacto positivo en el aprendizaje de los estudiantes.

En la investigación realizada el impacto causado en los alumnos fue positivo tanto en las calificaciones como en el aprendizaje percibido al tener que enfrentarse a las nuevas propuestas metodológicas. Los estudiantes lograron captar los contenidos establecidos por la metodología, tales como el desarrollo de las actividades de forma grupal, los tiempos de entregas, la interpretación de roles y la exposición sobre el trabajo realizado. También se observó mediante la bitácora de observaciones que los estudiantes pudieron trabajar de la mejor manera al momento de establecer las reuniones de grupo, determinar los sprints que iban a realizar para obtener el resultado exigido y poder sobreponerse a las dificultades que se presentaban a medida que desarrollaban el producto.

La implementación de la metodología Scrum facilitó a los docentes el mantenimiento de la dinámica de la clase, consiguiendo que todos los estudiantes pudiesen realizar un seguimiento constante de los contenidos presentados en la asignatura.

Por otra parte, el impacto causado en los estudiantes fue mayor con la metodología Scrum obteniendo un rango promedio de 10,07 y 9,5 en cuanto al aprendizaje percibido y la estrategia de aprendizaje respectivamente.

Con la implementación de la metodología Scrum, se observó mayor compromiso de parte de los estudiantes para cumplir con las directrices de la asignatura, al generar un ambiente de trabajo ideal dentro del aula donde los estudiantes pueden aprovechar los conocimientos que les proporcionan los docentes y sus pares.

5. Referencias

1. Costelloe E. (2004). "Teaching Programming The State of the Art. CRITE. Technical Report". Department of Computing, Institute of Technology Tallaght, Dublin, Ireland.
2. Barberis, Á. R., & Moral Sachetti, L. E. D. (2016, August). Scrum como herramienta metodológica en el entrenamiento cooperativo de la programación: de la teoría a la práctica. In XI Congreso de Tecnología en Educación y Educación en Tecnología (TE&ET 2016), recuperado de: http://sedici.unlp.edu.ar/bitstream/handle/10915/54603/Documento_completo.pdf-PDFA.pdf?sequence=1&isAllowed=y:
3. Perazo C. (25/10/2013) Tecnología. Reporte del Ministerio de Educación. Diario La Nación. Recuperado de: <http://www.lanacion.com.ar/1632045-el-80-de-los-estudiantes-de-carreras-informatica>
4. Albaladejo X. (2008) Fundamentos de scrum. Recuperado de <https://proyectosagiles.org/fundamentos-de-scrum/>
5. Gallego, M. T. (2012). Metodologia scrum. Universitat Oberta de Catalunya.

6. Kuz, A., Falco, M., & Giandini, R. S. (2018). Comprendiendo la aplicabilidad de Scrum en el aula: herramientas y ejemplos. *Revista Iberoamericana de Tecnología en Educación y Educación en Tecnología*, (21), e07-e07.
7. Miller J. (2016, 1 de noviembre), 5 Elements of Agile Classrooms. Recuperado de <http://blog.agileclassrooms.com/2016/11/5-elements-of-agile-classrooms.html>
8. Tejada Fernández, J., & Ferrández Lafuente, E. (2007). La evaluación del impacto de la formación como estrategia de mejora en las organizaciones. *Revista electrónica de investigación educativa*, 9(2), 1-15.
9. Chávez Andrade, J. V. (2019). Estandarización de los procesos de desarrollo de software utilizando buenas prácticas de programación y Scrum como marco de trabajo ágil en departamentos de TI (Master's thesis, Universidad Técnica de Ambato. Facultad de Ingeniería en Sistemas, Electrónica e Industrial. Maestría en Gerencia de Sistemas de Información).
10. Hervás Lucas, R. (2012). Método para el aprendizaje de entornos y lenguajes de programación basado en prototipado ágil. *Jornadas de Enseñanza de la Informática (18es: 2012: Ciudad Real)*.
11. Tumino, M. C. y Bournissen, J. M., (2020). Integration of Information and Communication Technologies (ICT) in the classroom and its impact on students: construction and validation of measurement scales. *Journal of Educational Research and Innovation (IJERI)*, 13, en Prensa.

Transición desde programación basada en bloques a basada en texto: una revisión del campo

Jorge Rodríguez¹, Gerardo Parra¹, Daniel Dolz¹, and Rubén Ramírez²

¹ Grupo de Investigación en Lenguajes e Inteligencia Artificial
Departamento de Teoría de la Computación - Facultad de Informática
Universidad Nacional del Comahue
Buenos Aires 1400, Neuquén, Argentina

² Consejo Provincial de Educación
Ministerio de Educación de la Provincia de Neuquén
Belgrano 1300, Neuquén, Argentina

j.rodrig@fi.uncoma.edu.ar, gparra@fi.uncoma.edu.ar, ddolz@fi.uncoma.edu.ar,
rhubenariel@gmail.com

Resumen El enfoque de programación por bloques demuestra ser una opción efectiva para dar los primeros pasos en el arte de programar. Se espera que en el transcurso de la escuela secundaria los estudiantes logren desarrollar programas usando lenguajes basados en texto. El paso a un entorno de programación tradicional, como Netbeans+Java, suele resultar un proceso frustrante para la mayoría de los estudiantes.

Este artículo presenta una revisión sobre entornos diseñados para favorecer y facilitar los procesos de transición de programación basada en bloques a programación basada en texto.

Estos entornos buscan hacer más accesible la programación tradicional para estudiantes con experiencia previa en ambientes basados en bloques y posiblemente reducir la curva de aprendizaje como también los niveles de frustración.

Además, este artículo presenta un modelo que permite evaluar estos recursos y constituye una herramienta para orientar los procesos de especificación y desarrollo de este tipo de entornos.

Palabras Clave: LENGUAJES DE PROGRAMACIÓN BASADOS EN BLOQUES, LENGUAJES DE PROGRAMACIÓN BASADOS EN TEXTO, ESCUELA SECUNDARIA, EDUCACIÓN EN CIENCIAS DE LA COMPUTACIÓN, TRANSICIÓN BLOQUE A TEXTO.

1. Introducción

La enseñanza de la computación en la escuela secundaria está transitando una profunda transformación. En muchos países se promueve el desarrollo de propuestas curriculares que adopten prácticas y conceptos fundamentales sobre Ciencias de la Computación como contenidos escolares para la educación secundaria [12,15,16].

En este contexto, las prácticas y conceptos relacionados al área de conocimiento Algoritmos y Programación reciben una atención creciente en el campo educativo. En los últimos años se realizan numerosas investigaciones que buscan definir enfoques y construir entornos destinados específicamente a la enseñanza y el aprendizaje de la programación en ambientes formales y no formales.

El enfoque y los entornos de programación basada en bloques demuestran ser una opción efectiva para introducir conceptos y prácticas fundamentales sobre algoritmos y programación a estudiantes sin formación previa en el área de conocimiento [19].

Sin embargo, existen desafíos emergentes para la enseñanza de las ciencias de la computación al momento de plantear una incorporación sostenida, rigurosa y progresiva a lo largo de los años de escolaridad [16].

Se espera que en el transcurso de la educación secundaria los estudiantes logren construir habilidades para desarrollar programas utilizando lenguajes de programación convencionales basados en texto. El paso a un entorno de programación tradicional suele resultar un proceso frustrante para la mayoría de los estudiantes [13].

En este sentido se justifica la necesidad de concretar trabajos de investigación y desarrollo sobre entornos que aborden el problema de la transición y se constituyan en soporte a prácticas pedagógicas coherentes con las necesidades de formación.

En este trabajo se presenta una revisión sobre entornos de programación que favorecen la transición de bloques a texto. Por otra parte, se expone un modelo donde se incluye la descripción de las características y componentes fundamentales de este tipo de entornos y cómo se relacionan con los procesos de transición.

El resto de este documento está organizado de la siguiente manera. En la sección 2, se describen trabajos relacionados que aportan un marco de referencia para el estudio. En la sección 3, se presenta el modelo para los ambientes de transición elaborado en el marco de este trabajo. La sección 4, muestra una revisión y análisis sobre las características de un conjunto de ambientes. Finalmente, en la sección 5, se presentan conclusiones y líneas de trabajos futuros.

2. Trabajos relacionados

Durante los últimos años, el problema de la transición desde la programación basada en bloques a la tradicional, que usa lenguajes basados en texto, está ganando atención como campo de investigación y desarrollo en el ámbito de la Educación en Ciencias de la Computación. Los ambientes desarrollados en este contexto, sus características y estrategias de enseñanza son algunas de las perspectivas en esta área emergente de investigación [11,17,13,18].

Varios estudios plantean que tanto los entornos basados en bloques como los basados en texto tienen beneficios y límites; además describen que existe una brecha importante entre los ambientes que dificulta los procesos de transición [13,14].

Las investigaciones acuerdan en relación a que los entornos de programación basados en bloques aportan ventajas significativas para construir habilidades de pensamiento algorítmico y comprender conceptos fundamentales sobre algoritmos y programación. Al mismo tiempo, establecen que no resultan adecuados para enseñar conceptos abstractos y prácticas complejas [13,18].

Por otro lado, la programación basada en texto constituye el estándar actual para la mayoría de los lenguajes y es un aprendizaje esperado en el contexto de la educación secundaria. Sin embargo, adoptar un lenguaje tradicional como primer contacto con la programación tiene varias limitaciones que causan dificultades y frustración. El hecho de producir código en un ambiente profesional implica una carga cognitiva que frecuentemente distrae de la complejidad intrínseca de la tarea de programación [13].

3. Modelo propuesto

En este trabajo, nos proponemos estudiar entornos que favorezcan y faciliten la transición de programación basada en bloques a programación basada en texto. En tal sentido, se han realizado experiencias satisfactorias con herramientas de la categoría mixta con ambientes del tipo de caja de arena.

Por herramientas de la categoría mixta se hace referencia a las que posibilitan trabajar con bloques visualizando, en forma inmediata, el código producido automáticamente, y viceversa.

En el contexto de este estudio se trabaja en la definición de un modelo que permita orientar el diseño y construcción de ambientes destinados a facilitar los procesos de transición. En este sentido, se busca sostener las ventajas de la representación gráfica y la característica de mapeo directo de las operaciones sobre el escenario con intención de posibilitar una retroalimentación ágil y la detección y corrección de errores. Al mismo tiempo que se mantiene la tarea en el plano de lo concreto se proporciona la flexibilidad, versatilidad y legibilidad del texto [19].

Se trabaja en la especificación de alguna herramienta que permita este paso. Idealmente dicha herramienta debería contener:

- Sandbox: Proponemos que el ámbito de aplicación de la herramienta sea contenido dentro de entornos definidos y aislados del sistema subyacente con límites claros pero con libertad dentro de los mismos.
- Multilenguaje: la herramienta debería proveer la posibilidad de generar e interactuar con más de un lenguaje de programación. Preferentemente, que algunos de ellos sean los mismos que luego serán utilizados en ambientes solo texto para continuar con la profundización de la disciplina.
- Modalidad dual: debería ser posible generar código a partir de bloques, y viceversa, en tiempo real. Las construcciones obtenidas a partir de los bloques, y viceversa, deben ser lo más directas posibles, para que la equivalencia entre ambas estructuras sea clara y no haya dudas que representan al mismo algoritmo de alto nivel.

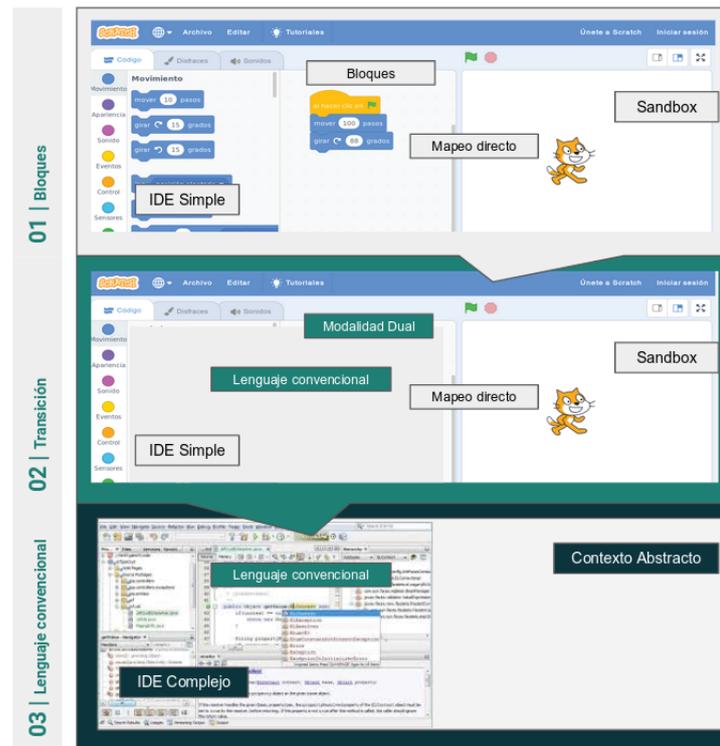


Figura 1. Modelo de ambiente de transición propuesto

- Autoevaluativa: no debería ser necesaria la ayuda de un docente para que el estudiante sepa si su algoritmo resolvió el problema o no. De todas maneras, esto no excluye una instancia de evaluación posterior de eficiencia, elegancia y posibles alternativas de solución diferentes.
- Instalable: esta línea siempre tiene en cuenta estudiantes sin acceso permanente a internet. Por lo tanto, una herramienta puramente *web-enabled* no fácilmente instalable de manera local no estaría al alcance de toda la comunidad educativa.
- Conjunto reducido del lenguaje: especialmente si uno o más de los lenguajes de traducción es de uso comercial como Java, Python, JS, etc, es importante acotar el conjunto de instrucciones para no dispersar con las mismas el objetivo del aprendizaje. Siendo las instrucciones excluidas posibles de incorporar en etapas posteriores con otros ambientes.
- Mapeo directo: Las operaciones programadas se mapean directamente sobre el mundo ofreciendo una retroalimentación instantánea de la actividad en la pantalla o el entorno físico.
- IDE simple: Los entornos de desarrollo profesionales como Netbeans o Eclipse se suelen ver en las aulas como un “entorno de aprendizaje”, aunque no se

han desarrollado con fines educativos. Estas herramientas ofrecen una amplia gama de funciones e interfaz compleja de las cuales la educación informática secundaria, en un primer momento, sólo requiere de una pequeña parte. Se propone trabajar con IDEs simples que integren algunas funciones de alerta visual de errores, reconocimiento de sintaxis y posibilidad de ejecutar el código.

4. Revisión de ambientes para la transición desde la programación basada en bloques a basada en texto

A continuación, se presentan diez ambientes de programación que tienen la posibilidad de facilitar la transición desde la programación basada en bloques a basada en texto.

En el contexto de este estudio se realizó una búsqueda en las bases de datos bibliográficas internacionales en línea, como ACM Digital Library, IEEE Xplore Digital Library y Google Scholar, con intención de identificar los principales ambientes. En un segundo momento, se realizó un análisis sobre las características de los ambientes seleccionados. Finalmente, se elaboró un estudio comparativo sobre los ambientes.

4.1. Ambientes

CodeCombat. CodeCombat[2] es un ambiente basado en juegos diseñado con el objetivo de aprender a programar. Los estudiantes escriben fragmentos de código y, de esa manera, pueden observar cómo los personajes del ambiente reaccionan en tiempo real. El ambiente consiste en un juego estilo RPG (Role Playing Game) estilo *Calabozos y Dragones*, donde en vez de jugar con los cursores o joysticks, las instrucciones se dan en Python y JavaScript mientras la ejecución se observa en un ambiente visual estilo juego de video.

CodinGame. CodinGame[4] es una plataforma online de entrenamiento para programadores y desarrolladores que les permite jugar y resolver problemas mediante la programación. El entorno provee desafíos cada vez más complejos, diseñados con el objetivo de aprender a codificar mejor con una aplicación de programación en línea que admite hasta veinticinco lenguajes de programación diferentes.

PythonTurtle. PythonTurtle[8] intenta proporcionar una de las formas más simples de aprender y enseñar el lenguaje de programación Python. Los estudiantes emplean un *shell* interactivo y utilizan las funciones de Python para mover una tortuga (en inglés, *turtle*) que se muestra en la pantalla.

El *shell* en PythonTurtle es un *shell* completo de Python. Es posible construir bucles (*loops*), definir funciones, crear clases, etc. PythonTurtle es completamente auto-contenido y no requiere tener instalado el lenguaje Python.

Processing. Processing[7] es un ambiente orientado a artistas y estudiantes. Con entorno similar a Wiring, se asemeja a lenguajes como Java y C++. Su simplicidad reside en que a partir de librerías y funciones disponibles se puede crear aplicaciones audiovisuales y juegos interactivos en muy poco tiempo.

Greenfoot. Greenfoot[5], es un entorno de desarrollo interactivo diseñado específicamente con propósitos educativos. Permite, en una primera instancia, la manipulación simple de objetos sobre escenarios bidimensionales. En un segundo momento, se opera el editor de código Java, donde se codifica el comportamiento de los actores reutilizando las librerías.

Pencil. Pencil Code[6] es un sitio de programación colaborativa para diseñar gráficos, reproducir música y crear juegos. Además, es posible experimentar con funciones matemáticas, geometría, páginas web, simulaciones y algoritmos. Los programas están disponibles para que todos puedan ver y remixar.

El sitio permite aprender y experimentar con lenguajes de programación profesionales usando un editor que admite trabajar en bloques o mediante texto. El lenguaje principal es CoffeeScript.

Monkey. CodeMonkey[3] es un juego educativo en el que los estudiantes aprenden a programar en un lenguaje real de programación, sin que se requiera experiencia previa. Es un juego *online* que enseña programación por medio del lenguaje CoffeeScript. El ambiente está diseñado para niños desde los 8 años y permite introducir, de manera amena, conceptos básicos de programación como: variables, operadores lógicos, sentencias condicionales, sentencias repetitivas, argumentos, llamadas a funciones, entre otros.

Tynker. Tynker[10] es un entorno de programación híbrido, basado en bloques, Java y Python, para diferentes edades. Proporciona unos grados más de tangibilidad a la propuesta de enseñanza de las Ciencias de la Computación al permitir programar dispositivos como drones y robots. Potencia en los estudiantes el empleo de modularización y abstracción de alto nivel.

BrickLayer. BrickLayer[1] es un ambiente que permite programar manipulando bloques estilo Lego y armando objetos complejos a partir de los mismos. En este ambiente, los programas son escritos en un lenguaje de programación funcional, denominado SML.

RoboMind. RoboMind[9] es un entorno de programación educativa con un lenguaje de scripting similar a JavaScript. Presenta una metáfora de un mundo cuadrículado donde los estudiantes interactúan con un robot simulado.

4.2. Estudio comparativo sobre los ambientes

En esta sección se presenta un análisis de carácter comparativo sobre las características observadas en los ambientes revisados. El Cuadro 1 muestra una síntesis de las características presentes en cada ambiente.

Característica	Ambiente									
	01	02	03	04	05	06	07	08	09	10
Sandbox			Sí	Sí	Sí	Sí		Sí		Sí
Lenguaje	PJs	PJJs+	P	PJJs	J	Pr	Pr	PJ	Pr	Pr
Modalidad dual						Sí		Sí		
Autoevaluativa	Sí	Sí					Sí	Sí	Sí	
Instalable (INS)/On line (OLn)	OLn	OLn	INS	INS	INS	OLn	OLn	OLn	OLn	INS
Conjunto reducido del Lenguaje	Sí		Sí			Sí	Sí	Sí	Sí	Sí
Mapeo Directo	Sí	Sí	Sí	Sí	Sí	Sí	Sí	Sí	Sí	Sí
IDE Simple	Sí		Sí	Sí	Sí	Sí	Sí	Sí	Sí	Sí

Cuadro 1. Características observadas en los ambientes estudiados. 01: CodeCombat, 02: CodinGame, 03: PythonTurtle, 04: Processing, 05: Greenfoot, 06: Pencil, 07: CodeMonkey, 08: Tynker, 09: BrickLayer y 10: RoboMind. Lenguajes: P: Python, J: Java, Js: JavaScript, Pr: Lenguaje propio del ambiente y +: otros lenguajes

La característica de caja de arena, que ofrece un espacio libre y al mismo tiempo contenido de trabajo, está presente en todos los ambientes explorados excepto en CodeCombat, CodinGame, BrickLayer y CodeMonkey. La mayor parte de los ambientes otorga un grado de libertad amplio a los estudiantes, lo que permite diversificar el tipo y complejidad de los proyectos.

CodeCombat, CodinGame, BrickLayer, CodeMonkey y Tynker cuentan con características que mejoran las posibilidades de autoevaluación y asistencia automática, estos ambientes trabajan con desafíos de complejidad creciente con verificación automática de la solución. CodeCombat, CodinGame, CodeMonkey y Tynker incorporan además características de gamificación trasladando la mecánica de los juegos al ámbito del aprendizaje.

Todos los ambientes incluyen diferentes formas de mapeo directo, esta característica logra sostener la experiencia de aprendizaje en el plano de lo concreto y visible contribuyendo al aplanamiento de la curva de aprendizaje.

CodeCombat, CodinGame, PythonTurtle, Processing y Greenfoot utilizan uno o más lenguajes tradicionales como Python, Java o JavaScript. Esto los pone en mejor posición para transitar hacia entornos de programación reales.

4.3. Experiencia

En el contexto de este estudio, se buscó conocer la percepción de los docentes que se desempeñan en espacios curriculares destinados a la enseñanza de la computación en el escuela secundaria acerca de las posibilidades para favorecer

el proceso de transición que ofrecen dos ambientes específicos: CodeCombat y CodinGame.

La población consultada se compuso por 54 docentes de informática que poseen conocimientos y habilidades previamente adquiridos en el campo disciplinar y didáctico disciplinar. Además, cuentan con experiencia en la enseñanza de la computación en la escuela secundaria.

Se completaron 2 sesiones de 140 minutos en cada caso destinadas a la presentación, reconocimiento y estudio de los ambientes. La actividad buscó que los docentes comprendan el funcionamiento de cada ambiente y construyan una apreciación acerca de sus características y posibilidades en la escuela secundaria.

Cada sesión se organizó en tres etapas: breve presentación del ambiente, estudio utilizando el rol de estudiante y retrospectiva individual acerca de la potencialidad percibida.

Concluidas las dos sesiones de estudio, se consultó sobre la accesibilidad, los motivos de su opinión y las dificultades y utilidades percibidas. Luego, acerca de las posibilidades de participar de situaciones de enseñanza y aprendizaje en contextos reales. Finalmente, se pidió a los participantes que valoren las oportunidades con que cuenta el ambiente para ubicarse como un dispositivo útil para la transición.

La Figura 2 muestra la percepción relativa acerca la accesibilidad de los ambientes. Los docentes aprecian que CodeCombat resulta altamente accesible a estudiantes secundarios (85.19%). En menor medida (37.04%) perciben que CodinGame es accesible, posiblemente esta opinión se debe a que la interfaz y desafíos son más complejos.

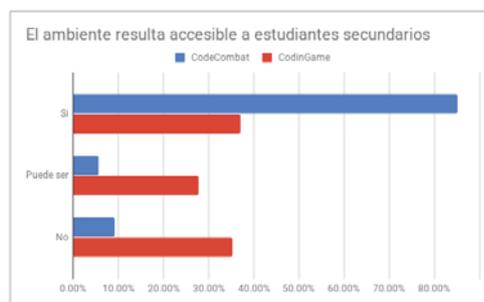


Figura 2. Accesibilidad de los ambientes CodeCombat y CodinGame

El segundo hallazgo se muestra en la Figura 3 donde, en forma coherente con el anterior, CodeCombat claramente es apreciado con mayores posibilidades (79.63% sobre 18.52%) de integrarse a procesos reales de enseñanza de la programación en el ámbito de la escuela secundaria.

Para cada ambiente seleccionado, se pidió a los docentes que expresaran si lo consideraban una buena opción para favorecer la transición de programación

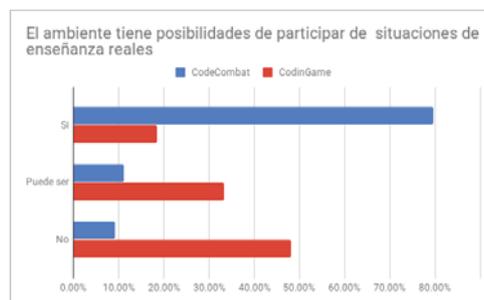


Figura 3. Posibilidades de participar contextos reales de enseñanza de los ambientes CodeCombat y CodinGame

basada en bloques a basada en texto. La Figura 4 muestra que CodeCombat es considerado un dispositivo con altas posibilidades (83.33 %) de constituirse en un puente entre la programación basada en bloques que sucede en los primeros años de la escolaridad secundaria y la programación utilizando lenguajes y entornos convencionales que se espera suceda en el tramo final de la escuela secundaria.

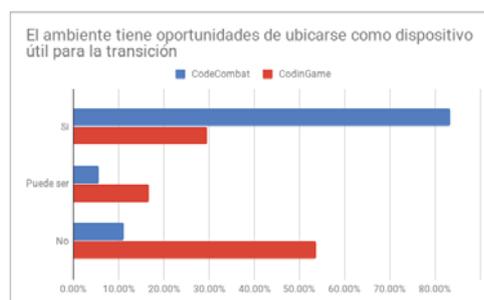


Figura 4. Oportunidad de ubicarse como dispositivo de transición de los ambientes CodeCombat y CodinGame

5. Conclusiones y trabajo futuro

Este documento busca contribuir a mejorar la comprensión acerca del problema de la transición desde la programación basada en bloques a la programación basada en texto y propone un modelo para la selección y definición de ambientes. Por otro lado, se realiza una revisión y análisis de una colección de recursos que tienen diferentes potencialidades en este contexto.

Finalmente, se resume un informe sobre las percepciones elaboradas por la docencia secundaria sobre dos ambientes en particular. Estas apreciaciones aportan

indicios preliminares que confirman el modelo propuesto, en tanto el ambiente que cubre mejor las características esperables recibió aceptación más fuerte.

El objetivo principal de esta línea de investigación es el de aportar nuevos y mejores elementos que colaboren en la definición de próximos entornos destinados a favorecer la enseñanza de las Ciencias de la Computación.

Referencias

1. BrickLayer Homepage. Último acceso Agosto 2019, website <https://bricklayer.org>.
2. CodeCombat Homepage. Último acceso Agosto 2019, website <https://codecombat.com/>.
3. CodeMonkey Homepage. Último acceso Agosto 2019, website <https://www.playcodemonkey.com/>.
4. CodinGame Homepage. Último acceso Agosto 2019, website <https://www.codingame.com>.
5. Greenfoot. Último acceso Agosto 2019, website <https://www.greenfoot.org/>.
6. Pencil Code Homepage. Último acceso Agosto 2019, website <https://pencilcode.net>.
7. Processing Homepage. Último acceso Agosto 2019, website <https://processing.org/>.
8. PythonTurtle Homepage. Último acceso Agosto 2019, website <http://pythonturtle.org>.
9. RoboMind Homepage. Último acceso Agosto 2019, website <https://www.robomind.net>.
10. Tynker Homepage. Último acceso Agosto 2019, website <https://www.tynker.com/>.
11. N. C. Brown, A. Altadmri, and M. Kölling. Frame-based editing: Combining the best of blocks and text programming. In *2016 International Conference on Learning and Teaching in Computing and Engineering (LaTICE)*. IEEE, 2016.
12. K-12 Computer Science Framework Steering Committee. *The K-12 Computer Science Framework*. ACM, 2016.
13. M. Kölling, N. C. Brown, and A. Altadmri. Frame-based editing. *Journal of Visual Languages and Sentient Systems*, 3(1), 2017.
14. W. Robinson. From scratch to patch: Easing the blocks-text transition. In *Proceedings of the 11th Workshop in Primary and Secondary Computing Education*, pages 96–99. ACM, 2016.
15. F. Sadosky. *CC - 2016 Una propuesta para refundar la enseñanza de la computación en las escuelas Argentinas*. Fundación Sadosky, Argentina, 2013.
16. M. Webb, N. Davis, T. Bell, Y. J. Katz, N. Reynolds, D. P. Chambers, and M. M. Syslo. Computer science in k-12 school curricula of the 21st century: Why, what and when? *Education and Information Technologies*, 22(2):445–468, 2017.
17. D. Weintrop and N. Holbert. From blocks to text and back: Programming patterns in a dual-modality environment. In *Proceedings of the 2017 ACM SIGCSE Technical Symposium on Computer Science Education*, pages 633–638. ACM, 2017.
18. D. Weintrop and U. Wilensky. Comparing block-based and text-based programming in high school computer science classrooms. *ACM Transactions on Computing Education (TOCE)*, 18(1):3, 2017.
19. D. Weintrop and U. Wilensky. How block-based, text-based, and hybrid block/text modalities shape novice programming practices. *International Journal of Child-Computer Interaction*, 17:83–92, 2018.

Una propuesta metodológica para evolucionar hacia un diseño curricular basado en competencias

Sonia Rueda, Jessica Carballido,
Luciano Tamargo, Alejandro J. García, Guillermo R. Simari
Departamento de Ciencias e Ingeniería de la Computación, Universidad Nacional del Sur

Resumen. Este trabajo propone una metodología para evolucionar hacia un modelo educativo basado en competencias, partiendo de un plan de estudios estructurado por contenidos conceptuales, distribuidos en asignaturas. La metodología establece un conjunto genérico de competencias de egreso y captura la especificidad de cada carrera a partir de la definición de resultados del aprendizaje. La propuesta plantea la modificación de los formatos curriculares, la integración de la teoría y la práctica, y la aplicación de estrategias de aprendizaje centradas en un estudiante activo.

Palabras clave: diseño curricular, aprendizaje activo, enfoque basado en competencias.

1 Introducción

Existen diversos factores que condicionan y guían el diseño curricular de las carreras de Informática. Uno de estos factores es la normativa vigente, en particular, la *Ley de Educación Superior* [1], la resolución que aprueba los *Contenidos curriculares básicos, carga horaria mínima, criterios de intensidad de la formación práctica y estándares para las cinco terminales de Informática* [2] y la resolución que establece los *Alcances y actividades profesionales reservadas exclusivamente al título* [3].

Los estándares aplicados por la Comisión Nacional de Evaluación y Acreditación Universitaria en los procesos de acreditación de carreras, prescriben los contenidos que deben incluirse en los planes de estudio asociados a una terminal. Cada institución universitaria ha completado sus planes con los temas que ha considerado relevantes, según el enfoque elegido, los alcances específicos que ha establecido y la carga horaria total. Las universidades que ofrecen una única carrera de Informática probablemente han adoptado una visión más generalista que aquellas que dictan dos o más terminales. El ingreso irrestricto impone algunas restricciones adicionales porque exige articular el primer tramo de las carreras, con el diseño curricular del secundario. El ámbito socio-productivo también plantea requerimientos para las carreras de Informática, aunque por lo general no se basan en contenidos conceptuales, sino más bien en las competencias que deberían desarrollar los estudiantes durante su trayecto formativo. Siendo la demanda de profesionales mayor a la oferta, las búsquedas laborales de alumnos avanzados y graduados recientes se orientan a competencias genéricas, como la resolución de problemas y el trabajo en equipo, más que a conocimientos específicos. La flexibilidad y el aprendizaje autónomo son valoradas particularmente porque permitirán la adaptación a un entorno dinámico.

El crecimiento de la disciplina es otro factor que impulsa el diseño y actualización de los currículos, siendo los estándares internacionales una referencia valiosa. Las recomendaciones del ACM sugieren mantener alineado el “tamaño” de los contenidos con la carga horaria [6]. La incorporación de nuevos temas debería estar acompañada por la eliminación de otros o la reducción de la intensidad con la cual se abordan.

Hasta el momento el diseño curricular de las carreras de Informática se ha orientado a la elaboración de planes de estudio estructurados en base a contenidos. Un enfoque alternativo es adoptar un modelo educativo basado en competencias, ya sea para proponer un nuevo trayecto formativo o bien para transformar el plan de estudios de una carrera implementada.

Este trabajo propone una metodología para evolucionar hacia un modelo basado en competencias, partiendo de un plan de estudios estructurado en asignaturas, cada una de las cuales tiene asociado un programa que especifica sus contenidos conceptuales. La primera tarea consiste en definir un conjunto de competencias genéricas de egreso. Posteriormente se especifican en los programas los resultados del aprendizaje que cada asignatura se compromete a alcanzar y las estrategias pedagógicas que se aplicarán, para contribuir con el desarrollo de las competencias definidas. Las Actividades Profesionales Reservadas establecidas por la normativa estarán garantizadas como resultados del aprendizaje.

La metodología ha sido elaborada en el contexto del seguimiento y actualización del plan de estudios de la Licenciatura en Ciencias de la Computación (LCC), que realiza la Comisión Curricular del Departamento de Ciencias e Ingeniería de la Computación de la Universidad Nacional del Sur (UNS).

La siguiente sección describe las principales características de un modelo educativo basado en competencias. A continuación, se presenta la metodología propuesta, y finalmente se describen conclusiones y líneas de acción para el futuro.

2 Un modelo educativo basado en competencias

En 1999 se inició en Europa un proceso de transformación educativa que se formalizó en la “*Declaración de Bolonia*” [4]. El acuerdo impuso a las universidades del Espacio Europeo de Educación Superior, la obligación de orientar la educación de grado a la obtención de *competencias*. En 2005 la Comisión Europea definió competencia como “una combinación dinámica de atributos, habilidades y actitudes”.

Algunos países ya habían experimentado transformaciones hacia un modelo educativo flexible, antes de que se iniciara el proceso de Bolonia. En 1996 la Facultad de Medicina de Brown University implementó un cambio curricular que denominaron *outcome-based education*. Los contenidos distribuidos en un currículo de materias o asignaturas, se reemplazaban por *resultados del aprendizaje* que se asociaban a unidades curriculares y se evaluaban en términos de la capacidad de *hacer*.

En América Latina en el año 2004 se inició el Proyecto Tuning, con el objetivo de propiciar la reflexión y el intercambio en el ámbito la educación superior, en torno a la mejora de los procesos de enseñanza y aprendizaje para la formación integral del estudiante con un enfoque de competencias. Desde la perspectiva de este proyecto las competencias representan una combinación de atributos que integran el *saber*, *saber hacer* y *saber ser* [5]. El *saber* se refiere al conocimiento teórico, el *saber hacer* a la aplicación práctica de este conocimiento y el *saber ser* a los valores que inciden en la percepción y la acción sobre el contexto. El proyecto distingue las *competencias genéricas*, compartidas por todas las titulaciones, de las *competencias específicas*, que determinan la identidad de cada disciplina particular.

En Argentina, la Red de Universidades con Carreras de Informática (RedUNCI) elaboró en 2018 una propuesta de estándares especificando un conjunto de competencias genéricas para las cinco terminales de la disciplina y competencias

específicas para cada una [7]. Previamente, el Consejo Federal de Decanos de Ingeniería (CONFEDI) había publicado su propuesta de estándares, distinguiendo también las competencias comunes a todas las titulaciones, de las específicas [8].

En línea con el modelo educativo propuesto en Tuning, en este trabajo definimos *competencia* como la “capacidad de integrar conocimientos, habilidades, actitudes y valores para aplicarlos en un contexto determinado”. La metodología que presentamos, propone partir de la especificación de un conjunto de competencias genéricas de egreso. Las competencias genéricas no aluden a una disciplina en particular, son transferibles a cualquier ámbito académico o laboral, por lo general comienzan a desarrollarse en la infancia, dentro y fuera del ámbito educativo formal, y están relacionadas entre sí.

La singularidad de cada disciplina quedará especificada por *resultados del aprendizaje*. De acuerdo a las recomendaciones del ACM [6], los resultados del aprendizaje capturan los procesos cognitivos que los estudiantes desarrollan en cada unidad curricular y permiten alcanzar gradualmente las competencias de egreso.

En términos del diseño curricular, los resultados del aprendizaje deben ser específicos, medibles, relevantes y alcanzables en un tiempo acotado. Cuando una unidad curricular asume el compromiso de alcanzar un resultado del aprendizaje, especifica explícitamente las estrategias que van a favorecer el aprendizaje centrado en el estudiante activo y las formas de evaluación que medirán los resultados, consistentes con las estrategias.

3 La propuesta metodológica

A continuación, se presenta la metodología para evolucionar desde un plan de estudios estructurado por contenidos hacia un modelo basado en competencias, utilizando como ejemplo, la transformación del plan de estudios de la Licenciatura en Ciencias de la Computación de la UNS. La metodología propuesta consta de nueve tareas:

1. Establecer un conjunto de competencias genéricas y el rango de niveles de intensidad (NI) para cada una.
2. Generar la matriz de competencias en base a niveles de intensidad (MCNI).
3. Evaluar la consistencia de la MCNI y ajustar la articulación de los niveles de intensidad entre asignaturas asociadas por requisitos de correlatividad.
4. Establecer los resultados del aprendizaje y organizarlos en unidades curriculares.
5. Evaluar la articulación entre unidades curriculares en base a resultados de aprendizaje (RA).
6. Generar la matriz de competencias en base a RA (MCRA) garantizando la cobertura de las actividades reservadas.
7. Evaluar la consistencia entre MCNI y MCRA considerando correlatividades.
8. Especificar las estrategias pedagógicas y los formatos curriculares en el plan.
9. Definir un mecanismo de mejora y evaluación continua.

En este trabajo las unidades curriculares son las asignaturas del plan, entre las cuales se han distribuido, antes de aplicar la metodología, los contenidos conceptuales, tanto los básicos que exige la reglamentación, como los que agrega la Universidad.

3.1 Establecer competencias y niveles de intensidad

En un modelo educativo basado en competencias, la integración de conocimientos, habilidades, actitudes y valores se consolida progresivamente, en forma transversal al aprendizaje de los contenidos conceptuales. La definición del conjunto de competencias genéricas de egreso es fundamental porque compromete a la institución universitaria a implementar estrategias pedagógicas y mecanismos de evaluación que promuevan su desarrollo. El diseño curricular especifica cómo y con qué nivel de intensidad se va a ir desarrollando cada competencia a través del plan de estudio.

La metodología propone definir entonces un rango de *niveles de intensidad* para cada competencia y especificar con precisión cómo se lo interpreta. Como partimos de un conjunto genérico es posible que cada competencia haya comenzado a desarrollarse en la escuela. Si asumimos un nivel de desarrollo previo, los rangos para los NI comenzarán con un valor mayor a cero. Por ejemplo, cada estudiante ingresa a una carrera universitaria habiendo alcanzado en la escuela secundaria cierto nivel de comprensión lectora, que se profundizará a medida que avance en el plan de estudios, hasta lograr el nivel requerido para un graduado.

En carreras sometidas a procesos de acreditación que establezcan requisitos para las competencias de egreso, el conjunto deberá ser consistente con la normativa del Ministerio de Educación, Ciencia y Tecnología. Hasta tanto la reglamentación no prescriba competencias, el conjunto puede definirse considerando únicamente los alcances del título, que incluyen a las Actividades Profesionales Reservadas [3].

Comenzamos con nuestro ejemplo estableciendo el siguiente conjunto de competencias genéricas: Comprensión Lectora (CL) / Producción Escrita (PE) / Oralidad (O) / Pensamiento Crítico (PC) / Pensamiento Abstracto (PA) / Razonamiento Lógico (RL) / Resolución de Problemas (RP) / Autonomía en el Aprendizaje (AA) / Administración del Tiempo (AT) / Trabajo en Equipo (TE).

La propuesta metodológica puede mantenerse, aun cuando se modifique el conjunto de competencias. Hemos adoptado el criterio de incluir competencias que puedan asociarse a resultados del aprendizaje medibles, valiosos en todo ámbito profesional.

El desarrollo de estas competencias ha guiado el diseño curricular de la LCC desde su creación. Sin embargo, no había una especificación explícita, documentada y articulada. En un modelo basado en competencias el compromiso se establece en el plan de estudios y se propaga a los programas de las asignaturas.

Siguiendo con el ejemplo de la LCC, es posible definir tres niveles de intensidad para el desarrollo de la Comprensión Lectora:

NI 1: Lectura literal, es el nivel básico de lectura centrado en la identificación de la información que está explícita en el texto, como por ejemplo las secuencias de hechos, acciones y relaciones de causa o efecto.

NI 2: Lectura inferencial, se basa en el reconocimiento de lo que está implícito en el texto y requiere un alto nivel de abstracción. Las inferencias se construyen cuando se comprende por medio de relaciones del texto con los saberes previos. El resultado es la elaboración de hipótesis, conclusiones y nuevas ideas.

NI 3: Lectura crítica de carácter evaluativo, intervienen los saberes previos del lector, su criterio y el conocimiento de lo leído, tomando distancia del contenido del texto para lograr emitir juicios valorativos desde una posición documentada y sustentada.

Cada asignatura contribuirá a fortalecer algunas competencias, mientras que otras, aun cuando las apliquen, no asumirán el compromiso de profundizarlas. En particular, probablemente todas las asignaturas van a requerir un nivel de comprensión lectora, pero solo algunas se comprometerán a aumentarlo. Además, no todas las

competencias se van desarrollar de acuerdo a un mismo rango. La especificación se completa cuando se ha definido el rango de NI para cada competencia del conjunto.

3.2 Generar la matriz de competencias por niveles de intensidad

Establecido el conjunto de competencias y definidos los rangos de niveles de intensidad, es posible determinar para cada unidad curricular y cada competencia, el nivel que se asume adquirido previamente y el que se aspira desarrollar. Una representación adecuada es una matriz cuyas columnas corresponden a las competencias y las filas consignan la información de cada asignatura. Por ejemplo:

	C1	C2	C3	C4	C5	C6
A1	½	1/1	1/1	1/1	1/1	1/2
A2	1/1	½	1/1	1/1	1/1	1/1
A3	3/3	2/2	1/2	1/1	2/2	2/3
A4	2/3	2/3	1/2	1/1	1/1	2/2
A5	2/2	1/1	1/1	1/1	1/1	2/2

En el ejemplo, todos los rangos comienzan en 1 y la matriz indica que:

- la asignatura A1 asume adquirido previamente el nivel de intensidad 1 para todas las competencias y se compromete a profundizar C1 y C6 hasta alcanzar el nivel 2.
- la asignatura A2 tiene las mismas premisas que A1 y desarrolla C2.
- la asignatura A3 asume adquirido el nivel 3 en C1, el nivel 2 en C2, C5 y C6 y el nivel 1 en C3 y C4. Se compromete a desarrollar el nivel 2 de C3 y el nivel 3 de C6.
- la asignatura A4 asume nivel 2 en C1, C2 y C6 y nivel 1 en las demás. Contribuye a desarrollar las competencias C1, C2 y C3.
- la asignatura A5 asume nivel 1 para C2, C3, C4 y C5 y nivel 2 para C1 y C6.

Ilustramos esta tarea con tres asignaturas de la LCC de la UNS:

	CL	PE	O	PC	PA	RL	RP	AA	AT	TE
Elementos de Álgebra y Geometría	1/2	1/2	1/1	1/1	½	1/2	1/1	1/1	1/2	1/1
Resolución de Problemas y Algoritmos	1/2	1/2	1/1	1/1	½	1/2	1/2	1/1	1/2	1/1
Introducción a la POO	2/3	2/2	1/1	1/1	2/3	1/1	2/3	1/2	2/2	1/1

Reiteramos que cuando a una competencia se le asigna un rango que comienza en 1, se asume algún desarrollo adquirido en la escuela, probablemente validado en la etapa de nivelación.

3.3 Evaluar y ajustar la articulación de los niveles de intensidad

Luego de completar la MCNI corresponde decidir si está articulada o si existen *cortocircuitos*. Para ello tomamos como insumo las restricciones de correlatividad que impone el plan de estudios. En una matriz articulada, cada asignatura asumirá que el nivel de desarrollo de cada competencia es:

- el que brinda la escuela secundaria o la etapa de nivelación si no tiene correlativas.
- las asignaturas correlativas, si tiene requisitos de correlatividad.

En el ejemplo propuesto en la subsección anterior, si A3 tiene a A1 y A2 como correlativas, se observa cortocircuitos en C1 y C5 porque parten de una premisa que no está garantizada por las asignaturas previas. Si la asignatura A5 tiene a A1 como

correlativa, no hay cortocircuito, pero de acuerdo a la matriz no desarrolla ninguna competencia. Además, ninguna de las asignaturas del ejemplo asume el compromiso de desarrollar C4; si al completar la tabla, la columna C4 se mantiene en 1, el graduado tendrá la misma capacidad en C4 que al ingresar. Esto nos dice que, o bien la competencia no es relevante, se ha cometido un error en la generación de la matriz o hay un déficit en el diseño del plan de estudios.

3.4 Establecer y organizar los resultados del aprendizaje

Esta tarea es central en la metodología, porque permite explicitar la vinculación entre competencias genéricas y contenidos conceptuales. El punto de partida es la definición de un conjunto de verbos que aludan a procesos cognitivos. Siguiendo las recomendaciones de Curriculum Guidelines for Undergraduate Degree Programs in Computer Science [6], en este trabajo proponemos utilizar la taxonomía de Bloom como insumo [9], aplicándola de manera indicativa y no prescriptiva.

Retomando como ejemplo la LCC de la UNS seleccionamos los siguientes verbos:

Conocer, Observar, Organizar/Estructurar, Buscar/Explorar, Seleccionar, Procesar, Clasificar, Priorizar, Identificar/Reconocer, Definir, Describir, Interpretar/Comprender/Entender, Establecer, Comparar, Argumentar, Especificar, Documentar, Elaborar/Producir/Construir/Implementar, Elaborar/Producir/Construir /Implementar, Desarrollar, Analizar/Reflexionar, Sintetizar, Aplicar, Evaluar/Validar/Valorar, Distribuir/Dividir, Componer, Planificar, Proyectar, Dirigir, Controlar, Certificar, Modelar, Diseñar, Descubrir, Innovar, Crear

Verbos como *conocer*, *diseñar* o *evaluar* refieren a la complejidad del proceso cognitivo y a la profundidad del aprendizaje del contenido conceptual que acompaña a dicho verbo. A modo de sugerencia, se recomienda evitar expresiones como *ser capaz de*, *realizar*, *saber*, como así también *adquirir conocimiento acerca de ...*, *aprender a ...* o *ejercitar* porque difícilmente puedan expresar resultados medibles.

En carreras acreditadas, el conjunto de verbos debe incluir a todos aquellos que permitan especificar como resultados del aprendizaje a las Actividades Profesionales Reservadas. Cuando los alcances del título reúnen a dos o más procesos cognitivos, es necesario separarlos para asociarlos a unidades curriculares. Consideremos una carrera cuyos alcances incluyan *Diseñar*, *implementar* y *validar sistemas de software*. Al intentar trasladarlos al plan de estudios, nos encontramos con que se trata de distintos procesos cognitivos, con diferente complejidad. Por lo general, una única asignatura no debería comprometerse a enseñar y evaluar todos. Este alcance podría asociarse a tres resultados del aprendizaje: *Diseñar sistemas de software*, *Implementar sistemas de software* y *Validar sistemas de software*.

Una manera de garantizar la cobertura de los alcances, es asociar resultados del aprendizaje a las asignaturas, comenzando por el último tramo del plan de estudios y aplicando “backward design” [10]. Así los alcances para el título, quedarán probablemente vinculados a las materias avanzadas, que asumirán desarrollados otros procesos cognitivos previamente.

Cada unidad curricular va a comprometerse a alcanzar los resultados específicos que quedan definidos en su programa. El compromiso incluye seleccionar y aplicar estrategias pedagógicas que aseguren que los resultados sean alcanzables dentro de la carga horaria asignada y mecanismos de evaluación que permitan medirlos. A modo de ejemplo, especificamos algunas instancias para los resultados del aprendizaje:

- Interpretar enunciados de problemas que incluyan representaciones visuales.
- Diseñar un sistema aplicando una metodología orientada a objetos.
- Seleccionar casos de prueba para validar un sistema, una clase o un método.
- Diseñar un modelo de datos para representar información persistente.
- Representar estados, metas y acciones en un sistema de planeamiento.
- Conocer algoritmos de planificación de procesos centralizados y distribuidos.

Observemos que el último ítem implica conocer la existencia de ciertos algoritmos, sin aludir a procesos cognitivos más específicos como analizar, evaluar o comparar.

3.5 Evaluar la articulación en base a resultados del aprendizaje

Una vez que se han establecido los resultados del aprendizaje y han quedado asociados a unidades curriculares, es necesario evaluar la articulación siguiendo los hilos de correlatividad. Cada unidad curricular asume el compromiso de alcanzar ciertos resultados del aprendizaje, partiendo de la base que los estudiantes han desarrollado previamente algunos procesos cognitivos que les permiten aplicar los conocimientos adquiridos en determinados contextos.

Así, el programa de cada asignatura incluirá como precondiciones, un subconjunto de los resultados del aprendizaje de las asignaturas ligadas por requisitos de correlatividad. Los procesos cognitivos, asociados a contenidos conceptuales, enhebran las unidades curriculares que conforman un trayecto formativo, siguiendo los hilos que imponen las restricciones de correlatividad. Una unidad curricular que se compromete a que los estudiantes aprendan a “Elaborar diagramas de clases para modelar los datos de un sistema” puede asumir que los estudiantes han adquirido previamente la capacidad para “Interpretar diagramas de clases que modelan los datos de un sistema”, si alguna asignatura correlativa tiene asociado este resultado.

3.6 Generar la matriz de competencias por resultados del aprendizaje

La relación entre los resultados del aprendizaje y las competencias puede visualizarse también mediante una matriz. Cada una de las columnas representa nuevamente a una competencia. Las filas corresponden a los RA que se asociaron a las asignaturas.

		C1	C2	C3	C4	C5	C6
A1	RA11	X					
	RA12						X
A2	RA21		X				
A3	RA31						X
	RA32			X			
	RA33			X			X
A4	RA41	X					
	RA42	X		X			

- A1 se compromete a alcanzar RA11 que desarrolla C1 y RA12 que desarrolla C6.
- A2 se compromete a alcanzar RA21 que desarrolla C2.
- A3 tiene asociado RA31 que desarrolla C6, RA32 que desarrolla C3 y RA33 que desarrolla las dos competencias mencionadas antes.
- A4 tiene asociados RA41 y RA42, el primero desarrolla C1 y el segundo C1 y C3.

Observemos que las columnas que corresponden a C4 y C5 no están marcadas, de modo que ninguna de las asignaturas de la tabla contribuye a su desarrollo.

3.7 Evaluar la consistencia entre las matrices

Generada la matriz que vincula resultados del aprendizaje con las competencias, debemos analizar su consistencia con relación a la MCNI. En esta tabla observamos una inconsistencia, ya que la asignatura A4 debería desarrollar C2, pero no se ha especificado ningún resultado de aprendizaje que contribuya a fortalecer esta competencia. Claramente hay dos posibilidades, se identifica un nuevo resultado de aprendizaje para A4, ligado a C2; o bien se elimina el compromiso de A4 con relación al desarrollo de C2 en la MCNI y se rehace la tarea 3.

3.8 Especificar estrategias pedagógicas y formatos curriculares

En un modelo educativo orientado al desarrollo de competencias el énfasis está en el “hacer”, el estudiante tiene un rol activo y está comprometido con su proceso de aprendizaje. Cada unidad curricular seleccionará las estrategias pedagógicas que resulten más adecuadas para alcanzar los resultados del aprendizaje comprometidos.

Las siguientes estrategias pedagógicas están alineadas con el aprendizaje activo: *Aprendizaje basado en el análisis de casos*, *Aprendizaje basado en la resolución de problemas*, *Aprendizaje basado en el desarrollo de proyectos*, *Aprendizaje basado en la investigación*, *Clase invertida*. El debate, los juegos, las aplicaciones interactivas son también estrategias centradas en el estudiante.

En el caso de la LCC numerosas asignaturas de diferentes áreas han adoptado alguna de estas estrategias. Algunas de ellas las han especificado en el programa, sin embargo, hasta el momento no se habían asociado explícitamente estrategias y resultados de aprendizaje, alineados a su vez con el desarrollo de competencias.

Las unidades curriculares pueden adoptar diferentes formatos:

- *Asignatura o Materia*, se estructuran en clases teóricas y prácticas. Las clases teóricas están a cargo del profesor y son más bien expositivas. Las prácticas asumen un rol activo del estudiante, las coordina un asistente con el cual colaboran algunos ayudantes. La articulación entre la teoría y la práctica permite que los estudiantes puedan aplicar los contenidos conceptuales presentados en las clases teóricas, a la resolución de los trabajos prácticos.
- *Taller*, está basado en la vivencia y la interacción. La teoría y la práctica se integran para promover la creatividad y el trabajo colaborativo de estudiantes y docentes.
- *Seminario*, es un formato frecuente en carreras de posgrado, aspira a profundizar el conocimiento en un tema o problemática específica, a partir de la exploración de información abundante y diversa. Promueve la autonomía y el pensamiento crítico.
- *Proyecto*, en este formato la actividad central es la elaboración, ejecución y evaluación de un plan establecido a partir de un objetivo, con recursos asignados. Por lo general se plantean proyectos para abordar en equipo para desarrollar la colaboración.
- *Laboratorio*, está basado en la realización de experiencias que comprueben o rebatan una hipótesis, promueve la búsqueda de demostraciones sistemáticas y la formulación de conclusiones. La incorporación de simulaciones y aplicaciones interactivas ha permitido introducir otros formatos para la experimentación.
- *Ateneo*, aspira a conformar un espacio de reflexión para profundizar el conocimiento de un tema en base al análisis de casos, situaciones o problemas. Promueve el debate, el pensamiento crítico y la integración de aprendizajes.

- *Trabajo de campo*, se basa en el desarrollo de actividades en un entorno laboral, puede ser dentro del ámbito público o privado, en el comercio, la industria o en cualquier organización que permita tener una experiencia profesional.

Al evolucionar hacia un modelo basado en competencias, es probable que el formato curricular no cambie drásticamente, sino que vaya transformándose en forma paulatina. Así, aunque mantengamos un plan estructurado en asignaturas, las actividades experimentales podrán organizarse como laboratorios, parte de la carga horaria puede adoptar el formato de taller y destinarse a un proyecto, algunos contenidos teóricos se presentarán a partir de la resolución de un problema o el análisis de un caso real. Todas estas alternativas promueven el aprendizaje entre pares, el docente orienta el proceso de aprendizaje, pero no es la única fuente de conocimiento. Cuando los grupos son reducidos y los estudiantes han adquirido cierto nivel de autonomía, la clase invertida es una estrategia valiosa. El docente brinda material o impulsa la búsqueda de material, que va a ser leído antes de la clase, en la cual rol protagónico lo tendrán los estudiantes, por ejemplo mediante un debate.

La infraestructura y la normativa son dos aspectos que inciden en la elección del formato curricular. La relación entre el número de docentes y estudiantes es otro factor que puede favorecer o bien obstaculizar la aplicación de una estrategia u otra.

3.9 Definir un mecanismo de mejora y evaluación continua

La aplicación de la metodología propuesta implica un esfuerzo considerable de coordinación y la participación activa de docentes y alumnos. La búsqueda de consensos es esencial para definir y enhebrar resultados del aprendizaje entre asignaturas ligadas por requisitos de correlatividad. El diseño curricular basado en competencias requiere seleccionar y aplicar estrategias pedagógicas que promuevan el aprendizaje centrado en el estudiante activo, el compromiso de los docentes es fundamental en este aspecto.

Como todo cambio, la transformación del modelo educativo va a requerir realizar un seguimiento de la implementación, analizar el impacto en el rendimiento académico de los estudiantes y más a largo plazo evaluar la inserción de los graduados en el ámbito socio productivo. A partir de estos resultados se definen ajustes y mejoras en el diseño. Una de las cuestiones a valorar es si los resultados del aprendizaje especificados son efectivamente medibles, relevantes en términos de las competencias y alcanzables en los tiempos establecidos. La medición del tiempo real invertido por los estudiantes en un trayecto formativo es un tema pertinente.

La transparencia es un factor importante para el seguimiento y evaluación. Los planes de estudio deben especificar competencias, alcances y rangos de niveles de intensidad. En los programas de las materias deben estar explicitados objetivos, resultados del aprendizaje, contenidos conceptuales, estrategias pedagógicas, mecanismos de evaluación, carga horaria y bibliografía. Esta documentación debe ser accesible a estudiantes, docentes, órganos de gestión y a la sociedad en su conjunto.

Un aspecto muy significativo a evaluar es la articulación de las asignaturas que se dictan simultáneamente en un mismo ciclo lectivo. Si tres materias, por ejemplo, adoptan una estrategia de aprendizaje basada en proyectos y la entrega final de todas se planifica para las últimas dos semanas del cuatrimestre, es probable que los resultados no sean alcanzables o al menos no para una proporción significativa de alumnos. El trabajo de campo es también sumamente valioso, pero debe coordinarse

de modo tal que integre actividades de diferentes asignaturas o bien que todas estén informadas cuando los alumnos están realizando prácticas en territorio.

Las encuestas son un recurso útil para evaluar la articulación horizontal y vertical de las asignaturas porque permiten conocer la opinión de los estudiantes, antes, durante y después de completarse una unidad curricular. Por supuesto el análisis de los datos que brindan los sistemas de información es indispensable para identificar dificultades y oportunidades de mejora.

4 Conclusiones y trabajo futuro

El desarrollo de competencias que articulen la formación académica con el desarrollo profesional, es una aspiración de toda institución universitaria que brinde carreras de Informática. Sin embargo, hasta el momento los planes de estudio se han estructurado principalmente en base a contenidos conceptuales.

La evolución hacia un modelo educativo basado en competencias plantea varios desafíos, requiere la participación y el consenso de diversos actores y, por sobre todas las cosas, exige flexibilidad para adaptarse a los cambios.

En este trabajo hemos propuesto una metodología para reformular un plan de estudios explicitando competencias genéricas y resultados del aprendizaje que permitan desarrollarlas. La modificación en las especificaciones de planes y programas debe visibilizarse en el aula, y para ello es necesario transformar la práctica pedagógica con un enfoque centrado en el estudiante activo. El próximo paso será aplicar la metodología a todas las carreras que gestiona el Departamento de Ciencias e Ingeniería de la Computación de la UNS. En el momento que se modifique la normativa ministerial que guía a los procesos de acreditación, volveremos a aplicarla utilizando los nuevos estándares.

5 Referencias

- [1] Ley de Educación Superior de la Nación Argentina. [↗](#)
- [2] Resolución 786/09 Contenidos curriculares básicos, carga horaria mínima, criterios de intensidad de la formación práctica y los estándares para las cinco terminales de Informática. [↗](#)
- [3] Resolución ME-1254/18: Alcances de título y actividades profesionales reservadas exclusivamente al título [↗](#)
- [4] Bologna Declaration. The European Higher Education Area. Joint Declaration of the European Ministers of Education (Bologna), 1999. [↗](#)
- [5] Reflexiones y perspectivas de la Educación Superior en América Latina, Tuning América Latina, 2007. [↗](#)
- [6] Recomendaciones Curriculares del ACM para Ciencias de la Computación, 2013 [↗](#)
- [7] Propuesta de Estándares para la disciplina Informática, RedUNCI, 2018 [↗](#)
- [8] Propuesta de Estándares de Segunda Generación para la acreditación de Carreras de Ingeniería “LIBRO ROJO DE CONFEDI” CONFEDI 2018 [↗](#)
- [9] Taxonomy of Educational Objectives: The Classification of Educational Goals. Benjamin Bloom, David McKay Company, 1956.
- [10] Morell Lueny, Transforming Engineering Education for the 21st Century, Encuentro Internacional Universidad-Empresa, Colombia, Septiembre 2018 [↗](#)



CACIC 2019
Congreso Argentino de Ciencias de la Computación

VIII Workshop Seguridad Informática Real

Coordinadores

Díaz, Javier (UNLP)
Ramón, Hugo (UNNOBA)
Aciti, Claudio (UNCPBA)

Comité Científico

Acquistapace, Carlos (Argentina)	Echaiz, Javier (Argentina)	Molinari, Lia (Argentina)
Benítez, Carlos (Argentina)	García Garino, Carlos (Argentina)	Saroka, Raúl (Argentina)
Castro Lechtaler, Antonio Ricardo (Argentina)	García Villalba, Javier (España)	Tarouco, Liane (Brasil)
Cipriano, Marcelo (Argentina)	García, Edith (Argentina)	Venosa, Paula (Argentina)
Díaz, Javier (Argentina)	Liporace, Julio César (Argentina)	
	Maiorano, Ariel (Argentina)	

Temas

Administración de Configuración de Software y Despliegue	Mantenimiento de Software
Administración de Proyectos de Software	Metodologías de Desarrollo de Software
Arquitecturas y Diseño de Software	Métricas de Software
Calidad de Software	Modelamiento de Software
Dependabilidad y Confiabilidad del Software	Modelamiento de Procesos de Negocio
Desarrollo Ágil de Software	Orientación a Aspectos
Desarrollo de Software basado en Componentes	Patrones y Frameworks
Entornos para el Desarrollo de Software	Plataformas como Servicios
Evolución del Software	Políticas para el Software y para las TICs
Estándares para el Desarrollo de Software	Privacidad
Herramientas de Software	Procesos de Software
Herramientas y Tecnologías de la Web 2.0	Reusó de Software
Ingeniería de Requerimientos de Software	Sistemas de Información
Ingeniería de Software Distribuido	Sistemas de Software basados en Cloud Computing
Ingeniería de Software Embebido y Tiempo Real	Sistemas de Software Sensibles al Contexto
Ingeniería de Software Empírica	Software como Servicio
Ingeniería de Software para Sistemas Ubícuos y Móviles	Soporte Computacional para Tareas Colaborativas
Ingeniería de Software para la Web	Testeo de Software
Ingeniería Reversa	Teoría y Métodos Formales
	Verificación y Validación de Software
	Workflows

Análisis del anonimato aplicado a criptomonedas

Ignacio Gallardo ^{1,2}, Patricia Bazan ², Paula Venosa ²

igallardo@info.unlp.edu.ar, pbaz@info.unlp.edu.ar, pvenosa@info.unlp.edu.ar

¹ Universidad de la Defensa Nacional - Facultad de Ingeniería del Ejército, Palermo, CABA, Argentina.

² Universidad de la Plata - Facultad de Informática - Laboratorio de Investigación de Nuevas Tecnologías Informáticas, La Plata, Argentina.

DOI: 10.17013/risti.n.pi-pf

Resumen: Este trabajo muestra un estudio en profundidad de la tecnología Bitcoin, en el cual se aborda un análisis cualitativo de sus funcionalidades y arquitectura. También describe diferentes técnicas y fines de usos que potencialmente un usuario podría llegar a realizar con esta tecnología. Por último presenta un análisis de la posibilidad de aplicación de diferentes prácticas de descubrimiento ante la presencia de una investigación del uso cibercriminal de criptomonedas.

Palabras-clave: Anonimato; privacidad; criptomonedas; blockchain; bitcoin.

Analysis of anonymity applied to cryptocurrencies

Abstract: *This work shows an in-depth study of Bitcoin technology, in which a qualitative analysis of its functionalities and architecture is addressed. It also describes different techniques and uses purposes that a user could potentially achieve with this technology. Finally, it presents an analysis of the possibility of applying different discovery practices in the presence of an investigation of the cybercriminal use of cryptocurrencies.*

Keywords: Anonymity; privacy; cryptocurrencies; blockchain; bitcoin.

1. Introducción

El derecho a la privacidad y el anonimato son temas que hoy en día despiertan el interés no solo de empresas particulares, sino que también para ciberdelinquentes o eventos similares a los ocurridos en los últimos tiempos, como conocer detalles delicados acerca de los documentos filtrados por Edward Snowden y las actividades desempeñadas por organizaciones como la NSA o la CIA. Se trata de cuestiones que ponen de manifiesto lo que muchos ya saben cuando navegan por internet: Toda actividad en la red está siendo monitoreada.

Muchas de estas actividades de monitoreo o vigilancia son llevadas a cabo por entidades gubernamentales, cuya finalidad es la de intentar detectar amenazas terroristas, actuar en consecuencia para minimizar su impacto, o trazar la amenaza para ejecutar acciones ofensivas, disuasivas y/o defensivas.

Evidentemente muchas personas obran por tener privacidad y no ser monitoreados, por esto en la actualidad han surgido diferentes herramientas que ayudan a proporcionar privacidad y anonimato en internet. Estas soluciones tecnológicas poseen además de una vía alternativa a los sistemas de monitoreo utilizados por los gobiernos, sistemas de tracking que son utilizados por organizaciones de marketing para perfilar a los usuarios y ofrecer productos acorde a sus patrones de navegación.

Dichas soluciones también suponen una amenaza para la privacidad, ya que se encargan de recolectar datos de navegación, información sobre realización de pagos y análisis de diferentes patrones de uso que cualquier usuario desearía mantener en secreto, ya que evidentemente forman parte de su vida privada. No obstante, el caso en donde este tipo de herramientas adquieren una relevancia mayor, es en países donde restringen el acceso a internet o donde los habitantes no poseen la posibilidad de informar al mundo sobre los abusos que sufren por parte de sus respectivos gobiernos, como por ejemplo, China o Norcorea. Finalmente, estos beneficios que aportan los sistemas de anonimato, también suelen ser utilizados por grupos de activistas o cibercriminales para realizar extorsiones o estafas financieras en la red.

Es útil plantear una distinción básica entre la privacidad y el anonimato en el contexto de las transacciones financieras. Se define una transacción financiera “anónima” al desconocimiento total por parte del contexto hacia el actor que la realiza. Por otra lado, se llama operación financiera “privada”, si el producto de compra y su cantidad son desconocidos, pero no sus actores.

El dinero en efectivo o trueque proporcionan máximas características de privacidad y anonimato esencialmente al momento de realizar transacciones. Por otra parte, y de manera contraria, existen las transacciones que no son ni privadas ni anónimas, y esto contempla, por ejemplo, donaciones de cierta cantidad de dinero, compras mediante tarjetas de débito/crédito, pago por transferencias bancarias, etc., en donde la identidad del ente comprador se encuentra almacenada y relacionada con la entidad vendedora junto al detalle de la transacción y, no obstante, ante ciertas situaciones esta información podrá eventualmente ser accedida.

Desde el punto de vista de **Bitcoin**, las transacciones no presentan características de privacidad, pero sí de anonimidad, es decir, las identidades no se registran en ninguna parte del **protocolo Bitcoin**, pero cada transacción realizada es visible en un “libro electrónico público” y distribuido, conocido como **blockchain** o **cadena de bloques**.

Estas características proporcionadas por **Bitcoin** alteran el principio de la regulación financiera y se convierte potencialmente en un mecanismo de pago generalizado entre los cibercriminales. Es por ello que en muchas ocasiones los investigadores forenses se tienen que enfrentar ante los desafíos que conlleva su investigación debido a la utilización de esta tecnología para el cobro anónimo de extorsiones o de servicios fraudulentos llevados a cabo por algunos grupos de ciberdelincuentes.

2. Estado del Arte

Los malwares o virus informáticos se presentan de diferentes formas y los ataques que adoptan los mismos se llevan a cabo de manera cada vez más sofisticada y difíciles de identificar. El motivo principal puede ser el de conseguir datos personales, encriptar archivos para luego poder extorsionar o simplemente desestabilizar una organización o gobierno. Los targets principales de estos ataques se clasifican en dispositivos inteligentes de cualquier tipo como: móviles, computadoras, cámaras de vigilancia y vehículos conectados.

Detrás de estos delitos no hay un cibercriminal, ni dos, ni tres; hay toda una industria que trabaja en red. Se trata de un entramado que se mueve en la **deep web** donde muchos de “los trabajos” se cobran en **criptomonedas**.

En el nivel más bajo de la pirámide está el **script kiddie**, un término despectivo para describir a quienes utilizan programas o scripts de otros para vulnerar sistemas informáticos. Ellos no desarrollan malwares, sino que se hacen de archivos o datos que obtienen en foros o por otra vía para realizar sus ataques.

En un nivel más avanzado se encuentran los **hackers** con ciertos conocimientos técnicos, algunos, incluso están graduados en alguna carrera de informática o sistemas. Ellos, por ejemplo, se encargan de publicar **exploits**, que son programas que se aprovechan de un agujero de seguridad en una aplicación o sistema. Un exploit no es, en sí, un código malicioso, sino "la llave" o el modo en que se puede acceder al sistema.

Encontrar y vender un exploit es legal siempre y cuando se utilice con fines éticos. De hecho hay empresas como **Zerodium** que compran los exploits para desarrollar soluciones de seguridad basada en esa información. El camino ilegal sería vender esos datos a cibercriminales que la utilizan para realizar ataques.

Los **exploits** para antivirus hoy en día se encuentran en torno a los 4,43 bitcoins (40 mil dólares) y los que son para sistemas operativos como el de Apple tienen un tope de hasta 166 bitcoins (1,5 millones de dólares).

También se rentan **botnets** por entre 0,02 y 0,04 bitcoins (170 y 350 dólares) por hora para enviar **Spam** o hacer ataques de **DNS** como el que ocurrió a fines de 2016 y que dejó a los principales sitios web del mundo sin servicio.

La realidad es que es cada vez más fácil y más masivo el acceso y utilización de las criptomonedas para cobros anónimos de dinero, especialmente **Bitcoin**, lo que converge en moneda corriente a la hora de hablar de cibercriminales, pagos extorsivos y lavado de dinero. Por este motivo surge la necesidad de contar con conocimientos y procedimientos al momento de toparse con la necesidad de realizar un análisis activo o pasivo ante un cibercrimen.

3. Metodología

Se realizará un estudio en profundidad de la tecnología Bitcoin abordando un análisis cualitativos de sus procedimientos, arquitectura y core de funcionamiento: **Blockchain** o **Cadenas de Bloques**.

Se describirán diferentes técnicas y medios de uso, con el fin de determinar los fines con los que potencialmente un usuario podría llegar a utilizar esta tecnología.

Se analizará la posibilidad de poder presentar diferentes prácticas a realizar ante la posibilidad de investigar el uso de las criptomonedas.

3. Criptomonedas

El comercio en Internet ha llegado a depender casi exclusivamente de las instituciones financieras que actúan como terceros de confianza para procesar pagos electrónicos. Si bien el sistema funciona lo suficientemente bien para la mayoría de las transacciones, todavía sufre las debilidades inherentes del modelo basado en la confianza. Las transacciones completamente irreversibles no son realmente posibles, ya que las instituciones financieras no pueden evitar disputas mediadoras. El costo de la mediación aumenta los costos de transacción, limitando prácticamente al tamaño mínimo de la transacción y eliminando la posibilidad de pequeñas transacciones eventuales, y hay un costo más alto en la pérdida de capacidad para realizar pagos no reversibles por servicios no reversibles. Con la posibilidad de reversión, la necesidad de confianza se extiende.

Se acepta un cierto porcentaje de fraude como inevitable. Estos costos e incertidumbres de pago se puede evitar en persona mediante el uso de la moneda física, pero no existe ningún mecanismo para realizar pagos sobre un canal de comunicaciones sin una parte confiable. Lo que se necesita es un sistema de pago electrónico basado en pruebas criptográficas en lugar de confianza, permitiendo que dos partes interesadas tramiten directamente entre sí sin la necesidad de un tercero confiable. Las transacciones que no son viables desde el punto

de vista informático protegerían a los vendedores contra el fraude, y los mecanismos de custodia de rutina podrían implementarse fácilmente para proteger a los compradores. Las criptomonedas, proponen una solución a problemas como el de doble gasto utilizando una arquitectura distribuida (peer to peer) para generar una prueba computacional del orden cronológico de las transacciones. El sistema es seguro siempre que los nodos “honestos” controlen colectivamente más potencia de CPU que cualquier otro grupo cooperante de nodos atacantes.

Una criptomoneda es un medio de cambio digital que utiliza además de la arquitectura descrita arriba, una tecnología criptográfica para asegurar la veracidad de las transacciones.

Se puede entender a la transferencia de una moneda digital como el endoso de un cheque, es decir, análogamente a que una persona escriba en el dorso el destinatario del dinero, y este puede a su vez endosarlo nuevamente. También se puede saber si la persona que se lo dió o bien fue el dueño del cheque, o bien fue el último en endosarlo antes que él.

En el ciberespacio, se puede lograr algo similar con firmas digitales y hashes criptográficos, es decir, cuando una persona quiere transferir dinero digital a otra, se crea una transacción, que no es más que la firma digital del hash criptográfico de la transacción anterior que usó ese dinero y la clave pública del destinatario. De esta forma, el destinatario puede verificar que el emisor era realmente el dueño del dinero, verificando la firma digital contra la transacción con el hash dado, y además, puede volver a transferirla usando él su clave privada.

La primera aparición pública de las criptomonedas tuvo origen con **Bitcoin**, donde un usuario con el pseudónimo “Satoshi Nakamoto” anunció, el 1 de noviembre de 2008, su investigación en un nuevo sistema de dinero digital, resumiendo sus propiedades y el contenido del artículo original que describía su trabajo, el cual se encontraba disponible en el portal de Bitcoin.

El 11 de febrero de 2009, un perfil creado en el portal P2P foundation, también con el nombre de “Satoshi Nakamoto”, publicó el siguiente mensaje: “Bitcoin open source implementation of P2P currency”. En el texto, “Nakamoto” daba a conocer el portal oficial de Bitcoin, las características fundamentales de éste, el artículo donde se describe el diseño e, incluso, el cliente inicial con el que comenzar a participar en la red.

3.1 Bitcoin

Antes de entrar en detalles técnicos, resulta útil resaltar que la novedad y éxito de Bitcoin radica en su forma de funcionamiento distribuido y sin una autoridad central que regule la emisión de moneda, o acepte o deniegue transacciones. Básicamente son los nodos pertenecientes a la red los que implícitamente toman estas decisiones de forma “democrática”. Por medio de conceptos que serán expuestos a continuación, es posible lograr entender el paradigma a través de los siguientes ejemplos:

- Los usuarios reciben bitcoins a modo recompensa por haber colaborado con la red (más adelante se verá cómo se produce esto). Hasta acá, puede parecer que los usuarios podrían engañar al sistema para aumentar su recompensa pero, por construcción del sistema, la mayoría de los usuarios tendrán que validar posteriormente esa recompensa. Así, si el usuario la aumentase de forma oculta, esa acción sería rechazada por el resto.
- Un usuario A realiza una transferencia de bitcoins a B. Para evitar que posteriormente A vuelva a utilizar esos mismos bitcoins para pagar a un tercer usuario C, en Bitcoin, las transacciones se hacen públicas, no obstante, cuando el resto de la red detecte esta inválida segunda transacción, la rechazará, imposibilitando dicha reutilización de bitcoins por parte del usuario A.

Como se aprecia en estos ejemplos, son los mismos usuarios los que toman las decisiones que normalmente corresponden a una única autoridad central. Esto hace que Bitcoin sea una moneda “democrática”. Como en cualquier democracia, su evolución se adapta a lo que la mayoría de la población quiere. Por consiguiente, en este caso no hay una equivalencia de “un usuario = un voto”, ya que el peso de cada usuario depende de la potencia de cómputo que éste dedica a la red. Así, la ecuación anterior en Bitcoin, sería más bien “x% de cómputo = x% de votos”. Por lo tanto, siempre y cuando más de un 50% de la potencia de cómputo de la red sea controlada por usuarios honestos, la red seguirá la evolución que estos decidan. La idea puede contemplarse como una “democracia ponderada” en función de las implicancias en el sistema.

3.1.1 El papel de la Criptografía

Las funciones criptográficas de las que **Bitcoin** hace uso, son responsables principales de que se consigan las propiedades de seguridad esta tecnología persigue. El aspecto más importante de la criptografía que le compete a esta criptomoneda es la **asimétrica** o de clave pública y sus capacidades de **firma digital**.

Bitcoin implementa el algoritmo ECDSA para firmar digitalmente las transacciones, utilizando los parámetros recomendados por el Standards for Efficient Cryptography Group (SECG), secp256k1. Las firmas utilizan la codificación DER para empaquetar sus componentes en un único flujo de bytes.

ECDSA ofrece ventajas frente a otros esquemas de firma que lo hacen ideal para su utilización en un protocolo distribuido en Internet, como son: longitudes de clave, de firmas muy cortas, y generación y verificación de firmas muy rápidas.

En los cálculos de hashes realizados en Bitcoin se utilizan los estándares SHA-256 y, cuando se requiere que el hash sea más corto, RIPEMD-160. Generalmente el cálculo de hashes se realiza en dos fases: la primera con SHA-256 y la segunda, dependiendo de las necesidades de longitud del resultado, con SHA-256 o RIPEMD-160.

La generación de números aleatorios es esencial para la criptografía y más en la aplicada a Bitcoin. Los **nonces** o números aleatorios que sólo se utilizan una vez son utilizados de forma directa para la generación de bloques en la **blockchain** de Bitcoin que se verá más adelante.

Las **pruebas de trabajo**, el principal componente de Bitcoin que garantiza que la red tenga un comportamiento legítimo. Básicamente, esta idea hace que validar/calcular nuevos bloques de transacciones conlleve un coste computacional muy elevado, de forma que, para hacerse con el control de la red (y por tanto de qué se valida y qué no), un atacante necesitaría una potencia de cómputo extremadamente difícil de conseguir.

En síntesis, en Bitcoin este control de complejidad en los cálculos para los nuevos bloques se realiza obligando a que el hash de cada nuevo bloque deba comenzar con un número determinado de ceros. Como se verá más adelante, para el cálculo de este hash se combinan datos de bloques anteriores y un nonce. Dado que las funciones hash criptográficas no son invertibles, para encontrar un bloque válido la única alternativa será ir obteniendo diferentes nonce hasta encontrar uno que cumpla el requisito preestablecido.

3.1.2 Contexto Operativo

Con el objetivo de contextualizar el medio operativo del sistema, es necesario describir los actores que interactúan en el mismo y la forma de cómo se efectúan las transacciones.

En cuanto a los actores que actúan en el sistema, se clasifican en tres tipos:

- **Nodos Normales:** realizan compras y pagos de bienes y servicios utilizando como moneda a los bitcoins, produciendo transacciones en el sistema.
- **Nodos Mineros:** son nodos normales que a parte dedican potencia de cómputo para validar nuevas transacciones, creando lo que se conoce como bloques de transacciones. Este rol, como antes se nombró, recibe recompensas en bitcoins por haber colaborado y proporcionado poder computacional a la red.

Un usuario, equivalente a un nodo, se identifica en el sistema por medio de una o más **direcciones Bitcoin**. Esta misma, se representa como una dirección virtual similar a una cuenta bancaria y se materializa como una clave pública de criptografía asimétrica. Estas direcciones pertenecientes a un usuario se almacenan y gestionan en un monedero virtual, que equivale a un **monedero** físico.

Una **transacción** es una transferencia de dinero de un **usuario** hacia otro, incluso a sí mismo, es decir, representa a la asignación de bitcoins de una **dirección Bitcoin** a otra. La constitución de una **transacción**, consiste en que si un usuario **A** transfiere dinero a uno **B**, el **usuario** de la **dirección Bitcoin** asignante (**usuario A**) firme una transcripción de la **dirección Bitcoin** del **usuario B** con la clave privada asociada a la dirección del **usuario A**, de esta forma se determinará que el nuevo propietario de esos bitcoins transferidos es la **dirección Bitcoin** del **usuario B**.

Estas transacciones, una vez que estén pendientes a confirmar, se agrupan a su vez en un conjunto de transacciones pertenecientes a un **bloque**, el cual será sometido posteriormente al proceso de minería y luego adherido a la **cadena de bloques** o **blockchain**.

La **blockchain**, representa al core del funcionamiento de esta criptomoneda. Es básicamente un registro público de las transacciones validadas en orden cronológico, es decir, cuando un bloque ya fue confirmado por medio del proceso de minería, éste pasa a formar parte de este registro público. La **cadena de bloques**, presenta una característica peculiar que es la de público acceso, distribuida, descentralizada y de sólo modo lectura; esto lo logra gracias a su constitución de **Árbol de Merkle**.

Los nodos que integran la arquitectura de **Bitcoin** constituyen un sistema de comunicaciones **P2P** o **peer-to-peer**. Como ya se ha mencionado, la filosofía aquí es evitar la existencia de autoridades centrales que controlan la red.

Como toda arquitectura **P2P**, **Bitcoin** dispone de una serie de mecanismos para descubrir nuevos nodos en la red, y mantener una lista actualizada de los mismos. Además, distintos clientes de **Bitcoin** pueden también ofrecer mecanismos adicionales, como por ejemplo mensajes de tipo **addr** y **getaddr**, mediante los cuales un cliente envía (o solicita) a otro un listado de clientes actualmente conectados a la red. También, en el código de los clientes se suele incluir un listado de nodos semilla, que se utilizarán para iniciar el proceso de conexión a la red en caso de que el resto de mecanismos fallen.

Además de los mecanismos para descubrir otros nodos en la red, hay otros tipos de mensajes de uso frecuente en **Bitcoin**. Por ejemplo, los mensajes **tx** y **block**, utilizados para enviar datos de transacciones y bloques, respectivamente, de manera que los nodos de la red puedan mantener la sincronía requerida por el protocolo. O los mensajes de tipo **inv**, que se utilizan para anunciar (y retransmitir) nuevas transacciones.

Como se describió con anterioridad, unos de los elementos que hace posible el funcionamiento de **Bitcoin** es la **criptografía asimétrica**. En ella, los distintos algoritmos funcionan a partir de una clave compuesta por dos elementos relacionados de modo que son fácilmente computables en una dirección (cifrado, descifrado y verificación de una firma digital) pero difícilmente computables en la contraria si se desconoce de la información secreta.

Una dirección **Bitcoin** convencional (P2PKH) es simplemente una cadena de texto codificada en **Base58Check** que tiene hasta 20 bytes de longitud y que consiste en el **hash**

de la **clave pública** asociada con la dirección. Este formato es similar al **Base64**, con la diferencia que no solo pretende mantener la información codificada lo más legible posible para el usuario, sino que también permite verificar de forma más eficiente si una cadena arbitraria que satisfaga dicha expresión se corresponde con una dirección real o no, aplicando un mecanismo de validación redundante que ya se emplea en los números de tarjeta de crédito o documentos de identidad.

$$\begin{aligned}
 & \text{Versión} = 1 \text{ byte de ceros} \\
 & \text{HashDeClave} = \text{Version} + \text{RIPEMD-160}(\text{SHA-256}(\text{ClavePública})) \\
 & \text{Checksum} = \text{SHA-256}(\text{SHA-256}(\text{HashDeClave})) \\
 & \text{DirecciónBitcoin} = \text{Base58Encode}(\text{HashDeClave} + \text{Checksum})
 \end{aligned}$$

Figura 1 - Formato de Dirección Bitcoin

Las direcciones cumplen con las siguientes características:

- Generación en tiempo computacionalmente reducido (milisegundos).
- La clave privada asociada a dicha dirección debe ser un problema computacionalmente complejo, con el fin de ofrecer garantías de que un tercero no logre generar una clave privada asociada a dicha dirección.
- Generación offline, con el fin de proporcionar una capa de seguridad a dicha creación.

Un usuario podría crear una o varias direcciones, no obstante, un conjunto de dichas direcciones constituye un **monedero** Bitcoin, mediante los cuales se realizan las transacciones que se verán a continuación.

3.1.3 Transacciones

Con el fin de representar el flujo de las criptomonedas en la red existe el concepto de transacciones.

Las transacciones en Bitcoin son estructuras de datos firmadas digitalmente que cambian el propietario de unidades de bitcoins asignándolas a otra dirección o propietario.

La estructura de datos de una transacción se encuentra formada por **entradas, salidas, hash de transacción, firma digital del emisor, clave pública del emisor, total entradas, total salidas, bloqueo, versión:**

- Entradas: registros que referencian los fondos de transacciones previas. Las mismas se encuentran firmadas digitalmente por el “pagador” (proceso necesario y suficiente para desbloquear los fondos transferidos). Campo de tamaño variable.
- Salidas: registros que determinan el nuevo o los nuevos propietarios de las bitcoins transferidas. Estas salidas se utilizan como entradas de transacciones próximas. Campo de tamaño variable.
- Hash de transacción: resumen de toda la estructura de datos.
- Firma digital del emisor: encriptación del **hash de la transacción** con la clave privada del emisor.
- Clave pública del emisor: se añade dicha para que se pueda verificar la firma digital cuando la transacción llegue a un nodo de la red que la deba procesar.
- Total entrada y salidas: número que indica cantidad de entrada y salidas adheridas a la transacción. Cada uno de estos campos puede contener entre 1 y 9 bytes.

- **Versión:** posee 4 bytes e indica el número de versión Bitcoin utilizado para esa transacción.
- **Bloqueo:** indica la fecha mínima en la cual dicha transacción puede ser agregada a la **cadena de bloques**. Si el valor indicado en este campo está entre cero y quinientos (incluidos ambos) indica la cantidad de bloques que deben agregarse a la cadena de bloques antes de agregar esta transacción; y si indica un valor mayor a quinientos, entonces se interpreta como una fecha límite en formato UNIX. En una transacción de transferencia de bitcoins, se deben utilizar todas las que se encuentran asignadas a la dirección origen.

Por ejemplo, si A posee 10 bitcoins y desea enviar sólo 5 bitcoins a B, pues entonces las salidas de la transacción van a ser 5 bitcoins para la dirección de B y 5 bitcoins para la dirección de A; donde esta última toma el rol de una “dirección de devolución”. Por consiguiente, en una transacción siempre se “gastan” todos los bitcoins asignados.

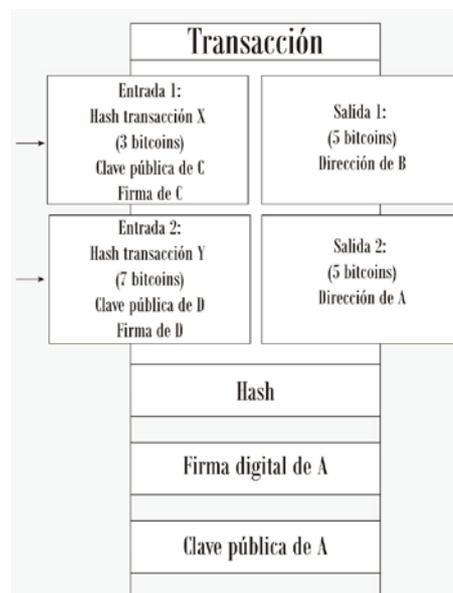


Figura 2 - Esquema de transacción Bitcoin

La suma de la totalidad de las entradas debe ser igual o mayor que la suma de la totalidad de las salidas. En el caso de que la cantidad de bitcoins de la entrada sea mayor que la de la salida, la diferencia se considera una “comisión”, y quien incluya esa transacción en la **cadena de bloques** o **blockchain** (base de datos distribuida y descentralizada) puede disponer de esa cantidad. Esta recompensa es una manera de motivar a los nodos **mineros**, que obtienen beneficios por su trabajo en forma de bitcoins. Las transacciones que poseen “comisiones” tienen prioridad por los nodos mineros al momento de elegir cual de ellas procesar primero, y en consecuencia, las transacciones que posean mayor monto en comisiones serán procesadas de forma más veloz en la red.

Cada **salida** y **entrada**, al igual que las transacciones, tienen su estructura interna. Como ya se vió, las **entradas** son referencias o “punteros” a **salidas** anteriores, es decir, cada **entrada** hace referencia a un identificador perteneciente a una **salida** (UTXO, Salida de Transacción Sin gastar) que se encuentra almacenada en la base de datos distribuida. Para gastar una **UTXO**, la **entrada** de la transacción también incluye una condición de desbloqueo que satisface la condición especificada por la **UTXO**. Este código de desbloqueo

normalmente consta de una firma la cual prueba la posesión de la dirección que se encuentra especificada en el código de bloqueo de la **UTXO**.

Las **entradas** están compuestas por los siguientes campos:

- Hash de transacción: puntero a la transacción que posee la **salida** perteneciente a esta **entrada**. Posee 32 bytes.
- Índice de la **salida**: índice de la **salida** perteneciente a esta **entrada**, es decir, el índice de la **UTXO** que se quiere gastar. Tiene 4 bytes.
- Tamaño del código de desbloqueo: especifica el tamaño en bytes que tiene el código de desbloqueo. De 1 a 9 bytes.
- Código de desbloqueo: el cual cumple las condiciones del código de bloqueo de la **UTXO**. Tamaño variable.

Las **salidas** están compuestas por:

- Monto: cantidad de bitcoins que se desean transferir. Posee 8 bytes.
- Tamaño del código de bloqueo: tamaño en bytes. De 1 a 9 bytes.
- Código de bloqueo: el cual define las condiciones que se deben de cumplir para poder gastar el monto. Generalmente, el código perteneciente a este campo realiza una transferencia de bitcoin a una dirección parametrizable. Tamaño variable.

Cada transacción crea salidas, las cuales son almacenadas en la base de datos distribuidas. Todas las salidas (excepto una) crean **UTXOs** las cuales son reconocidas por toda la red y están disponibles para que el poseedor haga uso de las mismas.

En síntesis, la transferencia de un monto en bitcoins es básica y sencillamente crear una **UTXO** asignada a la dirección bitcoin de destino.

3.1.4 Manipulación de Bitcoins

Para comenzar a manejar *bitcoins* es necesario crear una dirección Bitcoin que podrá ser realizado desde un cliente Bitcoin. En función a este cliente, existirán diversos formatos de almacenamiento de la información asociada a una cuenta. Por ejemplo, en el caso del cliente Bitcoin Core, esta información se almacena en un fichero denominado por defecto *wallet.dat*. De esta forma, las principales *wallets* o carteras disponibles son: las locales, en la nube, almacenamiento en frío y carteras mentales.

3.1.4.1 Carteras locales

Son todas aquellas que se instalan en un equipo y que generan y almacenan las claves privadas sin depender de algún almacenamiento externo.

Existen dos tipos de carteras locales, por un lado tenemos las que fueron diseñadas para funcionar de forma independiente como un nodo dentro de la red, por ejemplo: cartera oficial de Bitcoin (Bitcoin Core), y por otro lado están las que dependen de un tercero de confianza para operar con ella, por ejemplo: Green Address.

3.1.4.2 Carteras en la nube

Otra opción es la confianza en servicios de terceros. Existen varias alternativas en la red que ofrecen monederos online por medio de una plataforma. Estas, son intermediarios innecesarios desde el punto de vista técnico en la estructura del protocolo Bitcoin, pero su existencia tiene sentido por el ofrecimiento de servicios complementarios para los usuarios

novatos, por ejemplo: diferentes tipos de cambio entre divisas y criptodivisas, compras y ventas programadas, etc. La desventaja principal de optar por esta opción es que uno no tiene el control total de sus criptodivisas, tiene que confiar en un tercero que centraliza las operaciones y cobra una comisión por la gestión de las mismas.

3.1.4.3 Almacenamiento en frío

Esta opción pretende un almacenamiento de las criptomonedas en equipos offline, a modo de evitar el riesgo de que los bitcoins almacenados se vean afectados por ataques que se realizan a equipos que se encuentran conectados a Internet.

3.1.4.4 Carteras Mentales

Como la palabra lo indica, una forma de almacenar bitcoins en la mente de su dueño. Permite acceder a los bitcoins por medio de un passphrase que se almacenaron en una *brainwallet* y enviarlos a cualquier persona sin necesidad de exponer las claves privadas fuera de la extensión. Es decir, no es necesario intentar mantener segura la *wallet.dat*, ni almacenar largas e ininteligibles URLs. Desde el punto de vista de la seguridad este tipo de carteras son las menos recomendables ya que cualquiera que conozca la contraseña (determinando por ataques de fuerza bruta o diccionario) en cualquier parte del mundo podría firmar cualquier transacción sin consentimiento del propietario.

4. Análisis del anonimato

Las diferentes modalidades de extorsión se acercan cada vez más a ambientes de delincuencia organizada que a la de un delito informático común y corriente.

Un claro ejemplo de esto es la explosión de grupos dedicados a extorsionar a grandes objetivos como plataformas de juego online y sectores financieros mediante la amenaza de preparar ataques distribuidos contra su infraestructura.

El uso de las criptomonedas, en especial Bitcoin, permite también la posibilidad de crear escenarios ficticios en los que se simule la realización de otros pagos para hacer creer a sus víctimas que verdaderamente se están generando transacciones hacia sus direcciones con el objetivo de dar una sensación de que otros están pagando para conseguir convencerlos de que paguen también.

Por otro lado, la extorsión por medio de la amenaza del secuestro de bases de datos con información crítica de las empresas se ha convertido en una moda reciente. Los rescates de estas suelen realizarse en *bitcoins* a través de plataformas de compra y venta ubicadas en la *deep web* y creadas como servicios ocultos de las mismas. Paralelamente, las amenazas de filtración de grandes bases de datos con información de usuarios y contraseñas son inhibidas con pagos en *bitcoins*.

Por otro lado, las fuerzas de seguridad son conscientes de que se están produciendo intercambios de material de abuso infantil a cambio de esta criptomoneda a través de plataformas anónimas.

Para finalizar, todos estos casos son formas de obtener un beneficio ilícito cuyo cobro se realiza en *bitcoins* con el fin de anonimizar el rastro de dicho beneficio. Por lo tanto, no existen dudas de que los organismos de seguridad tienen mucho trabajo en este campo ya que muchas investigaciones probablemente hayan quedado frustradas debido al anonimato que proporciona implícitamente el uso de Bitcoin.

4.1 Traceando Bitcoins

Bitcoin, como visto en la sección anterior anterior, es un sistema con una total transparencia en sus operaciones, principalmente porque el histórico global, es decir, la *blockchain* está disponible y redundante en todos los nodos de la red y accesible para cualquier usuario de Internet. Bajo estas circunstancias, es útil resaltar que dada la posibilidad existente de purgar transacciones para optimizar espacios de almacenamiento, existen también multitudes de nodos, que contienen el histórico completo. Esto resulta funcional para poder verificar con total certeza que no ha habido irregularidades en las transacciones habituales.

Las direcciones en la *blockchain* están pensadas para funcionar como seudónimos con el fin de evitar que el carácter público del historial de transacciones implique de forma directa el poder identificar a alguien, pero eventualmente un usuario que quiera realizar un pago en *bitcoins*, tendrá que proporcionar algún dato identificativo a quien le proporcione el servicio en cuestión, por lo que su identidad quedará enlazada con la dirección que utilice para la realización del pago. En ocasiones similares a ésta, por más que se haya realizado una gestión adecuada de las direcciones Bitcoin, la dirección de pago se podrá utilizar para trazar otras direcciones relacionadas. Por tanto, son varias las fuentes que concluyen que es imposible que una dirección Bitcoin permanezca completamente anónima.

No obstante, desde una mirada criptográfica, esto no se refiere literalmente a que la identidad del propietario de dichas claves permanezca anónima, sino que se refiere a que dichas claves no contienen una identidad real “dentro” de ellas. A pesar de ello, como se ha avanzado y se verá con más detalle a continuación, esto no evita que sea posible (incluso probable, en ocasiones), deducir la identidad real de quien maneja una dirección Bitcoin.

En cualquiera de los casos, resulta relevante destacar que Bitcoin, no requiere la introducción de datos identificativos y que, a diferencia de los sistemas de comercio tradicionales, no existe una autoridad central a la que se pueda preguntar por la identidad real del propietario de una dirección o *wallet*.

4.1.1 Tracking basado en análisis de tráfico

Es posible mediante el análisis del tráfico TCP/IP descubrir la identidad de quien realiza un pago en Bitcoin. Debido al diseño de Bitcoin, la primera persona en anunciar una transferencia será, con alta probabilidad, el pagador de la misma. Por lo tanto, descubriendo quién fue el primero en publicarla, se podrá deducir con gran probabilidad quién es el pagador de dicha transacción y por tanto el propietario de las direcciones de entrada utilizadas.

También es destacable que este ataque está basado en la naturaleza de Bitcoin (que el primero en anunciar una transacción probablemente sea el pagador). Por ello, a Bitcoin por sí mismo le resulta difícil evitar este ataque. Para solucionarlo, no obstante, bastaría con utilizar algún sistema de anonimización de las comunicaciones, como la *darknet*.

4.1.2 Tracking basado en heurísticas

Otro tipo de análisis que se destaca bastante es el que se basa en las relaciones que se pueden establecer entre direcciones que, en algún momento, aparecen como entradas comunes a una transacción. Y es que, dada la construcción de Bitcoin, el hecho de que una entidad utilice varias direcciones Bitcoin como entrada a una misma transacción es garantía de que dicha

entidad controla las claves privadas asociadas a dichas direcciones. Por lo tanto, parece seguro asumir que todas esas direcciones pertenecen a la misma persona.

4.1.3 Análisis de grafo

Se crea un grafo $\tau (T, L)$, donde T es el conjunto de transacciones en la *blockchain* y L es el conjunto de asignaciones directas (relaciones de entrada salida en transacciones) entre estas transacciones. Cada asignación $l \in L$ lleva un número de monedas C_l . De forma inherente las transacciones tienen un orden total definido por el *blockchain*, y no pueden existir ciclos en τ .

4.1.4 Análisis de grafo de direcciones

Por medio del grafo de transacciones se pueden determinar los pares de direcciones origen-destino. Por medio de estas relaciones se obtiene el grafo $\alpha (A, L_0)$ donde A es el conjunto de direcciones de Bitcoin y L_0 es el conjunto de asignaciones directas, pero esta vez conectando direcciones en lugar de transacciones. Opcionalmente se puede transformar a α en un multigrafo al añadir la fecha como un atributo a cada $l \in L_0$ para poder distinguir entre múltiples asignaciones y un par de direcciones.

4.2 Acceso al dispositivo

Para determinar si un dispositivo ha operado con cuentas Bitcoin, es necesario hurgar en en los directorios donde se almacenan las carteras, que si bien la ubicación de las mismas pueden ser configuradas por el usuario, las rutas por defecto estará determinada por el cliente Bitcoin utilizado y el sistema operativo en el que se desplegaron. Por ejemplo:

- Cliente Bitcoin Core con sistema operativo Windows: “C:/Usuarios/<nombre usuario>/AppData/Roaming/Bitcoin”.
- Cliente Bitcoin Core con sistema operativo Linux: “/home/<nombre usuario>/.bitcoin/”.
- Cliente Electrum con sistema operativo Linux: “/home/<nombre usuario>/.electrum/wallets”.

En fin, una vez ubicadas las carpetas, se deberá realizar una copia del archivo con el objetivo de no manipular el archivo original.

Ya obtenido el duplicado del fichero este podrá o no estar cifrado. En el primer caso conlleva la posibilidad de que se cuente o no con el passphrase de descifrado. En el caso de que el fichero esté cifrado y no se cuente con la contraseña, aún así se podrá obtener cierta información de la misma ya que la única información que permanece encriptada es la clave privada con la que el portador firma las transacciones.

No obstante, la información que brinda el archivo obtenido es la siguiente:

- Historial de transacciones: de acá se podrán extraer las direcciones origen, destino y el monto en *bitcoins*.
- Fichero de la cartera: acá se encontrarán las etiquetas/comentarios/información adicional de las transacciones en caso de haberlas agregadas al momento, clave privada genérica y clave pública en Base58Check, si es que ha tenido recepciones de *bitcoins* programadas (monto, partes implicadas y fecha), semillas que permitirán la recuperación de la clave privada (si es que la cartera no fué protegida), tipo de cartera y si está encriptada o no.
- Contraseña: en caso de que el usuario no haya protegido la cartera podría extraerse fácilmente la clave privada.

En caso de que la cartera se encuentre cifrada, habrá que recurrir a diferentes métodos como la fuerza bruta, ataques de diccionario o alguna herramienta de cracking de contraseñas como ser “JohnTheRipper” que incluye un módulo que permite la utilización de la potencia de la herramienta con carteras Bitcoin. Por ejemplo, en una máquina en la que se tuviese funcionando el cliente Bitcoin Core, se podría utilizar el cliente bitcoin-cli con el comando **walletpassphrase** para realizar búsquedas sucesivas de un número determinado de palabras.

En cualquier caso habrá que determinar si el objetivo de la investigación es la incautación de los fondos que contiene la cartera o no, ya que para el momento de la incautación o acceso al equipo, el propietario de la cuenta bitcoin podría haber realizado un backup y realizar las transferencias de sus *bitcoins* a otra dirección no intervenida.

4.3 Identificando carteras

Los nodos con carteras locales completas necesitan tener conectividad para acceder a información de la red de Bitcoin y mantener sincronizada la *blockchain*. No obstante, al tener información que una máquina está en la red de Bitcoin es un indicio suficiente para determinar que allí hay en circulación bitcoins.

En Bitcoin Core, los nodos más cercanos de la red se identifican en el archivo *peers.dat*. Para que el cliente conozca más nodo, el protocolo de descubrimiento envía a estos nodos una petición para descubrir más nodos. Cuando estos nodos reciban dicha petición, estos informarán de aquellos nodos más cercanos a ellos, y así sucesivamente.

Existen aplicaciones de acceso público que permiten dimensionar la red a nivel mundial por medio de la utilización de este protocolo de descubrimiento, por ejemplo: *Bitnodes*, que presenta una vista en el mapa mundial de todos los nodos Bitcoin descubiertos.



Figura 3 -Vista sitio web bitnodes.earn.com

Dada una investigación forense, la aplicación web de la *Figura 3* podría ser utilizada para determinar a partir de una dirección IP si es que esta conectada a esta red o bien consultar el registro histórico de la misma cruzando los datos con la información proveída de *blockchain.info*.

No obstante, también se podrían cruzar los datos la información proveída por servicios como Shodan¹ o Mr. Looquer² sabiendo que en general estos nodos responden como un User-Agent que incluye la palabra Satoshi junto con la versión del protocolo que están ejecutando.

En consiguiente, podrían realizarse búsquedas por puertos por defectos de los diferentes clientes Bitcoin, como es el caso de Bitcoin Core con puertos por defecto en 8332 y 8333.

4.4 Sin Acceso al dispositivo

Generalmente a la hora de realizar investigaciones que tengan que ver con criptomonedas, no se cuenta con acceso físico al equipo, por lo tanto, es conveniente comenzar por las fuentes públicas disponibles, es decir, la *blockchain*.

En función del número de consultas que se desea realizar se podrá optar por diferentes exploradores en Internet, como ser blockexplorer.com o blockchain.info (ya anteriormente nombrado). En el caso que se deseen realizar consultas masivas y automatizadas también es posible y no muy complejo implementar un explorador de la cadena de bloques con una API Rest incluida.

Dentro del apartado de los bloques minados recientemente se podrá extraer la altura del bloque en la *blockchain*, tiempo transcurrido desde que se minaron los bloques, número de transacciones dentro de cada bloque, cantidad de bitcoins transferidos, nombre del pool de minería que ha resuelto el bloque y tamaño de los mismos.

Existe la posibilidad también de realizar búsquedas más específicas por medio del buscador de la cadena de bloques, por ejemplo: por altura de bloque, dirección Bitcoin, identificador del bloque, identificador de transacción o direcciones IP.

Otra funcionalidad muy interesante que proveen estas páginas de consultas es la categorización de operaciones en base a la información de contexto que se conoce. Estas mismas tienen la capacidad de etiquetar si un bloque fue sido añadido por uno u otro pool de minería en base a las direcciones IP.

4.5 Pagos en la Deep Web

Una de las soluciones más potentes en el campo de la privacidad y el anonimato, son las redes anónimas y la posibilidad de acceder a servicios que solamente se encuentran disponibles dentro de dichas redes. Actualmente existen algunas soluciones que son interesantes desde el punto de vista de TOR, I2P o Freenet. Se trata de soluciones avanzadas y maduras que existen actualmente en el campo del anonimato y privacidad, en consiguiente, estas tecnologías cuentan con un gran apoyo por parte de la comunidad usuaria.

En la última década el término “deep web” se ha ido popularizando y extendiendo tanto entre la comunidad *hacker*, como entre los usuarios comunes en Internet. No obstante son muchas las premisas erróneas sobre el término en cuestión, ya que en muchas ocasiones se utiliza de forma indistinta a otros términos como “dark web” o “darknet”.

El término “deep web” hace referencia principalmente a contenidos que no se encuentran indexados por los más grandes motores de búsqueda en Internet, no obstante, resulta imposible localizarlos. Los motivos por los cuales cierto contenido no se encuentra indexado por los grandes buscadores como Google y Yahoo pueden ser muy variados, sin embargo, generalmente uno de los motivos es debido a que el contenido es demasiado antiguo, existe mejor contenido y mayormente accedido que otro, o simplemente se encuentra protegido por sistemas de encriptación y/o autenticación. En la *deep web* el contenido se encuentra en

¹ <https://www.shodan.io/>

² <http://mrloquer.com/>

Internet, pero dicho contenido se encuentra accesible por fuera de los motores de búsqueda convencionales, y si dado el caso estuviesen indexados por los mismos, estos serían muy difíciles de hallar.

Por otra parte, el término *dark web* se refiere a contenido no indexable por decisión de los autores, los cuales comparten dicho contenido en otros medios como ser redes privadas o sitios web protegidos por sistemas de autenticación.

Finalmente el término *darknet* pertenece a un subconjunto de *deep web* que representa un espacio protegido por una red privada o al que solamente un número reducido de usuarios autorizados pueden acceder. Estos contenidos no se encuentran indexados por ningún buscador convencional, de hecho, en algunos casos el acceso a algún servicio determinado requiere una configuración especial y poco convencional para acceder a dicha red, por ejemplo el nodo cliente-servidor TOR.

Aunque existe una gran porción de la comunidad que aprovechan estas funcionalidades de privacidad y anonimato para actividades delictivas, como ser pagos con criptomonedas a cambio de algún servicio ilícito, o utilización de estos servicios anónimos para realizar pagos que sean muy difíciles de “*tracear*”, estos proyectos fueron llevados a cabo para todo lo contrario, es decir, para que las personas puedan ejercer derecho a la privacidad y libertad en la navegación, sin censuras ni restricciones al momento de acceder a contenidos que se encuentran prohibidos por el país u organización en donde se encuentran.

No obstante, no significa que estas herramientas sean buenas o malas, simplemente aporta los medios para conseguir diversos fines y lamentablemente, en algunos casos dichos fines incluyen actividades ilegales.

Cuando se realizan acciones ocultas en cualquiera de las *darknets* que incluyan por ejemplo tráfico de contenidos ofensivos o denigrantes, lo mejor que se puede hacer es denunciarlo, o también es posible intentar detectar ciertas vulnerabilidades que puedan ser utilizadas por los cuerpos de seguridad o los organismos pertinentes para la identificación de los administradores del sitio en cuestión. Aunque se trata de servicios que se encuentran en la *darknet*, pueden verse afectados por cualquiera de las vulnerabilidades a las que enfrenta cualquier servicio en Internet, en este sentido no existe ninguna diferencia entre sitios web en Internet y una *darknet*.

6. Trabajos Futuros

En este trabajo final se hizo hincapié en ciertos clientes de criptomonedas en especial de Bitcoin, pero la realidad es que hoy en día, a los pasos a los que avanza la tecnología y crece la comunidad usuaria, es necesario contar con un conocimiento de análisis forenses en todas las formas de utilización, como por ejemplo: dispositivos móviles, otras criptomonedas y las diferentes implementaciones de clientes de las mismas.

Por lo tanto, entre los trabajos futuros para esta línea de investigación, se identifica la estandarización de procedimientos de análisis forense profundo, no solo para criptomonedas, sino orientado a medios por los cuales se utilizan las mismas, como por ejemplo la *darknet*.

7. Conclusiones

Son cada vez más frecuentes aquellos actos de cibercriminalidad cuyo pago se realizan mediante monedas virtuales. Como se vió en el desarrollo de este trabajo final, resaltan los diferentes tipos de actos delictivos como los ataques de denegación de servicio distribuidos, cifrado de discos, robo de información, venta de bases de datos de usuarios y contraseñas, extorsión con contenido sexual, venta de drogas, armas y lavado de dinero.

En el desarrollo de este trabajo se demostró que existe mucha información para extraer desde dispositivos que han estado operando con criptomonedas, no obstante las mismas también suelen utilizarse en mercados ocultos como los ubicados en la *darknet* con el fin de lograr mayor anonimidad posible de forma tal de dificultar la realización de trackings con resultados positivos por parte de los investigadores forenses.

No obstante, también se demostró que en cualquiera de los casos, siempre es posible intentar verificar si una red o equipo está ejecutando algún tipo de cliente relacionado con criptodivisas aprovechando las características del protocolo de descubrimiento y diferentes herramientas de búsquedas existentes en Internet, así como también determinar si además de estar utilizando servicios de criptomonedas están utilizando servicios anónimos como TOR.

Para finalizar, la conjunción *darknet* y **Bitcoin** plantea un reto de gran complejidad desde el punto de vista de la anonimidad y privacidad para las agencias de investigación forense y fuerzas policiales. Su propia naturaleza hace que los procedimientos convencionales sean insuficientes para dicha tarea, siendo imprescindible acomodarlos a un entorno estandarizado para lograr un eficaz y eficiente análisis de las mismas para converger a resultados positivos dado el caso de la ocurrencia de actos criminales.

8. Referencias Bibliográficas

- Adams, M. (2016). *Blockchain: The history, mechanics, technical implementation and powerful uses of blockchain technology*. Matthew Adams.
- Antonopoulos, A. M. (2018). *Mastering bitcoin: Programming the open blockchain*. Beijing ; Boston ; Farnham ; Sebastopol ; Tokyo: O'Reilly.
- Bahga, A., & Madiseti, V. (2017). *Blockchain applications a hands-on approach*. Verlag nicht ermittelbar.
- Blake, I. F., Seroussi, G., & Smart, N. P. (1999). *Elliptic curves in cryptography*. Cambridge University Press.
- Capoti, D., Colacchi, E., & Maggioni, M. (2015). *Bitcoin revolution: La moneta digitale alla conquista del mondo*. Milano: U. Hoepli.
- Choi, Kyung-shick, and Toro-Álvarez Marlon Mike. *Cibercriminología: guía Para La investigación Del Cibercrimen y Mejores prácticas En Seguridad Digital = Cybercriminology: Guide for Cybercrime Investigation and Best Practices in Digital Security*. Universidad Antonio Nariño, 2017.
- Finch, V. (2017). *Bitcoin: The only complete quick & easy guide to mastering Bitcoin and digital currencies*. Auva Press. *Mastering bitcoin for beginners*. Publisher not identified.
- Hoskinson, Charles (2013), "The Mathematician's Defense of Bitcoin: It's Just Another Option".
- Jayawardena, Kasun P. *A Criminological Analysis: Using Real-Time Monitoring to Gather Data on Online Predators*. 2011.
- Jenkins, Y., & Jenkins, Y. (2015). *Bitcoin: Millionaire maker or monopoly money?* Publisher not identified.
- Johnson, E. C. (2017). *Cryptocurrency: The beginner's guide to investing and trading in cryptocurrency*. CreateSpace Independent Publishing Platform.
- Papacharissi, Zizi. *A Networked Self: Identity, Community and Culture on Social Network Sites*. Routledge, 2011.

Satoshi, Nakamoto. (2009). *Bitcoin: A Peer-to-Peer Electronic Cash System*. Retrieved from <https://bitcoin.org/bitcoin.pdf>.

Shane, P. M., & Hunker, J. A. (2013). *Cybersecurity: Shared risks, shared responsibilities*. Carolina Academic Press.

Steinmetz, R.; Wehrle, K (2005). 2. *What Is This "Peer-to-Peer" About?*. Springer Berlin Heidelberg. pp. 9-16.

The Dark Web: Breakthroughs in Research and Practice. IGI Global, Disseminator of Knowledge, 2018.

Vacca, J. R., & Vacca, J. R. (2013). *Computer and information security handbook*. Morgan Kaufmann is an imprint of Elsevier.

Vigna, P., & Casey, M. (2015). *The age of cryptocurrency: How bitcoin and digital money are challenging the global economic order*.

Construcción de un Generador Aleatorio Binario mediante Combinación No Lineal de LFSR

Germán Montejano¹, Pablo García², Andrés Farías.³

¹ FCFMyN (UNSL) – Ejercito de Los Andes 50, San Luis,
gmonte@unsl.edu.ar

² FCEyN (UNLPam) – Uruguay 151, Santa Rosa, La Pampa
pablogarcia@exactas.unlpam.edu.ar

³ DCE (Unlar) – Av. De La Fuente S/N - La Rioja
afarias@yahoo.com.ar

Abstract. El objetivo de este documento es mostrar los primeros avances en el desarrollo de un software generador de bits, a partir de la combinación no lineal de generadores Geffe, que produzca secuencias de alto nivel de aleatoriedad. Dicha combinación persigue la finalidad de romper la linealidad de los registros; para ello se utiliza una función booleana que cumple con una serie de requisitos matemáticos para asegurar las mejores prestaciones criptográficas. Al final del proceso se debe someter a diversas pruebas de aleatoriedad a la secuencia obtenida, para verificar que el generador realmente está entregando los resultados aleatorios esperados, para lo cual se recurre a herramientas estadísticas.

Palabras Claves: Generación de Bits Aleatorios, LFSR, Geffe, Prueba de Aleatoriedad, Complejidad Lineal.

1 Generador Aleatorio mediante Combinación No Lineal de LFSR

1.1 Descripción del Generador Aleatorio

El generador propuesto en este trabajo, está conformado por tres generadores Geffe, cuyas secuencias son combinadas de forma no lineal mediante una función booleana de tres variables $f_{booleana} = f_{booleana}(x, y, z)$. El esquema del generador se indica en la figura 1.

Para esta propuesta se analizan distintas funciones booleanas que son obtenidas mediante un procedimiento de búsqueda.

Las variables x, y, z de la función son las secuencias obtenidas de cada generador Geffe $x = S_{Geffe_1} = S_x$; $y = S_{Geffe_2} = S_y$; $z = S_{Geffe_3} = S_z$.

El propósito de combinar las secuencias $S_{Geffe_1}, S_{Geffe_2}, S_{Geffe_3}$, obtenidas por cada generador Geffe, consiste en lograr una secuencia final $w = S_{Generador}$ con un período y una complejidad lineal muchos mayores de las que se obtendría con cada

uno de ellos trabajando de manera individual.

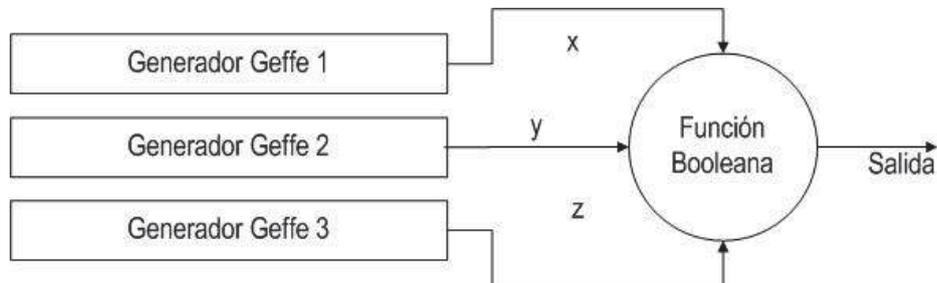


Figura 1. Esquema del Generador Aleatorio

2 Generador Geffe

Como las secuencias obtenidas de Registros de Desplazamiento Lineal con Retroalimentación (RDLR), conocidos por sus siglas en inglés LFSR (Linear Feedback Shift Register) son lineales, se desarrollaron distintas estructuras que permiten obtener secuencias no lineales a partir de los LFSR que componen el generador. Existen tres metodologías principales en ese sentido, (Massodi (et al.) [1], Canteaut [2] y Menezes (et al.) [3]). En particular, el generador Geffe (fig. 2), combina tres LFSR mediante una función booleana que tiene la siguiente forma:

$$f = f(a, b, c) = (c \cdot b) \oplus (b \cdot a) \oplus a$$

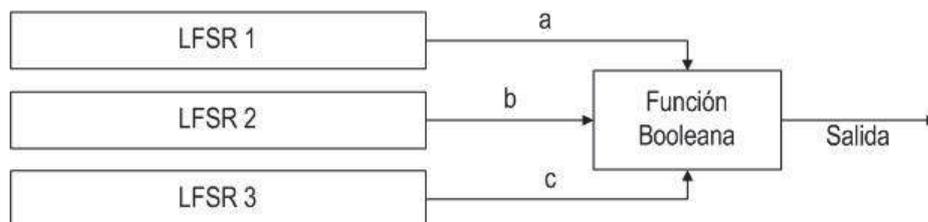


Figura 2. Generador Geffe

2.1 LFSR

La gran mayoría de los sistemas de cifrado de flujo modernos utilizan uno o varios LFSR como bloques de construcción. La principal razón es que tienen un período grande, buenas propiedades estadísticas, y se pueden implementar rápidamente tanto en hardware como en software.

Los LFSR, tienen un estado inicial de carga llamado semilla (clave) y están asociados a un polinomio de conexión que preferentemente debería ser un polinomio

primitivo.

Polinomios Primitivos. Según Parr (et al.) [4]: “Los polinomios primitivos son un tipo especial de polinomio irreducible. Los polinomios irreducibles son aproximadamente comparables con los números primos, es decir, sus únicos factores son 1 y el propio polinomio”

Recomiendan Stahnke [5], Seroussi [6], Mioc y Stratulat [7] la elección de un polinomio primitivo con la mínima cantidad de XOR, ya que de esa manera se incrementa la velocidad del procesamiento.

Período de un LFSR. Las características de cada LFSR son las siguientes. Según indican Wardlaw [8] y Guarino [9], desde una secuencia recurrente lineal de orden L homogénea cuyo polinomio característico sea primitivo. Entonces es periódica y su período es: $p = 2^L - 1$

Complejidad Lineal y Algoritmo de Berlekamp Massey. La complejidad lineal de la secuencia s generada por un LFSR, es el tamaño del LFSR más pequeño que genera esa misma secuencia. Para calcular la complejidad lineal, se recurre al algoritmo de Berlekamp Massey.

Es un algoritmo que proporciona un polinomio de conexión, dado solo unos pocos términos de una secuencia de LFSR, según detallan Campbell [10] y Constantinescu [11]

2.2 Estructura de los Generadores Geffe

Para la selección de los LFSR, que componen cada generador Geffe, además del requisito que los polinomios de conexión sean primitivos, hay otra condición adicional a cumplir y se refiere a que las complejidades lineales de los LFSR que componen cada generador Geffe, sean primos relativos de a dos.

En la tabla 1, se indican los componentes de los generadores:

Tabla 1: Características de los LFSR

Geffe	LFSR	Longitud	Polinomios primitivos	Períodos
1	LFSR 1	23	$P(x)_1 = x^{23} + x^5 + 1$	$2^{23} - 1 = 8388607$
1	LFSR 2	19	$P(x)_2 = x^{19} + x^5 + x^2 + x^1 + 1$	$2^{19} - 1 = 52487$
1	LFSR 3	18	$P(x)_3 = x^{18} + x^5 + x^2 + x^1 + 1$	$2^{18} - 1 = 262143$
2	LFSR 4	22	$P(x)_4 = x^{22} + x^1 + 1$	$2^{22} - 1 = 4194303$
2	LFSR 5	17	$P(x)_5 = x^{17} + x^3$	$2^{17} - 1 = 131071$
2	LFSR 6	25	$P(x)_6 = x^{25} + x^3 + 1$	$2^{25} - 1 = 33554431$
3	LFSR 7	21	$P(x)_7 = x^{21} + x^2 + 1$	$2^{21} - 1 = 2097151$
3	LFSR 8	16	$P(x)_8 = x^{16} + x^5 + x^3 + x^2 + 1$	$2^{16} - 1 = 65535$
3	LFSR 9	23	$P(x)_9 = x^{23} + x^5 + 1$	$2^{23} - 1 = 8388607$

Claves. Para originar las semillas de los distintos LFSR se realiza un proceso, que parte de la clave de una longitud de 32 caracteres, que expresados en código ASCII

(American Standard Code for Information Interchange), tiene 256 bits.

Para simplificar el procedimiento de introducción de la clave, se aceptan solamente las letras del alfabeto (minúsculas y mayúsculas) y los números del sistema de numeración decimal, es decir un total de 62 caracteres permitidos.

3 Generador

3.1 Período del Generador

Gammel, Gottfert y Kniffler [12], proponen que el período del generador es igual al orden del polinomio mínimo m_{S_w} de la secuencia S_w , entonces $período(S_w) = orden(m_{S_w})$. La secuencia S_w , se obtiene de la combinación de las secuencias: S_w, S_y, S_z , que a su vez tienen polinomios mínimos: $m_{S_x}, m_{S_y}, m_{S_z}$, y en el caso que sean primos relativos de a dos, el orden del polinomio mínimo de la secuencia S_w , resulta igual al mínimo común múltiplo del orden de cada polinomio mínimo, es decir $orden(m_{S_w}) = mcm(orden(m_{S_x}), orden(m_{S_y}), orden(m_{S_z}))$.

Además, para los polinomios mínimos se cumple $orden(m_{S_i}) = per(S_i)$. Por esta razón se obtiene: $período(S_w) = orden(m_{S_w}) = mcm(per(S_x), per(S_y), per(S_z))$.

Entonces las tres secuencias obtenidas de los generadores Geffe, al ser combinadas con una función booleana final, tendrán una secuencia resultante cuyo período será igual a $período(S_w) = mcm(per(S_x), per(S_y), per(S_z))$. El valor de los períodos, según al diseño adoptado, será: $período(S_w) = 1,4272E+45$.

3.2 Complejidad Lineal del Generador

Según Gammel, Gottfert y Kniffler [12], la complejidad lineal del generador es igual al grado del polinomio mínimo de la secuencia S_w , $\Lambda(S_w) = grado(m_{S_w})$

El algoritmo de Berlekamp-Massey, permite encontrar el polinomio de menor grado que puede generar una secuencia igual a la que se está analizando. Para el caso en estudio: $\Lambda(S_w) = 1836740$.

3.3 Funciones Booleanas

Las formas en las que se pueden representar las funciones booleanas incluyen: tabla de verdad, vector y forma polar, además de la forma algebraica normal (FNA), las cuales resultarán de interés en este caso. Las funciones booleanas que se necesitan deben ser balanceadas.

Tablas de Verdad de Funciones Balanceadas. Para seleccionar las funciones booleanas a utilizar, se parte de la tabla de verdad balanceadas para tres variables, donde se presentan distintas alternativas de salidas que tiene igual cantidad de ceros y unos. En total se obtuvieron 70 funciones balanceadas, indicadas en la tabla 2.

Funciones Expresadas en FNA. Se determina la forma normal algebraica, a partir de las tablas de verdad anterior, mediante la expresión: $f = A_n \cdot \text{tabla verdad}(f)$

En la tabla 3 se indican los coeficientes de la FNA:

$$f_{FNA} = a_1 1 \oplus a_2 x \oplus a_3 y \oplus a_4(xy) \oplus a_5 z \oplus a_6(xz) \oplus a_7(yz) \oplus a_8(xyz)$$

Tabla 2: Funciones balanceadas

f()	f()	f()
A 1 1 1 1 0 0 0 0	Z 1 1 0 0 0 1 1 1	AY 0 0 1 1 0 1 0 1
B 1 1 1 0 1 0 0 0	AA 1 0 1 0 0 1 1 1	AZ 1 0 0 0 0 1 1 0
C 1 1 0 1 1 0 0 0	AB 0 1 1 0 0 1 1 1	BA 0 1 0 0 0 1 1 0
D 1 0 1 1 1 0 0 0	AC 1 0 0 1 0 1 1 1	BB 0 0 1 0 0 1 1 0
E 0 1 1 1 1 0 0 0	AD 0 1 0 1 0 1 1 1	BC 0 0 0 0 1 1 1 0
F 1 1 1 0 0 1 0 0	AE 0 0 1 1 0 1 1 1	BD 1 1 0 0 0 0 0 1
G 1 1 0 1 0 1 0 0	AF 1 0 0 0 1 1 1 1	BE 1 0 1 0 0 0 0 1
H 1 0 1 1 0 1 0 0	AG 0 1 0 0 1 1 1 1	BF 0 1 1 0 0 0 0 1
I 0 1 1 1 0 1 0 0	AH 0 0 1 0 1 1 1 1	BG 1 0 0 1 0 0 0 1
J 1 1 0 0 1 1 0 0	AI 0 0 0 1 1 1 1 1	BH 0 1 0 1 0 0 0 1
K 1 0 1 0 1 1 0 0	AJ 1 1 1 0 0 0 0 0	BI 0 0 1 1 0 0 0 1
L 0 1 1 0 1 1 0 0	AK 1 1 0 1 0 0 0 0	BJ 1 0 0 0 1 0 0 1
M 1 0 0 1 1 1 0 0	AL 1 0 1 1 0 0 0 0	BK 0 1 0 0 1 0 0 1
N 0 1 0 1 1 1 0 0	AM 0 1 1 1 0 0 0 0	BL 0 0 1 0 1 0 0 1
O 0 0 1 1 1 1 0 0	AN 1 1 0 0 1 0 0 0	BM 0 0 0 0 1 1 0 1
P 1 1 1 0 0 0 1 0	AO 1 0 1 0 1 0 0 0	BN 1 0 0 0 0 0 1 1
Q 1 1 0 1 0 0 1 0	AP 0 1 1 0 1 0 0 0	BO 0 1 0 0 0 0 1 1
R 1 0 1 1 0 0 1 0	AQ 1 0 0 1 1 0 0 0	BP 0 0 1 0 0 0 1 1
S 0 1 1 1 0 0 1 0	AR 0 1 0 1 1 0 0 0	BQ 0 0 0 0 1 0 1 1
T 1 1 0 0 1 0 1 0	AS 0 0 1 1 1 0 0 0	BR 0 0 0 0 0 1 1 1
U 1 0 1 0 1 0 1 0	AT 1 1 0 0 0 1 0 1	
V 0 1 1 0 1 0 1 0	AU 1 0 1 0 0 1 0 1	
W 1 0 0 1 1 0 1 0	AV 0 1 1 0 0 1 0 1	
X 0 1 0 1 1 0 1 0	AW 1 0 0 1 0 1 0 1	
Y 0 0 1 1 1 0 1 0	AX 0 1 0 1 0 1 0 1	

Transformada Rápida de Walsh-Hadamard. El espectro correspondiente puede ser calculado por la expresión: $F(\omega) = \sum_{x \in \mathbb{Z}_2^n} (-1)^{f(x) \oplus x \cdot \omega}$.

3.2 Propiedades Criptográficas Deseables en Funciones Booleanas

A continuación se indican las propiedades criptográficamente más significativas, según Elhosary, Hamdy, Farag y Rohiem [13] y Clark, Jacob, Maitra y Stanica [14].

Función Balanceada. Una función booleana de n-variables f se dice ser balanceada si $w(f) = 2n - 1$. Esta propiedad es deseable para evitar ataques criptodiferenciales. La función es balanceada cuando el primer coeficiente del espectro de Walsh-

Hadamard, es igual a cero: $F(0) = 0$

No Linealidad. Esta propiedad reduce el efecto de los ataques por criptoanálisis lineal. La No Linealidad de una función booleana puede ser calculada directamente de la transformada de Walsh-Hadamard $NL_f = \frac{1}{2} \cdot (2^n - |WH_{max}(f)|)$

Tabla 3: Funciones en FNA (Coeficientes a_i)

f()	1	2	3	4	5	6	7	8	f()	1	2	3	4	5	6	7	8	f()	1	2	3	4	5	6	7	8
A	1	0	0	0	1	0	0	0	Z	1	0	1	0	1	1	0	0	AY	0	0	1	0	0	1	1	0
B	1	0	0	1	0	1	1	0	AA	1	1	0	0	1	0	1	0	AZ	1	1	1	1	0	1	0	0
C	1	0	1	1	0	1	0	0	AB	0	1	1	0	0	0	0	0	BA	0	1	0	1	1	1	1	0
D	1	1	0	1	0	0	1	0	AC	1	1	1	0	1	0	0	0	BB	0	0	1	1	1	0	0	0
E	0	1	1	1	1	0	0	0	AD	0	1	0	0	0	0	1	0	BC	0	0	0	1	1	0	1	0
F	1	0	0	1	1	1	0	0	AE	0	0	1	0	0	1	0	0	BD	1	0	1	0	1	0	0	0
G	1	0	1	1	1	1	1	0	AF	1	1	1	1	0	1	1	0	BE	1	1	0	0	1	1	1	0
H	1	1	0	1	1	0	0	0	AG	0	1	0	1	1	1	0	0	BF	0	1	1	0	0	1	0	0
I	0	1	1	1	0	0	1	0	AH	0	0	1	1	1	0	1	0	BG	1	1	1	0	1	1	0	0
J	1	0	1	0	0	0	0	0	AI	0	0	0	1	1	0	0	0	BH	0	1	0	0	0	1	1	0
K	1	1	0	0	0	1	1	0	AJ	1	0	0	1	1	0	0	0	BI	0	0	1	0	0	0	0	0
L	0	1	1	0	1	1	0	0	AK	1	0	1	1	1	0	1	0	BJ	1	1	1	1	0	0	1	0
M	1	1	1	0	0	1	0	0	AL	1	1	0	1	1	1	0	0	BK	0	1	0	1	1	0	0	0
N	0	1	0	0	1	1	1	0	AM	0	1	1	1	0	1	1	0	BL	0	0	1	1	1	1	1	0
O	0	0	1	0	1	0	0	0	AN	1	0	1	0	0	1	0	0	BM	0	0	0	1	1	1	0	0
P	1	0	0	1	1	0	1	0	AO	1	1	0	0	0	0	1	0	BN	1	1	1	1	1	0	0	0
Q	1	0	1	1	1	0	0	0	AP	0	1	1	0	1	0	0	0	BO	0	1	0	1	0	0	1	0
R	1	1	0	1	1	1	1	0	AQ	1	1	1	0	0	0	0	0	BP	0	0	1	1	0	1	0	0
S	0	1	1	1	0	1	0	0	AR	0	1	0	0	1	0	1	0	BQ	0	0	0	1	0	1	1	0
T	1	0	1	0	0	1	1	0	AS	0	0	1	0	1	1	0	0	BR	0	0	0	0	1	0	0	0
U	1	1	0	0	0	0	0	0	AT	1	0	1	0	1	1	1	0									
V	0	1	1	0	1	0	1	0	AU	1	1	0	0	1	0	0	0									
W	1	1	1	0	0	0	1	0	AV	0	1	1	0	0	0	1	0									
X	0	1	0	0	1	0	0	0	AW	1	1	1	0	1	0	1	0									
Y	0	0	1	0	1	1	1	0	AX	0	1	0	0	0	0	0	0									

Grado algebraico. Es el grado algebraico máximo de una función booleana.

SAC. El criterio de avalancha estricto (SAC por sus siglas en inglés), requiere los efectos avalancha de todos los bits de entrada.

Una función booleana se dice que satisface SAC sí y solo sí $f(x) \oplus f(x \oplus u)$ es balanceada para toda u con $w(u) = 1$.

3.3 Tablas de resultados

Son rechazadas las funciones que tienen los siguientes resultados: espectro de Walsh-Hadamard $F(0) > 0$, no linealidad $Nf = 0$, grado algebraico $g(f) = 1$ y no

cumplen el SAC, ($SAC = No$).

Son seleccionadas, por lo tanto, las funciones indicadas en la tabla 5.

Tabla 4: Resultados

f()	F(0)	NL _f	gf()	SAC	f()	F(0)	NL _f	g(f)	SAC	f()	F(0)	NL _f	gf()	SAC
A	0	0	2	No	Z	0	2	2	No	AY	0	2	2	Sí
B	0	2	2	Sí	AA	0	2	2	No	AZ	0	2	2	Sí
C	0	2	2	Sí	AB	0	0	2	No	BA	0	2	2	Sí
D	0	2	2	Sí	AC	0	0	2	No	BB	0	2	2	No
E	0	2	2	No	AD	0	2	2	No	BC	0	2	2	Sí
F	0	2	2	Sí	AE	0	2	2	No	BD	0	0	2	No
G	0	2	2	Sí	AF	0	2	2	Sí	BE	0	2	2	Sí
H	0	2	2	No	AG	0	2	2	Sí	BF	0	2	2	No
I	0	2	2	Sí	AH	0	2	2	Sí	BG	0	2	2	No
J	0	0	2	No	AI	0	2	2	No	BH	0	2	2	Sí
K	0	2	2	Sí	AJ	0	2	2	No	BI	0	0	2	No
L	0	2	2	No	AK	0	2	2	Sí	BJ	0	2	2	Sí
M	0	2	2	No	AL	0	2	2	Sí	BK	0	2	2	No
N	0	2	2	Sí	AM	0	2	2	Sí	BL	0	2	2	Sí
O	0	0	2	No	AN	0	2	2	No	BM	0	2	2	Sí
P	0	2	2	Sí	AO	0	2	2	No	BN	0	2	2	No
Q	0	2	2	No	AP	0	0	1	No	BO	0	2	2	Sí
R	0	2	2	Sí	AQ	0	0	2	No	BP	0	2	2	Sí
S	0	2	2	Sí	AR	0	2	2	No	BQ	0	2	2	Sí
T	0	2	2	Sí	AS	0	2	2	No	BR	0	0	2	No
U	0	0	2	No	AT	0	2	2	Sí					
V	0	2	2	No	AU	0	0	2	No					
W	0	2	2	No	AV	0	2	2	No					
X	0	0	2	No	AW	0	2	2	No					
Y	0	2	2	Sí	AX	0	0	2	No					

3.3 Funciones seleccionadas y Complejidad Lineal

En la tabla 5, se detallan las funciones seleccionadas, conforme a los criterios de tabla 4, agrupadas, en orden descendente, según la complejidad lineal estimada.

Tabla 5: Funciones seleccionadas y su complejidad lineal

Función	Complejidad	Función	Complejidad	Función	Complejidad	Función	Complejidad
AF	1836741	BJ	1255291	AZ	1241516	AT	1178222
AM	1836740	I	1255290	S	1241515	Y	1178221
G	1836664	AK	1255214	AL	1241420	BE	1178203
BL	1836663	AH	1255213	AG	1241419	N	1178202
R	1836645	D	1254470	C	1240714	T	1177497
BA	1836644	BO	1254469	BP	1240713	AY	1177496

B	1835118	P	1254393	F	1240618	K	1177478
BQ	1835117	BC	1254392	BM	1240617	BH	1177477

4 Pruebas

4.1 Pruebas Básicas para Secuencias Binarias

Se utilizarán cinco pruebas estadísticas básicas, recomendadas por Dragomir [15], para secuencias binarias.

Prueba de Frecuencia (Monobit). El propósito de esta prueba es determinar si el número de 0 y 1 en s es aproximadamente el mismo, como se esperaría para una secuencia aleatoria, el estadístico χ_1 se determina con la expresión: $\chi_1 = \frac{(n_0 - n_1)^2}{n}$. El valor calculado se compara con la distribución χ^2 con un grado de libertad $\nu = 1$.

Prueba de Series. El propósito de esta prueba es determinar si el número de ocurrencias de las cuatro formas (00, 01, 10, 11) de dos bits es aproximadamente el mismo número esperado para una secuencia aleatoria.

Las secuencias aleatorias son uniformes, por lo que cada forma de 2 bits tiene la misma probabilidad de ocurrencia, el estadístico χ_2 se determina con la expresión: $\chi_2 = \frac{4}{n-1} \cdot (n_{00}^2 + n_{01}^2 + n_{10}^2 + n_{11}^2) - \frac{2}{n} \cdot (n_0^2 + n_1^2) + 1$. El valor calculado se compara con la distribución χ^2 con un grado de libertad $\nu = 2$.

Prueba de Póker. Esta prueba se usa para probar la aleatoriedad de un patrón de 3 bits en una secuencia, el estadístico χ_3 se determina con la expresión $\chi_3 = \sum_i^d \frac{(A^{(d)}_i - E_i)^2}{E_i}$, donde $E_i = C_i^d \cdot \left(\frac{L}{2^{d,d}}\right)$. El valor calculado se compara con la distribución χ^2 con un grado de libertad $\nu = 2^d - 1$.

Prueba de Runs. El objetivo de esta prueba es calcular el número total de secuencias ininterrumpidas de bits igual a cero o uno. Una secuencia ininterrumpida de bits idénticos de longitud k es una subsecuencia compuesta por k bits del mismo valor (cero o uno) y que está limitada por un bit de valor diferente. El propósito de la prueba de ejecuciones es determinar si el número de ejecuciones (de ceros o unos) de varias longitudes en la secuencia s es el esperado para una secuencia aleatoria. El número esperado de espacios (o bloques) de longitud i en una secuencia aleatoria de longitud n viene dado por la fórmula $e_i = \frac{(n-i+3)}{2^{i+2}}$.

El estadístico χ_4 se determina con la expresión $\chi_4 = \sum_{i=1}^k \frac{(B_i - e_i)^2}{e_i} + \sum_{i=1}^k \frac{(G_i - e_i)^2}{e_i}$.

El valor calculado se compara con la distribución χ^2 con un grado de libertad $\nu = 2k - 2$.

Prueba de Autocorrelación. El propósito de esta prueba es verificar las

correlaciones entre la secuencia s y las versiones desplazadas (no cíclicas) de la misma. Sea d un entero fijo $1 \leq d \leq \lfloor \frac{n}{2} \rfloor$. El número de bits en s que no es igual a los desplazados d es $A_{(d)} = \sum_{i=0}^{n-d-1} s_i \oplus s_{i+d}$.

El estadístico χ_5 es $\chi_5 = \frac{2 \cdot (A_{(d)} - \frac{n-d}{2})}{\sqrt{n-d}}$.

4.2 Pruebas de los Generadores con las Funciones Booleanas Seleccionadas

A las funciones de la tabla 5, se las somete a pruebas estadísticas en los párrafos anteriores, con 1.000 claves distintas y secuencias de 100.000 de bits y un nivel de significancia de $\alpha = 0,10$. Los resultados obtenidos se muestran en tabla 6.

Cualquiera de ellas puede ser utilizada como función booleana de combinación de las secuencias provenientes de los tres generadores Geffe.

Tabla 6: Funciones que superaron las pruebas estadísticas

$f_{booleana}(x, y, z)$	$f_{booleana}(x, y, z)$
$f_{AF} = 1 \oplus x \oplus y \oplus (x \cdot y) \oplus (x \cdot z) \oplus (y \cdot z)$	$f_C = 1 \oplus y \oplus (x \cdot y) \oplus (x \cdot z)$
$f_{AM} = x \oplus y \oplus (x \cdot y) \oplus (x \cdot z) \oplus (y \cdot z)$	$f_{BP} = y \oplus (x \cdot y) \oplus (x \cdot z)$
$f_G = 1 \oplus y \oplus (x \cdot y) \oplus z \oplus (x \cdot z) \oplus (y \cdot z)$	$f_{AT} = 1 \oplus y \oplus z \oplus (x \cdot z) \oplus (y \cdot z)$
$f_{BL} = y \oplus (x \cdot y) \oplus z \oplus (x \cdot z) \oplus (y \cdot z)$	$f_Y = y \oplus z \oplus (x \cdot z) \oplus (y \cdot z)$
$f_B = 1 \oplus (x \cdot y) \oplus (x \cdot z) \oplus (y \cdot z)$	$f_{BE} = 1 \oplus x \oplus z \oplus (x \cdot z) \oplus (y \cdot z)$
$f_{BQ} = (x \cdot y) \oplus (x \cdot z) \oplus (y \cdot z)$	$f_N = x \oplus z \oplus (x \cdot z) \oplus (y \cdot z)$
$f_{BJ} = 1 \oplus x \oplus y \oplus (x \cdot y) \oplus (y \cdot z)$	$f_T = 1 \oplus y \oplus (x \cdot z) \oplus (y \cdot z)$
$f_I = x \oplus y \oplus (x \cdot y) \oplus (y \cdot z)$	$f_{AY} = y \oplus (x \cdot z) \oplus (y \cdot z)$
$f_D = 1 \oplus x \oplus (x \cdot y) \oplus (y \cdot z)$	$f_K = 1 \oplus x \oplus (x \cdot z) \oplus (y \cdot z)$
$f_{BO} = x \oplus (x \cdot y) \oplus (y \cdot z)$	$f_{BH} = x \oplus (x \cdot z) \oplus (y \cdot z)$

5 Conclusiones

Se desarrolló un generador de secuencias binarias pseudoaleatorias de elevado período y complejidad lineal. Para ello se diseñó un dispositivo que combina en forma no lineal las secuencias producidas por generadores Geffe, mediante una función booleana de tres variables.

Los LFSR que componen cada generador Geffe tienen polinomios de conexión primitivos y las longitudes de los mismos son primos relativos de a pares, lo que asegura un elevado período en la secuencia resultante.

La función booleana que es la responsable de la combinación no lineal, debe asegurar las mejores prestaciones criptográficas, partiendo de funciones balanceadas expresadas en la forma normal algebraica.

Realizado el proceso de selección, las funciones fueron incorporadas al generador y luego puestas a funcionar para generar las secuencias respectivas con distintos valores de claves y ser sometidas a las pruebas de aleatoriedad.

Las funciones que superaron las pruebas, son la que en definitiva pueden utilizar

como función de combinación del generador propuesto.

Los generadores de secuencias pseudoaleatorias, en nuestro caso binarias, son útiles en una amplia variedad de aplicaciones: simulaciones por computadoras, estudio de fenómenos físicos, investigación operativa, simulación de Monte Carlo, pruebas de primalidad, estadísticas computacionales, juegos de azar, juegos de computadora y en los cifrados de flujo.

Referencias

1. Massodi, F., Alam, S., Bokhari, M.: An Analysis of Linear Feedback Shift Registers in Stream Ciphers. *International Journal of Computer Application*. 16 (17). 0975 – 8887 (2012).
2. Canteaut, A. y Filio, E.: Ciphertext only reconstruction of stream ciphers based on combination generators. *Fast Software Encryption 2000, Lecture Notes in Computer Science*. 1978. 165–180 (2001).
3. Menezes, A.; Van Oorschot, P.; Vanstone, S.: *Handbook of Applied Cryptography*. USA: Massachusetts Institute of Technology (1996).
4. Paar, C.; Pelzl, L.: *Understanding Cryptography*. Alemania: Springer. (2010).
5. Stahnke, W.: Primitive Binary Polynomials. *Mathematics of computation*. 27. 124. 977-980 (1973).
6. Seroussi, G.: *Table of Low-Weight Binary Irreducible Polynomials*. Computer Systems Laboratory (1998).
7. Mioc, M., Stratulat, M.: Study of Software implementation for Linear Feedback Shift Register based on 8th degree irreducible polynomials. *International Journal of Computers*. 8 (2014).
8. Wardlaw, W.: *A Matrix Model for the Linear Feedback Shift Register*. Identification Systems Branch Radar Division. NRL Report 9179 (1989).
9. Guarino, S.: *Ciphertext-only reconstruction of LFSR-based stream ciphers*. Facoltà di Scienze Matematiche Fisiche e Naturali, Università degli studi di Roma Tre (2010).
10. Campbell, P.: *An Implementation of the Berlekamp-Massey Linear Feedback Shift-Register Synthesis Algorithm in the C Programming Language*. Sandia National Laboratories (1999).
11. Constantinescu, N.: Combining Linear Feedback Shift Registers. *Annals of University of Craiova, Math. Comp. Sci. Ser.* 36 (2). 42–46 (2009).
12. Gammel, B., Gottfert, R. y Kniffler, O.: *The Achterbahn Stream Cipher*. Infineon Technologies AG (2005).
13. Elhosary, A., Hamdy, N., Farag, I., Rohiem, I.: State of the ART in Boolean Functions Cryptographic Assessment. *International Journal of Computer Networks and Communications Security*. 1. (3). 88-94 (2013).
14. Clark, J., Jacob, J., Maitra, S. y Stanica, P.: Almost Boolean Functions: The Design of Boolean Functions by Spectral Inversion. *Computational Intelligence*. 20. (3). 450-462 (2004).
15. Dragomir, I.: Statistical Assessment of Binary Sequences Generated by Cryptographic Algorithms. *Social Economic Debates*. 5 (2) (2016).

Criptografía Ligera en Internet de las Cosas para la Industria

Eterovic, Jorge; Cipriano, Marcelo; García, Edith, Torres, Luis

Instituto de Investigación en Ciencia y Tecnología. Facultad de Ingeniería.
Vice-Rectorado de Investigación y Desarrollo.
Universidad del Salvador
Lavalle 1854. C1051 AAB. Buenos Aires, Argentina.
{jorge.eterovic; cipriano1.618; edithxgarcia}@gmail.com, torreslu@ar.ibm.com

Abstract. La Criptografía Ligera o Liviana (Lightweight Cryptography) es uno de los temas de actualidad de la Criptología. Una gran variedad de algoritmos “livianos” han sido diseñados para garantizar Confidencialidad, Autenticidad e Integridad de los datos en dispositivos de lo que se ha dado en llamar Internet de las Cosas (IoT por sus siglas en inglés). Algunos de ellos surgen del ámbito académico y se aplican en la Industria; otros son propietarios, desarrollados por las empresas para satisfacer sus requerimientos de seguridad. En este trabajo se presenta el estado del arte de algunos de tales algoritmos empleados en diferentes dispositivos IoT. Se describen brevemente sus características criptológicas generales y se muestran los diferentes ataques a los que fueron sometidos. Finalmente se enumeran algunas de las tendencias para el diseño e implementación de dichas primitivas.

Keywords: Criptografía Ligera, Lightweight Cryptography, Internet de las Cosas, IoT, Internet Industrial de las Cosas, IIoT, Criptosistemas Livianos.

1. Introducción

En la actualidad el concepto “*Internet de las Cosas*” o *IoT*¹ es una de las expresiones más usadas en diferentes ámbitos. Es un término muy amplio que abarca a diferentes dispositivos y tecnologías. Aunque ya se observa un presente con una plétora de estos dispositivos, en un futuro cercano habrá más y más dispositivos aún conectados entre sí. El concepto de *IoT* surgió en 1999 cuando el investigador *Kevin Ashton* se refirió a él en una conferencia; y su relación a las investigaciones en el campo de la *Identificación por Radio Frecuencia* o *RFID*² y en la Tecnología de las Redes de Sensores. En este último caso se conectan una gran cantidad de nodos muy simples con un *Centro de Control (CC)*.

Algunos dispositivos IoT funcionan a batería y/o generan su propia energía usando por ejemplo, paneles solares. Para garantizar la *Confidencialidad*, *Autenticidad* e *Integridad* de los datos e información que viaja a través de los canales de comunicación entre los sensores y su CC, se deben emplear algoritmos criptográficos.

¹ Internet of Things: interconexión de sensores y objetos cotidianos, mediante Internet.

² Radio Frequency IDentification.

Algunos de estos equipos tienen grandes procesadores por lo que es posible usar los mismos algoritmos criptográficos usados en las PCs y Servidores. Por ejemplo los teléfonos celulares, tablets y demás objetos similares. Otros, sin embargo, necesitan procesadores de muy baja potencia, de la cual sólo puede dedicarse una pequeña parte a la seguridad. Asimismo podrían estar limitados en el consumo de energía, capacidad de memoria o en su reducido tamaño, debido a las funcionalidades para los que fueron diseñados. Este tipo de equipos se caracterizan por sus capacidades restringidas o limitadas.

Los algoritmos tradicionales podrían incurrir en un alto consumo de recursos o provocar retardos en la transmisión de datos cuando se los hace trabajar en estos dispositivos. Por lo tanto, dadas las capacidades restringidas y que la seguridad es fundamental para su funcionamiento, los algoritmos criptográficos utilizados deben ser tan “livianos” como sea posible.

Dado que los chips *RFID* son usados para identificación, la información asociada a los chips debe ser cifrada para, entre otras cosas, prevenir el acceso no autorizado a la misma. Debido a que estos cuentan con un número muy pequeño de compuertas lógicas (*GE: Gate Equivalent*) y además tienen muy poca capacidad de consumo energético, se requieren algoritmos criptográficos con diseños específicos.

Es aquí, donde la Criptología Liviana se ha propuesto cubrir las necesidades de seguridad en IoT. Sus diseños varían ampliamente, pero en todos los casos procuran satisfacer los recursos limitados de las “cosas”. Siendo muy amplia la propuesta, es posible en todos los casos establecer una serie de tendencias y criterios que persiguen los algoritmos a la hora de ser implementados.

2. Algoritmos livianos basados en Stream Ciphers

A continuación se muestran algunos algoritmos livianos de tipo *Stream Cipher* utilizados en el ámbito industrial. Se describen las características generales de los mismos y los ataques criptográficos a los que fueron sometidos.

A5/1.

El diseño exacto del algoritmo no fue claro en sus comienzos, una primera aproximación de su funcionamiento interno fue publicada en 1994 [1]. El algoritmo genera un *Keystream* a partir de un *Vector de Inicialización (IV* por sus siglas en inglés) de 22 bits y de una clave secreta de 64 bits, utilizando tres *Registros de Desplazamiento con Realimentación Lineal (LFSR)*³ diferentes cuyas longitudes suman 64 bits. Ataques prácticos se han implementado usando *Time-Memory Trade-Offs* a través de *Rainbow Tables (RTs)*, sacando ventaja del hecho que la *Función de Actualización* de su estado interno no es biyectiva [2]. Se requieren sólo 2^{24} pasos luego de la construcción de las *RTs*. Además 10 bits de la clave fueron siempre seteados a 0 en muchas implementaciones. El protocolo 2G GSM todavía usa este algoritmo.

³ Linear Feedback Shift Registers.

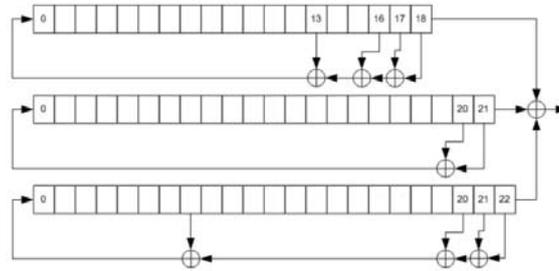


Figura 1: esquema del algoritmo A5/1

A5/2.

El algoritmo es de alguna manera similar al A5/1 pero aún mucho más débil. Fue destinado para ser usado en algunos países con restricciones de exportación de Estados Unidos. Es vulnerable a ataques de “texto cifrado solamente” con una complejidad de 2^{16} pasos, usando la redundancia de los códigos correctores-detectores de errores. Los ataques posibles requieren un pre-cálculo de complejidad práctica [3, 4]. Por su debilidad, el consorcio 3GPP recomienda fuertemente la no utilización de este algoritmo a partir de mediados de 2006.

Familia A5-GMR-x

Estos algoritmos criptográficos fueron adoptados por protocolos de telefonía satelital. Los algoritmos A5-GMR-1 y A5-GMR-2 fueron obtenidos por ingeniería inversa por Driessen *et al.* [5]. Aunque son diferentes entre sí, son fácilmente vulnerados.

A5-GMR-1 es una variante de A5/2 con un estado interno que consiste en cuatro LFSRs cuyas longitudes suman 81bits. Los registros son cloqueados irregularmente como en el A5/2. Puede ser atacado usando solamente “texto cifrado conocido”, invirtiendo 2^{21} matrices triangulares de tamaño 532×532 , teniendo una complejidad aproximada a $2^{38.1}$ operaciones, previamente haciendo un significativo pre-cálculo.

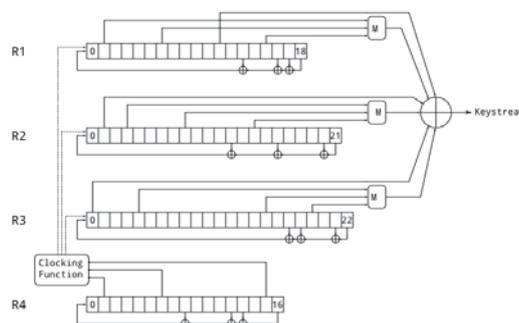


Figura 2: esquema del algoritmo A5-GMR-1

A5-GMR-2 es un stream cipher en bytes, con una estructura más sofisticada basada en 3 componentes llamadas F, G y H por Driessen *et al.* La estructura H utiliza dos S-boxes del algoritmo DES: las cajas 2 y 6. Un ataque práctico con pocos datos y baja

complejidad en tiempo se puede encontrar en [6]. Se necesita suponer a lo sumo 32 bits usando sólo un frame de 15 bytes con un complejidad promedio de 2^{28} .

Atmel Ciphers.

SecureMemory (SM), *CryptoMemory (CM)* y *CryptoRF(CR)* son una familia de chips con amplias aplicaciones. Utilizan stream ciphers similares llamados *Atmel Ciphers*. Son algoritmos propietarios que fueron obtenidos por ingeniería inversa y atacados por *García et al.* [7]. Otros ataques mucho más eficientes fueron propuestos por *Biryukov et al.* [8] quebrando el cifrado de *SM* con una complejidad en tiempo de $2^{29.8}$ segundos, usando 1 frame y el cifrado de *CM* con una complejidad en tiempo 2^{30} segundos, usando 30 frames y alrededor de 530 Mb de memoria. El algoritmo consiste en 3 *NLFSRs*⁴ con un tamaño total 117 bits.

Crypto-1.

Es un stream cipher usado por las tarjetas inteligentes de la línea *Mifare classic* de *NXP Semiconductors*, antes *Philips Semiconductors*. Fue recuperado por ingeniería inversa por *Nohl et al.* [9]. Está basado en un *LFSRs* de 48 bits combinado con varias funciones Booleanas no lineales. Fue sucesivamente atacado por varios grupos de investigación [10, 11] con una complejidad en tiempo de 2^{32} . Estas tarjetas han sido usadas desde 1998 pero la fecha exacta de su diseño no es clara.

Content Scrambling System (Css).

Este algoritmo era aplicado en el cifrado de DVDs para considerar los *Derechos Digitales (DRMs Digital Rights Managements)*. Consiste en dos *LFSRs* de longitudes de 17 y 25 bits que generan dos bytes en paralelo. Luego eran sumados módulo 2^8 para obtener 1 byte de *keystream*. A diferencia de la mayoría de los cifradores de flujo, este *keystream* no es simplemente xoreado con el texto claro. El texto claro primero se modifica pasando por una *S-box* biyectiva de 8 bits cuyo resultado es sumado con el *keystream* para obtener el texto cifrado. Esta operación es a menudo llamada *Mangling Step*. Una descripción completa del algoritmo puede encontrarse en [12, 13]. El algoritmo recibió varios ataques importantes, puesto que la longitud de la clave secreta es de 40 bits. El cifrado solamente es vulnerable a un ataque de fuerza bruta con una complejidad en tiempo de 2^{40} .

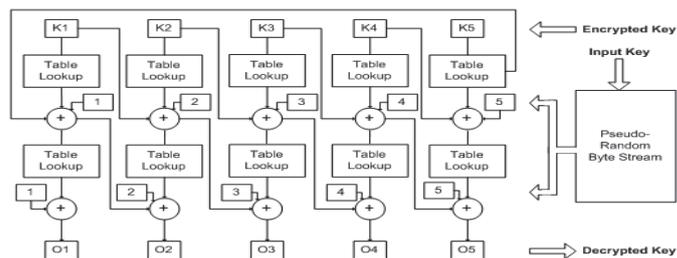


Figura 3: esquema del algoritmo Ccs.

⁴ NonLinear-Feedback Shift Register: Registros de Desplazamiento con Realimentación No Lineal.

Common Scrambling Algorithm (Csa-SC).

El *CSA* es utilizado para proteger la emisión de televisión digital. Consta de dos cifrados en cascada [14]. El primer cifrado es un block cipher el cual se llama *Csa-BC*, el segundo es un stream cipher llamado *Csa-SC*. El stream cipher se basa en dos *FSRs* (*Feedback Shift Registers*) de 20 celdas de 4 bits de contenido. La función de realimentación de los registros involucra, entre otras cosas, varias *S-boxes* de 5x2 bits. Se combinan usando suma módulo 2^4 para extraer 2 bits del *keystream* del estado interno y de los dos registros, en cada clock. Este algoritmo tiene varias vulnerabilidades [14]. A menudo el *keystream* tiene ciclos muy cortos. Es posible recuperar la clave secreta solucionando alrededor de 2^{28} sistemas de 60 ecuaciones lineales con 40 incógnitas lo cual demanda una complejidad de a lo sumo $2^{45.7}$ operaciones.

Dsc.

El *Dect Standard Cipher* (*Dect* “*Digital Enhanced Cordless Telecommunications*”) o *Dsc* es un algoritmo usado para cifrar las comunicaciones de los teléfonos inalámbricos. Los primeros ataques se centraron en su protocolo y en sus fallas en la implementación [15]. Fue paulatinamente descrito por ingeniería inversa y se encontraron ataques que requieren sólo alrededor de 2^{15} muestras de *keystream* y 2^{34} cifrados, lo cual lleva un par de horas para recuperar la clave secreta. En ese trabajo [16] los autores lo describen como “un stream cipher asincrónico con baja complejidad no lineal que utiliza una clave secreta de 64 bits y un IV de 35 bits”. Su estructura basada en *LFSRs* que cloquean irregularmente recuerda al algoritmo *A5/I*.

E0.

La privacidad del protocolo *Bluetooth* está basada actualmente en el algoritmo *AES*, que reemplazó al stream cipher llamado *E0*. Su estado interno de 128 bits está dividido en 4 *LFSRs* y su función filtro tiene un diseño propio. Una descripción completa de *E0* se puede encontrar en los trabajos en donde se muestran ataques contra el algoritmo [17, 18, 19]. *Lu et al.* Encontraron un ataque que permite recuperar la clave secreta usando los primeros 24 bits de $2^{23.8}$ frames, con una complejidad de 2^{38} operaciones.

Espresso.

Es un algoritmo diseñado por *Elena Dubrova* y *Martin Hell* [20] propuesto para proteger las comunicaciones *5G* (*Wireless Communication Systems*). Consiste en dos partes fundamentales: un *NLFSR* de 256 bits en la configuración *Galois* y una *Función de Salida* no lineal de 20 variables. Utiliza una clave secreta de 128 bits y un IV de 96 bits. De acuerdo a los autores, es sumamente liviano: tiene 1497 GE. Procesa 2.22 Gbits/seg y una latencia de 232 ns. La estructura del algoritmo *Grain* y *Espresso* son muy similares a partir de lo cual *Wang* y *Lin* [21] aplicaron un *Slide Attack* usando el análisis del *Grain*, desarrollaron un ataque de IV elegido que les permite recuperar la clave secreta de 128 bits con solo dos pares de IVs relacionados, no más que 2^{42} IVs elegidos y con una complejidad computacional de 2^{64} . Así el stream cipher *Espresso* no es seguro para 128 bits de clave.

Hitag2 y Megamos.

Estos stream ciphers son usados en los *Sistemas de Inmovilización de Automóviles (car immobilizers)* implementados por diferentes empresas que evitan que el motor del auto se encienda a menos que el “*transponder*” se encuentre cerca del vehículo. Inicialmente los algoritmos se mantenían secretos. Luego *Hitag2* fue publicado por *Wiener* y detallado en [22] y *Megamos* obtenido por ingeniería inversa por *Verdult* et al. [23]. Ambos algoritmos cuentan con un estado interno pequeño, de 48 y 57 bits respectivamente. Los tamaños de clave, otras debilidades en los cifradores y en el protocolo que los utiliza, permiten ataques prácticos contra los dispositivos cuya seguridad depende de estos algoritmos. Por ejemplo, es posible atacar una clave de *Hitag2* usando 1 minuto de comunicación entre la clave y el automóvil con alrededor de 2^{35} mensajes encriptados. La clave secreta de *Megamos* puede ser recuperada con una complejidad en tiempo de 2^{48} , aunque ataques aún más eficientes son considerados cuando se tiene en cuenta el método de manejo de las claves usadas en el dispositivo.

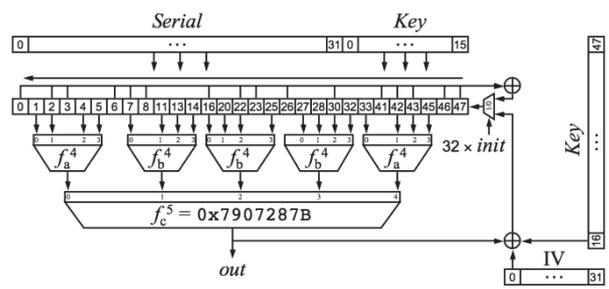


Figura 4: Diagrama del algoritmo HITAG2

iClass.

Es una familia de tarjetas inteligentes comercializadas en 2002. El stream cipher utilizado fue obtenido por ingeniería inversa y atacado por *García et al.* [24]. Tiene un estado interno de 40 bits, registrando 2^{22} intentos de autenticación, la clave puede ser recuperada con 2^{40} cifrados.

Kindle Cipher (PCI).

Este algoritmo fue primero publicado en *Usenet* por *Alexander Pukall* en 1997. Este algoritmo no fue técnicamente diseñado en la industria ni por académicos. La empresa *Amazon* lo utilizó al menos hasta 2012 para los productos con derechos digitales *DRM*, protegiendo así sus e-books usando el formato de archivo *MOBI*. Utiliza una clave secreta de 128 bits y un estado interno de 24 bits que se actualiza usando diferentes operaciones incluyendo multiplicación modular. En cada clock el algoritmo genera 1 byte, resultando así un *keybystream*. Ha sido quebrado por *Biryukov et al.* [25] usando por ejemplo 2^{20} textos claros conocidos y una complejidad en tiempo de 2^{31} . Ataques mucho más prácticos de *texto cifrado conocido* se pueden aplicar en ciertos contextos.

Oryx.

Mientras *A5/1* aseguraba las comunicaciones *GSM* en Europa, el algoritmo *Oryx* fue elegido por *Telecom Industry Association Standard (TIA)* para proteger las comunicaciones telefónicas en USA. Una descripción del algoritmo se puede encontrar en [26] donde ataques prácticos son presentados. El algoritmo utiliza una clave secreta de *96 bits*, un estado interno también de *96 bits* repartidos en *3 LFSRs* de *32 bits* cada uno y una *S-box* de *8 bits* variable. Es posible atacarlo con una complejidad en tiempo de 2^{16} usando *25 bytes* de texto claro conocido.

3. Algoritmos livianos basados en Block Ciphers

A continuación se muestran algunos Block Ciphers livianos utilizados en el ámbito industrial. Se describen sus características y los ataques criptográficos a los que fueron sometidos.

BeepBeep.

El algoritmo *BeepBeep (Embedded Real-Time Encryption)* es un block cipher *autokey*. Trabaja con arreglos de *32 bits* y un IV que tiene diferentes opciones de acuerdo a la aplicación en la que se esté implementando. Fue diseñado por *Kevin Driscoll* de *Honeywell Laboratories* [27] para cubrir los requerimientos de seguridad que incluyen las comunicaciones wireless, manejo remoto de sistemas de control para plantas químicas y de energía, para manejo de redes distribuidas, acceso y control remoto. Salvo la presentación de *BeepBeep* y los requerimientos de criptografía en tiempo real *FSE (Fast Software Encryption)* en 2002, no se ha encontrado información acerca de análisis y ataques de este algoritmo.

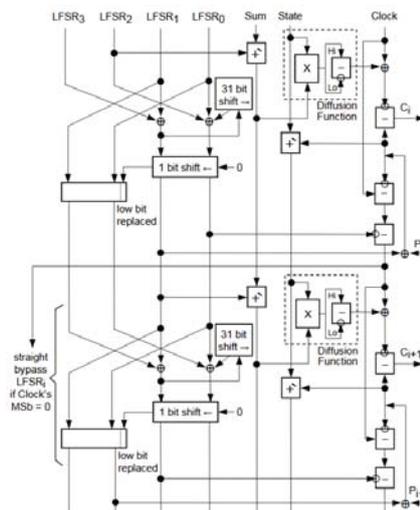


Figura 5: Diagrama de cifrado del algoritmo BeepBeep.

CMEA.

Este algoritmo fue usado por *TIA* para asegurar la transmisión de números de teléfono a través de las líneas telefónicas. Una buena descripción de este algoritmo se puede encontrar en [28], en el cual accidentalmente se describe un ataque contra él. Cifra un bloque de longitud arbitraria en bytes (usualmente en la práctica de 2 a 6 bytes) usando una clave de 64 bits. Es vulnerable a ataques de “texto claro conocido”, necesitando entre 40 y 80 bloques de datos, considerando 2^{24} y 2^{32} cifrados. Sus *S-boxes* parecen contener una estructura oculta según [29].

Cryptomeria.

En forma abreviada se le llama “C2”. Comparte la misma estructura que el *DES* (*Data Encryption Standard*): cifra bloques de 64 bits usando una clave secreta de 56 bits y utiliza una función *Feistel* de 32 bits. Opera mezclando una sub-clave de 32 bits a través de una suma modular, luego usa una caja *S-box* de 8 bits seguida por una permutación lineal de 32 bits. La *S-box* es secreta, de manera tal que un ataque de recuperación de la caja ha sido propuesto en [30]. El mismo trabajo presenta un ataque de recupo de clave de complejidad en tiempo de 2^{48} . Este algoritmo fue destinado a ser usado en reproductores de DVD, en cuyo caso podría entenderse como un sucesor del algoritmo *Css*, y también para algunas tarjetas SD. En total, se utilizan 10 rondas lo que implica que sólo 10 llamadas a la caja son necesarias para cifrar un bloque de 64 bits, comparado con las 160 llamadas a la caja que son necesarias para cifrar un bloque de 128 bits usando *AES-128*.

Common Scrambling Algorithm (Csa-BC).

El algoritmo *Csa* usa un stream cipher (ya mencionado anteriormente) y un block cipher llamado *Csa-BC*. Cifra bloques de 64 bits usando una clave de la misma longitud. Su estructura recuerda a una *Red Feistel* de 56 rondas, usando ocho sub-bloques de 8 bits cada uno. Las funciones están basadas en una única *S-box* aleatoria de 8 bits (llamada *B*), componiendo una variante de ella con una permutación simple de bits llamada *sigma* (σ), obteniéndose $\sigma \circ B$. Una especificación completa del algoritmo se puede encontrar en [14]. Al momento, no se tiene conocimiento de algún tipo de ataque.

Dst40.

Este algoritmo fue recuperado por ingeniería inversa a través de información parcial descubierta en una patente y de su implementación física en un dispositivo [31]. Fue usado en dispositivos de *RFID* vendidos por *Texas Instrument*, para inmovilizadores de automóviles y pago electrónico. El cifrador trabaja con un bloque de 40 bits con una clave de igual longitud, también usando 200 rondas de una *Red Feistel* asimétrica. La función *Feistel* aplica 38 bits del estado interno y 40 bits de una subclave y produce 2 bits de salida componiendo varias funciones booleanas. Debido al tamaño de su clave de 40 bits es posible efectuar un ataque por fuerza bruta.

Keeloq.

También llamado “code-hopping encoder”. Diseñado en 1985, se le concede la patente en el año 1996. Es empleado por muchas compañías de automóviles como *Chrysler*, *Daewoo*, *Fiat*, *GM*, *Honda*, *Toyota*, *Volvo*, *Volkswagen*, en los sistemas de

apertura remota de puertas. Utiliza una clave secreta de 64 bits , y bloques de 32 bits . Es un cifrado iterativo, que consiste en 528 rondas. En sus comienzos el algoritmo era secreto. Pero en 2006 se filtraron sus especificaciones. Con esta información varios grupos de investigación presentaron ataques contra el algoritmo [32]. Por ejemplo, la clave secreta puede recuperarse usando 2^{16} textos claros conocidos y $2^{44.5}$ cifrados. Ataques mucho más eficientes por *canal lateral* se han propuesto para las implementaciones comerciales del cifrado. Una observación interesante es que en algunas marcas de automóviles cifra el texto claro 0 e incrementa la clave sumando 1 al entero representado por los bits de la clave secreta, de esta forma no utilizaría nunca la misma clave. Esto es por supuesto, bastante raro y se podría modelar *Keeloq* por un algoritmo dual en donde se tiene un texto claro de 64 bits y una clave de sólo 32 bits , con lo cual se rompería el algoritmo fácilmente por fuerza bruta.

4. Estrategias de Diseño de Algoritmos Livianos

La *Liviandad* puede ser vista como un conjunto específico de propiedades para tener en cuenta en el diseño de los algoritmos criptográficos livianos. Considerando los avances y los actuales resultados académicos, estos aspectos son adaptados de manera diferente en cada uno de ellos. Si bien hay nuevos aportes, prevalecen técnicas, metodologías y operaciones que son utilizadas en los criptosistemas convencionales.

A la hora de seguir una estrategia de diseño liviano, surgen entre otras cosas, una serie de características criptológicas a considerar como ser: Estructura General del Algoritmo, Esquema de Generación de Sub-claves y las Características de las Operaciones Lineales y No Lineales.

Los algoritmos diseñados a partir de estas propiedades, deben ser seguros, simples, rápidos y de alta performance.

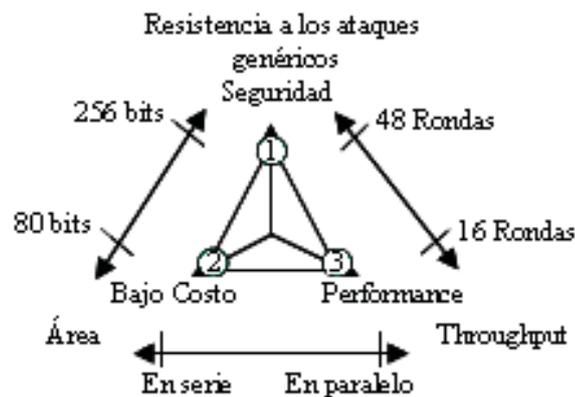


Figura 6: Inspirada en A. Poschmann, Lightweight Cryptography: Cryptographic engineering a pervasive world.

4.1 Estructura General del Algoritmo

De acuerdo a la forma de procesar la información se puede optar por un Cifrado en Bloques (Block Cipher), un Cifrado en Cadena (Stream Cipher) o un mecanismo “*híbrido*” entre los dos principios.

En el caso de los Block Ciphers una de las construcciones más utilizadas es la de SP-Networks (Redes de Sustitución-Permutación) y dentro de ellas las redes tipo Feistel.

Para los *Stream Ciphers*, la construcción es del tipo Vernam: el *texto claro* es sumado XOR con la secuencia de salida binaria pseudoaleatoria del algoritmo.

4.2 Esquema de Generación de Subclaves (Key Schedule)

El *KS* es el área de diferencia más notable entre los algoritmos livianos y los convencionales. Los algoritmos con *Generación de Subclaves* convencionales implementados sobre PCs tienden a ser más complejos por lo que requieren más tiempo de procesamiento y memoria, es por ello que el diseño de *KS livianos* no debe ser desatendido. Una construcción muy habitual de *KS livianos* meramente “selecciona” diferentes bits de una clave maestra secreta en cada ronda del algoritmo. El propósito de esta selección es que se necesite un cálculo muy sencillo para producir una subclave para cada ronda.

4.2 Operaciones Lineales y No Lineales

La no linealidad es una propiedad indispensable para cualquier primitiva criptográfica. Esta puede ser proporcionada por *Tablas de Sustitución (S-Boxes)* o a través del uso de operaciones aritméticas no lineales. Estas deben estar bien diseñadas para resultar ser resistentes a los ataques conocidos. A la vez deben ser *livianas*, esto significa que deben utilizar la menor cantidad de compuertas (*GE*) en su implementación. Para el caso de operaciones aritméticas no lineales, se considera la suma con acarreo módulo 2^{16} , 2^{32} o 2^{64} y también la multiplicación modular.

En la *Tecnología ARX (Addition, Rotation, XOR)* sólo estas tres simples operaciones son usadas en el diseño del algoritmo. Mostrándose como el nuevo paradigma a la hora de construir *rondas livianas*. Se puede agregar a este concepto de no linealidad el uso de *NLFSRs* y *LFSRs clockeados* irregularmente.

4.3 Operaciones Lineales

Para poder garantizar una buena *difusión* de los bits de clave y de los bits del texto claro a lo largo del proceso de cifrado, transformaciones lineales deben aparecer combinadas con las operaciones no lineales. Esta propiedad puede alcanzarse con distintos procedimientos: con transformaciones lineales, con *LFSRs (Registros de Desplazamiento con Realimentación Lineal)* y con matrices *MDS (Maximum Distance Separable)* conocidas en los códigos detectores-correctores de errores, tales como se puede observar en muchos de los algoritmos presentados.

5. Conclusiones y Futuros Trabajos

La Criptografía Ligera es el nuevo paradigma de la Criptología al poder dar respuesta a las necesidades crecientes de seguridad en dispositivos restringidos. En ellos no es posible aplicar los algoritmos más robustos de la Criptografía convencional ya que estos pueden llegar a consumir demasiados recursos.

En este trabajo se han presentado distintos algoritmos livianos que han sido diseñados para cubrir las necesidades específicas de la Industria. Asimismo se destacan sus características criptológicas, la resistencia a diferentes ataques genéricos y los ataques en los que algunos han sido vulnerados.

Finalmente se enumeran algunas tendencias en el diseño de estas primitivas, al observarse las mismas atravesando los principios y filosofías que les dieron origen.

Futuros trabajos de investigación podrían llevar adelante pesquisas respecto a las características en particular de las estrategias aquí presentadas, como así también indagar acerca de nuevas filosofías de diseño que pudieren aparecer en los próximos años.

6. Agradecimientos

El equipo de investigación agradece a las autoridades de la Facultad de Ingeniería de la Universidad del Salvador y al Vicerrectorado de Investigación y Desarrollo (VRID) a través de la Dirección de Investigación y del Instituto de Investigación en Ciencia y Tecnología, (en el cual se enmarca este proyecto Código VRID 1935 – Código Académico100091) por el apoyo recibido para poder llevar adelante esta investigación.

7. Referencias.

- [1] Ross Anderson. A5 (Was: HACKING DIGITAL PHONES). uk.telecom (Usenet), <https://groups.google.com/forum/?msg/uk.telecom/TkdCaytoeU4/Mroy719hdroJ#!msg/uk.telecom/TkdCaytoeU4/Mroy719hdroJ>, June 1994.
- [2] Golic, J. Cryptanalysis of alleged A5 stream cipher. In Walter Fumy, editor, *Advances in Cryptology – EUROCRYPT’97*, volume 1233 of *Lecture Notes in Computer Science*, pages 239–255. Springer, Heidelberg, May 1997.
- [3] Barkan, E. Biham, E. Keller, E. Instant ciphertext-only cryptanalysis of GSM encrypted communication. In Dan Boneh, editor, *Advances in Cryptology – CRYPTO 2003*, volume 2729 of *Lecture Notes in Computer Science*, pages 600–616. Springer, Heidelberg, August 2003.
- [4] Barkan, E. Biham, E. Keller, E. Instant ciphertext-only cryptanalysis of GSM encrypted communication. *Journal of Cryptology*, 21(3):392–429, July 2008.
- [5] B. Driessen, R. Hund, C. Willems, C. Paar, and T. Holz. Don’t trust satellite phones: A security analysis of two satphone standards. In *2012 IEEE Symposium on Security and Privacy*, pages 128–142, May 2012.
- [6] Li, R. Li, H. Li, C. Sun, B. A low data complexity attack on the GMR-2 cipher used in the satellite phones. Pages 485–501.
- [7] Garcia, F. van Rossum, P. Verdult, R. Schreur, R. Dismantling SecureMemory, CryptoMemory and CryptoRF. In *Proceedings of the 17th ACM Conference on Computer*

- and Communications Security, CCS '10, pages 250–259, New York, NY, USA, 2010. ACM.
- [8] Biryukov, A. Kizhvatov, I. Zhang, B. Cryptanalysis of the Atmel cipher in SecureMemory, CryptoMemory and CryptoRF. pages 91–109.
- [9] Nohl, K. Evans, D. Starbug, S. Plötz, H. Reverseengineering a cryptographic RFID tag. In USENIX security symposium, volume 28, 2008.
- [10] Courtois, N. Nohl, K. O’Neil, S. Algebraic attacks on the crypto-1 stream cipher in mifare classic and oyster cards. Cryptology ePrint Archive, Report 2008/166, 2008. <http://eprint.iacr.org/2008/166>.
- [11] Golic, J. Cryptanalytic attacks on MIFARE classic protocol. In Ed Dawson, editor, Topics in Cryptology – CT-RSA 2013, volume 7779 of Lecture Notes in Computer Science, pages 239–258. Springer, Heidelberg, February / March 2013.
- [12] Becker, M. Desoky, A. A study of the DVD content scrambling system (CSS) algorithm. In Proceedings of the Fourth IEEE International Symposium on Signal Processing and Information Technology, pages 353–356. 2004.
- [13] Pedersen, L. Munk, K. Andersen, L. Cryptography – the rise and fall of DVD encryption. Available online at <http://citeseerx.ist.psu.edu/viewdoc/download;jsessionid=3672D97255B2446765DA47DA97960CDF?doi=10.1.1.118.6103&rep=rep1&type=pdf>, 2007.
- [14] Weinmann R., Wirt, K. Analysis of the DVB Common Scrambling Algorithm. In David Chadwick and Bart Preneel, editors, Communications and Multimedia Security: 8th IFIP TC-6 TC-11 Conference on Communications and Multimedia Security, Springer US. 2004.
- [15] Lucks, S. Schuler, A. Tews, E. Weinmann, R. Wenzel. M. Attacks on the DECT authentication mechanisms. In Marc Fischlin, editor, volume 5473 of Lecture Notes in Computer Science, pages 48–65. Springer, Heidelberg, April 2009.
- [16] Nohl, K. Tews, E. Weinmann, R. Cryptanalysis of the DECT standard cipher. In Hong and Iwata [HI10], pages 1–18.
- [17] Fluhrer, S. Lucks, S. Analysis of the E0 encryption system. In Vaudenay and Youssef [VY01], pages 38–48.
- [18] Lu, Y. Vaudenay, S. Faster correlation attack on Bluetooth keystream generator E0. In Matthew Franklin, editor, Advances in Cryptology – CRYPTO 2004, volume 3152 of Lecture Notes in Computer Science, pages 407–425. Springer, Heidelberg, August 2004.
- [19] Lu, Y. Meier, W. Vaudenay, S. The conditional correlation attack: A practical attack on bluetooth encryption. In Victor Shoup, editor, Advances in Cryptology – CRYPTO 2005, volume 3621 of Lecture Notes in Computer Science, pages 97–117. Springer, Heidelberg, August 2005.
- [20] Dubrova, E. Hell, M. Espresso: A stream cipher for 5g wireless communication systems. Cryptography and Communications, 9(2):273–289, 2017.
- [21] Wang, M. Lin, D. Related Key chosen IV Attack on Stream Cipher Espresso Variant. IEEE International Conference on Computational Science and Engineering (CSE) 2017.
- [22] Verdult, R. Garcia, F. Balasch, J. Gone in 360 seconds: Hijacking with hitag2. In Proceedings of the 21st USENIX Conference on Security Symposium, Security’12, pages 37–37, Berkeley, CA, USA, 2012. USENIX Association.
- [23] Verdult, R. Garcia, F. Ege, B. Dismantling Megamos crypto: Wirelessly lockpicking a vehicle immobilizer. In Supplement to the 22nd USENIX Security Symposium (USENIX Security 13), pages 703–718. USENIX Association, August 2013.
- [24] Garcia, F. de Koning Gans, G. Verdult, R. Wirelessly lockpicking a smart card reader. International Journal of Information Security, 13(5):403–420, 2014.
- [25] Biryukov, A. Leurent, G. Roy, A. Cryptanalysis of the “kindle” cipher. In Knudsen and Wu [KW13], pages 86–103.
- [26] David Wagner, Leone Simpson, Ed Dawson, John Kelsey, William Millan, and Bruce Schneier. Cryptanalysis of ORYX. In Stafford E. Tavares and Henk Meijer, editors, SAC

- 1998: 5th Annual International Workshop on Selected Areas in Cryptography, volume 1556 of Lecture Notes in Computer Science, pages 296–305. Springer, Heidelberg, August 1999.
- [27] Driscoll. BeepBeep Embedded Real-Time Encryption. International Workshop on Fast Software Encryption. In FSE2002:Fast Software Encryption. pp.164-178.
- [28] Wagner, D. Schneier, B. Kelsey, J. Cryptanalysis of the cellular encryption algorithm. In Burton S. Kaliski Jr., editor, Advances in Cryptology – CRYPTO’97, volume 1294 of Lecture Notes in Computer Science, pages 526–537. Springer, Heidelberg, August 1997.
- [29] Perrin, L. More reverse-engineered S-boxes. Presentation at the rump session of ESC’2017. Slides available at <https://www.cryptolux.org/mediawiki-esc2017/images/2/2e/Rump.pdf>, 2017.
- [30] Borghoff, J. Knudsen, L. Leander, G. Matusiewicz, K. Cryptanalysis of C2. In Halevi [Hal09], pages 250–266.
- [31] Bono, S. Green, M. Stubblefield, A. Juels, A. Rubin, A. Szydlo, M. Security analysis of a cryptographically-enabled RFID device. In Proceedings of the 14th Conference on USENIX Security Symposium - Volume 14, SSYM’05, pages 1–1, USA, 2005.
- [32] Indestege, S. Keller, N. Dunkelman, O. Biham, E. Preneel, B. A practical attack on KeeLoq. In Nigel P. Smart, editor, Advances in Cryptology – EUROCRYPT 2008, volume 4965 of Lecture Notes in Computer Science, pages 1–18. Springer, Heidelberg, April 2008.

Detección de canales encubiertos en la Capa de Red

Ignacio Martín Gallardo Urbini^{1,2}, Santiago Manuel Rudi³

¹ Centro de Investigación y Desarrollo de Software Operacional del Ejército Argentino

² Facultad de Ingeniería del Ejército - Universidad de la Defensa Nacional

³ Facultad de Ingeniería - Universidad de Palermo

⁴ Facultad de Ingeniería - Universidad Nacional de La Plata

Ignacio Martín Gallardo Urbini, igallardo@est.iue.edu.ar

Santiago Manuel Rudi, srudi@palermo.edu

Resumen. La presente investigación muestra el desarrollo e implementación de una técnica para detectar canales encubiertos en la capa de red del modelo OSI, por medio de la aplicación de algoritmos de machine learning. Junto con la definición del Protocolo de Internet, se realiza un análisis a fondo de aplicabilidad esteganográfica en el mismo, implicando posibles comunicaciones encubiertas entre un emisor y un receptor. Se abordan también tecnologías de control, análisis de tráfico y aprendizaje automático con el objetivo de detectar canales laterales de comunicación dentro del marco de Internet Protocol. Las aplicaciones de esta técnica quedarán asentadas con un caso de estudio para validar el modelo.

Palabras-clave: Esteganografía, Protocolo de Internet, Comunicaciones Encubiertas, Criptografía, Machine Learning.

1 Introducción

Eran mediados del siglo XIV, la dinastía Yuan reinaba el territorio chino, y los gobernantes, a fin de asegurar su dominio, imponían la orden de que cada diez familias usaran un solo cuchillo en su vida cotidiana (entre otras atrocidades y represiones) con el objetivo de que la gente quedase sin armas de metal por si querían levantarse en rebelión. Los habitantes hartos de las injusticias, decidieron sublevarse; para ello, los organizadores concibieron la idea de promover a los vecinos a regalarse mutuamente unas tortas llamadas “moon” en vísperas de las fiestas de otoño. Dentro de las tortas se puso una pequeña octavilla con un mensaje táctico para actuar en conjunto. La rebelión fue un éxito total y derivó en el establecimiento de la dinastía Ming.

El individuo, desde el momento en que experimentó la motivación de desenvolverse socialmente, sintió la necesidad de comunicarse, pero muchas veces, dicha comunicación debiera ser secreta para cierta porción de la sociedad; es decir, de alguna u otra forma debió comunicarse de manera oculta o encubierta con algunas personas, sin que otras se enteren de dicha interacción.

Eficientemente, los pobladores chinos pudieron comunicarse de manera incógnita para llevar a cabo la organización de la revuelta, logrando que dicha interacción pase desapercibida para los integrantes de la dinastía Yuan.

En la presente investigación, se abordará el concepto de *comunicación esteganográfica* como la ciencia de la *comunicación encubierta*. Se la llamará “ciencia”, y no “arte”, ya que este último pertenece a una condición humana personal que pudiendo o no involucrar conocimientos previos, tiene aptitud para generar hechos nuevos, pero cuando comienza a regirse por sus leyes, deja de ser arte para transformarse en ciencia.

El objetivo de este desarrollo apunta a realizar un análisis de aquellos campos en el encabezado IP donde es factible adicionar información de forma tal de crear un canal lateral, sin afectar la funcionalidad del protocolo.

Para comenzar, se presentará un breve marco teórico con el objetivo de incorporar las diferentes definiciones necesarias para comprender el desarrollo. No obstante, se presentarán conceptos sobre IPv4 (Internet Protocol Versión Cuatro), esteganografía, análisis de flujo de paquetes de datos en la red y machine learning.

Luego, para validar la propuesta, se llevará a cabo el diseño contextual de aplicación de un componente de software de comunicación encubiertas vía capa de red, otro módulo de software para análisis de flujos de datos y otro para aplicar algoritmos de machine learning con el fin de detectar canales laterales de comunicación.

Para finalizar, se mostrarán y compararán los resultados de las pruebas, se abordarán las conclusiones, líneas futuras de investigación y referencias bibliográficas utilizadas para la redacción del contenido de este documento.

2 Esteganografía y Criptografía

Se definirá entonces, *comunicación encubierta* o *esteganográfica*, al acto de comunicación entre dos o más actores, a través de un canal determinado de comunicación, sin que algún tercero con acceso a ese canal tenga conocimiento de dicha interacción. Se llamará *esteganografía*, a la disciplina que permite escribir en forma encubierta utilizando procedimientos científicos, y ocultando así mensajes u objetos dentro de otros llamados *portadores*, de modo que no se perciba su existencia.

Si bien la *esteganografía* suele confundirse con la *criptografía*, por ser ambas parte de los procesos de protección de información, son disciplinas distintas, tanto en su forma de implementar como en su objetivo mismo. La *criptografía* es la ciencia de la escritura enigmática, se utiliza para cifrar información de manera que sea ininteligible para un intruso subrepticio, pero éste mismo sabe que hay información cifrada, en cambio la *esteganografía* oculta la información en un portador de modo que no sea advertido el hecho mismo de su existencia y envío, y de esta forma, un intruso ni siquiera sabrá que se está transmitiendo información sensible. No obstante, la *criptografía* y la *esteganografía* podrían utilizarse de manera que una complementa a la otra.

El objetivo principal de la comunicación esteganográfica es ocultar y enviar información sensible a través de un portador, logrando que ésta pase totalmente inadvertida para quienes no participan de la comunicación encubierta.

Para que un *emisor* y un *receptor* logren una comunicación esteganográfica, es menester la existencia de diferentes *canales*: *canal oculto* o *encubierto* y *canal público* o *legítimo*, donde el primero se define como el canal de comunicación en el que no se envía la información pertinente para la comunicación convencional. El segundo término o *canal público*, se refiere al ducto de datos que tienen como fin que la información de una comunicación convencional viaje por medio del mismo. Para ejemplificar, en una foto, una marca de agua no visible a simple vista en la foto pertenecería al canal oculto de comunicación, y la foto en sí al canal público.

El *emisor* es quien comienza el envío de mensajes por medio de un canal encubierto, siendo así, el *receptor* quien realizará el procedimiento inverso para obtener el mensaje.

Las comunicaciones encubiertas tienen sus bases dadas la existencia de una interacción oculta entre el emisor y receptor. Esto implica básicamente, que existe una interacción a la vista de todos o pública, que explota una interacción oculta para comunicar un mensaje de manera disimulada a cierta porción de ese público. Por ende, a la vista de cualquiera simula que la comunicación real, consiste en los mensajes que actúan como *portadores*.

El *portador*, se define como el conjunto de datos que va a ser utilizado para adulterarlo. Al introducir un mensaje sensible, se mantiene oculto a partir de la aplicación del *algoritmo esteganográfico*; directamente relacionado con una *clave esteganográfica*, que permite determinar la forma en que se aplica. Tanto el emisor como el receptor, deberán predefinir antes de la comunicación el *algoritmo esteganográfico* con su *clave esteganográfica*.

Al acto de introducir un mensaje en un portador llamaremos *infiltrar*, mientras que al acto inverso, en donde se recupera la información oculta del portador se denominará *filtrar*.

El término *mensaje legítimo* se utilizará para referirse a la información transportada por el portador en sí, mientras que el mensaje a *infiltrar* en el acto esteganográfico se definirá como *mensaje sensible*.

Estos conceptos se entienden mejor a partir del relato de *Heródoto*, en su libro *Historias*, contextualizado en la rebelión de las ciudades jonias de Asia Menor, donde Aristágoras, tirano de Mileto, es solicitado por el rey de Persia a organizar una expedición contra la isla griega de Naxos. Pero Histaeus, griego, sabiendo que Aristágoras no permanecerá leal a Persia, quiere ponerse en contacto con él para pedirle que se una a la rebelión contra el rey persa. Para ello, hizo cortar el pelo al más leal de sus esclavos hasta dejarlo rapado, luego tatuó el mensaje secreto en su cabeza y esperó que le crezca el pelo. Con el pelo largo, el esclavo fue enviado a Aristágoras, pasando todos los controles, dado a que no llevaba nada extraño. Para leer el mensaje, Aristágoras le cortó el pelo nuevamente, culminando así con el éxito de la rebelión jonia ante el dominio persa.

Analizando el relato, el esclavo fue utilizado como *portador* por parte del *emisor*, *infiltrando* un *mensaje sensible* en su cabeza representada como *clave esteganográfica*, utilizando el acto de rapar la cabeza como *algoritmo esteganográfico* y al esclavo pasando los controles como el *mensaje legítimo*, *filtrado* posteriormente por el *receptor*.

Actualmente la esteganografía, desde una perspectiva moderna, y en términos informáticos, se refiere a la información o a un *archivo* cualquiera que se encuentre oculto dentro de otro, normalmente multimedial, es decir, el portador es una imagen, *archivo* de audio o de video. Se hace referencia a *archivo* de manera general para referirse a un conjunto de bits que conforman una entidad de información almacenada en algún lugar.

Haciendo referencia a la Fig. 1, el emisor comienza con el proceso de infiltración de un mensaje sensible dentro de un portador determinado, aplicando un algoritmo esteganográfico parametrizado con una clave esteganográfica, creando así un *mensaje esteganográfico*. Este último, es enviado al receptor, donde posteriormente a la recepción aplicará el algoritmo esteganográfico junto con la clave esteganográfica, llamando a esto el proceso de filtrado, obteniendo así el mensaje sensible.



Fig. 1. Arquitectura de una comunicación esteganográfica.

Cualquier actor con acceso al medio de transmisión, tendrá acceso a los *mensajes esteganográficos*, pero solo podrá visualizar el *mensaje legítimo*, y no el *sensible*, ya que este último viaja abstractamente por el *canal oculto*, a diferencia del mensaje convencional que lo hace por medio del *canal legítimo*.

Un canal de comunicación permite explotar posibilidades de comunicaciones esteganográficas cuando es posible asignar otras interpretaciones a los campos que lo conforman, es decir, permite que en ciertos lugares fluya información diferente a la información que debería poseer por especificación. Por ejemplo, si la especificación técnica no aclara qué acción tomar si un campo determinado, toma el valor cero o uno en un contexto en particular.

En material legal, aún no se ha encontrado algún estándar que respalde o especifique a la esteganografía; lo que tiene sentido, ya que gran parte del éxito de una comunicación encubierta o esteganográfica depende estrictamente del factor “secreto”, por lo que no tendría sentido adoptar o establecer un estándar esteganográfico. Dicho esto, los *portadores* de apariencia inocentes no despertaran sospechas a un *espía*, y los *mensajes sensibles* llegarán a destino sin problemas.

En la actualidad, los requerimientos y/o requisitos de un problema de comunicación segura, no toleran siquiera que un espía sospeche que se tiene algo que ocultar. No es suficiente con cifrar por métodos criptográficos la información que fluye en el medio, ya que con esto, lo único que se lograría es que el enemigo o espía no tenga acceso directo a dicha información, pero sabe de su existencia; lo que sería terriblemente problemático en situaciones de guerra, en donde, abundan secuestros, torturas, y abusos del poder.

Por otro lado, en el acto de una comunicación esteganográfica, donde el objetivo primordial es que el enemigo ignore la existencia de información; si éste sospecha de que la misma existe, o conoce el método esteganográfico utilizado, podrá extraerla sin demasiadas dificultades. No obstante, la idea principal es que a una primera vista, se presente la información en alguna forma aleatoria comprimiéndola y codificándola por algún método criptográfico previamente a la aplicación de alguna técnica esteganográfica. Sometiendo los *mensajes sensibles* a estos dos procesos, si un *espía esteganográfico* o criptoanalista sospecha de la existencia de los mismos, y conoce o logra descifrar el método esteganográfico utilizado, extraerá la información y se encontrará con un conjunto de bits de aspecto aleatorio, por lo tanto no probará con certeza si existe un mensaje sensible, si aplicó el método de *filtrado* correcto, o si simplemente extrajo “basura”.

Queda a destacar, que dada la facilidad algorítmica de detección, y la complejidad de hallazgo que presenta la esteganografía, se caracterizará por ser sutilmente una gran *estrategia evasiva*. Se interpretará una *estrategia evasiva*, como el conjunto de acciones planificadas sistemáticamente en el tiempo, que se llevan a cabo para lograr un determinado fin o misión, para eludir cierta dificultad, compromiso o peligro.

De todas formas, no existe seguridad absoluta. Hoy en día, los espías experimentados son capaces de burlar lo impensado. Es por eso que dependerá de la habilidad o ingeniería personal que uno presente en el acto de ocultar información en una comunicación crítica o de la habilidad y astucia del espía que intenta descubrir la existencia de un canal encubierto de información.

3 End to End over IP

Desde la consolidación de internet las consideraciones iniciales que se tomaron en las definiciones del modelos OSI¹ se determinó que la función de los protocolos que coexisten sobre los niveles inferiores de un sistema de comunicación entre emisores y receptores intermedios deben encargarse del direccionamiento y envío de los paquetes, sin preocuparse por su contenido, su seguridad y la correcta recepción.

El avance en las telecomunicaciones y el incremento de tráfico en las redes de datos, junto con una serie de hechos en torno a la seguridad hacen que el concepto descrito con anterioridad tenga que ser modificado o nuevamente evaluado dado que es factible sacar provecho de determinadas estructuras de protocolos dentro del stack TCP/IP² y así utilizar el mismo para fines a los que no fue creado.

Dentro estos protocolos de capas inferiores del modelo OSI se encuentra Internet Protocolo (IP) el cual vive en la capa de red y cuya responsabilidad radica en el envío y recepción de los paquetes de datos, dejando a capas superiores el control de errores y congestión. En otros términos, la red no posee inteligencia en cuanto a conexiones, retransmisiones, control de errores ni secuenciamiento y solo se encarga de transmitir lo que los extremos le indican, ya que cada uno de los paquetes de datos son tratados independientemente..

La capa de red provee el servicio de *direccionamiento*, que consiste básicamente en la transmisión de datagramas de bits llamados *paquetes de datos* entre dos extremos. Las *direcciones IP* poseen una longitud fija de treinta y dos bits, divididas en cuatro octetos. Están compuestas por una dirección de red y otra de host.

IP es un protocolo no orientado a la conexión lo que significa que no mantiene ningún registro de los datagramas sucesivos sino que cada uno es tratado de manera independiente , además comúnmente se lo denomina de mejor esfuerzo dado que todos los hosts dentro de la red reciben el mejor servicio posible en ese momento, lo que significa que obtendrán diferentes anchos de bandas y tiempos de respuesta en función del tráfico en la red. Dentro del encabezado IP que puede variar entre 20 y 40 bytes según se utilice o no el campo Opciones, tiene como único parámetro de seguridad una suma de control de cabecera que cumple el servicio de detectar errores que pueden haberse introducido durante la transmisión o el armado del mismo.

0-3	4-7	8-15	16-18	19-31
Versión	Tamaño Cabecera	Tipo de Servicio	Longitud Total	
Identificador			Flags	Posición de Fragmento
Time To Live	Protocolo		Suma de Control de Cabecera	
Dirección IP de Origen				
Dirección IP de Destino				
Opciones				Relleno

Fig. 2. Cabecera Paquete de Datos IP.

4 Análisis de tráfico de red

En estos últimos años se han vivido muchos cambios desde el punto de vista de las comunicaciones digitales. No obstante, por el mismo desarrollo y masiva utilización, las redes son cada vez más susceptibles a ciberataques que mutan en función del tiempo.

Existen una gran variedad y diversidad de ciberataques posibles debido a la cantidad de protocolos y algoritmos que están en juego a la hora de enviar información y muchos de los cuales no fueron definidos centrándose en la seguridad. Por ello, se hace imprescindible diseñar un sistema de seguridad para tratar de detectar y mitigar posibles amenazas que pongan en riesgo la seguridad. Una de las principales herramientas que permite localizar intentos de ataques son los comúnmente llamados “sniffers”, que permiten explorar los

¹ Modelo de Interconexión de Sistemas Abiertos

² Protocolo de Control de Transmisión / Protocolo de Internet

datagramas dentro del stack TCP/IP. Los DPI³ y DFI⁴ son un claro ejemplo a la hora de hablar de analizadores de tráfico, utilizados hace varios años en aplicaciones especializadas en detección de spam, malware y ataques tipo DoS (Denial of Service), etc. En la literatura se suele hablar de DPI para hacer mención tanto al análisis como a las aplicaciones que hacen uso de ese estudio, a su vez previo a este concepto existe un tipo de análisis más simple denominado SPI⁵. Ambos difieren en que capa del modelo OSI se sitúan, haciendo esta aclaración con la Fig. 3.

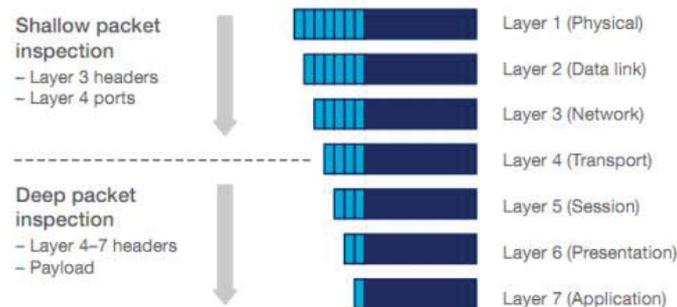


Fig. 3. DPI y SPI ubicados en las capas del modelo OSI

Por tanto, según esta definición se podría catalogar SPI como un conjunto reducido de DPI. Para evitar malentendidos, se limita el uso de SPI para el análisis hasta la cabecera de la capa de transporte, y el uso de DPI como el análisis más allá de la cabecera de ésta, es decir, el análisis de la capa de aplicación e incluso los datos de usuario.

La motivación para el desarrollo de DPI se fundamenta en que la clasificación tradicional basada en el análisis de las cabeceras de los niveles 2 a 4 del modelo OSI (SPI) no es un mecanismo fiable a día de hoy para determinar información sobre el protocolo y la aplicación que transporta un paquete.

DPI permite clasificar el tipo de tráfico en tiempo real, según el protocolo, pudiendo llegar a analizar en base a una base de firmas el tipo de información que se envía en el payload del datagrama.

El avance en la ingeniería de tráfico trajo consigo nuevos conceptos como ser el enmascaramiento (técnica que utilizan ciertos protocolos que hacen un uso no acorde a los “puertos bien conocidos” para evitar su reconocimiento), y el cifrado (técnica por la cual se oculta información obteniendo un criptograma que solo puede ser legible por un tercero que conozca la clave). Debido a esto los métodos de análisis de tráfico nombrados se encontraron con deficiencias en el procesamiento de nuevos tipos de tráfico, por esta razón surge el desarrollo del análisis DFI, que lleva a cabo un análisis heurístico o de comportamiento para determinar qué tipo de tráfico está fluyendo por la red. DFI es capaz de detectar una aplicación (o amenaza) a partir del comportamiento del flujo de paquetes, en lugar de buscar el protocolo o el uso de puertos dentro del mismo paquete. Este hecho es importante, porque cada vez más tráfico está cifrado o es transmitido mediante ‘tunneling’ a través de la red, y más aplicaciones son enmascaradas.

Debido al caso de estudio de este paper se centrará en SPI, dado a que permite como hacer un estudio de los encabezados IP para extraer la información necesaria de los campos que lo componen.

5 Machine learning

El concepto de ciencia de datos engloba una serie de principios, problemas, definiciones y algoritmos para procesar y extraer información de un conjunto de datos (datasets) y encontrar en estos patrones.

La definición común entre las diferentes herramientas es la toma de decisiones en base a un análisis de datos que determina si los eventos se reiteran o están relacionadas en una serie de tiempo, pero también permite predecir comportamiento anómalo o extraño en función de un aprendizaje previo, y es en este punto en que el machine learning se convierte en herramienta para la investigación del siguiente estudio.

Para considerar la complejidad de la obtención de patrones se debe definir en relación con las habilidades humanas. Un humano puede definir reglas en base a pocas variables pero cuando ese número crece se vuelve

³ Deep Packet Inspection

⁴ Deep Flow Inspection

⁵ Statefull Packet Inspection

inviabile obtener una relación entre ellas. En consecuencia la ciencia de datos se aplica a la existencia de grandes volúmenes de variables y la clave del resultado es la correcta obtención de los atributos que se evaluarán.

EL machine learning (ML) se compone de dos fases. Primero se aplican algoritmos a los datasets para identificar patrones, estos se pueden representar de diferentes maneras como ser árboles de decisión, redes neuronales y modelos de regresión. Esta representación se la conoce como modelo. Luego, una vez que el modelo está creado se lo utiliza para el análisis. El desafío del ML es encontrar el algoritmo que mejor sesgo de aprendizaje obtenga para un conjunto de datos.

Dentro del ML se diferencian dos tipos de aprendizajes, supervisado y no supervisado. El aprendizaje supervisado trabaja con datos "etiquetados" intentando encontrar una función que dadas las variables de entrada (inputs) les asigne una etiqueta de salida adecuada. El algoritmo se entrena con un histórico de datos y así "aprende" a asignar la etiqueta de salida adecuada a un nuevo valor, es decir, predice el valor de salida. El otro tipo tiene lugar cuando no se dispone de datos "etiquetados" para el entrenamiento. Sólo conocemos los datos de entrada, pero no existen datos de salida que correspondan a un determinado input. Por tanto, sólo se podrá describir la estructura de los datos, para intentar encontrar algún tipo de organización que simplifique el análisis. por ello, tienen un carácter exploratorio.

Dentro de los algoritmos supervisados que componen el ML se utilizara árboles de decisión por su conveniencia ante problemas de clasificación. Estos se encuentran formados por nodos de decisión (está asociado a uno de los atributos y tiene 2 o más ramas que salen de él, cada una de ellas representando los posibles valores que puede tomar el atributo asociado) y nodos respuesta (está asociado a la clasificación que se quiere proporcionar, y devuelve la decisión del árbol con respecto al ejemplo de entrada).

En los arboles de decision, la decision para una instancia se toma comenzando por la parte superior del árbol y va navegando hacia abajo aplicando una secuencia de pruebas de atributos a la instancia. Cada nodo en el árbol especifica un atributo de prueba, y el proceso continúa nodo a nodo eligiendo qué rama seguir dependiendo del resultado de las pruebas de atributos en cada instancia.

El objetivo del algoritmo de aprendizaje del árbol de decisión es encontrar un conjunto de reglas de clasificación que divide el conjunto de datos de entrenamiento en conjuntos de instancias que tengan el mismo valor para el atributo objetivo.

6 Aplicación Esteganográfica en IPv4

Al analizar la estructura detallada del encabezado de este protocolo (explícito en la **Fig. 2**), el objetivo primordial será hallar segundas interpretaciones de los campos que lo conforman, es decir, interpretaciones que no estén contempladas en la especificación técnica formal del IP, sin que impacten de alguna forma "negativa" en la implementación del mismo. Esto servirá para saber que campos tener en cuenta a la hora de aplicar el algoritmo de aprendizaje.

Es de relevancia aclarar que la cabecera del *paquete de datos IP* resultante de una aplicación esteganográfica (con un *mensaje sensible oculto*) junto con su *carga útil*, será correctamente interpretado en todas las implementaciones de IP, es decir, respetará lo que indican los RFC.

Versión- Campo compuesto por 4 bits. Su objetivo es especificar la estructura que va a tener el paquete de datos, es decir, la estructura soportada por la versión especificada, que en este caso el número de versión será el 4 (cuatro), expresado en binario (0100). Cada terminal o gateway al que llegue este paquete, verificará que el número de versión coincida con su estructura, por lo tanto, si se realiza alguna modificación, el campo no coincidirá con lo especificado, y el paquete IP será descartado. Entonces, este campo no se tendrá en cuenta.

Tamaño Cabecera- Este campo está también compuesto por cuatro bits, fue diseñado para especificar el largo en cuanto a cantidad de filas del encabezado del paquete de datos IP (cada fila con 32 bits). Según la especificación, el número mínimo de filas es "5", y éste, puede variar hasta como máximo "15" (1111 en binario) con el uso de un campo opcional denominado *Opciones*. Ya que el valor de este campo debe coincidir con el largo real del encabezado IP, entonces no permite aplicar técnicas esteganográficas alterando los valores, por lo tanto, no se tendrá en cuenta.

Identificador- Como bien se visualiza en la **Fig. 2**, es un campo de 16 bits, que si se lo eleva al cuadrado provee 65536 combinaciones posibles. Este campo sirve para asignar un identificador único a los paquetes de datos que serán fragmentados para viajar por la red debido a su gran tamaño. El protocolo IP, determina que todos los fragmentos que tengan el mismo identificador, *dirección IP de origen* y *dirección IP de destino* pertenecerán a un mismo paquete de datos, por lo tanto, de esta forma se logrará el reensamblado de los mismos. El procedimiento correcto para que este campo sea interpretado correctamente según su especificación técnica, simplemente consiste en que el emisor asigne un valor entre 0 y 65535 único en el momento que el paquete de

dato se encuentra “viajando” desde el emisor hasta el receptor. En concordancia, se puede inferir que en el *identificador* se podrá ingresar cualquier valor, siempre y cuando se cumpla con la condición de unicidad nombrada antes en ese contexto determinado. No obstante, es posible aplicar eficientemente esteganografía en este campo, ingresando en alguna combinación binaria que traducido a un lenguaje de alto nivel representa un carácter o símbolo.

Tipo de Servicio- Campo formado por 8 bits. El mismo expresa una métrica representada en 2^8 combinaciones posibles para determinar el tipo de servicio que se va a utilizar para la transmisión de los diferentes paquetes de datos en la red, en síntesis, asignar diferentes prioridades en el momento de envío.

En la especificación del protocolo, se puede apreciar que las diferentes combinaciones expresan diferentes opciones importantes de asignación de prioridades, pero lo relevante a destacar es que el bit 6 y el 7 fueron reservados para uso futuro. De acuerdo a ello, para esta versión, norma y contexto de utilización de protocolo, se podrá utilizar para esteganografiar por medio de sus 4 combinaciones (2^2).

Longitud Total- Este campo utiliza 16 bits para determinar el largo total del paquete de datos en sí, medidos en octetos, incluyendo encabezado y carga útil. La mayoría de las implementaciones IP son capaces de procesar paquetes de datos de hasta 576 octetos, por lo tanto, en la especificación se recomienda que los paquetes enviados no superen un tamaño de 64 octetos para la cabecera y 512 octetos para la carga útil (576 octetos en total). Al igual que el campo *Tamaño de Cabecera*, es un campo derivado, ya que va a depender del largo real del paquete de datos tratado en ese momento. No obstante, si llegase un paquete con un largo distinto al especificado en este campo, el gateway lo descartará.

Flags- Representa su funcionalidad mediante 3 bits, de los cuales el bit 1 (el bits que se encuentra en la posición 0, es decir, el más significativo) se encuentra reservado, y según la especificación, en dicho campo debe ir “0”, pero no se aclara nada de qué sucede si dicho flag setea en “1”. El segundo bit, si se encuentra en “1” significa que el paquete de datos no es fragmentable, y si está en “0”, determina que el mismo se puede fragmentar. En general no se recomienda fragmentar dado a la alta demanda de procesamiento y recursos que requiere esta funcionalidad; pero si se da el caso de que este bit sea “1”, y el router en ese momento decide que es necesario fragmentar (por condiciones del medio), pues entonces lo descartará. El bit 3 va a determinar la existencia o no de más fragmentos, por lo tanto, tiene sentido sólo si el bit “no fragmentar” se encuentra en “0”. Aclarar esto, si el bit de “más fragmentos” es igual a “1”, quiere decir que existen más fragmentos, y si está en “0”, es el último fragmento, o que es lo mismo, no existen más fragmentos. Tampoco, en la documentación del protocolo, se especifica sobre qué sucede si el bit de “no fragmentar” y “más fragmentos” están en “1” en el mismo paquete de datos, por lo tanto, si se fuerza al mismo a que suceda esto, no se espera que se altere la integridad del paquete de datos, ni interfiera en la comunicación correcta del protocolo. Se definirá a *Flags* como posible a esteganografiar, ya que atenta con la semántica definida en la especificación técnica del protocolo, pero no rompe con la integridad del datagrama. Vale aclarar que con la modificación de este campo, se traería inconvenientes en la funcionalidad *Path MTU Discovery*⁶, pero como no es considerada esencial, entonces podría darse por alto.

Posición de Fragmento- Formado por 13 bits. Básicamente determina a qué posición del paquete de datos pertenece este fragmento en particular. El primer fragmento poseerá la posición en “0”, ya que este campo posee la posición del inicio del contenido del fragmento respecto del paquete de datos. Este parámetro, está directamente relacionado con el campo *Flags*, y en la especificación no aparecen determinaciones con respecto a que se debe hacer si el campo “no fragmentar” es “1” y en este campo existe un valor diferente a “0”. *Posición de Fragmento*, tiene significado únicamente si el paquete de datos procesado es un fragmento, es decir, “no fragmentar” es “0”. Según la especificación, se indica únicamente que si el campo “no fragmentar” es “1”, entonces tanto *Posición de Fragmento* como el campo “más fragmentos”, deben ir en “0”. Este campo entonces posee posibilidades esteganográficas.

Si bien los dos últimos campos descritos, a simple vista no presentan un gran aporte en materia esteganográfica, una vez desarrollados los campos *Opciones* y *Suma de Control de Cabecera*, se podrá presentar tangiblemente un ejemplo de aplicación esteganográfica en el proceso de fragmentación y reensamblado de paquetes de datos, ya que la utilización de estos campos conforman el conjunto de información de control necesaria para que este procedimiento pueda llevarse a cabo.

Time to Live- Campo compuesto por 8 bits. El objetivo primordial es darle un tiempo de vida (TTL) a los paquetes de datos, y esto se cuantifica en la cantidad de gateway IP por los que se le está permitido “pasar” o ser procesados. Si éste se encuentra en “0”, en el primer gateway al que llega el paquete de datos será destruido/descartado. Según la especificación, el TTL es medido en unidades de tiempo de segundos, y cada host con implementación IP, por la cual es procesado el paquete de datos, deberá descontar en al menos una

⁶ Por sus siglas en inglés, Unidad Máxima de Transmisión, es una operación de control en donde se envían paquetes de datos con tamaños igual al máximo permitido.

unidad a este campo, por más que el tiempo de procesamiento del mismo sea diferente a una unidad de tiempo. El valor máximo que puede tomar este campo es 2^8 , por lo tanto, si un paquete de datos posee este valor, significará que el mismo no podrá atravesar más de “255” gateways. Sobre las bases descritas, se podrá aplicar esteganografía pero tanto, el emisor como el receptor deberán interactuar en un escenario controlado y conocido, como por ejemplo, dado un contexto en el que el emisor y el receptor están separados por dos gateways, si el primero filtra un mensaje en este campo, se sabrá y definirá previamente que el paquete de datos pasará por medio de dos gateway, decrementándose este campo en dos unidades de tiempo al llegar al receptor. En síntesis, teniendo en cuenta las variables, se podría llegar a establecer un canal encubierto con este campo.

Protocolo- Con ocho bits describe el protocolo utilizado en el paquete de datos. Al tratar de modificar este campo, se alteraría la integridad y semántica del paquete. No obstante, no se podrá aplicar esteganografía.

Suma de Control de Cabecera- Se comenzará definiéndolo como no aplicable, ya que cumple con el objetivo de proteger al encabezado del paquete de datos de posibles errores de procesamientos o transmisión. Básicamente, aplica un algoritmo donde toma el encabezado del paquete en palabras de 16 bits (con este mismo campo en “0”), luego se calcula la suma aritmética del complemento a “1” (se suma bit a bit, y al resultado se le agrega el valor del acarreo de la suma) de todas las palabras, y se calcula el complemento a “1” (el/los “0/s” se cambian por “1/s”, y el/los “1/s” por “0/s”) del resultado. Este procedimiento se repite cada vez que el paquete se es procesado por los enrutadores.

Dirección Destino y Dirección Origen- Cada campo posee un tamaño de 32 bits. El primero determina la dirección del receptor, y el segundo, del emisor. Por cuestiones evidentes, no se utilizará este campo tampoco.

Opciones- Campo de longitud variable, puede tomar un valor máximo de 40 bytes, y en algunos paquetes de datos IP la existencia de este campo es optativa. Puede presentarse de dos formas: La primera, un único octeto que determina el tipo de opción. La segunda, es un octeto con el tipo de opción, opcionalmente un octeto con el largo de la opción, y el contenido de la opción. A su vez, el octeto perteneciente al tipo de opción está compuesto por tres partes: 1 bit correspondiente al *flag de copia*, los 2 siguientes determinan la *clase de opción*, y los 5 últimos codifican el *número de opción*. Al mismo tiempo, el *flag de copia* indica si esa opción se copia en todos los fragmentos (si es que existe fragmentación), donde si el mismo es igual a “0” indica que no se copia, y si vale “1” indica lo contrario. La *clase de opción*, posee 4 combinaciones posibles por el hecho de que está compuesto por 2 bits, y las mismas representan: *opción de control*, *reservado para uso futuro*, *opción de depuración y mediciones*, y la última combinación también es *reservado para uso futuro*. La especificación no determina el orden en que deben aparecer las diferentes combinaciones dentro de *clase de opción*. Con estas bases descritas, será posible utilizar este ordenamiento para infiltrar un mensaje esteganográfico.

Relleno- Este campo se utiliza para asegurar de que el encabezado tenga una longitud igual a 32 bits, o más bien, que sea múltiplo de 4 octetos. Este campo se presenta como uno diferente dentro de la descripción del mismo en la especificación técnica de IP, y su estructura es igual a la primer combinación entre *clase de opción*, y *número de opción* descriptos en el campo anterior. No obstante, la esteganografía se aplicará de la misma forma que el campo anterior.

7 Caso de Estudio

El diseño del componente esteganográfico, fue realizado con las características necesarias para lograr una comunicación encubierta entre dos o más participantes. Aplicado a la práctica, se infiltraran los mensajes pertenecientes a la comunicación encubierta en paquetes de datos IP creados específicamente para realizar esta comunicación (proceso de infiltración, desde el lado del emisor), y posterior a la recepción. Con respecto al protocolo a utilizar en capas superiores se definió TCP, pero podría ser indiferente ya que el posterior análisis de flujos de datos será sobre la capa de red. En segundo lugar se diseñó un componente de machine learning donde se lo entrenó con tráfico específicamente generado para su customización, donde el mismo está construido con el algoritmo Random Forest.

Se generó un tráfico libre de anomalías, donde el mismo se montó sobre un entorno de red que a través de un port-mirroring del switch se envía al “netflow probe” el tráfico con el cual generamos las estadísticas. Este dataset se enviará a un colector de datos que es donde posteriormente se realizarán los procesos de ML. Cómo trabajamos en un enfoque de anomalías, lo que se persigue es que las características definan cuál es el comportamiento normal del tráfico y por tanto, lo que se desvíe de ahí se considera anómalo. Estos se definió dado que al tener múltiples variantes de infiltración de información se haría muy dificultoso el entrenamiento del algoritmo para tráfico no benigno.

El caso de prueba se realizó en un ambiente controlado (Red LAN) en donde IP no sufre variaciones considerables y con esto descartamos una posible fragmentación de los datagramas que puedan distorsionar los resultados. En la Fig. 4 se aprecia un diagrama arquitectónico de la solución.

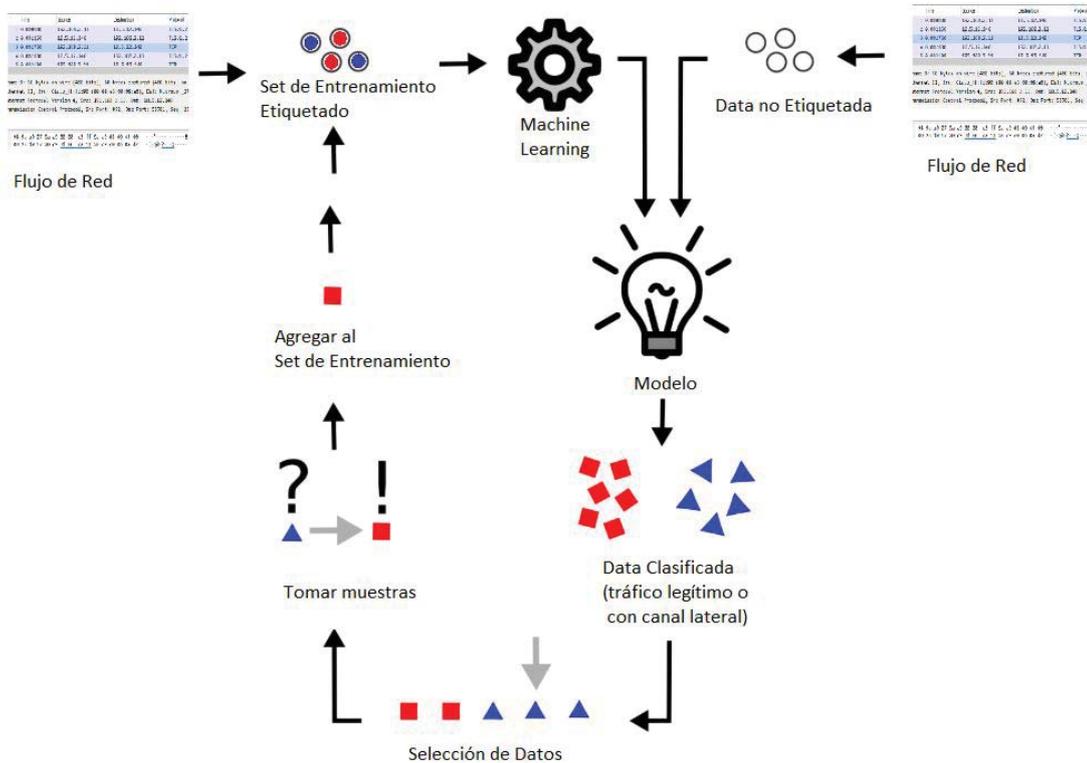


Fig. 4. Arquitectura para la detección de canales encubiertos.

8 Análisis de resultados

Partiendo de un volumen considerable de tráfico de red se entrenó el algoritmo random forest de aprendizaje en donde en cada nodo del árbol se evalúa un campo específico del header IP. En base a lo explicado en los apartados anteriores se determina si coincide con tráfico estipulado por el estándar del protocolo o existe una comunicación lateral.

Según los datos aprendidos por cada campo, se confeccionó en base a los aprendidos un algoritmo de formato top-down en donde en cada instancia de nodos se compara a través de una estructura “if/else” si los datos fueron modificados intencionalmente o no. Dependiendo de la decisión tomada si es positiva la secuencia finaliza en una hoja o nodo de decisión, arrojando que el datagrama se vio comprometido, en caso contrario la iteración continúa nodo a nodo evaluando los campos a analizar.

En las pruebas se obtuvo una efectividad de entre el 98% (con las tramas de 3 días de muestreo) y 91% (con las tramas de 5 días de muestreo) en la identificación de tráfico con canal lateral. Aún así, la versatilidad y los tiempos de ejecución más rápidos del algoritmo clasificador seleccionado lo convierten en una alternativa muy competitiva. Además, se estableció que el método propuesto en el caso de estudio y replicado sigue siendo útil en un contexto controlado dejando para futuras líneas de investigación diferentes topologías o entornos de red en donde entren en juego diferentes flujos de tráfico. El rendimiento de clasificación no puede competir con otros sistemas clasificadores, pero tiene la ventaja de ser totalmente no intrusivo, transparente y no requiere hardware adicional. El método seleccionado parece prometedor, y todavía hay espacio para mejoras adicionales en este contexto particular.

9 Conclusiones y Futuras líneas de investigación

En este trabajo de investigación, se estudió y analizó al Protocolo de Internet desde el punto de vista de su aplicabilidad esteganográfica, para así poder determinar e identificar los campos necesarios a analizar para detectar posibles comunicaciones encubiertas. En el desarrollo, se utilizan reglas o patrones no definidos por la especificación técnica del protocolo para realizar segundas interpretaciones que viola la política de seguridad del mismo. El objetivo principal fue detectar mensajes infiltrados en un canal de comunicación convencional por medio de la aplicación de algoritmos de machine learning.

Sobre estas bases estudiadas, se puede determinar que la posibilidad de aplicabilidad esteganográfica es un factor que se encuentra presente en el protocolo analizado y que también es totalmente detectable bajo la técnica implementada. Así mismo, estas características, deberían ser tenidas en cuenta por los encargados de gestionar la seguridad en el flujo de información en la red, ya que hoy en día no son muy conocidas las potencialidades de explotación de canales encubiertos, en caso de su utilización con fines no legales.

A partir de este estudio crítico de la especificación del Protocolo de Internet, implanta un punto de partida para posibles mejoras en el/los protocolo/s, y/o anticipar futuros inconvenientes en diferentes implementaciones.

Por último, como trabajos a futuro se definieron las siguientes:

- a. Aplicación esteganográfica y detección de canales encubiertos en protocolos de otras capas, como por ejemplo: TCP (protocolo de capa de transporte), o HTTP (protocolo de capa de aplicación).
- b. Implementar la detección de canales encubiertos por medio de otros algoritmos de machine learning y aprendizaje automático.
- c. Aplicabilidad esteganográfica y detección de canales encubiertos en IPv6.
- d. Abordar el estudio del diseño de un componente de análisis de flujo integrado donde también se analice la carga útil del protocolo de comunicaciones.
- e. Investigar aplicabilidad para detección de tráfico malicioso y/o detección de malwares “0 days”.

Referencias

1. Tuomas A.: “Invisibilidad Práctica en Comunicaciones Digitales”. HUT Seminar on Network Security, (1995)
2. Cachin: “Un Modelo de Información Teórico para Esteganografía”, (2004)
3. Pitas I., Voyatzis G.: “Chaotic mixing of digital images and applications to watermarking”. European Conference on Multimedia Applications Services and Techniques, vol. 2, pp. 687–695, (1996)
4. RFC 791 Internet Protocol. <https://tools.ietf.org/html/rfc791> . Accedido el 1 de Julio 2019.
5. Ahsan K. : “Covert Channel Analysis and Data Hiding in TCP/IP”, M.A.Sc. thesis, Dept. of Electrical and Computer Engineering, University of Toronto, (2002).
6. Lenti J.: “STEGANOGRAPHIC METHODS”, (2005)
7. Cole E.: “Ocultamiento de Vista Llana – Esteganografía y el Arte de la Comunicación Encubierta”, (2003)
8. Handel T., Sandford M.: “Hiding data in the OSI network model”, (1996)
9. Kalker T.: “Defending Against Statistical Steganalysis”, Niels Provos, (2002)
10. Cox J., Miller L., Bloom J. A., Fridrich J., Kalker T. : “Marcas de Agua Digital y Esteganografía”, (2008)
11. Shiroshita T., Takahashi O., Yamashita M., “Integrating layered security into reliable multicast, (1996)
12. Kipper G.: “Guía del Investigador de Esteganografía”, (2004)
13. Rowland C. H., “Covert channels in the TCP/IP protocol suite”, (1997)
14. Newman R.: “Informática Oculta y Comunicaciones de la red”, (2007)
15. Fabien A.P., Ross J.: “En los Límites de la Esteganografía”, (1997)
16. Stalling W.: “Comunicaciones y Redes de Computadores”. Décima Edición, (1997)
17. Blitz A.: “Computer Forensics and Investigations”, Cuarta Edición, (2013)
18. Clair B.: “Esteganografía: Cómo enviar mensajes secretos”, (2001)

Ensembling to improve infected hosts detection

Paula Venosa¹, Sebastián García², Francisco Javier Díaz¹

¹ LINTI - Facultad de Informática - Universidad Nacional de La Plata

² Stratosphere Laboratory, AIC, FEL, Czech Technical University

{pvenosa,jdiaz}@info.unlp.edu.ar

sebastian.garcia@agents.fel.cvut.cz

Abstract. In this paper we describe the main ensemble learning techniques and their application in the cybersecurity threats detection. The state of the art in the use of ensemble learning techniques is presented here as an alternative to the current intrusion detection mechanisms, analyzing their advantages and disadvantages. We propose to incorporate ensemble learning to SLIPS [3], a behavioral-based intrusion detection and prevention system that uses machine learning algorithms to detect malicious behaviors, to obtain better results, taking advantage of the benefits of the SLIPS classifiers and modules. As part of this work we extend ensembling by considering algorithms from different domains (not machine learning domains), as Thread Intelligence. As a first stage of this project, performance tests of ensemble learning algorithms were performed to detect malware from flows evaluating its accuracy. The results of these tests are presented here, as well as the conclusions obtained and the future work.

Keywords: Ensemble learning,cybersecurity,malware,intrusion detection

1 Introduction

Detecting malware and attacks by analyzing network traffic remains a challenge. Although there are several detection mechanisms to accurately separate the malicious behavior from the normal one, it is still extremely difficult to have a detection system that can handle all the situations that arise in the network with low error rate. These techniques include machine learning algorithms, static signatures and experts rules based. In particular, the most used method today is based on the contribution of rules by a large community of analysts, called Threat Intelligence. The most important obstacles for a successful detection are: First, normal traffic is extremely complex,

diverse and changing. Second, malicious actions change continuously, adapting, migrating and hiding as normal traffic. Third, the amount of data to analyze is huge, forcing analysts to lose data in favor of speed. Fourth, detection must occur in near real time to be useful. Fifth, there is a large imbalance in the amount of normal traffic compared to the amount of malicious traffic, making very difficult to have good detection results. Sixth, the cost of False Positives errors and False Negative errors is different, further complicating the decision process.

To solve some of these problems, the security learning community proposed the use of ensemble algorithms. These algorithms implement techniques for using, adding and summarizing information about several different detectors in a final decision.

Although there were some good proposals for ensemble learning techniques applied to the security of the network [2], there are two aspects of them that were not fully studied. First, the application of ensemble learning algorithms with Threat Intelligence data (e.g. VirusTotal [3]). Secondly, there are no ensemble learning algorithms that work as a function of time in the detection of the same source hosts.

In the current scenario where it is impossible to stop all threats, the effort should not focus only on protection and prevention, but be directed towards detection and response[4]. In this context, malware continues to represent the main threat and its detection remains one of the main concerns [5].

Given the large amount and exponential growth of malware samples [6] [7] [8] an effective way to detect them is required. The tools that use signature-based methods require maintaining a database to store patterns based on the characteristics of the malware extracted by experts. They present their limitations since a small change in the malware produces a different signature.

Intrusion detection systems have the ability to detect threats in the network but sometimes they are not effective. It is of vital importance that the algorithm used is reliable and can provide high detection accuracy. There are many research works that address this problem using several methods [9].

In order to benefit from multiple different classifiers, and exploit their strengths, we propose the use of ensembling algorithms [10] [11], which combine the results of the individual classifiers into a final result to achieve greater precision and thus a better result. By combining a set of classifiers to decide, the ensemble learning methodology imitates the human nature of seeking various opinions before making a crucial decision, weighing individual opinions and combining them for a better final decision.

2 Ensemble Learning Techniques

Ensemble learning techniques allow combining multiple models, both homogeneous and heterogeneous, with the aim of classifying new instances.

In practice, after constructing a set of classifiers that use some parts of the original dataset, the predictions of the different classifiers are combined to make a final decision. Different schemes are used, either the same algorithm can be trained using different datasets or different algorithms can be trained using the same dataset.

Ensemble learning models combine the decisions of multiple models to improve overall performance. Minimize noise errors, bias and variance. Among the main techniques are some very simple as Majority Voting, Average and Average with Weight[12]. **Majority voting:** In this technique, multiple models are used to make predictions for each tuple. The predictions of each model are considered as a separate vote. The prediction that is obtained from most models is used as the final prediction.

Average: This technique takes an average of predictions of all models and is used to make the final prediction. **Average with weight:** In this technique all models are assigned different weights that define the importance of each model for prediction.

And these are added others such as bagging, boosting, random forest and stacking [13]. **Bagging** (also known as bootstrap aggregating): each classifier is trained with a random subset with replacement of the original training set. With each sample a model is constructed. The results of these models are combined using average or majority voting. **.Bootstrap or boosting:** It is an iterative technique that adjusts the weight of an observation based on the last classification. If an observation was incorrectly classified, try increasing the weight of this observation and vice versa. The first algorithm is trained in the entire data set and subsequent algorithms are constructed by adjusting the residuals of the first algorithm, thus giving greater weight to those observations that the previous model predicted poorly. It is based on the creation of a series of weak algorithms, each of which may not be good for the entire data set, but is good for a part of the data set. Therefore, each model actually increases the performance of the set. In bagging, the samples used are independent, so the algorithms can be run in parallel. Boosting works in a sequential way. **Random forest:** It is a supervised machine learning algorithm based on ensemble learning. It is used for regression and classification. The idea is to build multiple decision trees and

add them to give an exact result. Decision tree is a deterministic algorithm that tend to overfit because it build the best possible tree for given data and fail to generalize when unknown data is provided. Each tree is constructed with a different random subset of our data. Random forest is more accurate than a simple decision tree because it minimizes the overfitting. In the classification problems majority voting is used to combine the predictions while in the regression, the predictions are made taking the average of the tree predictions.

Stacked generalization [13] is a method that uses a different way of combining multiple models introducing the meta learner concept. Stacking is the generalization of other teaching methods. The process is: 1) Split the training into a sample for training and a sample for testing. 2) A set of base learners are trained with the training sample. 3) The models are tested using the test sample. 4) The predictions are used as input and the current response as output to train the highest level learner.

3 Ensemble Learning Applied to Network Security

Ensemble learning techniques represent an improvement over machine learning techniques, we seek to combine them to obtain better classifiers taking advantage of the potential of each one.

The IDSs are today with large volumes of data and with the difficulty of detecting new attacks that arise day by day. In this regard, there are several investigations that propose that they are nourished by machine learning, taking the advantages that this entails: 1) The ability of machine learning to generalize to detect new types of intrusions, 2) Attack signatures can be automatically extracted from tagged traffic data, 3) Ability to adapt to new attacks. At the same time, since in an attack they differ: intrinsic features (general information), traffic features (connection features) and content features (package info), there are proposals around implementing models with armed dataset samples from the different sets of features and different classifiers, which are then combined with ensemble learning techniques[14][15].

[14] presents the state of the art of the teaching methods used in modern IDSs describing different works that use basic Machine Learning techniques as Ensemble Learning techniques and tests are carried out using the KDD'99 dataset in most of

them. They conclude that ensemble learning presents an improvement to basic machine learning techniques since they improve detection given the possibility of using parallel architectures, such as GPGPU. In [14] a similar analysis is made from previous work, emphasizing that given the importance of the selection of features, it should be studied in this regard. It is also important to evaluate which base classifiers to use and how it should be combined in a way to design architectures that make multiple classifiers collaborate with each other instead of competing. This problem is addressed in a more specific way in [15], where ensemble learning is applied for intrusion detection to detect DoS, R2L, U2R and Probing attacks in FTP service traffic. The work performs the tests based on the UCI KDD and Neural Networks dataset with different feature sets: intrinsic features, content features and traffic features using MPLM and Majority, Bayesian Average, Belief, as ensemble learning algorithms. In this work the metrics to compare the results were: % error, average cost, % false alarms. And it was concluded that ensemble learning reduces the percentage of error but also reduces the capabilities of generalization.

MPLM proposes to analyze the different phases or facets of a problem and build on this perspective of the problem. Perspectives are represented by a set of dataset features. Models are obtained from the different perspectives that are then combined with ensemble learning techniques. A problem to solve in this framework is the criteria and the implementation of the selection of features for each model to build the perspectives. MPLM can be applied to the detection of various attacks such as DoS, R2L, U2R and Probing [17] and for the detection of botnet activity [18], where the perspectives represent the different stages of their life cycle. In [18] a new model for applying learning in multifaceted problems is presented, focusing on the selection of features to be included in each perspective (network-based perspectives, host-based perspectives, DNS-based perspectives), one of the aspects that is raised as an aspect to be solved. In addition, the criteria of what features make sense to include in each model, taking into account that the inclusion of strongly correlated features may not contribute to the classification process.

There are also proposals as [19] based on modified Stacking to detect network intrusions (Probe, DoS, UR2 and R2L) where models are generated using samples from the random selection of dataset features, then select the best models according to a defined criteria (accuracy, information gain, recall mean and true positive rate) and combine them with stacking as the ensemble learning technique. And other more

innovative works such as [20] that proposes to apply ensemble learning clustering to detect botnets. They generate partitions and use link based algorithm to combine them (in that step is where the ensemble learning takes place) and apply machine learning to perform the classification in each cluster of the final partition.

No proposals described include Thread Intelligence in the classification process.

4 Ensemble Learning to Improve Detection of Infected Machines

SLIPS [1] is a behavioral-based intrusion detection and prevention system that uses machine learning algorithms to detect malicious behaviors. In addition to the different classifiers that this tool has today (MLDetection and Portscan detector) other modules such as Thread Intelligence are in development to incorporate more information to the detection, and thus improve the accuracy of the tool from its use.

The proposal presented in this article and which is currently being worked on consists of the incorporation of ensemble learning algorithms that allow combining the information obtained from the different classifiers in order to improve the results in the detection of infected IPs, taking into account the anomalies that the classifiers are able to detect, in each of the stages, from the analysis of the flows that represent the connections of the hosts of the network on which SLIPS[1] operates.

The contribution consists in improving the detection process by implementing ensemble learning to take decisions based on different data provided by SLIPS[1], considering that to determine if an IP is infected in a time window, it has: 1)For a flow, different predictions, one for each classifier. 2)A set of flows associated with the given IP (the source IP of those flows), with its corresponding prediction. 3)A set of malicious behavior alerts associated with the given IP (that have this IP as source IP). 4)Information from different Thread Intelligence sources that indicate destinations of the analyzed flows that are malicious (with some confidence percentage).

In 1 we propose to apply ensemble learning to decide whether the flow is classified as malware or normal. In 2 to determine if the IP is infected or not, our proposal consists in applying ensemble learning to the set of flows and their predictions. Regarding the information provided by 3, the proposal is to take into account the percentage of alerts reported for that time interval. While, with respect to 4, it is also planned to implement ensemble learning to the information provided from the

different IT sources, and evaluate whether to incorporate this information into each flow or associate it with the source IP directly, and take it into account in the decision.

5 First tests -Ensemble learning applied to network security

As we mentioned in previous section, within the framework of our proposal, a possible application of ensemble learning is to combine the results of multiple classifiers for a flow, where each one predicts whether it corresponds to malware or normal traffic. Then in SLIPS[1] we have n predictions for a flow, where n is equal to the number of classifiers in operation. As described in section 4 of this article, the advantages that can be obtained from applying ensemble learning to intrusion detection depends on the attack to be detected, the machine learning techniques to be combined and the features to be taken into account in the classification of the flows.

Thus, as a first stage of this project, performance tests of the teaching algorithms were performed to detect malware from flows evaluating its accuracy compared to a set of classic Machine Learning (ML) algorithms. To carry out the tests, the Stratosphere dataset [21] was used. It is a mixed dataset with tags corresponding to normal traffic and malware traffic, from a botnet known as Rbot. The following ML algorithms were tested: Logistic Regression(LR), Naive Bayes(NB), Random Forest(RF), Kneighbords(KN) and Decision Tree(DT). And the ensemble learning techniques used: voting hard (majority voting), voting soft (Using the sum of the predicted probabilities), voting with weight, boosting and bagging [22]. Tests with the Scikit-learn library [23] were implemented and the accuracy obtained from applying the `cross_val_score` function to the model was used as metric.

Table 1. Tests without applying ensemble learning

Algorithm	Accuracy
Logistic Regression	0.9945719941 (+/- 0.0064141127)
Random Forest	0.9999557803 (+/- 0.0000413645)
naive Bayes	0.9899729966 (+/- 0.0124655918)
Kneighbords	0.9997567974 (+/- 0.0002254790)
Decision Tree	0.9999336723 (+/- 0.0000644568)

Table 2. Tests with voting techniques including LR, Naive Bayes and RF:

Algorithm	Accuracy
Ensemble [Majority Voting]	0.9973136314 (+/- 0.0028659845)
Ensembling [Voting with probabilities sum]	0.9975347114 (+/- 0.0029525249)
Voting con peso (LR=1, RF=3 y NB=1)	0.999934 (+/-0.000041)

Algorithm	Accuracy
Voting con peso (LR=1, RF=3 y NB=2)	0.999912 (+/-0.000075)
Voting con peso (LR=2, RF=3 y NB=1)	0.999889 (+/-0.000092)

The tests with both voting techniques give the same results. Both improve the LR and the NB but not the RF. This happens because when combining RF that gives good results with two other algorithms that are worse, and then majority decides badly.

Voting with weight does not present improvements for RF although it does show improvements compared to the other voting techniques tested. Those that best classify are that give RF the greatest weight and the least weight to the other two algorithms.

Table 3. Tests by changing Random Forest to KNeighbors:

Algorithm	Accuracy
Voting con peso (LR=1, KN=2 y NB=1)	0.999757 (+/-0.000254)
Voting con peso (LR=1, KN=3 y NB=1)	0.999757 (+/-0.000254)
Voting con peso (LR=2, KN=3 y NB=1)	0.999746 (+/-0.000254)

Table 4. Tests changing Random Forest to Decision Tree:

Algorithm	Accuracy
Voting con peso (LR=1, DT=3 y NB=1)	0.999934 (+/- 0.000064)
Voting con peso (LR=1, DT=3 y NB=2)	0.999912 (+/- 0.000075)
Voting con peso (LR=2, DT=3 y NB=1)	0.999934 (+/- 0.000064)

Instead of using voting combining LR, RF and NB, it was tried to combine LR and NB with KN and with DT, and improvements could also be seen when applying teaching regarding not applying it, although they were not significant but they were significant. there were them for all the algorithms involved.

Table 5. Tests using a seed value = 8

Algorithm	Accuracy
Logistic Regression (con bagging)	0.9941959620 (+/- 0.0173140208)
Random Forest (con bagging)	0.9999778908 (+/- 0.0001326553)
Naive Bayes(con bagging)	0.9916532294 (+/- 0.0272774708)
Decision Tree (con bagging)	0.9999778908 (+/- 0.0001326553)
Kneighbors (con bagging)	0.9998010158 (+/- 0.0003396534)

The precedent table shows that improvements are obtained in all tested algorithms.

Table 6. Tests of boosting (Adaboost) tests:

Algorithm	Accuracy
Logistic Regression (con boosting)	0.9855731305 (+/- 0.0170767141)

Algorithm	Accuracy
Random Forest (con boosting)	0.9999557803 (+/- 0.0000541578)
Naive Bayes(con boosting)	0.7788494492 (+/- 0.2931913628)
Decision Tree (con boosting)	0.9999336711 (+/- 0.0001326557)

Adaboost improves for DT is the same for RF and does not improve either NB or LR.

6 Conclusions and future work

From the previous section it is concluded that the teaching improves the results against the algorithms that do not apply it, and that the improvement is significant when the base algorithms are weak. If any of the base algorithms has a high percentage of successes, heavy voting, giving more weight to this algorithm, is the better. This technique is appropriate to apply to the problem of deciding the prediction for a flow, given a set of predictions resulting from the different SLIPS[1] classifiers.

Of the works that describe the use of ensemble learning applied to cybersecurity, none incorporates the valuable information provided by Thread Intelligence sources, which is part of our proposal.

On the other hand, to carry out the proposal described here, as we have described in section 4, the most appropriate techniques to apply depend on the problem given, so in order to find the best options for the implementation of the proposal, specific tests must be carried out with the data that SLIPS[1] provides. These tests, the design of the solution and its implementation are the keys to the future work of this project.

References

1. <https://www.stratosphereips.org/technology>. Stratosphere IPS Project site.
2. Vanerio, J., & Casas, P. (2017, August). Ensemble-learning Approaches for Network Security and Anomaly Detection. Workshop on Big Data Analytics and Machine Learning for Data Communication Networks (pp. 1-6). ACM.
3. <https://www.virustotal.com/>. File and suspicious urls analysis site to detect malware.
4. Kasey Panetta. (2017). 5 Trends in Cybersecurity for 2017 and 2018. Diciembre de 2018, Smarter with Gartner website: <https://www.gartner.com/smarterwithgartner/5-trends-in-cybersecurity-for-2017-and-2018/>

5. ENISA. (2017). ENISA Threat Landscape Report 2017 15 Top Cyber-Threats and Trends. December 2018, de ENISA Sitio web: <https://www.enisa.europa.eu/publications/enisa-threat-landscape-report-2017>
6. PANDA. (2018). 2017 in Figures: The Exponential Growth of Malware. December 2018, de PANDA Website: <https://www.pandasecurity.com/mediacenter/malware/2017-figures/>
7. AV-Test. (2018). Malware. Diciembre de 2018, de AV-Test The Independ IT-Security Institute Sitio web: <https://www.av-test.org/en/statistics/malware/>
8. McAfee Labs. (2017). McAfee Labs Threat Report. December de 2018, de McAfee website: <https://www.mcafee.com/enterprise/en-us/assets/reports/rp-quarterly-threats-dec-2017.pdf>
9. R. Z. A. Mohd, M. F. Zuhairi, A. Z. A. Shadil and Hassan Dao, "Anomaly-based NIDS: A review of machine learning methods on malware detection," 2016 International Conference on Information and Communication Technology, Kuala Lumpur, 2016, pp. 266-270.
10. Zhi-Hua Zhou. (2012). Ensemble Methods: Foundations and Algorithms (Chapman & Hall/Crc Machine Learnig & Pattern Recognition) 1st Edition. US: Chapman and Hall/CRC.
11. [Cha Zhang Yunqian Ma](#). (2012). Ensemble Machine Learning: Methods and Applications 2012th Edition. United States: Springer.
12. Lior Rokach (2009). Ensemble-based classifiers. Published online: 19 November 2009. United States: Springer.
13. Martin Sewell (2011). Ensemble learning. United Kindom: UCL Research Note (UCL Department of Computer Science).
14. Gianluigi Folino, Pietro Sabatino (2016). Review Ensemble based collaborative and distributed intrusion detection systems: A survey. Journal of Network and Computer Applications. Elsevier Journal.
15. Luca Didaci, Giorgio Giacinto and Fabio Roli (2002). Ensemble Learning for Intrusion Detection in Computer Networks. ACM Journal.
16. Chih-Fong Tsai, Yu-Feng Hsu, Chia-Ying Lin, Wei-Yang Lin (2009). Intrusion detection by machine learning: A review. United States: Expert Systems with Applications 36 (2009) 11994–12000.
17. Miller, Sean & Busby-Earle, Curtis. (2018). Multi-Perspective Machine Learning (MPML) A Machine Learning Model for Multi-faceted Learning problems 10.1109/CSCI.2017.60.
18. Miller, Sean & Busby-Earle, Curtis. (2018). Multi-Perspective Machine Learning A Classifier Ensemble method for intrusion detection. 10.1145//CSCI.2017.
19. Necati DEMİR * , Gökhan DALKILIÇ (2018). Modified stacking ensemble approach to detect network intrusion Necati DEMİR * , Gökhan DALKILIÇ. Turkish Journal of Electrical Engineering & Computer Sciences 26: 418 – 433
20. Mai, Long & Kun Noh, Dong. (2017). Cluster Ensemble with Link-Based Approach for Botnet Detection. Journal of Network and Systems Management. 10.1007/s10922-017-9436-x.
21. <https://mcfp.felk.cvut.cz/publicDatasets/CTU-Malware-Capture-Botnet-51/capture20110818.pcap.netflow.labeled>. Stratosphere Datasets download site.
22. Aishwayra Singh (2018) A Comprehensive Guide to Ensemble Learning (with Python codes). Published online in Analytics Vidhya: June 18, 2018.
23. <https://scikit-learn.org/stable/>. Scikit-learn python library website.

Forensic Readiness: Guía de buenas prácticas

Mónica D. Tugnarelli ⁽¹⁾, Francisco Javier Díaz ⁽²⁾

⁽¹⁾ Facultad de Ciencias de la Administración –

Universidad Nacional de Entre Ríos

⁽²⁾ Facultad de Informática – Universidad Nacional de La Plata

e-mail: montug@fcad.uner.edu.ar, jdiaz@unlp.edu.ar

Abstract. En este artículo se presenta una guía de buenas prácticas para la implementación del enfoque preventivo denominado Forensic Readiness, el cual propone aplicar tratamiento de evidencia digital a los datos que se recolecten antes de la ocurrencia de un incidente de seguridad. La guía está dividida en cinco etapas con actividades y resultados para cada una de ellas y puede considerarse una base de partida para la adopción de Forensic Readiness en cualquier tipo de organización que quiere proteger su información y pretende iniciarse en este proceso. Esta guía está estrechamente relacionada e integrada a las políticas de seguridad y a las de continuidad del negocio, requiriendo del compromiso de todos los niveles organizacionales para su correcto cumplimiento.

Keywords: Forensic Readiness, buenas prácticas, continuidad digital, seguridad

1 Introducción

Forensic Readiness o Preparación Forense es un enfoque preventivo que plantea que la evidencia digital se recolecte y asegure de manera anticipada, es decir antes de la ocurrencia de un incidente de seguridad. Este término fue enunciado por John Tan [1] quien lo describió principalmente a través de dos objetivos: maximizar la capacidad del entorno para reunir evidencia digital confiable y minimizar el costo forense durante la respuesta a un incidente.

En trabajos anteriores se ha realizado un análisis comparativo de las prestaciones de este enfoque preventivo contrastándolo con un enfoque reactivo, donde los datos se recolectan a posteriori de un incidente, es decir, sobre un evento consumado, arribando a la conclusión de que Forensic Readiness proporciona un mejor mecanismo activo de anticipación y de respuesta a los incidentes, a la protección de activos y al resguardo de evidencia digital. [2] [3].

Establecer el momento de la recolección de datos es donde radica la principal diferencia con la informática forense, que actúa luego de cometido el delito o incidente, cuando se ha incautado el dispositivo sospechoso y se han establecido puntos de pericia. En este último escenario se tiende a ignorar qué ha sucedido con el objeto de la investigación antes del incidente de seguridad y antes de la decisión de realizar una investigación.

En coincidencia, ambos enfoques consideran a la evidencia como el insumo primario para cualquier investigación forense y que por su carácter preponderadamente volátil debe contemplar aspectos básicos en cuanto a su recolección, integridad, relevancia, almacenamiento, trazabilidad y mantenimiento de la cadena de custodia. Para ello toman lo especificado por normativas tales como la RFC 3227 “*Directrices para la recopilación de evidencias y su almacenamiento*” [4] y la ISO/IEC 27037 “*Information technology — Security techniques — Guidelines for identification, collection, acquisition and preservation of digital evidence*” [5], entre otras.

La Preparación Forense, trata a los datos seleccionados como evidencia forense desde el momento de la recolección y, para seleccionar esos datos que serán resguardados como evidencia, es necesario identificar los activos primarios, determinar métodos de recolección, asegurar la cadena de custodia y mantener a toda la organización trabajando en conjunto para asegurar que cada prueba recolectada pueda ser utilizada tanto en el proceso de investigación forense como ante un proceso judicial en caso de ser requerido.

La continuidad digital, entendida como la capacidad de usar la información de la organización de la manera en que se necesite y durante el tiempo que se requiera [6], tiene una relación intrínseca con Forensic Readiness, planteada como tal desde el inicio de la planificación de la continuidad o desde el diseño de la política de seguridad, convirtiéndose esta preparación en uno de los requisitos del negocio y una actividad clave en la identificación de activos de información de la organización.

No existe una normativa estandarizada para implementar la preparación forense, pero en lo que sí coinciden los autores que han avanzado en modelos conceptuales de Forensic Readiness, tales como Poee y Labuschagne [7] y Jan Collie [8], es que la protección de la información es un punto crítico al que cualquier organización debería prestar atención y en ese sentido preocuparse por adoptar mecanismos para asegurar tanto el resguardo de información como su propia continuidad como negocio.

El enfoque preventivo puede ser muy efectivo, lo que ante un incidente de seguridad redundará en un ahorro de tiempo y dinero. Sin embargo, la complejidad del entorno objeto del análisis exige que la definición de los detalles del procedimiento sean precisados desde el inicio con el fin de identificar los activos de la organización, seleccionar los datos a preservar y determinar aspectos de trazabilidad de los datos, entre otros, motivos por los cuales en este trabajo se propone una guía de buenas prácticas, que se presenta en el punto siguiente, y que puede ser de utilidad para la implementación de Forensic Readiness en una organización.

2 Guía de Buenas Prácticas para la implementación de Forensic Readiness

La Guía de Buenas Prácticas que se propone puede considerarse como una base de partida para la implementación de Forensic Readiness y está organizada en cinco etapas. Esta estructura en etapas permite minimizar la complejidad del proceso, así la organización puede avanzar en la medida de su capacidad pero de manera constante, sabiendo que a mayor nivel de madurez alcanzará procesos y procedimientos más detallados y de mejor calidad.

En la Figura 1 se representa gráficamente las etapas propuestas, las actividades mínimas que comprenden y lo que se debería lograr como producto de cada etapa:

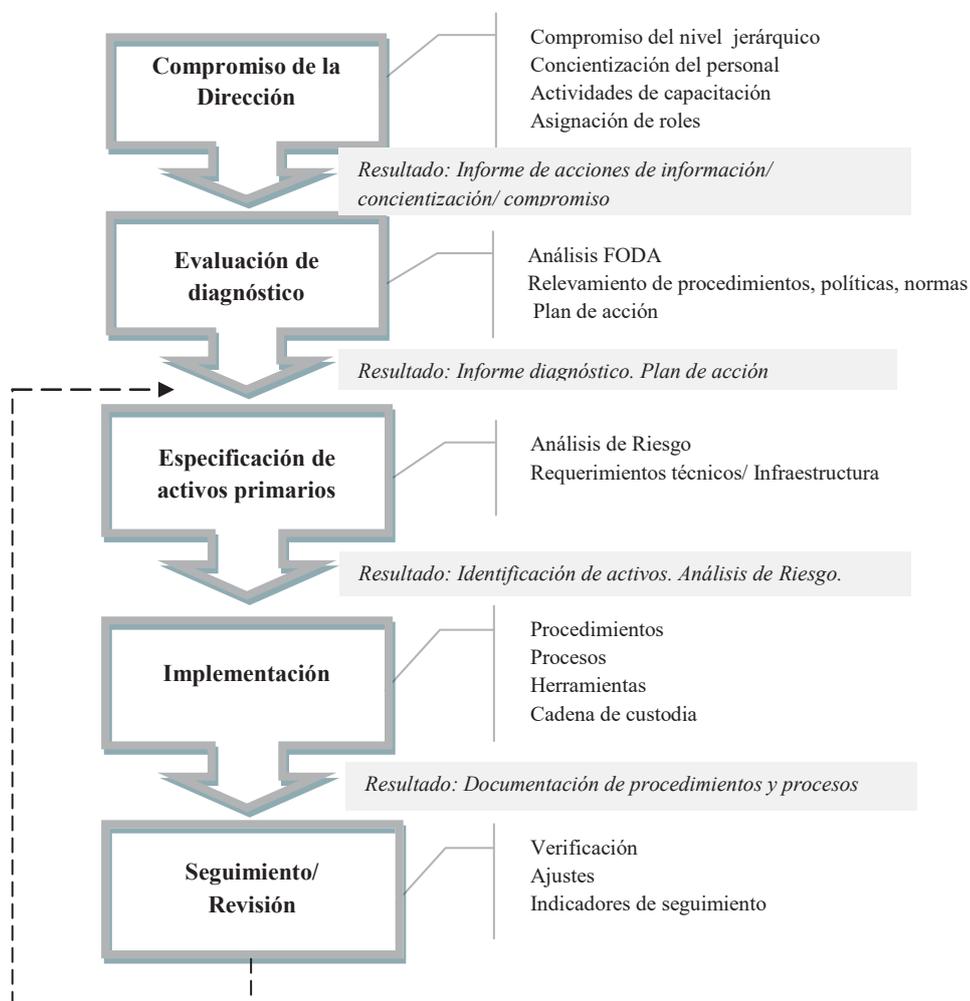


Figura 1. Etapas para la implementación de Buenas Prácticas en Forensic Readiness

A continuación se describen brevemente cada una de las etapas.

2.1 Compromiso de la Dirección

Cuando se decide implementar un enfoque Forensic Readiness, donde los controles sobre la información son muy estrictos, la dirección de la organización tiene una tarea esencial: comprometerse y tratar de lograr el compromiso de todos los niveles. Es fundamental que los niveles jerárquicos asuman la responsabilidad porque son los que tienen una visión estratégica y entienden claramente los objetivos de negocio.

Se deben realizar reuniones informativas, capacitaciones en seguridad informática y acciones de concientización para que cada persona esté al tanto de los procedimientos adoptados, que pueda participar en las diferentes fases de la implementación y así de esta manera, avanzar hacia la meta de que toda la organización trabaje en pos de la seguridad de los datos.

Un rol destacado tendrán los puestos de administrador del sistema de información, el administrador de redes de datos y el administrador de la seguridad. Sobre ellos recae la responsabilidad técnica de aplicar los procedimientos específicos sobre los datos a recolectar. Deben estar capacitados profesionalmente en seguridad informática, en ciberseguridad, en herramientas de Ethical Hacking, entre otros, para contar con las competencias y habilidades adecuadas para este proceso de preparación forense. Serán los primeros que deberán detectar las fallas de seguridad y reaccionar en consecuencia, tratando de mantener operativa la infraestructura de la organización, aplicar las herramientas de forensia, asegurar la evidencia y dar rápida respuesta para mantener la continuidad digital.

Como resultado de esta etapa se debe producir un documento que detalle las acciones de capacitación y concientización llevadas a cabo, indicando el grado de cumplimiento y la cantidad de personal que ha participado en las diversas instancias.

2.2 Evaluación de diagnóstico

De acuerdo a las acciones realizadas en la Etapa 1, el personal de la organización estará en condiciones de participar de este diagnóstico aportando desde las características propias de sus puestos de trabajo en relación a cuestiones de seguridad y actividades relacionadas con el tratamiento de los datos. Como herramientas participativas de diagnóstico pueden emplearse entrevistas, reuniones por sector y ampliadas, encuestas, checklists y demás instrumentos que ayuden a lograr una evaluación integral de la organización y su contexto.

Por otra parte, se deben analizar los procedimientos y políticas existentes con respecto a la seguridad informática, como así también las normas y leyes vigentes que correspondan aplicar.

Si es necesario puede completarse el diagnóstico con un análisis FODA para conocer las amenazas, debilidades, oportunidades y fortalezas de la organización, sobre todo en lo relativo a la seguridad interna y externa.

Como producto de esta etapa se debe lograr un diagnóstico completo y un plan de acción que identifique los objetivos y el alcance que guiarán la implementación de la Preparación Forense. Este plan de acción debe detallar además, como mínimo, acciones prioritarias, asignación, responsables y tiempo previsto para cada tarea, recursos asignados, etc. Dependiendo del tamaño de la organización este plan de acción puede ser para un sector de la misma o para su totalidad

2.3 Especificación de activos primarios

Los datos, la información, los procesos, los sistemas y servicios son activos fundamentales para una organización. Tienen requisitos de seguridad tales como confidencialidad, integridad y disponibilidad indispensables para mantener la continuidad del negocio, la rentabilidad, la competitividad y la reputación ante los usuarios/clientes, aspectos que son requeridos en el logro de los objetivos propios de la organización.

Hay varias metodologías que sirven de base para el análisis de riesgo y la identificación de activos, especialmente de los activos primarios que marcan los requisitos de seguridad para los demás componentes del sistema.

Este análisis se inicia con un inventario de activos, información que puede tomarse de la etapa anterior de diagnóstico, lo cual servirá para diferenciar entre activos estratégicos, tácticos y operacionales. Los activos estratégicos son los servicios o productos que sustentan la capacidad competitiva de la organización, los activos tácticos los procesos involucrados para la correcta entrega de esos servicios o productos, y los activos operacionales aquellas actividades que se determinan como críticas en cada proceso.

El inventario también puede emplearse para conocer el estado de situación de infraestructura y las posibles adecuaciones necesarias con vista a la implementación de la Preparación Forense, sobre todo la infraestructura de almacenamiento requerida para dar soporte a la metodología preventiva.

De ser necesario se puede ajustar el plan de acción luego de seleccionados los activos primarios, con indicación de prioridades, datos a recolectar, tiempo de recolección y responsable asignado.

Es imprescindible comprender la operatoria, protocolos asociados, la estructura y demás características del activo seleccionado, porque ese conocimiento ayudará a detectar rápidamente un comportamiento sospechoso o anómalo.

Como resultado de esta etapa se contará con un conocimiento detallado de los activos y de los riesgos asociados a los mismos, la probabilidad de ocurrencia, el impacto y los controles que se deberán efectuar para su mitigación. Cada uno de los

riesgos identificados deberá estar relacionado con una o más de las dimensiones de seguridad informática establecidas como requisitos del sistema, lo cual permitirá establecer el nivel de riesgo y los controles específicos que correspondan.

2.4 Implementación

Definidos los activos primarios y efectuado el análisis de riesgo correspondiente, se puede iniciar con la etapa de implementación de Forensic Readiness.

Esta es una instancia más bien técnica donde, sobre los activos identificados, se iniciará la recolección de datos y el tratamiento de los mismos como evidencia digital.

Los procedimientos que se apliquen sobre los datos recolectados deben orientarse a resguardar la integridad de la evidencia y a asegurar la confiabilidad y la disponibilidad como requisitos a cumplir por parte del sistema de seguridad que comienza a funcionar.

Se reitera la importancia de mantener la cadena de custodia desde el inicio, para asegurar la plena admisibilidad como prueba legal de la evidencia. Esta cadena de custodia implica el registro detallado de cada paso dado con respecto a los datos, es decir, asegurar la trazabilidad de la información y mantener la integridad de los archivos con algoritmos de hash que producirán una huella digital unívoca para el conjunto de datos.

Por idénticos motivos deben documentarse las herramientas tecnológicas utilizadas tanto para recolectar como las aplicadas en el análisis de datos, las responsabilidades asignadas y las prioridades consideradas con respecto al resguardo de datos.

El valor temporal que la organización le otorgue a sus datos determinará las necesidades de infraestructura de almacenamiento para el registro centralizado de los datos considerados evidencia. Este repositorio debe recibir tratamiento con el rango de activo primario.

La aplicación de herramientas de forensia informática se realiza en esta etapa tanto para la vigilancia de incidentes como para el análisis de datos. Existen numerosas soluciones comerciales y open source que deberán ser cuidadosamente seleccionadas dependiendo de sus prestaciones, utilizando también una combinación de varias herramientas diferentes para lograr una mayor potencia y performance.

Las vulnerabilidades descubiertas deben ser reportadas al responsable designado de manera inmediata para su tratamiento, como así también deberá hacerlo cualquier persona de la organización que crea estar frente a un fallo de seguridad.

Como resultado de esta etapa se obtendrán procedimientos detallados de la implementación de Forensic Readiness, incluyendo descripción de herramientas empleadas, formularios que registren cada instancia de la cadena de custodia y instrumentos de reporte de incidentes.

2.5 Seguimiento/Revisión

Se requiere un seguimiento activo de todo el proceso de la Preparación Forense para verificar su correcto funcionamiento y para detectar a tiempo errores o falencias que puedan ocurrir.

Es importante evaluar si las actividades desarrolladas se realizan conforme a lo previsto y la confidencialidad, integridad y disponibilidad de los datos se encuentra garantizada. Para esta evaluación se pueden construir indicadores de seguimiento que reflejen los resultados de eficiencia y efectividad de lo implementado y que ayuden a tomar decisiones al respecto, tales como: resultados de auditorías; resultados de análisis de intrusión; observaciones recibidas; informaciones de avance sobre acciones preventivas y correctivas; mediciones y actualización de vulnerabilidades entre otros.

Los cambios en la organización, en la tecnología y en las leyes aplicables pueden producir cambios en los objetivos de negocio, en las amenazas detectadas, en las oportunidades del entorno y en la efectividad de los controles implementados por lo que es conveniente la revisión periódica de los activos primarios y los riesgos que los afectan. Lo importante es que desde la Dirección se mantenga una actitud proactiva de vigilancia sobre el desarrollo de todas las actividades relacionadas, para asegurar los recursos requeridos y como mejora continua del proceso de Preparación Forense.

3 Conclusiones y trabajos futuros

En este artículo se ha presentado una propuesta de Guía de Buenas Prácticas para la implementación de Forensic Readiness en cualquier tipo de organización que quiera adoptar este enfoque para la protección de sus datos y como método de anticipación de incidentes de seguridad.

La característica de considerar a los datos como evidencia digital desde el momento de la recolección trae aparejados cambios operativos en el modo de trabajo y una variación en la forma de utilizar la información, por eso es adecuado contar con una guía básica para organizar la implementación.

Es un factor clave el compromiso de todas las personas de la organización, tanto para el cumplimiento de los estrictos requisitos que se proponen sobre los datos considerados evidencia como así también para la aplicación de las políticas de seguridad y el esquema de continuidad digital, cuestiones que están intrínsecamente ligadas a Forensic Readiness.

A futuro se podría avanzar con el análisis del grado de madurez necesario en una organización para adoptar este modelo de trabajo, como así también en el diseño de un framework que permita implementar Forensic Readiness en cualquier tipo de organización lo cual, además, puede ser una estrategia para incentivar la cultura de

la protección de datos y seguridad de la información tan demanda frente a la tendencia de crecimiento sostenido de los delitos informáticos.

Referencias

- [1] TAN, John. (2001). Forensic Readiness. http://isis.poly.edu/kulesh/forensics/forensic_readiness.pdf
- [2] Tugnarelli, M.; Fornaroli, M.; Santana, S.; Jacobo, E.; Díaz, F.J. Análisis de metodologías de recolección de datos digitales. Workshop de Investigadores en Ciencias de la Computación (WICC 2017). ISBN: 978-987-42-5143-5. <http://sedici.unlp.edu.ar/handle/10915/61343>
- [3] Tugnarelli, M., Fornaroli, M., Santana, S., Jacobo, E., Díaz, J. Analysis of Methodologies of Digital Data Collection in Web Servers. Communications in Computer and Information Science (Springer), Vol. 790, Pag.265. (2018) <https://link.springer.com/content/pdf/bfm%3A978-3-319-75214-3%2F1.pdf>
- [4] RFC 3227 Guidelines for Evidence Collection and Archiving. <https://www.ietf.org/rfc/rfc3227.txt>
- [5] Guidelines for identification, collection, acquisition and preservation of digital evidence” ISO/IEC 27037:2012
- [6] The National Archives. Managing digital continuity. <http://www.nationalarchives.gov.uk/information-management/manage-information/policy-process/digital-continuity>
- [7] Poee, A & Labuschagne, Les. A conceptual model for digital forensic readiness. (2012) Information Security for South Africa - Proceedings of the ISSA 2012 Conference. 1-8. 10.1109/ISSA.2012.6320452.
- [8] Collie, Jan. A Strategic Model for Forensic Readiness. (2018). Athens Journal of Sciences. <https://www.athensjournals.gr/sciences/2018-1-X-Y-Collie.pdf>

Marco para la Aplicación de Criptoanálisis

Antonio Castro Lechtaler¹, Marcelo Cipriano¹, Francisco Javier Díaz², Edith García¹,
Julio Liporace¹, Ariel Maiorano^{1,2}, Eduardo Malvacio¹

¹ Facultad de Ingeniería del Ejército (FIE), Universidad de la Defensa Nacional (UNDEF)

{acastro,marcelocipriano,egarcia,jliporace,
emalvacio,maiorano}@fie.undef.edu.ar,

² Facultad de Informática (FI), Universidad Nacional de la Plata (UNLP)

{ariel.maiorano,javierd}@info.unlp.edu.ar

Resumen. El objetivo del presente artículo es presentar y describir un proyecto de software libre que implementa un marco o *framework* para la aplicación de diferentes metodologías, técnicas o mecanismos de criptoanálisis, conjuntamente con análisis auxiliares o complementarios. La herramienta permite aplicar diferentes técnicas sobre archivos de "muestras" manejados por el sistema. Permite a su vez la revisión comparativa de resultados. El código fuente ha sido publicado en Github.com.

Palabras clave: seguridad informática, criptografía, criptoanálisis, herramientas de criptoanálisis.

1 Introducción

Como fuera adelantado en el resumen, este trabajo presenta un proyecto de software libre y código abierto, desarrollado por el GICSI (Grupo de Investigación en Criptografía y Seguridad Informática) de la FIE (Facultad de Ingeniería del Ejército), para la implementación de un marco o *framework*. Se referirá al sistema como "Marco para la Aplicación de Criptoanálisis", o por sus siglas, MAC. Este sistema permitirá aplicar diferentes técnicas o mecanismos de criptoanálisis, conjuntamente con análisis auxiliares o complementarios, como por ejemplo análisis estadísticos sobre secuencias binarias. La herramienta permite aplicar estos análisis sobre archivos que llamaremos "muestras", también manejados por el sistema. Implementa a su vez la funcionalidad necesaria para la revisión comparativa de resultados.

Continuando en línea con otras publicaciones de software libre y de código abierto realizadas por el GICSI, relacionados a la seguridad informática en general y a la criptografía en particular [1,2]; la primer versión de este proyecto, MAC, que al momento permite la aplicación de análisis estadísticos y la comparación de los resultados obtenidos para diferentes muestras- ya se encuentra publicada en su totalidad y disponible sin restricciones en la plataforma de alojamiento de proyectos de software libre Github.com [3], bajo la Licencia Pública General de GNU versión 3, o GPLv3 por sus siglas en inglés.

2 Presentación del proyecto

2.1 Contexto

Este es un proyecto académico de desarrollo de software libre y de código abierto del Grupo de Investigación en Criptografía y Seguridad Informática (GICSI), de la Facultad de Ingeniería del Ejército (FIE), de la Universidad de la Defensa Nacional (UNDEF); forma parte del proyecto “Herramientas para la Evaluación de Algoritmos Criptográficos” (HEAC), aprobado en el Programa de Acreditación y Financiamiento de Proyectos de Investigación (UNDEFI), de la UNDEF.

A su vez, el proyecto se encuentra enmarcado en el trabajo realizado por miembros del GICSI doctorandos en el área "Redes y comunicaciones", sub-área "Criptografía", del Doctorado en Ciencias Informáticas de la Facultad de Informática (FI), de la Universidad Nacional de La Plata (UNLP).

Como fuera adelantado en la presentación de las líneas de investigación relativas al proyecto “Herramientas para la Evaluación de Algoritmos Criptográficos” (HEAC)[4], los objetivos incluyen el diseño y desarrollo de una herramienta que permita llevar adelante estos análisis y la realización de pruebas que permitan el estudio de diferentes propiedades de esquemas criptográficos de manera ágil y eficiente. Como parte de las primeras instancias o etapas en la implementación del proyecto, esta herramienta permite al momento el análisis estadístico de secuencias binarias para aplicar en algoritmos de cifrado de flujo (*stream ciphers*) y generadores de números pseudaleatorios (*pseudo random numbers generators*). Tales secuencias pueden ser generadas además por LFSRs (*Linear Feedback Shift Registers*), NLFSRs (*Non-Linear Feedback Shift Registers*), CCGs (*Clock Controlled Generators*), entre otros esquemas y mecanismos. Se espera diseñar y desarrollar un software abierto que permita la evaluación de las propiedades criptográficas y de seguridad de secuencias pseudoaleatorias binarias procedentes de algoritmos y esquemas criptográficos. El enfoque propuesto para el desarrollo se centra en el estudio e implementación de las diferentes técnicas criptoanalíticas aplicables de acuerdo a la bibliografía, su conjunción y el desarrollo de un conjunto de herramientas que permitan analizar propiedades criptológicas.

2.2 Tecnologías utilizadas

Se trata de una aplicación Web, desarrollada en lenguaje de programación Python versión 3.6, utilizando el *framework* Web Django. Los “análisis”, que se integran a la aplicación configurándose como *plugins*, podrían desarrollarse en cualesquier otro lenguaje de programación ya que serán ejecutados como programas externos. Los ejemplos de *plugins* incluidos al momento se encuentran programados en los lenguajes Python y C.

En el repositorio público del proyecto en Github.com se encuentran las instrucciones necesarias para la instalación y puesta en marcha de la herramienta. Dadas las tecnologías mencionadas anteriormente, el software puede ser instalado y ejecutado en diferentes sistemas operativos (Windows, MacOS o Linux).

3 Funcionalidad de la herramienta

En el presente apartado se ejemplificará cómo un usuario podría operar con el sistema. Se describirán a continuación las principales funciones que provee la herramienta actualmente. Nos valdremos de dos muestras de secuencias binarias de aproximadamente la misma longitud; la primera constando del contenido de un archivo comprimido, y la segunda de la salida producida por el archivo de sistema `/dev/urandom` de un sistema Linux, utilizadas sólo como ejemplo y para permitir mostrar mediante capturas de pantalla la operatoria. En el apartado siguiente serán descriptas otras funcionalidades planeadas, a ser implementadas, para completar el alcance propuesto del proyecto.

3.1 Manejo de muestras

Dado que típicamente la primera acción que un usuario realizaría en el sistema sería la agregar muestras a analizar, como puede visualizarse en la figura siguiente, en el menú principal de opciones presentadas a la izquierda de todas las pantallas del sistema, la primera es la de “Muestras”.

Esta opción permite acceder al listado de muestras cargadas. Como fuera mencionado y se observa en la imagen, para ejemplificar el uso de la herramienta, se cargaron dos muestras del tipo “Secuencia binaria”. La primera que se visualiza en el listado corresponde a la salida producida por el archivo de sistema `/dev/urandom` de un sistema Linux, y la segunda al contenido de un archivo comprimido.

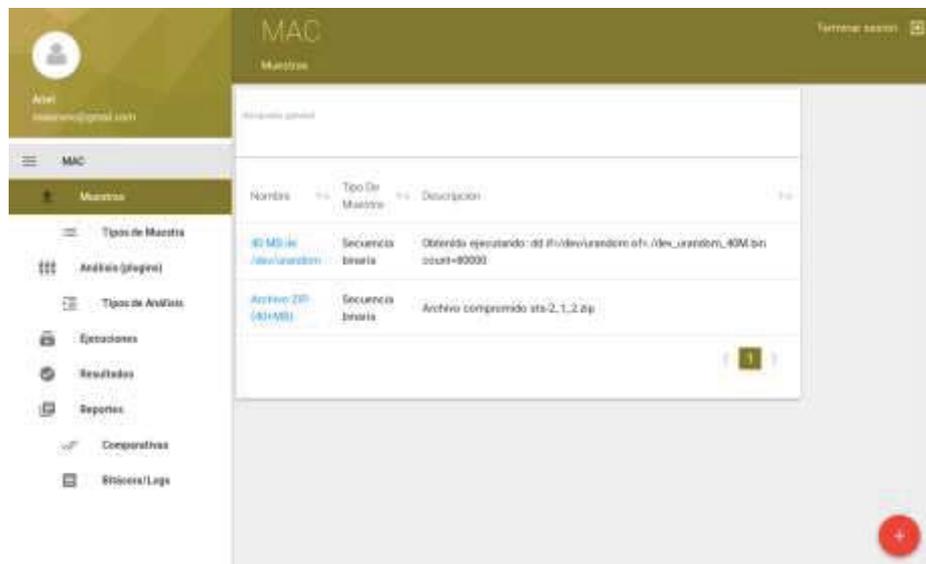


Figura 1. Captura la pantalla del sistema que lista las muestras ingresadas por el usuario para su posterior análisis.

En esta pantalla, utilizando el botón color rojo de agregar a pie de página y alineado a la derecha (símbolo +), se pueden agregar otras muestras a ser utilizadas en los análisis. Esto puede realizarse a partir de un archivo o también introduciendo el valor en formato de texto, pudiendo expresarlo en notación binaria, hexadecimal o codificación ASCII. Se especifica además un tipo de muestra (en ambos casos de ejemplo el tipo corresponde a “Secuencia binaria”). Estos tipos de muestra pueden ser configurados en el sistema utilizando la opción del menú principal “Tipos de muestra”. Por último cabe aclarar que el nombre y una descripción opcional se utilizan únicamente para poder referenciar la muestra desde las otras pantallas del sistema que la involucran.

3.2 Plugins de análisis

La funcionalidad principal del sistema se basa en la aplicación de diferentes técnicas de análisis sobre las muestras disponibles (que hayan sido cargadas en el sistema de acuerdo a lo descrito en el apartado anterior).

Los diferentes análisis que pueden realizarse a través del sistema son configurados como *plugins* o “complementos” (por utilizar una traducción posible).

Estos *plugins* o complementos constan básicamente de, por un lado, programas externos que serán ejecutados por el sistema, y por otro lado, de la configuración necesaria tanto para su ejecución como para la interpretación –y registro en el sistema- de sus resultados.

La siguiente figura muestra el listado de los *plugins* configurados en el sistema:

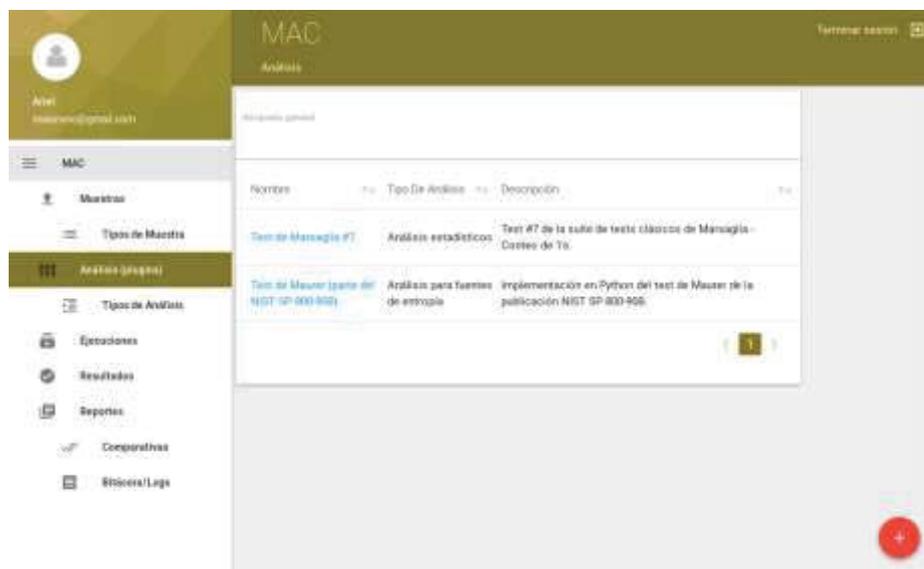


Figura 2. Captura de pantalla del sistema que muestra los *plugins* o “complementos” de análisis configurados.

En este caso, y como se observa en la figura de la captura de pantalla, los *plugins* configurados para la ejemplificación del uso de la herramienta son dos métodos, o tests de pseudoaleatoriedad, para la realización de análisis estadísticos de secuencias binarias: El séptimo Test de Marsaglia [5] y el Test de Maurer de acuerdo a la publicación SP 800-90B de la NIST [6].

3.2.1. Tests de Marsaglia

Como ejemplo inicial y dentro se incorporó originalmente uno de los llamados Tests de Marsaglia [5], particularmente el identificado con el número siete, también llamado de “conteo de unos”.

Este test considera el archivo de muestra como un flujo de bytes. Cada byte puede contener de 0 a 8 dígitos 1 (uno), con probabilidades 1, 8, 28, 56, 70, 56, 28, 8 y 1 sobre 256, respectivamente. Luego considera que el flujo de bytes proporciona un conjunto de palabras de 5 letras, que son superpuestas, donde cada letra puede tomar los valores A, B, C, D, E. Las letras se determinan por el número de 1 'en un byte: 0,1 ó 2 resultan en una A, 3 en una B, 4 unos, en una C, 5 en una D y 6,7 u 8 en una letra E. Así el autor originalmente explicó que se tendría un un mono en una máquina de escribir que golpea cinco teclas con varias probabilidades (37, 56, 70, 56 y 37 sobre 256). Hay 5^5 palabras posibles de 5 letras, y de un conjunto de 256,000 palabras de 5 letras (superponiéndose), se hacen conteos de las frecuencias de cada palabra. De lo anterior se deriva una prueba de chi cuadrado. El resultado del testeo que el sistema registra será el valor “p”.

3.2.2. Test de Maurer

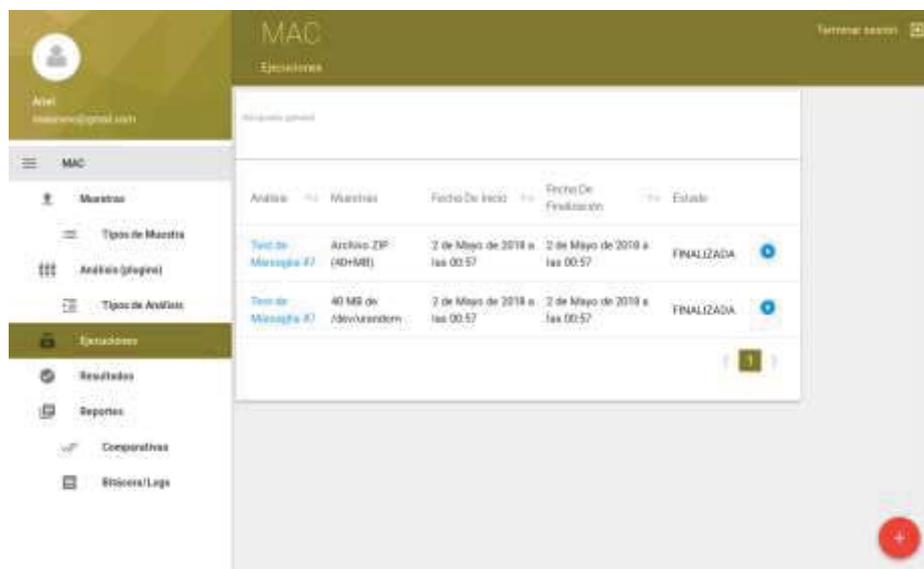
También considerándolo un ejemplo inicial entre pruebas de esta índole, actualizado en el repositorio de software del proyecto recientemente, la herramienta incorpora una implementación de referencia de la NIST (por sus siglas en inglés de *National Institute of Standards and Technology*) basado en el test de Maurer, o Test Universal de Maurer, o también Estadística Universal de Maurer; según lo especificado en su publicación NIST SP 800-90B, para análisis de fuentes de entropía.

De acuerdo a las especificaciones mencionadas, la prueba de compresión calcula la tasa de entropía de un conjunto de datos –en el caso del sistema, de los archivos de muestra–, según la cantidad de datos que se pueden comprimir. Basado en la Estadística Universal de Maurer, la prueba genera un diccionario de valores y luego calcula el número promedio de muestras requeridas para obtener una salida basada en el diccionario. Una de las ventajas de usar la Estadística de Maurer es que no se asume independencia. Cuando una salida con dependencias se prueba con esta estadística, la tasa de compresión se ve afectada (y, por lo tanto, la entropía), pero aún se obtiene una estimación de la entropía. El cálculo de la Estadística de Maurer requiere solo una “pasada” a través del conjunto de datos para proporcionar una estimación de entropía, por lo que es más eficiente que otros algoritmos de compresión. La estadística de Maurer es la media de los valores de compresión.

3.3 Ejecuciones de análisis

Cargadas las muestras que se desee analizar, y configurados los *plugins* de análisis, el paso siguiente dará un usuario de este sistema será el de ejecución de tales análisis, pruebas o tests.

Como puede verse en la figura que sigue, mostrando como ejemplo una captura de pantalla habiendo accedido a la opción del menú principal “Ejecuciones”, que lista las ejecuciones ingresadas e iniciadas por indicación del usuario en el sistema, se muestran dos ejecuciones finalizadas del análisis del Test de Marsaglia número siete sobre las dos muestras de ejemplo.



Análisis	Muestras	Fecha De Inicio	Fecha De Finalización	Estado
Test de Marsaglia #7	Archivo ZIP (40MB)	2 de Mayo de 2018 a las 00:57	2 de Mayo de 2018 a las 00:57	FINALIZADA
Test de Marsaglia #7	40 MB de Adventure2000	2 de Mayo de 2018 a las 00:57	2 de Mayo de 2018 a las 00:57	FINALIZADA

Figura 3. Captura de pantalla donde se visualiza el estado de las ejecuciones de análisis solicitadas por el usuario.

3.4 Presentación de resultados

Finalmente, luego de ejecutados los análisis, el sistema permite por supuesto la visualización de los resultados correspondientes. En primer lugar, a través del listado que puede observarse en la figura a continuación. En segundo lugar, para permitir una vista que facilite la comparación entre los resultados obtenidos, a través de los reportes, que son accesibles a partir de la última entrada de menú en el margen izquierdo de todas las pantallas.

En el ejemplo de la figura puede apreciarse que para el caso de la aplicación del Test de Marsaglia número siete a los dos archivos de muestras de ejemplo disponibles, se visualiza el valor “p” obtenido.

Resultado	Análisis	Muestra	Fecha	Eliminar
Valor p: 0.000000	Test de Marcaglia-87	Archivo ZIP (40+MB)	3 de Mayo de 2018 a las 00:57	X
Valor p: 0.000000	Test de Marcaglia-87	Archivo ZIP (40+MB)	3 de Mayo de 2018 a las 00:57	X
Valor p: 0.473333	Test de Marcaglia-87	40 MB de files/usuarios	3 de Mayo de 2018 a las 00:57	X

Figura 4. Captura de pantalla en la cual pueden apreciarse los resultados de las ejecuciones de los testeos utilizados como ejemplo.

4 Otras funcionalidades planeadas (Trabajo a futuro)

Se planea la continuación del desarrollo para ampliar la funcionalidad actual de la herramienta, en línea con los objetivos del proyecto.

Entre las funcionalidades a implementar consideradas se encuentran principalmente las relativas a la incorporación de nuevos *plugins* de análisis, para implementar, por ejemplo:

1. Análisis estadísticos adicionales a los ya implementados (secuencias binarias; generadas por generadores pseudoaleatorios –algoritmos de cifrado en flujo, registros desplazables, etc.-) [7,8,9].
2. Análisis de complejidad lineal (secuencias binarias) [7,8].
3. Análisis y ataques por correlación (secuencias binarias) [7,8].
4. Criptoanálisis lineal (algoritmos de cifrado) [9].
5. Criptoanálisis algebraico (algoritmos de cifrado) [10].
6. Criptoanálisis diferencial (algoritmos de cifrado) [11,12].
7. Otras técnicas de criptoanálisis de aparición reciente en la literatura, como ser, por ejemplo, la implementación del “*Cube-Attack*” [13,14] o métodos basados en *SAT-Solvers* [15,16].

5 Conclusiones

Este artículo presentó una herramienta para la aplicación de técnicas criptoanalíticas, configurables mediante *plugins* de análisis. Se describió el contexto del proyecto y sus

objetivos. Llamada MAC, por las siglas de Marco de Aplicación de Criptoanálisis, la herramienta se distribuye como un proyecto de software libre, de código abierto.

Desconocemos la existencia de otras herramientas de acceso libre que implementen la funcionalidad que actualmente implementa y que se planea implementar en nuestro proyecto, específicamente en relación a una interfaz común para el manejo de muestras y la posibilidad de aplicación –y revisión de resultados- de diversos tests o pruebas de manera modular, en un marco de trabajo común.

También se espera que su publicación pueda lograr que quienes pudieran valerse de una herramienta de este tipo para agilizar y normalizar estos análisis se interesen e involucren en el proyecto.

Referencias

1. Repositorio del proyecto de software MAC en Github. [En línea] <https://github.com/gicsi/mac>, accedido por últ. vez en julio de 2019.
2. Castro Lechtaler, A., Liporace, J., Cipriano, M., García, E., Maiorano, A., Malvacio, E., Tapia, N. Automated Analysis of Source Code Patches using Machine Learning Algorithms. XXI Congreso Argentino de Ciencias de la Computación (Junín, 2015). ISBN: 978-987-3806-05-6. [En línea] http://sedici.unlp.edu.ar/bitstream/handle/10915/50585/Documento_completo.pdf-PDFA.pdf?sequence=1, accedido por últ. vez en julio de 2019.
3. Proyecto AAP, Repositorio de GICSI en Github. [En línea] <https://github.com/gicsi/aap>, accedido por últ. vez en julio de 2019.
4. Cipriano, M., Malvacio, E., Estevez, C., Fernández, D., García, E., López, G., Liporace, J., Maiorano, A., Vera Batista, F. Software Abierto para la Evaluación de Sistemas Criptológicos Integrados. XX Workshop de Investigadores en Ciencias de la Computación. RedUNCI, UNNE. ISBN: 978-987-3619-27-4. [En línea] http://sedici.unlp.edu.ar/bitstream/handle/10915/68350/Documento_completo.pdf-PDFA.pdf?sequence=1&isAllowed=y, accedido por últ. vez en julio de 2019.
5. Marsaglia, G. The Marsaglia Random Number CDROM including the Diehard Battery of Tests of Randomness. [En línea] <http://www.csis.hku.hk/cisc/download/index2.htm>, accedido por últ. vez en julio de 2019.
6. Recommendation for the Entropy Sources Used for Random Bit Generation. Computer Security Resource Center. NIST. [En línea] <https://csrc.nist.gov/publications/detail/sp/800-90b/final>, accedido por últ. vez en julio de 2019.
7. Massey, J.L. "Shift-register synthesis and BCH decoding". IEEE Transactions on Information Theory 15, 1969.
8. Golomb. "Shift Register Sequences". Aegean Park Press, 1982.
9. Christopher Swenson. "Modern Cryptanalysis: Techniques for Advanced Code Breaking". Wiley Publishing. 2012.
10. Nicolas T. Courtois, Karsten Nohl, and Sean O'Neil. Algebraic attacks on the crypto-1 stream cipher in mifare classic and oyster cards. Cryptology ePrint Archive, Report 2008/166, 2008.
11. E. Biham and A. Shamir. "Differential Cryptanalysis of the Full 16-Round DES". In Advances in Cryptology — CRYPTO 1992, volume 740 of Lecture Notes in Computer Science, pages 487–496, 1992.

12. M. Albrecht and C. Cid. Algebraic Techniques in Differential Cryptanalysis. In Fast Software Encryption 2009 – FSE 2009, Lecture Notes in Computer Science. Springer-Verlag, 2009.
13. Dinur, I., Shamir, A.: Cube Attacks on Tweakable Black Box Polynomials. In: EUROCRYPT. pp. 278–299 (2009).
14. Dinur, I., Morawiecki, P., Pieprzyk, J., Srebrny, M., Straus, M.: Cube Attacks and Cube-Attack-Like Cryptanalysis on the Round-Reduced Keccak Sponge Function. In: Advances in Cryptology - EUROCRYPT 2015. 34th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Sofia, Bulgaria, April 26-30, 2015, Proceedings, Part I. pp. 733–761 (2015).
15. McDonald, C. and Pieprzyk, C. “Attacking Bivium with MiniSat”, Cryptology ePrint Archive, Report 2007/040, 2007.
16. The Z3 Theorem Prover, Repositorio en Github de los proyectos. [En línea] <https://github.com/Z3Prover>, accedido por últ. vez en mayo de 2017.

Mejora de la Integridad de Datos en el Modelo OTP-Vote

Silvia Bast
Universidad Nacional de La Pampa
Email:
silviabast@exactas.unlpam.edu.ar
www.exactas.unlpam.edu.ar

Germán Montejano, Mario Berón
Universidad Nacional de San Luis
Email: [{gmonte,mberon}@unsl.edu.ar](mailto:gmonte,mberon@unsl.edu.ar)
www.unsl.edu.ar

Abstract. El Modelo de OTP Vote presenta las características de anonimato incondicional y posibilita llevar la seguridad computacional a los niveles deseados. El esquema se compone de tres procesos: preparación, desarrollo y cierre de la elección-recuento de votos. La primera etapa incluye la configuración y parametrización de la elección, la segunda abarca el proceso de emisión de votos y la tercera el cierre y recuento de sufragios. Se presenta, en este artículo, una propuesta de mejora de seguridad e integridad de los datos del modelo, que permite detectar intentos de intrusión por parte de terceros con el objetivo de modificar el contenido de los votos. Este trabajo pretende ser un aporte al incremento en el nivel de confianza del electorado, aspecto fundamental para el éxito de los sistemas de voto electrónico.

Keywords: voto electrónico, integridad de datos, seguridad, funciones, bits de control

1 Introduction

La aplicación del voto electrónico presenta actualmente grandes discusiones, no sólo dentro del ámbito académico, sino también fuera de él. Quienes están en contra de su aplicación basan sus argumentos en la falta de transparencia del proceso y de los resultados. Esta postura es abonada por productos desarrollados y puestos en funcionamiento en distintas elecciones, que no cumplen con los requisitos establecidos para un sistema de eleccionario, por lo que gran parte de la sociedad percibe a tales sistemas como artefactos cerrados, no confiables, que procesan el voto de una forma oscura o desconocida y como consecuencia, opina que no deben ser usados.

Por definición, el voto electrónico es un método en el cual los votos son emitidos o tabulados por medios electrónicos. Un sistema de voto electrónico es un componente de software que mapea el procedimiento de voto electrónicamente [1] y que debe cumplir además con un conjunto de requisitos como los indicados en [2], [3], [4] y [5] que le son propios. En cuanto a los datos, los sistemas de voto electrónico deben proteger:

- La privacidad del votante (anonimato) de forma indefinida, aún después de finalizado el acto eleccionario, dado que, si algún atacante obtiene datos de los registros que mantienen la relación entre el voto y el votante podrá dedicar todo el tiempo para intentar el descifrado del mismo.
- La seguridad de los datos de los votos durante el transcurso del proceso electoral, ya que luego los resultados son de público conocimiento.

Desde el proyecto Aspectos de Seguridad en Proyectos de Software FCEyN se desarrolló el modelo OTP Vote [6] que asegura: anonimato incondicional y seguridad computacional que puede llevarse a cualquier nivel exigible durante el proceso eleccionario.

En el presente trabajo se realiza un aporte a la integridad de los datos de los votos durante el tiempo que dura el proceso electoral (10 horas), que tiene por objetivo detectar intentos de modificación fraudulenta.

El artículo está organizado como sigue: la sección 2 explica el modelo OTP Vote, la sección 3 describe la propuesta de mejora de integridad de los datos, la sección 4 presenta un caso de estudio y la sección 5 detalla las conclusiones y los trabajos futuros.

2 El Modelo OTP-Vote

Las principales características del modelo OTP Vote son:

- Uso de criptografía basada en One Time Pad (OTP, [7]) que cumple con las hipótesis y condiciones del “Secreto Perfecto” de Shannon, presentadas en [8].
- Uso de claves distribuidas que combinadas funcionan como una sola, concepto derivado de la propuesta de protocolos Broadbent y Tapp [9].
- Almacenamiento basado en el modelo de canales paralelos [10] denominado Múltiples Canales Dato Único (MCDU), que propone dividir en q canales el almacenamiento total y registra cada voto una vez en cada canal en posiciones aleatorias potencialmente distintas.

El modelo propone tres etapas, que se muestran en la Fig. 1 y se describen en las siguientes subsecciones.



Fig. 1. Proceso de OTP - Vote

Primera Etapa

La configuración o preparación de la elección incluye las siguientes actividades de configuración de los datos que se usarán durante el proceso eleccionario:

- 1 Definición de las dimensiones del Archivo Binario donde residirán los Votos (ABV) y Clave de Descifrado (CD) que hace uso de las fórmulas propuestas en [5].
- 2 Definición de las dimensiones de cada uno de los atributos que se almacenarán en las tuplas (Identificador de voto, Identificador del cargo, Identificador del candidato seleccionado, bits adicionales de control, redundancia de información que aporta a oscurecer el contenido de la tupla). Para llevar a cabo esta tarea, debe tenerse en cuenta la probabilidad de que un intruso pueda detectar una tupla válida de entre todas las tuplas posibles. Como se afirma en [6], [11] y [12], aumentando la redundancia en la cantidad de bits destinados a registrar cada atributo, la probabilidad de obtener una tupla válida de entre todas las combinaciones de valores posibles puede llevarse a cualquier valor deseado.
- 3 Generación de los códigos para cada uno de los atributos identificadores (Cargos, Candidatos e Identificadores de Votos). Dado que las tuplas (votos) que se almacenan usando el esquema MCDU, descrito en [11], por medio de la operación XOR [13], es necesario que los identificadores de las tablas cumplan con un conjunto de características para evitar que, a partir de votos que colisionan, el voto pueda interpretarse incorrectamente. Cada identificador (Id_i) que se agregue debe:
 - a. Ser distinto a los almacenados previamente:

$$Id_i \neq Id_j \quad \forall j \quad 1 \leq j \leq i-1 \quad (1)$$

- b. El XOR del Id_i con cada uno de los identificadores que ya residen en la tabla, no debe dar como resultado alguno de los Id existentes:

$$Id_i \oplus Id_j \neq Id_k \quad \forall j \quad 1 \leq j \leq i-1, \quad \forall k \quad 1 \leq k \leq i-1 \quad (2)$$

- c. El XOR del Id_i con cada una de las combinaciones de grupos de los identificadores que ya residen en la tabla, no debe dar como resultado alguno de los Id existentes

En [12] se desarrolla un algoritmo para la generación de códigos que cumplen con las características mencionadas anteriormente.

- 4 Configuración de la ubicación de los atributos en la tupla. Se denomina tupla a cada una de las filas del ABV en donde podría residir un voto. Dentro de la tupla, se guardan los identificadores de voto, cargo, candidato seleccionado, bits de control. Es posible almacenar los bits de cada uno de los atributos de manera consecutiva o dispersa dentro de la tupla, por ejemplo, un identificador de voto de 8 bits, podría estar dispuesto de la siguiente manera, los 2 primeros bits a partir de la posición 9 de la tupla, los 3 bits siguientes a partir de la posición 1 de la tupla, el siguiente bit en la posición 15 y finalmente los últimos 2 bits del identificadora partir de la posición 5.

- 5 Generación de las tablas: Identificadores de Votos, Cargos, Candidatos, Atributos, Ubicaciones, Votos Planos. En la Fig. 2 pueden observarse las tablas mencionadas con sus atributos.

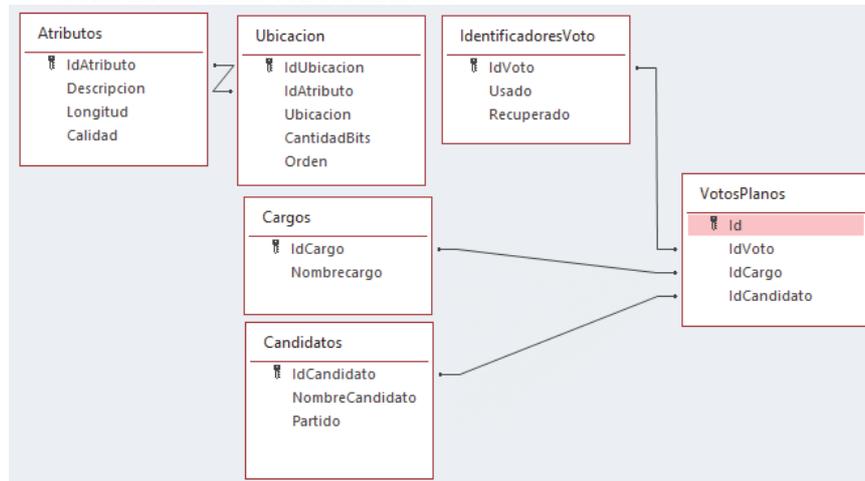


Fig. 2. Tablas de OTP - Vote

- 6 Finalmente, cómo última actividad de esta etapa cada autoridad aporta una clave para el ABV y una clave para CD que inicializarán ambos archivos binarios.

Segunda Etapa

Consiste en dos grandes actividades, bien diferenciadas y que se llevan a cabo por separado con el objetivo de que no se registre relación alguna entre el votante y el identificador de voto.

1. *Autenticación del elector*: se verifica que la persona que va a emitir el voto aparezca en el padrón electoral. El modelo propone que el usuario se registre en el lugar de la elección, en presencia de una autoridad de la Junta Electoral.
2. *Emisión del voto*: para el almacenamiento de los Votos se sigue el esquema MCDU. Para cada voto:
 - a. El sistema selecciona aleatoriamente de la tabla IdentificadoresVoto, un Id y en caso de no estar usado, lo marca como tal. Si ya fue utilizado anteriormente repite el proceso hasta encontrar uno disponible.
 - b. El sistema genera una Clave OTP ClaveVotov (CV_v) del mismo tamaño que ABV que:
 - Aportará a la CD Final de los votos, mediante operaciones XOR.
 - Cifrará la información del voto.
 - c. El elector genera su voto, seleccionado el cargo y el candidato que elige para el mismo, generándose entonces *CadenadeVoto_v*, que luego se combinará con la CV_v para producir la Contribución Final de Voto (CFV).
El aporte de la clave de voto a la CD se lleva a cabo a través de la siguiente operación:

$$CD = CD \oplus CV_v \quad \forall v \quad 1 \leq v \leq N \quad (3)$$

El sistema genera una cadena de *TBslot* bits *ContribucionInicial_v* (CI_v) con todos sus elementos en cero. Produce además un conjunto de números aleatorios $CjtoQ = \{q_i\}$ para cada uno de los Q canales, donde q_i representa el lugar donde se almacenará el voto en el canal i -ésimo.

Se realiza el XOR de la *CadenaDeVoto_v* con los slots que corresponden a los q_i de la CI_v . Esto es:

$$Contribución_{vi} = CadenadeVoto_v \oplus CI_{vi} \quad \forall i \in CjtoQ \quad (4)$$

Finalmente:

$$\begin{aligned} CFV_v &= Contribución_v \oplus CV_v \\ ABV &= ABV \oplus CFV_v \end{aligned} \quad (5)$$

Tercera Etapa

Para completar la etapa de Cierre de la Elección y Recuento de Votos se requiere la intervención de las Autoridades de la Junta Electoral. El proceso de descifrado de los votos consta de tres subprocesos:

1. Aplicación de las mismas claves que las Autoridades usaron en la etapa inicial tanto al ABV como a CD.
2. XOR entre el ABV y la CD resultantes de los pasos anteriores que genera el Archivo Binario de Votos Descifrado (ABVD).

$$ABVD = ABV \oplus CD \quad (6)$$

3. Se procede a la recuperación de los votos [6] generando el Archivo Binario de Votos Descifrado Versión 2 (ABVDv2) que incluye dos bits más por cada tupla con el siguiente significado: (00) tupla no revisada, (01) tupla de ceros, (10) colisión y (11) voto válido y revisa cada una de las tuplas.

3 Propuesta de Mejora de Integridad de Datos de las Tuplas

Como se ha mencionado previamente, en los sistemas de voto electrónico, debe preservarse la seguridad de los datos de los votos durante las 10 horas en las que transcurre el proceso electoral. Para fortalecer esta característica y aumentar la confianza del electorado se realiza un aporte que apunta a mejorar la integridad de los sufragios. A continuación, se especifican para cada una de las etapas, las actividades que se ven afectadas por la propuesta.

Etapa 1

Una de las actividades de esta etapa, es la configuración de la ubicación de los atributos de las tuplas que se almacenan en el archivo ABV. Los atributos de las tuplas pueden residir de forma consecutiva o alternada dentro de la misma. Por ejemplo:

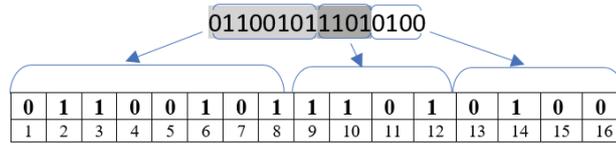


Fig. 3. Tupla con atributos en forma consecutiva

En la Fig. 3, se muestra una tupla con 8 bits para el Identificador de Voto, 4 para el Identificador de Cargo y 4 para el Identificador de Candidato, guardados de forma consecutiva. OTP Vote permite también que los atributos se almacenen en forma dispersa dentro de la tupla. A modo de ejemplo, en la Fig. 4 se muestra el Identificador de Candidato almacenado de forma dispersa.

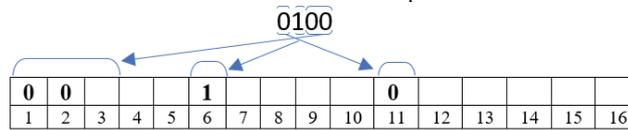


Fig. 4. Atributo Candidato almacenado en forma dispersa

Para fortalecer la integridad de los datos y de esta manera evitar intentos de modificación fraudulentos de los mismos, se propone agregar para cada atributo individual uno o más conjuntos de bits de control (BC), como así también BC a nivel de la tupla completa. Cada BC se calcula como resultado de una función $f(x)$. Dado que los mismos forman parte de la tupla, también pueden almacenarse de forma dispersa y deben cumplir con las mismas características que los demás atributos de datos. Sintéticamente y teniendo en cuenta las tablas de la Fig.2:

Sea LA_i , de longitud A_i

1. La suma de la cantidad de bits consecutivos ocupados de cada grupo de la tabla Ubicaciones para el atributo A_i debe ser igual a LA_i de la tabla Atributos

$$\sum_{j=1}^{\text{Cantidad de Grupos}} A_i LBC_j = LA_i \tag{7}$$

Donde

```
Cantidad de Grupos = SELECT
Count(Ubicaciones.IdUbicacion) AS CuentaDeIdUbicacion
FROM Ubicaciones
GROUP BY Ubicaciones.IdAtributo
HAVING (Ubicaciones.IdAtributo=Ai);
```

```
AiLBCj = SELECT Ubicaciones.CantidadBits,
FROM Ubicaciones WHERE (Ubicaciones.IdAtributo=Ai) AND
(Ubicaciones.IdUbicacion)=j);
```

2. Para todo grupo de tuplas de la tabla Ubicaciones que corresponden a un atributo A_i , debe cumplirse que la intersección de cada una de las instancias que indican bits consecutivos, con todas las demás, devuelva como resultado el conjunto vacío, esto es:

$$\bigcap_{j=1}^{\text{Cantidad de Grupos}} A_i B C_j = \emptyset \tag{8}$$

Donde $A_i B C_j = \{A_i B C_k, A_i B C_{k+1}, \dots, A_i B C_{k+d}\}$, el conjunto formado por los bits consecutivos correspondientes al atributo A_i , en la instancia j de la tabla Ubicaciones, donde:

k = Bit Inicial en la instancia j
 d = Bits consecutivos en la instancia j

- Además, debe controlarse que cada atributo tenga un conjunto propio de bits en el que se alojará y que la intersección de ese conjunto con los bits adjudicados a los demás atributos devuelva cómo resultado el conjunto vacío.

$$\bigcap_{i=1}^{\text{Cantidad de Atributos}} A_i B = \emptyset \tag{9}$$

Donde $A_i B = \{A_i B_j, A_i B_{j+1}, \dots, A_i B_{\text{Cantidad de Grupos}}\}$, el conjunto formado por los subconjuntos de bits consecutivos correspondientes al atributo A_i de la tabla Ubicaciones.

Un ejemplo de la Tabla Atributos puede observarse en la Tabla 1. En la misma se detallan los nombres de los atributos, la longitud, la calidad de los mismos (Datos, BC de atributos o BC de tuplas) y, en el caso de los BC se especifica la función que se aplicará y sobre qué atributo objetivo.

Tabla 1. Ejemplo de la Tabla Atributos.

IdAtributo	Descripción	Long	Calidad	Función	Atributo Objetivo
1	IdVoto	4	Dato		
2	IdCargo	2	Dato		
3	IdCandidato	4	Dato		
4	BCIdVoto1	16	BCAtributo	f(IdVoto)	1
5	BCIdVoto1	16	BCAtributo	g(IdVoto)	1
6	BCIdCargo1	16	BCAtributo	h(IdCargo)	2
7	BCIdCandidato1	32	BCAtributo	p(idCandidato)	3
8	CITupla1	32	BCTupla	K(tupla)	tupla
9	CITupla2	32	BCTupla	m(tupla)	tupla

Etapla 2

En esta etapa, la modificación radica en que luego de que se asigna aleatoriamente un Identificador de Voto al elector, y éste elige un Candidato para el Cargo objeto de la elección, el sistema calcula los BC aplicando las funciones definidas en la tabla Atributos (Fig.2), para cada uno de los atributos y también para la tupla en general.

Etapa 3

Luego de aplicar nuevamente las claves de las autoridades electorales al ABV y a CD se genera el ABVD, que se transforma en el AVBDv2 agregándole 2 bits extras a cada una de sus tuplas y se procede a la recuperación y generación de la tabla de Votos planos, como se detalla en [6]. El algoritmo de Recuperación se modifica incluyendo el proceso de verificación de los BC.

El proceso de *Recuperación* ($ABVDv2, VotosPlanos$), recibe el ABVDv2 y produce la tabla de Votos Planos que se publicará luego de terminado el acto electoral y a partir de la cual se calculará el resultado de la elección.

Este proceso recorre cada una de las tuplas del ABVDv2 y evalúa los bits adicionales que se han agregado a las mismas. En el caso de que estos bits sean (0,0) que significa que la tupla aún no fue revisada o (1,0) que indica que la tupla es el resultado de una colisión, se procede a la revisión de la tupla.

El primer paso de la revisión consiste en aplicar a la tupla la función *TupladeCeros*(*tupla*) que devuelve verdadero si la misma consiste en una secuencia de ceros, lo que indica que en este lugar no reside ningún voto y se le asigna (0,1) a los bits adicionales correspondientes.

En caso de que no sea una tupla de ceros, se recupera el voto que está almacenado mediante el proceso *ObtenerVoto*(*tupla*, *voto*), que obtiene los datos de una tupla y los convierte en el formato esperado para un voto de acuerdo a la configuración que se estableció para la tupla en la Etapa 1.

Luego se aplica la función *VotoValido*(*voto*) que, dado un voto recuperado por el proceso anterior, verifica si los datos son correctos o son el resultado de una colisión. Para ello debe chequear los datos almacenados en las tablas de la Fig. 2. Si la función devuelve falso, significa que el voto es inválido, se le asigna (1,0) a los bits adicionales indicando que fue el resultado de una colisión. En caso de que sea un voto válido (con atributos correctos: Identificador de voto, identificador de cargo e Identificador de candidato), se aplica la función *VerificarBC* (*voto*) que vuelve a calcular las funciones para cada uno de los BC de los atributos y de la tupla general y compara los valores resultantes con los almacenados en las tuplas. Si la función devuelve Verdadero, se almacena el voto en la tabla de votos planos y se asigna (1,1) a los bits adicionales.

4 Caso de Estudio

A continuación, se especifica un caso de estudio en el que puede observarse la detección de una alteración de los datos. Se detallan los datos de las tablas de Identificadores de Votos, Cargos y Candidatos, y el ABVDv2. Para facilitar la interpretación se destinarán los primeros 8 bits de las tuplas al Identificador de Voto, los 4 siguientes al identificador de Cargo, los 4 siguientes al Identificador de Candidato y los 4 restantes al almacenamiento del BC sobre el atributo Id.Candidato.

Archivo Binario de Votos Descifrado V2			
	Archivo Binario de Votos Descifrado	Bit Adic1	Bit Adic2
1	00101000 0110 0100 0001 1000 1001	1	1
2	10001101 0000 0010 0000 0000 0101	1	0
3	10100100 0110 0101 0001 1000 1001	1	0
4	00000000 0000 0000 0000 0000 0000	0	0
5	00000000 0000 0000 0000 0000 0000	0	0
6	00000000 0000 0000 0000 0000 0000	0	0
7	00000000 0000 0000 0000 0000 0000	0	0
8	00000000 0000 0000 0000 0000 0000	0	0
9	11011000 0110 0101 0010 1110 1001	0	0
10	00000000 0000 0000 0000 0000 0000	0	0
11	00000000 0000 0000 0000 0000 0000	0	0
12	01010101 0110 0100 0001 1001 1100	0	0

Identificadores de Voto		
Id Voto	Usado	Recuperado
00101000	Si	Si
10100100	Si	No
11011000	Si	No
01010101	Si	No

Tabla de Candidatos	
Id Candidato	Descripción
0100	A
0101	B

Tabla de Cargos	
Id Cargo	Descripción
0110	Cargo 1

Fig. 5. Datos del caso de estudio

Para el ejemplo se establece un BC para el atributo Identificador de Candidato, la función que se aplicará para la obtención del BC será $h(IdCandidato) = IdCandidato \text{ Mod } 3$. En Fig. 5 se muestra el estado de ABVDv2 al momento de verificación de la tupla 3. El voto residente en la tupla 1 tiene atributos correctos y además al evaluar la función sobre el IdCandidato (0100=4) devuelve 1, que coincide con el valor almacenado para el BC correspondiente, queda marcada como voto válido (1,1). La tupla 2, no corresponde a tupla de ceros, ni a un valor de voto válido, por lo que se marca como colisión (1,0). Al evaluar la tupla 3, devuelve valores válidos para Identificador de voto, de cargo y de candidato, pero, al aplicar la función sobre el IdCandidato (0101=5), devuelve 2 (0010), y el valor almacenado del BC es (0001) por lo que puede detectarse una anomalía, el voto no se marca como válido y se continúa la evaluación y el proceso de recuperación de los votos. Cabe aclarar que dado que el voto se almacena en q canales es posible su recuperación desde alguno de los $q-1$ canales restantes. [6].

En el caso de estudio puede observarse que hubo una modificación no autorizada en el Identificador del Candidato votado (en lugar de registrar el candidato A, se registró el B) que pudo detectarse gracias al uso de BC sobre ese atributo.

5 Conclusiones y Trabajos Futuros

Se presenta en este trabajo una propuesta para mejorar la integridad de datos del modelo OTP-Vote, a través del uso de bits de control (BC) para cada uno de los atributos del voto y también a nivel de tupla en general, que sumada a: el uso de claves de autoridades, las claves combinadas OTP para cada uno de los votos, la redundancia de bits usada en la configuración de los parámetros de la tupla y el proceso de recuperación en caso de colisiones, resultan un aporte integral a la seguridad de los datos usados en OTP Vote.

El modelo permite definir tantos BC como se crean necesarios para los atributos de datos, estableciendo para cada uno de ellos, una función que permita obtener un número que se almacenará en la tupla. Por otro lado, también es posible establecer BC con funciones que se aplican a la totalidad de los atributos de datos de la tupla.

Cuando el voto está en proceso de verificación y recuperación, se aplican nuevamente la/s función/es general/es a los atributos y en caso de que no devuelvan el mismo valor que está almacenado en los bits destinados al/los BC puede concluirse que hubo alguna alteración en los valores originales. Como quedó probado en el caso de estudio pudo detectarse una modificación en el dato del candidato votado, lo que podría afectar el resultado de la elección.

Esta contribución presenta un importante aporte a la integridad de los datos de los votos, reforzando el control sobre la modificación de los mismos. Además, teniendo en cuenta que se propone hacer uso de código abierto para el modelo, la mejora puede llevarse a cabo de forma sencilla, agregando al proceso una función de resultado booleano, que no afecta la legibilidad del algoritmo.

A futuro, será necesario realizar un profundo análisis sobre las funciones que resulten más apropiadas para aplicar a los atributos individuales y a la tupla en general. Además, se deberá trabajar para permitir que el modelo facilite el proceso de auditoría de terceros e incluya mejoras en la verificabilidad E2E.

References

1. Odrisek, B. "E-Voting Security Study" - Communications-Electronics Security Group, X/8833/4600/6/21, (Copyright The Crown) Issue 1.2 31 United Kingdom, 2002
2. Epstein, J. "Electronic Voting", Cyber Defense
3. Kazi Md. Alam Rokibul; Tamura, Shinsuke, "Electronic Voting - Scopes and Limitations" IEEE/OSA/IAPR International Conference on Infonnatics, Electronics & Vision.
4. Prince, A. "Consideraciones, aportes y experiencias para el Voto electrónico en Argentina". 2005.
5. van de Graaf, J., Henrich C., Müller-Quade J. "Requirements for secure voting". 2011.
6. Bast, S. "Confidencialidad e Integridad de Datos en Sistemas de E-Voting – Un Modelo para la Implementación Segura de un sistema de Voto Presencial" - Editorial Académica Española. <https://www.eae-publishing.com>-ISBN 978-3-639-53793-2. 2017.
7. Nagaraj N., Vaidya V., Vaidya P: "Revisiting the one-time pad," International Journal of Network Security, vol. 6, no. 1, pp. 94-102, 2008.
8. Shannon, C. "Communication Theory of Secrecy Systems"-Bell System Tec Journal-1949
9. Broadbent A., Tapp A.: "Information-Theoretically Secure Voting Without an Honest Majority". In Proceedings of the IAVoSS Workshop On Trustworthy Elections (2008).
10. García, P. "Una Optimización para el Protocolo Non Interactive Dining Cryptographers" - Editorial Académica Española (<https://www.eae-publishing.com/> - ISBN-13: 978-3-639-85270-7. ISBN-10: 3639852702. EAN: 9783639852707 – 2017.
11. Bast, Silvia Gabriela; García, Pablo Marcelo; Montejano, Germán Antonio. "Modelo de Datos del Sistema de Voto Electrónico Presencial OTP-Vote". Memorias de las 46 Jornadas Argentinas de Informática e Investigación Operativa (JAIIO). SIE 2017 Simposio de Informática en el Estado. 4 al 8 de setiembre de 2017 UTN. Córdoba. Arg.. ISSN: 2451-7534
12. Bast, Silvia, García Pablo, Montejano Germán: "Generación de Códigos para OTP - Vote". Actas de 5to Congreso Nacional de Ingeniería Informática / Sistemas de Información. Aspectos Legales y Profesionales y Seguridad Informática" ISSN: 2347-0372 © CONAIIISI 2017. Pág 12-22. Noviembre de 2017. Santa Fe.
13. Murdocca M., Heuring V. "Principles of Computer Architecture. Appendix A: Digital Logic". Editor: Addison Wesley; Edición: US ed (29 de noviembre de 1999) Idioma: Inglés - ISBN-10: 0201436647 - ISBN-13: 978-0201436648.

Modelo de Seguridad IoT

Monzon German, Todt Carolina Mariana, Bolatti Diego Angelo, Gramajo Sergio,
Scappini Reinaldo

Universidad Tecnológica Nacional, Facultad Regional Resistencia,
Centro de Investigación Aplicada en Tecnologías de la Información y la Comunicación
(CInApTIC)

French 414 – Resistencia (3500) Chaco - Argentina
{ german.monzon.95, carolinatodt, diegobolatti, sergiogramajo, rscappini}@gmail.com

Abstract. El Internet de las cosas (IoT) no solo conectará computadoras y dispositivos móviles, sino que también interconectará edificios, hogares y ciudades inteligentes, así como redes eléctricas, redes de agua y gas, automóviles, aviones, etc. IoT liderará al desarrollo de una amplia gama de servicios de información avanzados que deben procesarse en tiempo real. Sin embargo, las infraestructuras y servicios de IoT presentan grandes desafíos de seguridad debido al aumento significativo de la superficie de ataque, la complejidad, la heterogeneidad y la cantidad de recursos. En este documento, presentamos un marco de seguridad de IoT para infraestructuras inteligentes como Smart Homes, Smart Grid, Smart Connected Health y otras aplicaciones basadas en IoT.

Keywords: IoT, Ciudades Inteligentes, Seguridad.

1 Introducción

El internet de las cosas (IoT) actualmente vive una fuerte expansión, convirtiéndose en una tendencia irreversible, esto se refleja en la conexión diaria a Internet de miles de dispositivos, sensores y chips los cuales obtienen y distribuyen información por la web para brindar servicios en distintos ámbitos de la vida diaria.

Esta poderosa tecnología, capaz de recolectar y distribuir gran cantidad de información trae asociados nuevos riesgos los cuales podrían amenazar la seguridad de los datos, generando graves consecuencias.

El gran desafío es la complejidad del ecosistema IoT, que impide que la mayoría de las empresas elaboren un marco de seguridad y privacidad. A diferencia de los equipos de TI, los dispositivos conectados no son diseñados pensando en la seguridad, y muchos de ellos no poseen capacidades esenciales de encriptación o autenticación.

La seguridad es una necesidad para los sistemas de IoT para proteger los datos confidenciales e infraestructuras físicas críticas [1]. Sin un buen nivel de protección, los usuarios no pueden adoptar muchos sistemas y aplicaciones de IoT. La seguridad en los sistemas de red tradicionales sigue siendo un desafío, mientras que los sistemas de IoT plantean muchos más desafíos para los investigadores debido a varias

características especiales de estos sistemas. Un análisis profundo es esencial para desarrollar nuevas soluciones de seguridad y sus aplicaciones a sistemas.

En el presente trabajo exploramos cómo mejorar la seguridad en IoT a través de la implementación de un modelo de seguridad. A lo largo del desarrollo de este modelo vamos a considerar la integración con el mundo físico, los dispositivos y las comunicaciones heterogéneas, las restricciones de recursos, la privacidad, la gran escala, la gestión de confianza, el diseño de la seguridad de los dispositivos y las redes en sí mismas.

Este informe se estructura de la siguiente manera, en la sección 2, se hace una breve descripción de la arquitectura de seguridad de IoT. En la sección 3 se presentan las principales amenazas que afectan a el IoT. En la sección 4 se da paso a la presentación del modelo de seguridad de IoT propuesto, y finalmente, en la sección 5 se presentan las conclusiones.

2 Arquitectura de Seguridad de IoT

Como hemos mencionado anteriormente, los dispositivos de IoT se están volviendo omnipresentes, esto crea nuevos tipos de problemas y preocupaciones de seguridad más complejos. Si esos problemas de seguridad no pueden abordarse adecuadamente, se dificultará en gran medida una adopción más amplia de las aplicaciones de la IoT.

Por lo tanto, vamos a enfocar el problema de la seguridad en base a sus aplicaciones, como ser Smart Home, Smart Grid, Smart connected health y otras aplicaciones basadas en IoT como transporte, logística, seguridad pública, video vigilancia urbana, climatología, entre otras [1].

El primer aspecto que tomaremos en cuenta es el servicio de seguridad de capa perimetral (Edge layer security service -EdgeSec). Muchos dispositivos finales como las etiquetas RFID, no tienen recursos suficientes para administrar la seguridad de extremo a extremo. En lugar de que los dispositivos finales gestionen procesos de seguridad por sí mismos, las tareas de administración de seguridad pueden ir desde dispositivos de baja capacidad hasta dispositivos de borde más potentes. En este escenario, el dispositivo final puede tener que confiar en la capa de borde y usarla como el agente de seguridad.

Las ventajas de implementar seguridad en la capa de borde están relacionadas a mayores recursos para cómputo intensivo, cifrado de datos, generación de claves y detección de intrusos, desde dispositivos finales. Además, por estar cerca de los dispositivos finales, se reducen los costos de comunicación y mejoran el rendimiento de aplicaciones en tiempo real. Y, al tener mayor capacidad de información, que los dispositivos finales, es posible implementar una administración de seguridad más optimizada en la capa Edge que puede usarse para proteger la privacidad de los dispositivos finales [2].

3 Amenazas en IoT

Según la recomendación ITU-T Y.4806 podemos clasificar las distintas amenazas hacia los dispositivos IoT según el vector de impacto, que puede provenir del entorno virtual, el entorno físico o del objeto mismo. La figura 1 muestra los distintos vectores de ataque que pueden afectar al sistema [3].

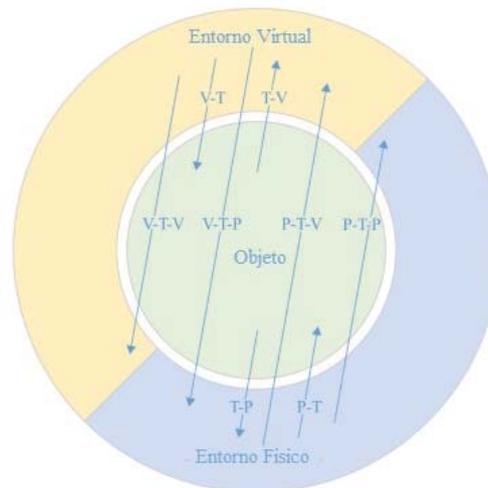


Fig. 1. Vectores de Ataque

El aspecto más importante a estudiar es el vector proveniente del entorno virtual, el cual a través del objeto afecta al entorno físico. Y los que tienen su origen en el comportamiento del objeto, debido a una mala implementación o a que pudiera estar comprometido, dando lugar a problemas de seguridad tanto en el aspecto informativo como funcional [3].

Las principales amenazas que afectan los sistemas de Internet de las Cosas son [3]:

- Manipulación intencional o eludir las restricciones establecidas por los mecanismos de seguridad.
- Desactivación de los mecanismos de seguridad separados debido a las acciones maliciosas del atacante.
- Aprovechar la falta de controles de seguridad inadecuados que, si se implementan adecuadamente, detectarían las acciones que son, o que pueden ser, engañosas (ingeniería social, solicitud de falsificación, etc.).
- La explotación de la falta de, o la aplicación inapropiada de seguridad, que fue limitado por requisitos o exigencias de seguridad relacionadas con la naturaleza de los procesos físicos y con la aplicación del objeto.
- Desactivación de los mecanismos de seguridad debido a las acciones maliciosas del atacante.

2.1 Capacidades para afrontar las amenazas

Para hacer frente a estas amenazas, vamos a recopilar las capacidades de seguridad que podrían dar soporte a la aplicación segura de Internet de las Cosas, como se menciona en la recomendación ITU-T Y.4401[3].

Según esta recomendación, podemos dividir las capacidades de seguridad en 6 grupos [3]:

- **Seguridad en la comunicación**
 - Validación de entradas, control de fuentes, protocolos y flujos.
 - Resistencia a la sobrecarga del canal y denegaciones de servicio.
 - Protocolos de encriptación de datos.
- **Seguridad en la gestión de datos**
 - Control de los comandos para las aplicaciones IoT, de los parámetros y la semántica.
 - Encriptación de datos de aplicación, sumas de comprobación y firmas digitales.
- **Seguridad en la provisión de servicios**
 - Mecanismos de monitoreo, aislamiento de los datos obtenidos y análisis de los componentes, mecanismos de alarmas, aislamiento del monitoreo.
- **Integración de la seguridad**
 - Integración de las reglas y políticas para la validación de entradas en diferentes capas.
- **Autenticación y Autorización mutua**
 - Autenticar y autorizar los sujetos antes de que intenten gestionar y controlar los mecanismos de protección.
- **Auditoría de la seguridad**
 - Monitorear los intentos de gestión y control de los mecanismos de protección.
 - Capacidad de detectar ataques.
 - Monitorear la carga del equipamiento y los canales de comunicación.
 - Detectar ataques en la capacidad de recuperación y respuesta a incidentes.

4 Modelo de Seguridad Propuesto

A continuación, vamos a proponer un modelo de seguridad para redes y dispositivos de Internet de las Cosas, para ello tomamos como base las recomendaciones de la IoT Security Foundation (Fundación de Seguridad IoT), que propone un marco de referencia para cubrir los aspectos de seguridad de los dispositivos IoT.

Este marco de referencia incluye, además de la seguridad del software de los dispositivos, a la seguridad misma del “objeto”, y los procesos de negocio que podrían ocasionar problemas en cuestiones de seguridad, y aspectos relacionados a los

puntos de agregación como Gateway(puertas de enlace) que, si bien no son dispositivos IoT, forman parte de la red por la cual éstos se comunican, y, desde este punto de vista, también incluiremos el cableado y las conexiones inalámbricas de los dispositivos[4].

La Fundación resalta 6 puntos claves para la seguridad en IoT: Gestión Responsable de la seguridad, Diseñado para la seguridad, Ajustado para la criptografía, Asegurar los frameworks de redes y aplicaciones, Asegurar los procesos de producción y la cadena de suministros, y Seguridad para los consumidores [4].

Si bien, este enfoque está orientado a las empresas industriales que proveen productos dentro del rubro de Internet de las Cosas, podemos extraer varios puntos importantes comparando con la recomendación anterior:

- **La gestión de la seguridad:** Desde el punto de vista de los recursos humanos, se recomienda tener personal dedicado especialmente a gestionar y controlar la seguridad de los dispositivos y sus aplicaciones.
- **Seguridad en las redes de comunicación:** Se debe prestar especial atención en que los canales de comunicación, así como las interfaces de red que se utilicen, apliquen respectivamente las medidas de seguridad necesarias para mitigar las amenazas mencionadas anteriormente.
- **Integración de los procesos de seguridad:** el hardware y el software deben estar diseñados para cumplimentar las medidas de seguridad establecidas, en caso contrario, exploraremos la posibilidad de extraer esos requerimientos a una capa de red diferente.

El proceso para determinar los requerimientos de seguridad depende no solo de la naturaleza de los dispositivos IoT que se vayan a utilizar, sino también del entorno objetivo para su implementación.

Como primera medida, se deben determinar los potenciales riesgos que afectarían la seguridad tanto de la información, como de las características físicas en las cuales los dispositivos puedan estar implicados.

Una vez determinada la lista de riesgos, se procede a asignar una probabilidad y un costo en caso de ocurrencia, para obtener un factor de riesgo, y de esta forma poder priorizarlos. Luego estudiaremos los riesgos de un caso de aplicación particular para nuestro proyecto.

El siguiente paso es determinar el nivel de cumplimiento requerido para cada uno de los dispositivos, para esto, vamos a analizar la tríada CIA, constituyendo así 5 clases de cumplimiento [4]:

Clase de Cumplimiento	Objetivo de Seguridad		
	Confidencialidad	Integridad	Disponibilidad
Clase 0	Baja	Baja	Baja
Clase 1	Baja	Media	Media
Clase 2	Media	Media	Alta
Clase 3	Alta	Media	Alta
Clase 4	Alta	Alta	Alta

Tabla 1. Nivel de Cumplimiento

Como podemos observar en la tabla 1, para cada clase de cumplimiento se asigna un nivel de Confidencialidad, Integridad y Disponibilidad. Haciendo un análisis de

esta tabla, vemos que el factor más determinante es la Disponibilidad, ya que está directamente relacionada con la función básica de los dispositivos, que es el de generar y proveer información, y en caso de una falla, se verá afectada la seguridad tanto de las personas como del entorno en el cual están desplegados. Y como segunda medida tenemos la confidencialidad, aunque este punto dependerá de la privacidad de los datos que se están compartiendo a través de la red.

A continuación, realizaremos una descripción de cada una de las clases de cumplimiento mencionadas anteriormente [4]:

- **Clase 0:** Donde la pérdida de control sobre los dispositivos, o el compromiso de los datos, resultan en un impacto poco perceptible para los usuarios.
- **Clase 1:** Donde la pérdida del control o el compromiso de los datos resulta en un impacto limitado para los usuarios.
- **Clase 2:** En adición a la clase 1, el dispositivo está diseñado para resistir ataques que afecten la disponibilidad, que puedan tener un impacto significativo para los usuarios, por ejemplo, limitar las operaciones dentro de la infraestructura a la cual están conectados.
- **Clase 3:** En adición a la clase 2, el dispositivo está diseñado para proteger datos sensibles incluyendo datos personales de los usuarios.
- **Clase 4:** En adición a la clase 3, en estos dispositivos el compromiso con los datos generados o la pérdida de control pueden afectar críticamente la infraestructura o causar lesiones en los usuarios.

Una vez determinada la clase de cumplimiento a la que pertenece nuestro dispositivo, debemos completar la lista de requerimientos para identificar cuales se cumplen, y cuales debemos implementar, variando su característica excluyente u opcional en base a la clase a la que pertenece.

Para abordar estos requerimientos, analizaremos las características principales que se deben estudiar:

- Procesos, políticas y responsabilidades.
- Hardware del dispositivo y seguridad física, incluyendo claves de encriptación para los dispositivos.
- Software del dispositivo y Sistema Operativo.
- Interfaces cableadas e inalámbricas del dispositivo.
- Autenticación y autorización, cumpliendo las normativas de protección de datos y privacidad.
- Seguridad en la capa de aplicación, incluyendo sistemas web y móviles, así como los sistemas de red y la nube.
- La configuración de los dispositivos, así como la guía de usuario para un uso seguro de los mismos.

En este punto, vamos a incluir en nuestro estudio otras recomendaciones, para completar nuestra lista de requerimientos. Primero extenderemos la clasificación de los mismos, analizando el caso de CISCO, que propone un modelo de seguridad para Internet de las Cosas (IoT) y comunicaciones máquina a máquina (M2M), compuesto de 4 componentes: **Autenticación, Autorización, Políticas aplicadas a la red y Analíticas de Seguridad** [5].

Si bien los factores de autenticación y autorización han sido mencionados anteriormente, observamos que el factor de red, que se refiere a los dispositivos que transportarán el tráfico generado por nuestros dispositivos, no deja de ser un factor

sumamente importante para nuestro estudio, estos funcionarán como capa de borde para aplicar las medidas de seguridad que los dispositivos IoT no puedan cumplir.

Como adición a las recomendaciones analizadas anteriormente, vemos que CISCO hace foco en los controles que se realizarán sobre los datos recolectados, para poder detectar anomalías, indudablemente debemos incluir este factor en nuestro estudio[5].

Por su parte la Online Trust Alliance, propone un Marco de confianza IoT que incluye un juego de principios estratégicos necesarios para asegurar los dispositivos IoT y sus datos cuando son enviados y a través de su ciclo de vida. El marco de confianza de IoT se divide en 4 áreas clave: Principios de seguridad, Acceso y credenciales del usuario, Privacidad, divulgaciones y transparencia y Notificaciones y mejores prácticas relacionadas. De estos principios, encontramos un nuevo ítem para nuestra clasificación, las notificaciones y avisos que proporciona el dispositivo al usuario final [6].

Por su parte, el CIS (Center for Internet Security), define una serie de controles de seguridad aplicables a Internet de las Cosas, y cada uno de estos controles vamos a incluirlos en nuestra clasificación. Como así también agregaremos una sección de respuesta a incidentes [7].

Como resultado, nuestra lista de requerimientos constará de 62 recomendaciones de seguridad, que estarán subdivididas en 10 áreas claves [8]:

- **Procesos, políticas y responsabilidades:** Es necesario divulgar completamente las políticas respecto de la recolección, el uso, y como se comparten los datos, además de los términos y condiciones de los parches de seguridad.
- **Hardware del dispositivo y seguridad física:** La seguridad física es probablemente más que un problema, dado que estos dispositivos suelen estar al aire libre o en ubicaciones remotas y cualquiera puede acceder físicamente a ellas.
- **Software del dispositivo y Sistema Operativo:** Es importante mantener un inventario actualizado del software autorizado como así también de los permisos de usuario sobre el sistema operativo.
- **Interfaces cableadas e inalámbricas del dispositivo:** Es necesario adoptar políticas que controlen el acceso y los servicios de red.
- **Autenticación y autorización:** adoptar mecanismos seguros para interactuar y establecer conexiones con dispositivos.
- **Seguridad en la capa de aplicación:** La capa de aplicaciones incluye a todos los dispositivos que tengan conectividad con el dispositivo IoT, lo que puede incluir las aplicaciones web locales, las basadas en la nube y las móviles.
- **Configuración de los dispositivos:** Como en casi cualquier dispositivo IoT, la principal medida de seguridad es una configuración correcta, es indispensable no dejar la configuración por defecto y utilizar usuarios y contraseñas apropiadas para evitar posibles amenazas.
- **Analíticas de Seguridad:** Es necesario contar con una recopilación, monitoreo y normalización de datos de los dispositivos IoT, como así también proveer de informes y alerta sobre actividades específicas o cuando las actividades caen fuera de las políticas establecidas.

- **Respuesta a Incidentes:** Para administrar y recuperarse de violaciones de datos y ataques a la red es necesario contar con servicios de respuestas a incidentes.
- **Notificaciones y Alertas:** Para mantener la seguridad del dispositivo es clave tener mecanismos y procesos que notifiquen rápidamente al usuario si hay una amenaza o una acción requerida.

5 Conclusiones y Trabajo Futuro

A pesar de la gran cantidad de aplicaciones potenciales de IoT y del hecho de que muchas de las tecnologías que soportan IoT ya están en uso, IoT aún no es una realidad en un sentido práctico. Esto puede atribuirse en gran medida a las preocupaciones sobre la seguridad. La naturaleza física inherente de la IoT implica que los ataques cibernéticos, donde el daño normalmente se limita a un mundo virtual, ahora puede presentar una amenaza física real. Esto podría ser una amenaza para la privacidad, una molestia o incluso una amenaza real para la seguridad personal.

En este documento, diseñamos un modelo de seguridad el cual proporciona autenticación esencial, garantiza comunicaciones seguras, admite la autorización del usuario y ofrece notificación de riesgos, satisfaciendo así las principales preocupaciones de los usuarios. El marco puede servir como base para respaldar la seguridad de las aplicaciones de IoT.

Si bien el marco que hemos desarrollado ayuda a cumplir con los requisitos básicos de seguridad, también nos permitió reafirmar los conceptos desarrollados durante la investigación:

- Los componentes que más nos interesan proteger son los que forman parte de las Interfaces cableadas e inalámbricas del dispositivo, ya que son el punto de acceso para establecer la comunicación con los dispositivos.
- Dentro de las interfaces, es muy importante implementar un buen mecanismo de autenticación, ya que es la forma más efectiva de evitar el acceso de usuarios maliciosos que puedan generar algún daño en el sistema.
- El segundo factor importante que se debe implementar es el de cifrado de los datos, esto se vuelve mucho más importante aún para las comunicaciones inalámbricas.
- El factor más difícil de proteger es el hardware del dispositivo, esto se debe a razones lógicas, ya que las prestaciones que ofrecen los mismos, se ven limitadas a cumplir su propósito con el menor costo en términos de fabricación e implementación.
- Aunque no podamos realizar mayores modificaciones en el hardware, este puede verse muy comprometido en el caso de que un usuario malicioso logre evadir la autenticación, y tenga acceso al software del dispositivo, este podría generar un comportamiento que afecte al componente físico del dispositivo, así como al entorno físico del mismo, incluyendo a los propios usuarios.

Vamos a analizar este último punto, si recordamos los vectores de ataque que mencionamos al inicio del documento, vemos que el caso mencionado se corresponde

con el vector V-T-P (Virtual-Thing-Physical), que es el proveniente del entorno virtual, que, a través del acceso a las funcionalidades del dispositivo, logra afectar al entorno físico. Concluimos en que nuestra clasificación se corresponde con la información recolectada previamente.

Además, logramos descubrir dos puntos importantes en la aplicación de un modelo de seguridad para Internet de las Cosas:

- Es muy importante mantener políticas de seguridad a las cuales responden los demás requerimientos.
- Se debe notificar al usuario la mayor cantidad de información posible relacionada con eventos que afecten a la seguridad del sistema, ya que la interacción entre estos está limitada a su funcionalidad específica.

Como trabajo futuro, se planea investigar y complementar al modelo de seguridad propuesto la seguridad de las redes LoraWan, el cual es una especificación técnica de una red LPWAN (Low Power Wide Area Network) propuesta por la Lora Alliance. Esta red nace para cubrir la necesidad de comunicar dispositivos IoT o M2M, de bajo coste y bajo consumo energético a grandes distancias.

Referencias

1. N. Ekedebe, W. Yu, C. Lu, H. Song, Y. Wan. Securing transportation cyber-physical systems. *Securing Cyber-Physical Systems*, CRC Press, Boca Raton, pp. 163–196. 2015.
2. L. Sweeney. K-anonymity: a model for protecting privacy, *Int. J. Uncertain. Fuzziness Knowl. -Based Syst.* 10 (5) 557–570.2002.
3. International Telecommunication Union (ITU). (2017). Recommendation ITU-T Y.4806.
4. IoT Security Foundation. (2018). IoT Security Compliance Framework. Release 2.
5. Cisco. (2019). Securing the Internet of Things: A Proposed Framework. [online] Available at: <https://www.cisco.com/c/en/us/about/security-center/secure-iot-proposed-framework.html>.
6. Marco de confianza y confidencialidad de IoT v2.5 | Internet Society. (2018). Retrieved from <https://www.internetsociety.org/es/resources/doc/2018/iot-trust-framework-v2-5/>
7. CIS Critical Security Controls. (2015). Internet of Things Security Companion to the CIS Critical Security Controls.
8. Monzon, G., Todt, C., Bolatti, D., Gramajo, S., & Scappini, R. (2019). Modelo de Seguridad IoT. Retrieved from: <http://monwatch.firre.utn.edu.ar/files/ListaDeRequerimientos>

Procedimientos de CiberSeguridad en un Laboratorio de Educación, Desarrollo e Investigación (EDI) para la identificación de vulnerabilidades en su red informática.

Fabián Gibellini; Roberto Muñoz; Analía Lorena Ruhl; Juliana Notreni;
Cecilia Sánchez; Ignacio Sánchez; Marcelo Auquer; Milagros Zea.

Universidad Tecnológica Nacional – Facultad Regional Córdoba – Maestro Marcelo Lopez esq. Cruz Roja Argentina, Ciudad Universitaria (X5016ZAA)- Córdoba, Argentina.

{fabiangibellini, robertmunioz, analialorenaruhl, julinotreni, csanchezjuriol, ignaciojsb, marcelo.auquer, milyzc}@gmail.com

Resumen

Cuando se habla de seguridad de una red informática se debe tener en cuenta tanto la seguridad física como la seguridad lógica de la misma. Se tiene que considerar todas las actividades, técnicas y herramientas relacionadas con el fin de proteger los datos que manejan los sistemas de la organización, tanto sistemas de uso interno como los sistemas expuestos a los usuarios finales, con la premisa que estos datos no sean accedidos o interferidos por personas no autorizadas, ni destruidos accidental o intencionalmente. Este estudio está inserto y forma parte de un proyecto de investigación denominado “Determinación de Indicadores, técnicas y herramientas que evidencian buenas prácticas en la ciberseguridad de la infraestructura tecnológica en un laboratorio de Educación, Investigación y Desarrollo de la UTN - FRC.”, homologado por la Secretaría de Ciencia y Tecnología bajo el código SIUTNCO0005366

La gran contribución de este paper es un modelo de ciberdefensa en profundidad que relaciona las vulnerabilidades latentes de la red informática y sus medidas de seguridad a través de la cual la vulnerabilidad es mitigada.

Palabras clave: ciberseguridad; redes informáticas; seguridad

1 Introducción

La seguridad informática engloba tanto la seguridad física, seguridad en los sistemas desarrollados y administración de la seguridad en una red informática. Según la norma ISO/IEC 27002 la seguridad física o ambiental tiene como objetivo “evitar el acceso físico no autorizado, los daños e interferencias a la información de la organización y las instalaciones de procesamiento de la información” [1].

Por otro lado, administrar la seguridad lógica en una red informática, es de vital importancia porque está generalmente tiene más de un servidor y maneja datos de

muchos sistemas. Han existido ataques a través de internet que han causado graves daños a diferentes sectores (por ejemplo: personal, económico, empresarial, organizaciones y gobierno), entre los que podemos recordar el ataque a Sony Pictures en el 2014 que le costó a la empresa quince millones de dólares [2].

Se han creado diferentes procedimientos, procesos o técnicas que ayudan a implementar seguridad en una red como procedimientos de autenticación y autorización a la red [3], sistemas de detección de intrusos (IDS), IDS basados en data mining y machine learning de forma complementaria [4].

Rojanakul y Liang [5] afirman que la estabilidad y seguridad de la red es muy importante enfocándose en una estrategia de asegurar y configurar políticas de seguridad para resguardar la infraestructura. Para lo cual divide la red en dos zonas, la de uso externo y la de uso interno.

Dees y Rahman [6] identifican ciertas contramedidas para estas amenazas, como implementar una infraestructura fuerte y luego asegurarla a través de la implementación de capas y políticas en cada capa, encriptar transmisiones (cableadas y no cableadas), educar y entrenar a los empleados en las cosas que pueden y no pueden hacer mientras están conectados a la red, educar en la necesidad de crearse usuario y contraseña que sean seguras.

Otro de los procedimientos que suman y se han vuelto clave al momento de implementar seguridad son los backups [7] de toda información (documentos, programas, servidores, máster de equipos) que se crea necesarios para el correcto funcionamiento.

Un laboratorio de Educación, Investigación y Desarrollo (EDI), cuenta con un gabinete informático que debe estar preparado para estudiantes, docentes, profesionales e investigadores, donde las necesidades y exigencias día a día son mayores en lo que concierne a software, aplicativos y hardware. Es indispensable contar con la infraestructura acorde para afrontar los diversos pedidos que se realizan por parte de todos los actores involucrados.

Teniendo presente la heterogeneidad de software que se debe implementar para satisfacer las necesidades de cada cátedra/materia/curso que solicita un servicio y de los investigadores que integran los Proyectos de Investigación, Desarrollo e Innovación que se llevan a cabo utilizando los mismos recursos, es indispensable llevar a cabo acciones que den continuidad a los servicios protegiendo la confidencialidad, integridad y disponibilidad de estos.

En el caso del Laboratorio de Ingeniería en Sistemas de Información (LabSis) de la UTN-FRC se atienden a todos los servicios nombrados anteriormente y se tiene especial cuidado en la disponibilidad e integridad de los datos que intervienen en los servicios prestados, ya que el dictado de clases, toma de parciales y exámenes finales e investigación dependen de estas características.

En LabSis, se aplican distintas técnicas y herramientas de seguridad en las diferentes capas de la red para evitar que, accidental o intencionalmente, algún servicio quede fuera de funcionamiento. Si bien el LabSis está en continuo funcionamiento, es real la necesidad de hacer un análisis de reingeniería de los métodos y procesos hasta ahora implementadas con el objetivo de alcanzar un conjunto de procedimientos que sinérgicamente colaboren a la ciberseguridad de la infraestructura tecnológica de un laboratorio de Educación, Desarrollo e Investigación (EDI) y desarrollar una base de

conocimientos basadas en indicadores de vulnerabilidades, técnicas y herramientas que fortalezcan la seguridad.

La principal necesidad identificada fue mitigar los riesgos relacionados a las vulnerabilidades de la red, en la cual coexisten tres modelos de uso de la infraestructura Educación, Desarrollo e Investigación (EDI), de forma tal de evitar un mal uso que genere pérdidas o daños, ya sea por desconocimiento o equivocación de los usuarios de la red informática.

En cambio, en un ambiente de educación, desarrollo e investigación, más precisamente un laboratorio EDI, un modelo de organización tradicional no es del todo representativo, dado que las actividades que se realizan en el mismo no son cotidianas, o mejor dicho el periodo para que puedan ser consideradas lo más cercano a rutinarias es más amplio que un día, por ejemplo, si se considera la parte de educación, en los servicios que se brindan en gabinete informático a docentes y estudiantes, la rotación de estudiantes cambian todos los años para cada cátedra, al igual que en los talleres eventuales, como lo son las competencias de programación [8], los seminarios y olimpiadas informáticas [9], y en lo que compete a investigación y desarrollo se tiene una variabilidad media debido a que los proyectos tienen una duración entre dos y tres años y, una vez implementados ó finalizados, su variabilidad pasa a ser constante.

Por todo lo expuesto se puede plantear que el objetivo principal presentado en este proyecto es “Determinar los factores que colaboren en el control de seguridad de la red informática en el ámbito de un laboratorio de Educación, Investigación y desarrollo (EDI), proponiendo técnicas, herramientas e indicadores que evidencian buenas prácticas en la ciberseguridad de la infraestructura tecnológica del Laboratorio de Ingeniería en Sistemas de Información de la UTN - FRC (LabSis)”.

2 Metodología

Para el desarrollo de este proyecto se utilizará el método empírico [10][11], siendo que en algunas actividades utilizarán el método científico durante su ejecución, las cuales culminarán en la aceptación o refutación de hipótesis sobre evidencias de buenas prácticas en LabSis.

Toda investigación o estudio llevado a cabo, fluctúan sobre la seguridad informática en redes de información que operan sobre infraestructuras tecnológicas de un ámbito público, como ser el Laboratorio de Ingeniería en Sistemas de Información (LabSis) de la Universidad Tecnológica Nacional de la Facultad Regional Córdoba.

Teniendo en cuenta que los servicios que debe prestar esta entidad a docentes y la protección de datos sensibles (uno de ellos son los parciales y exámenes finales) y cuya incumbencia concierne únicamente a la Universidad en que se realiza, es que este proyecto se inserta dentro de la línea de investigación de ciberseguridad en una infraestructura tecnológica.

Se realizará una investigación exploratoria a los fines de profundizar y poder cumplir con la detección de las técnicas y herramientas de ciberseguridad utilizadas en actividades relacionadas a la seguridad informática de la red, realizando un estudio y análisis de las técnicas y herramientas de ciberseguridad.

Posteriormente se analizarán las actividades que se realizan en el LabSis identificando las técnicas y herramientas de ciberseguridad utilizadas actualmente con el fin de determinar un conjunto de indicadores cuantificables relacionados a la ciberseguridad que sean aplicables en un laboratorio EDI, de esta manera daríamos cumplimiento al objetivo planteado.

Para determinar el conjunto de buenas prácticas del LabSis con las técnicas y herramientas que contribuyan a la ciberseguridad en laboratorios de Educación, desarrollo e investigación se dará categoría a los actores involucrados en el LabSis y a los servicios exigidos por los mismos, realizando una evaluación del análisis de riesgo con el que cuenta actualmente el LabSis. De la evaluación que se realice se procederá a analizar y seleccionar indicadores que muestren evidencias de buenas prácticas. Se formulará una descripción de la infraestructura y para poder crear una base de conocimiento relacionada a ciberseguridad que a partir de los registros de eventos de la infraestructura tecnológica del LabSis y de recolectar información pertinente a los indicadores identificados.

Se realizará el análisis de las técnicas de redes neuronales y machine learning que colaboren con la seguridad del LabSis, identificando a dichas técnicas como método de prevención y de colaboración con la ciberseguridad de infraestructura del LabSis. Con las recolecciones tomadas y las mediciones que se llevan a cabo, se definirá una guía de buenas prácticas para laboratorios EDI que fortalezcan la seguridad.

3 Desarrollo

En primera medida es necesario elaborar Mapas de Procesos que permitan conocer mejor el funcionamiento del laboratorio EDI, detectando procesos y en cada uno ellos los procedimientos que están relacionados con cada proceso. Con el mapa de procesos y el detalle de cada uno de los procesos identificados se debe realizar un análisis de riesgos basados en los procedimientos que detallan las actividades de cada proceso.

El análisis de riesgo obtenido debe ser evaluado y aprobado por el personal y dirección que integran el laboratorio EDI. Para el mismo se debe tener en cuenta la realización del inventario de activos más críticos, luego se deben identificar las amenazas que pueden afectar a los activos y valorando cada una de ellas basada en el impacto que puede tener si las misma ocurrieran, también se debe identificar las vulnerabilidades de los activos y ponderarlas. Una vez identificados los activos, amenazas y vulnerabilidades se debe realizar un análisis de las medidas de seguridad, técnicas y herramientas que ya están implementadas en el laboratorio EDI. Esto es la base para diseñar una metodología de ciberseguridad basado en un modelo de defensa.

3. Resultados, Avances/Discusión

Se plantea el siguiente esquema para identificar medidas existentes y necesarias para cubrir las amenazas identificadas, especificando las técnicas y herramientas a considerar en cada capa:

1- Capa de Políticas y Procedimientos <ul style="list-style-type: none"> - Política de Seguridad de la Información - Organización de la Seguridad - Roles y Responsabilidades de Seguridad - Procedimientos y estándares - Capacitación de seguridad 	2- Capa de Seguridad Física: <ul style="list-style-type: none"> - Control de Acceso Físico - Alarmas. - Seguridad Perimetral - Controles Ambientales
3- Capa de Perímetro <ul style="list-style-type: none"> - Test de Penetración - Firewall. - VPNs - Sistema de detección de intrusos (IDS) - Sistema de Protección de intrusos (IPS) 	4- Capa de Red Interna <ul style="list-style-type: none"> - Listas de Control de Acceso (ACLs) - Red de área local virtual (VLANs) - Secure Shell (SSH) - Secure Sockets Layer (SSL) - Transport Layer Security (TLS) - Auditorías de Seguridad
5- Capa de Host <ul style="list-style-type: none"> - Hardening del Sistema Operativo - Registros de Seguridad (Logs) - Control de contraseñas. - Antimalwares/Antivirus. - Escaneo de vulnerabilidades - Gestión de Parches de Seguridad 	6- Capa de Aplicación <ul style="list-style-type: none"> - Código Seguro - Controles de acceso - Políticas de contraseñas - Gestión interna de los ambientes de desarrollos. - Seguridad en el ciclo de vida del desarrollo del software
7- Capa de Datos <ul style="list-style-type: none"> - Criptografía/ofuscación - Data Leak Prevention (DLP) - Sanitización de los datos - Clasificación - Resguardo de los datos 	

Cada técnica o herramienta descrita es considerada un punto de control que colabora a la ciberseguridad en el modelo de defensa planteado. Teniendo en cuenta las amenazas identificadas en el análisis de riesgo debe ser siempre posible lograr una trazabilidad para cada una de estas amenazas a una medida de seguridad implementada en el laboratorio EDI (Tabla 1).

Vulnerabilidades identificadas	Ponderación de la vulnerabilidad	Medida de Seguridad implementada en el Laboratorio

Tabla 1. Cuadro de relación de trazabilidad entre amenaza identificada (Vulnerabilidad) y medida de seguridad implementada.

La ponderación es un atributo de la vulnerabilidad y está relacionado al impacto de la amenaza que representa dicha vulnerabilidad en el análisis de riesgo.

El trabajo expuesto tiene como finalidad obtener una visibilidad global de las amenazas o vulnerabilidades latentes en una red informática de un laboratorio EDI. De forma tal que cada amenaza identificada pueda ser rastreada a una medida de seguridad implementada en cada capa del modelo de defensa planteado. A partir de esto se puede detectar indicadores que puedan ser medidos y posteriormente conformar una guía de buenas prácticas con una base de ciberseguridad a Laboratorios de Educación, Desarrollo e Investigación de organizaciones, principalmente educativas como universidades, institutos educativos de educación media y superior, academias y otras organizaciones que tengan infraestructuras tecnológicas para brindar servicios.

El modelo de defensa presentado fue planteado teniendo en cuenta la heterogeneidad de los usuarios de un laboratorio EDI y la masividad de usuarios que pasan diariamente por estos laboratorios, identificando los perfiles de estos y asegurando que puedan realizar las actividades planificadas. Por lo que laboratorios con las mismas características pueden aplicarlo, lo que no quita el hecho que entornos más homogéneos también lo puedan hacer.

Referencias

- [1] Norma Chilena. Tecnologías de la información - Código de prácticas para los controles de seguridad de la información. Número de referencia NCh-ISO 27002:2013. Pág. 43. (2013).
- [2] The Interview revenge hack cost Sony just \$15m. Recuperado de <https://www.theguardian.com/film/2015/feb/04/guardians-peace-revenge-hacksony-finances-unscathed>. (2014).
- [3] Todd, M., Rahman, S. (Noviembre 2013). COMPLETE NETWORK SECURITY PROTECTION FOR SME'S WITHIN LIMITED RESOURCES. International Journal of Network Security & Its Applications (IJNSA). Vol. 5, Nro. 6.
- [4] Zurutuza, R., Uribeetxeberria, R. (). Revisión del estado actual de la investigación en el uso de data mining para la detección de intrusiones.
- [5] Rojanakul, K., Liang, H. (2009). Network Security Infrastructure Management. IEEE.
- [6] Dees, K., Rahman, S. (2011). ENHANCING INFRASTRUCTURE SECURITY IN REAL ESTATE. International Journal of Network Security & Its Applications (IJNSA), Vol.3, No.6.
- [7] Dharma, R., Sake, S., Manuel, M. (2013). Backup and Recovery in a SAN. Versión 1.2. EMC2 Techbooks.

- [8] Castillo, J., Cárdenas, M., Serrano, D. (2011). Experiencias en el Desarrollo de Competencias de Programación en UTN-FRC. Dpto. Ingeniería en Sistemas de Información - Universidad Tecnológica Nacional – Facultad Regional Córdoba. TE&ET 2011. Recuperado el 12, Junio, 2018 de <http://sedici.unlp.edu.ar/handle/10915/18414>.
- [9] Marciszack, M., Muñoz, R., Castillo, J., Delgado A., Serrano, D., Gatto, S. (2013) Colaboración entre el Gobierno de la Provincia de Córdoba y la UTN - FRC para el desarrollo de Olimpíadas Informáticas". Recuperado el 12, Junio, 2018 de <http://conaiisi.frc.utn.edu.ar/PDFsParaPublicar/1/schedConfs/4/234-650-1-DR.pdf>.
- [10] Bunge, M. (1998). La ciencia su Método y su Filosofía. Editorial Siglo Veinte. Buenos Aires
- [11] Barchini (2005). G. Métodos “I+D” de la Informática. Universidad Nacional de Santiago del Estero, Argentina.



CACIC 2019
Congreso Argentino de Ciencias de la Computación

II Track de Gobierno Digital y Ciudades Inteligentes

Coordinadores

Estévez, Elsa (UNS)

Pasini, Ariel (UNLP)

Temas

- Administración de cambios en instituciones públicas
- Administración de conocimiento en gobierno
- Administración de interesados
- Administración de registros públicos digitales
- Arquitecturas empresariales gubernamentales
- Asociaciones público-privadas
- Automatización de procesos administrativos gubernamentales
- Calidad de datos abiertos
- Censura y restricción de datos abiertos
- Ciberseguridad en gobierno
- Co-creación de datos abiertos para beneficios sociales
- Compartir información de gobierno
- Computación en la nube gubernamental
- Diseño y entrega de servicios públicos digitales
- Estándares de metadatos para datos abiertos
- Infraestructura tecnológica para datos abiertos
- Innovación con datos abiertos
- Integración de procesos administrativos gubernamentales
- Marcos de interoperabilidad
- Portales de información de gobierno
- Protección de datos personales
- Reingeniería de procesos administrativos gubernamentales
- Seguridad de la infraestructura de información de gobierno
- Servicios compartidos
- Sistemas y servicios de energía inteligente
- Sistemas y servicios de movilidad inteligente
- Sistemas y servicios para ciudades inteligentes
- Sistemas y servicios para gestión del agua
- Sistemas y servicios para protección del medio ambiente
- Soluciones para participación ciudadana
- Tableros de control para gestión municipal

Blockchain y gobierno digital

Juan Santiago Preisegger¹, Rocío Muñoz¹, Ariel Pasini¹, Patricia Pesado¹

Instituto de Investigación en Informática LIDI (III-LIDI)*
Facultad de Informática – Universidad Nacional de La Plata 50 y 120 La Plata Buenos Aires
*Centro Asociado Comisión de Investigaciones Científicas de la Pcia. de Bs. As. (CIC)
¹ Becario UNLP

{jspreisegger,rmunoz,apasini,ppesado}@lidi.info.unlp.edu.ar

Abstract. En los últimos años blockchain provocó un auge en lo que respecta a la seguridad, confiabilidad y transparencia de los datos. Los gobiernos ven la utilización de esta tecnología como una mejora en la prestación de servicios. Se propone clasificar las aplicaciones en una estructura de gobierno de seis áreas, *Seguridad, Producción y economía, Transporte, Relaciones interiores y exteriores, Salud y Educación, ciencia y tecnología*. Que permite analizar las aplicaciones en función de las áreas y las características de blockchain que le permite obtener un mayor beneficio. Argentina posee importantes avances a través de *Blockchain Federal Argentina* (BFA), una plataforma multiservicios abierta y participativa pensada para integrar servicios y aplicaciones sobre blockchain. Para determinar el avance en el área se relevaron 70 aplicaciones relacionadas con las áreas de gobierno, que se están ejecutando actualmente. Luego se analizó cada una de ellas según su tipo y el área de gobierno que se ve beneficiada con la utilización de esta tecnología.

Keywords: Ingeniería de Software, blockchain, gobierno digital.

1 Introducción

En los últimos años blockchain provocó un auge en lo que respecta a la seguridad, confiabilidad y transparencia de los datos. Los gobiernos ven la utilización de esta tecnología como una mejora en la prestación de los servicios de gobierno.

Blockchain se define como una “una base de datos distribuida entre diferentes participantes, protegida criptográficamente y organizada en bloques de transacciones relacionados entre sí matemáticamente.” [1]. Existen tres tipos de Blockchain [2], *públicas, privadas y federadas o híbridas*, la principal diferencia entre estos tres tipos es el acceso para formar parte de esta red, en las públicas puede participar cualquier usuario, en las privadas se requiere de un permiso otorgado por un administrador y las federadas son una combinación de las dos anteriores.

Los beneficios de esta arquitectura se relacionan con seis características: la *administración*, la *persistencia*, la *identificación*, la *auditabilidad*, la *participación* y la *transparencia*.

Analizando estas características aparece un conjunto de aplicaciones que se ven ampliamente beneficiadas, como, por ejemplo, aplicaciones relacionadas con *registros públicos*, sistemas relacionados con la *trazabilidad*, aplicaciones *financieras e IoT* (Internet de la Cosas).

Un gobierno digital se puede ver muy beneficiado con el uso de Blockchain [3]. Para facilitar la clasificación de las aplicaciones se propone una estructura de gobierno de seis áreas, *Seguridad, Producción y economía, Transporte, Relaciones interiores y exteriores, Salud y Educación, ciencia y tecnología*. Dicha estructura permite analizar las características mencionadas y los tipos de aplicaciones que pueden beneficiarse.

Argentina posee importantes avances en blockchain a través de la Secretaría de Gobierno de Modernización de la Nación [4], que forma parte de *Blockchain Federal Argentina* (BFA): una plataforma multiservicios abierta y participativa pensada para integrar servicios y aplicaciones sobre blockchain, que se encuentra en constante crecimiento [5].

Para determinar el avance en el área se relevaron más de 150 casos, quedándonos con 70 aplicaciones relacionadas con las áreas de gobierno que se están ejecutando actualmente. Luego se analizó cada una de ellas determinando las áreas de gobiernos y tipos de aplicaciones más avanzados.

En la sección 2 se presenta una descripción de blockchain, sus características y posibles aplicaciones, en la tercera sección, se presenta los beneficios de la tecnología en las diferentes áreas de gobierno. Luego, el avance de la tecnología en Argentina seguido por el relevamiento de aplicaciones en las áreas de gobierno y su correspondiente análisis. Finalizando con las conclusiones.

2 Blockchain

Blockchain se define como “una base de datos distribuida entre diferentes participantes, protegida criptográficamente y organizada en bloques de transacciones relacionados entre sí matemáticamente.” [1]. Dentro de blockchain tenemos diferentes partes: la *base de datos distribuida* donde se registran todas las transacciones (o guardan los datos, dependiendo de la infraestructura); el *bloque de transacciones*, que contiene las transacciones con las correspondientes identidades digitales de quien las realiza y quien las recibe y la identificación del bloque anterior; los *nodos*, que son computadoras que conforman la red con diferentes roles; el *algoritmo de encriptación con su correspondiente timestamp*, que permite la validación del bloque; y las *Wallet*, que son interfaces gráficas que permiten a los usuarios realizar transacciones y manejar sus identidades digitales.

2.1 Tipos estructurales de blockchain.

Blockchain posee tres tipos de estructuras: **1) Públicas:** No hay restricciones para la lectura de datos ni para la realización de operaciones por parte de los usuarios, es abierta para todos y cualquiera puede participar en ella. **2) Privadas:** La lectura de datos y las operaciones sobre la red están limitadas a participantes determinados, los usuarios

deben obtener permisos por parte del administrador de la red para poder utilizarla. **3) Híbridas/Federadas:** Utiliza lo mejor de las soluciones de las blockchain públicas y privadas; existe un número determinado de organizaciones, entidades o compañías que se encargan de administrar la red y mantener las copias del registro sincronizadas, el acceso a la red se hace mediante una interfaz web que los administradores ponen a disposición del usuario brindando, al mismo tiempo, un acceso controlado y libertad.

2.2 Características de una blockchain

En función de los tipos estructurales de blockchain se presentan las siguientes características:

Administración: La estructura de blockchain va a determinar de forma directa su administración. En las redes privadas hay un sistema donde una única entidad se encarga de mantener la cadena, a este se lo denomina *centralizado*. En las híbridas/federadas hay varias entidades que se encargan de esto, por lo que podemos decir que se trata de un sistema *semi-centralizado*. Por último, las redes públicas donde una transacción puede realizarse entre dos pares cualquiera (P2P), sin la intervención del ente central, conforman un sistema *descentralizado*, el cual da lugar al concepto de “Validación sin Intermediarios”, donde no se requiere de una persona, empresa o institución que legitime la información guardada.

Persistencia: Cada transacción que se extiende por la red blockchain debe ser validada por todos los nodos. Para esto es agrupada en bloques, junto con otras transacciones, los cuales son confirmados y registrados de manera distribuida, lo que hace imposible su manipulación debido a la forma de encriptación y linkeo entre los bloques. Si se intenta modificar un bloque, se deben modificar todos sus sucesores en la red. Se generan los conceptos de “Seguridad e inmutabilidad”, debido a que toda la información alojada es inamovible y de “Registro creciente”, ya que la información de la red no puede ser adulterada.

Identificación: En base al tipo de estructura de blockchain la identificación puede variar. En el caso de las blockchain privadas o Híbridas/Federadas, se requiere de la identificación de los usuarios para obtener permisos y poder realizar transacciones. Una red blockchain pública permite la interacción a través de la utilización de claves generadas, por lo que un usuario no se identifica con sus datos personales, pudiendo ocultar su identidad, aunque realizando un seguimiento podría revelarse cierta información de estos.

Auditabilidad: Las transacciones de todos los tipos de blockchain se validan y registran con una marca de tiempo. Los usuarios, según el nivel de permisos otorgados, pueden verificar y rastrear fácilmente los registros anteriores accediendo a cualquiera de los nodos de la red mejorando la “transparencia” y “trazabilidad” de los datos almacenados. Se garantiza un registro inalterable de transacciones debido a su solidez e inmutabilidad, permitiendo realizar un seguimiento de la información.

Participación: Esta característica determina el grado de intervención que se le permite a cada usuario que actúa en la red. En el caso de las redes públicas, cualquier usuario puede formar parte de la red colaborando como un nuevo nodo y utilizar sus servicios como simples usuarios. En las antípodas se encuentran las redes privadas, en las cuales solamente se les permite a los usuarios utilizar la red si el administrador lo permite. En las redes Híbridas/Federadas, por su parte, los administradores pueden permitir a los usuarios sumarse a participar siempre y cuando cumplan con ciertos acuerdos, y se hayan registrado previamente.

Transparencia: En función del nivel de participación y del modo de administración, se obtendrá un determinado grado en el nivel de transparencia de la blockchain. Esto determinará si cualquier persona puede velar por la integridad de la información contenida en la red o no. Se puede decir que las blockchain privadas tienen un muy bajo nivel de transparencia (o nulo) ya que generalmente los usuarios no tienen acceso a la información interna de la misma. En las redes Híbridas/Federadas el nivel de transparencia está determinado por el acceso a la información que los administradores brinden a los usuarios comunes, y las blockchain públicas tendrán el nivel de mayor transparencia ya que todos los nodos tienen acceso a toda la información almacenada.

En la tabla 1, se puede observar el resumen del análisis previamente explicado de las características para cada tipo de Blockchain.

Table 1. Características de blockchain según el tipo de estructura

		Tipos de Blockchain		
		Privada	Híbrida/Federada	Pública
Características	Administración	Centralizada	Semi-Centralizada	Descentralizada
	Persistencia	✓	✓	✓
	Identificación	✓	✓	✗
	Auditabilidad	✓	✓	✓
	Participación	✗	✗	✓
	Transparencia	✗	—	✓

✓ sí, ✗ no, — depende de las organizaciones

2.3 Aplicaciones de blockchain

El uso de blockchain habilita la programación de aplicaciones permitiendo realizar flujos de tareas programables entre partes interesadas, que pueden ejecutarse como consecuencia de transacciones y se apoyan en todas las garantías de confianza y transparencia que nos da blockchain. A esta característica se la denomina “**Smart Contracts**” (contratos inteligentes) y se la define como un “protocolo de transacción computarizado que ejecuta los términos de un contrato” [6] permitiendo comenzar a pensar procesos como seguimiento de licitaciones, sistemas de trazabilidad de productos, plataformas de documentos “vivos”, etc.

El uso de esta característica da origen innumerables aplicaciones que permiten explotar los beneficios de Blockchain. Analizando los modelos propuestos por [7] [8], se agruparon las aplicaciones en: Registros Públicos, Trazabilidad, Financieras e IoT. Como puede verse en la Figura 1.

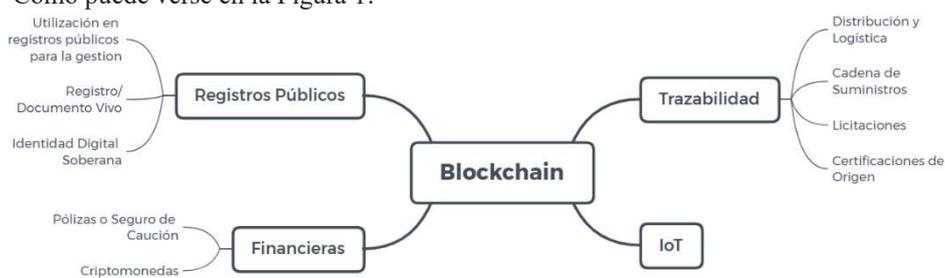


Fig. 1. Clasificación de aplicaciones de blockchain

Registros públicos: Institución que funciona como nexo entre los ciudadanos y los agentes de un Estado, bajo regulación y control de este. A través de los registros públicos el ciudadano puede presentar solicitudes, escritos y comunicaciones a las administraciones públicas. Además, mediante estos, se registran los documentos que remite al ciudadano, a entidades privadas o a la propia administración. La actualización constante de los documentos del ciudadano, reflejando los cambios correspondientes y su trazabilidad dan origen al concepto de “*Documento vivo*”. Un ejemplo de documento vivo es la “*Identidad digital soberana*”, los sistemas de identidad en línea se crean con la finalidad de asociar datos con los individuos. El usuario tiene el control de sus datos, es decir que pasa a ser el verdadero dueño de sus datos y puede elegir qué y a quién facilitárselos, como así también a quien revocarle el permiso para accederlos.

Sistemas que requieren trazabilidad: Este tipo de sistemas se caracteriza por poseer un conjunto de procedimientos preestablecidos y autosuficientes que permiten conocer el historial, la ubicación y la trayectoria de un producto, o lote de productos, a lo largo de la cadena de suministros en un momento dado. Entre ellos encontramos: *a) Distribución y logística:* blockchain permite establecer una red segura entre todas las partes implicadas, mediante esta se podría validar la autenticidad de cualquier documento electrónico al instante sin necesidad de la intervención de un tercero prescindiendo del papel. Desde un mismo canal se puede gestionar toda la documentación y los pagos minimizando los posibles errores de comunicación, reduciendo los trámites y aumentando la seguridad a todos los niveles. *b) Cadena de suministro:* blockchain permite hacer un seguimiento digital de los productos en cada fase de la cadena de valor, desde los proveedores a los consumidores, almacenando la información del producto, como el lugar de origen, número de lote, fecha de caducidad, temperatura de almacenamiento, etc. con la ventaja de que la información ha sido consensuada por todos los integrantes de la red. *c) Licitaciones:* es un proceso por el que se adjudica la realización de una obra o un servicio a la persona o empresa que ofrece las mejores condiciones. Se puede generar una licitación almacenando la documentación paso a paso dentro de la blockchain. También es posible automatizar los procesos de licitación a través de la generación de contratos inteligentes para la

evaluación de las propuestas. *d) Certificaciones de origen:* documento formal cuya función es determinar el origen en el cual ha sido fabricada la mercancía, bajo los términos establecidos por el acuerdo comercial que corresponda al destino. Mediante la aplicación de blockchain se asegura de manera unívoca la no alteración de los datos o falsificación del certificado de origen.

Financieras: Este campo se refiere al conjunto de actividades que tienen relación con los bienes que dispone una persona, especialmente en forma de dinero. A partir de la generación de las *Criptomonedas* como un tipo de divisa alternativa se generaron diversas aplicaciones para el intercambio de bienes o la obtención de estos. Podemos mencionar *plataformas de micropagos*, especialmente destinados a transacciones transfronterizas o intercambios de bienes y *pólizas o seguro de caución*, donde el asegurador se obliga a indemnizar al asegurado por los perjuicios que sufra en caso de que este incumpla sus obligaciones.

IoT - Internet de las Cosas: Es una red de dispositivos electrónicos interrelacionados, que tienen identificadores únicos y poseen la capacidad de transferir datos a través de una red, como por ejemplo, personas con implantes electrónicos que requieren ser monitoreados, animales de granja con un transpondedor de biochip, automóviles con sensores incorporados para alertar al conductor cuando la presión de los neumáticos es baja, o cualquier otro objeto al que se puede asignar una dirección IP y darle la capacidad de transferir datos a través de una red. Blockchain puede mejorar las redes de sensores inalámbricos distribuidos, minimizando sus deficiencias y maximizando su potencial. Proporciona un intercambio de datos seguro y auditable en escenarios heterogéneos.

3 Blockchain en áreas de gobierno

3.1 Áreas de gobierno

Existen diversas formas de clasificar las áreas de gobierno, la más relevante es la orientada al área de incumbencia y a las necesidades de los ciudadanos que atienden. Por ejemplo, un gobierno puede ser dividido en: *Seguridad, Producción y economía, Transporte, Relaciones interiores y exteriores, Salud y Educación, ciencia y tecnología.*

Seguridad: Tiene injerencia en la seguridad interior y exterior, la preservación de la libertad, la vida y el patrimonio de los habitantes, sus derechos y garantías en un marco de plena vigencia de las instituciones del sistema democrático.

Producción y economía: Encargada de diseñar y ejecutar los planes relacionados con el fomento de la producción industrial y el comercio exterior, la creación de empleo de calidad, la promoción de una inserción internacional a escala, la defensa de la competencia y la federalización de la producción. A su vez, interviene en todo lo inherente al financiamiento del Sector Público Nacional.

Transporte: Se encarga de la gestión y coordinación de todo lo inherente al transporte aéreo, ferroviario, automotor, fluvial y marítimo, y a la actividad vial. Se encarga de diseñar y ejecutar políticas para el desarrollo del transporte en el país y controla las regulaciones y concesiones en transporte, imponiendo sanciones a quienes las incumplan.

Relaciones Interiores y Exteriores: Encargada de todo lo relacionado al gobierno político interno y al ejercicio pleno de los principios y garantías constitucionales, asegurando y preservando el régimen republicano, representativo y federal, y en relación con la política de obras públicas, vivienda y hábitat. Esta área se encarga, además, de las relaciones exteriores de la Nación y su representación ante los gobiernos extranjeros.

Salud: Se encarga de atender las cuestiones administrativas relacionadas con el servicio de salud, entre ellas las cuestiones de epidemiología, campañas de vacunación, control sanitario de las fronteras, registro de los profesionales de la salud y el banco de drogas, entre otras.

Educación, Ciencia y Tecnología: Se encarga de implementar, diseñar y poner en práctica políticas para los distintos niveles educativos, además de gestionarlos. Además, estimula la labor en ciencia y tecnología, y difunde sus resultados para instalar el valor socioeconómico del conocimiento. Asimismo, impulsa la innovación de base tecnológica y la incorporación en la cultura productiva de las empresas argentinas. Su fin es contribuir, a partir de la ciencia, la tecnología y la innovación productiva, al desarrollo económico, social y cultural de la Nación.

3.2 Beneficios en las áreas de gobierno utilizando blockchain.

Relacionando las características de blockchain vistas en 2.2 y las necesidades de las áreas de gobierno planteadas en 3.1, la tabla 2 muestra las posibles aplicaciones que podrían explotar los beneficios de blockchain en un gobierno.

Table 2. Aplicación de blockchain en áreas de gobierno

		Áreas de gobierno					
		Seguridad	Producción y economía	Transporte	Relaciones Interiores y Exteriores	Salud	Educación, Ciencia y Tecnología
Aplicaciones de Blockchain	Registros Públicos	URP Cred. Armas de Fuego Publicación de Altura de Ríos	Registro Pyme o industria	Lic. conducción CNRT	Visas Pasaportes Edición digital del Boletín Oficial de la República Argentina	Reg. personas Actas de nacimiento	Patentes Prop. intelectual
		RV Antec. penales Reg. sumarial	Deducciones Impuesto a las Ganancias Pagares Digitales	Reg. multas	Pasaportes	Hist. clínica Prescripciones	Análisis de las materias, Actas Universitarias, Libros Digitales SAS
		ID Hist. Penal Tit. Armas	Monotributo, resp. Inscripto, S.A, S.R.L Pólizas Digitales,	Lic. Conducir Tit. Automotor	Visas, Pasaportes	Part. Nacimiento, Hist. clínica. Discapacidad	Títulos, Patentes Prop. Intelectual

Trazabilidad	DyL	Equipamiento, Cadena de custodia de evidencia	Productos	Sistemas de seguimiento de los medios de transporte	Aduanas, intercambio de información entre naciones	Equipamiento	Equipamiento
	CS	Cadena de custodia de evidencia	Etapas en la generación de productos	–	Gestión de migraciones	Administraciones de medicamentos	–
	L	Armas y equipamiento	Insumos varios	Insumos varios	Insumos varios	Equipos médicos	Insumos varios
	CO	–	Productos	Control de Cargas	Docs.	Drogas	–
Financieras	C	–	Pago productos	Sistema de Boleto Electronico	Intercambio de bienes entre naciones, inversiones	Tokenización y paga de la información médica o cobros generales	Utilización para financiamiento de investigaciones, Tokenización y paga de la información aportada para investigaciones
IoT	IoT	Cameras de Vigilancia Tobillera seguimiento	Luminarias	Peajes Camaras de multas	Pasaportes con chips	Información de sensores para el seguimiento de pacientes	–

URP: Utilización en registros públicos – RV: Registro/Documento vivo – ID: Identidad digital soberana – DyL: Distribución y logística – CS: Cadena de suministro – L: Licitaciones – CO: Certificaciones de origen – PC: Pólizas o seguro de caución – C: Criptomoneda – IoT: Internet of Things.

4 Blockchain en Argentina

Desde que la primera blockchain fue conceptualizada en 2008 y aplicada en 2009 como componente central de Bitcoin [9], la tecnología se masificó, al punto de llegar a explotar un polo de blockchain en nuestra región. En el contexto de Argentina existen varias empresas que se encargan de diversas aplicaciones sobre esta tecnología, entre las que se pueden mencionar las que realizan aplicaciones con fines económicos (Fintech) como *Ripio*, *Bitex* y *Wayniloans*, y las que generan, además, aplicaciones y/o servicios con fines no económicos, como por ejemplo garantizar la autoría, integridad y fecha de cualquier registro o documento, como contratos, workflows, certificaciones o diplomas académicos, propiedad intelectual, entre otras, como *Signatura*, *RSK* y/o *Coinfabrik*, *Agree Market* y *Everis Argentina*.

En las esferas gubernamentales se generó interés en el área con el fin de visualizar la apertura de los procesos y la transparencia a través de la verificación de la autenticidad y la incorruptibilidad de los datos. Como consecuencia de esto, a partir de mediados de 2017, las ediciones electrónicas del Boletín Oficial de la Nación se certifican mediante la utilización de la blockchain. También existen datos certificados dentro de la Municipalidad de Córdoba, entre otras.

En abril del corriente año, se anunció una colaboración conjunta entre el Instituto de Investigación Blockchain (BRI) y el Gobierno de Argentina para avanzar en la implementación de la tecnología en más áreas de gobierno y en diversos proyectos que se están llevando a cabo en el país. Dentro de estos, están las plataformas de compras y contrataciones del Estado Nacional: Compr.ar y Contrat.ar, y el programa de Trámites a Distancia (TAD).

En los últimos años surgió la iniciativa *Blockchain Federal Argentina (BFA)*, donde confluyen distintas partes de la sociedad, como individuos, organismos, instituciones y

empresas de cualquier sector, interesados en desplegar aplicaciones para generar aportes y avanzar en la implementación de blockchain. Se trata de una plataforma multiservicios abierta y participativa pensada para integrar servicios y aplicaciones de blockchain.

BFA es una iniciativa confiable y completamente auditable que permite optimizar procesos y funciona como herramienta de empoderamiento para toda la comunidad. Cuenta con aportes de sectores públicos, privados, académicos y de la sociedad civil. La participación de toda la comunidad es esencial, mantiene un modelo de gobernanza que asegura la representación de todos los sectores en la toma de decisiones. Al ser una plataforma pública, su uso no está restringido a las organizaciones que participan del consorcio, sino que toda la comunidad tiene las puertas abiertas para participar.

Entre las implantaciones que presenta BFA se encuentran: Sello de Tiempo (BFA), Smart Voting (BFA), Firma Conjunta de Documentos (GDE), Libros Digitales SAS (Secretaría de Modernización), Carpeta Ciudadana (CABA), Portadocumentos Digital (RedLink), Actas Universitarias (Universidad Nacional de Córdoba), Edición digital del Boletín Oficial de la República Argentina (Secretaría Legal y Técnica), Pagarés Digitales (Garbarino), Títulos Universitarios (SIU), Monitores de Ethereum (Última Milla, C&S), entre otros.

5 Estado de Blockchain en aplicaciones de gobierno

Para el relevamiento se tomó como base las aplicaciones propuestas por BFA a nivel Nacional y las aplicaciones propuestas por el portal “Blockchain Service” [10] donde se registran más de 150 casos (de éxito y de estudio) de utilización de blockchain en diferentes ámbitos. Del conjunto se seleccionaron las aplicaciones relacionadas con las áreas de gobierno planteadas anteriormente, obteniendo 70 casos (en uso y en fase de prueba). En 44 de los casos, las aplicaciones relevadas aplicaban a más de un área, por lo que en total para el análisis de consideraron 114 casos. La tabla 3 presenta los porcentajes en función de las aplicaciones relevadas para cada una de las áreas de gobierno propuesta.

Analizando los porcentajes obtenidos, lo primero que se destaca es el alto porcentaje en la subclase *Criptomonedas*, dentro de la clase *Financiera*, lo cual resulta lógico ya que la propia tecnología blockchain se originó a partir de la creación de estas.

En cuanto a las áreas de gobierno, la más avanzada es la de *Producción y economía* influida fuertemente por el avance de la tecnología, seguida por la de *Transporte* por la relación directa que tiene con la utilización de aplicaciones de criptomonedas y por último el área de *Seguridad* que solo presenta algunas aplicaciones en registros públicos y licitaciones.

Si observamos los valores obtenidos en base a la clasificación generada por la clase de aplicación, se ve un mayor avance en las aplicaciones generadas para su uso en los *Registros Públicos*, principalmente por la subclase *URP (Utilización de Registros Públicos)*. Los valores encontrados en esta subclase representan el interés del gobierno en todas sus áreas en generar confianza aprovechando los beneficios de las características en la utilización de blockchain. Es seguida por la clase *Financiera*, por

la subclase *C* (*Criptomonedas*) y, por último, por la clase IoT, la cual se encuentra en incipiente desarrollo.

Table 3. Valores del estado de Blockchain en aplicaciones de gobierno

		Áreas de gobierno						Total Individual	Total por área
		Seguridad	Producción y economía	Transporte	Relaciones Interiores y Exteriores	Salud	Educación, Ciencia y Tecnología		
Aplicaciones de Blockchain	Registros Públicos	URP	3,51%	3,51%	2,63%	5,26%	2,63%	2,63%	20,18%
		RV	0,00%	0,88%	0,88%	0,88%	1,75%	2,63%	7,02%
		ID	0,00%	2,63%	0,88%	0,00%	1,75%	4,39%	9,65%
	Trazabilidad	DyL	0,00%	2,63%	3,51%	1,75%	0,88%	0,88%	9,65%
		CS	0,00%	4,39%	2,63%	0,88%	0,88%	0,88%	9,65%
		L	0,88%	0,88%	0,88%	0,88%	0,88%	0,88%	5,26%
		CO	0,00%	0,88%	0,88%	0,00%	0,00%	0,00%	1,75%
	Financieras	PC	0,00%	0,88%	0,88%	0,00%	0,00%	0,00%	1,75%
		C	0,00%	11,40%	7,02%	4,39%	0,00%	6,14%	28,95%
	IoT	IoT	0,00%	2,63%	1,75%	0,00%	1,75%	0,00%	6,14%
Total		4,39%	30,70%	21,93%	14,04%	10,53%	18,42%	-	-

URP: Utilización en registros públicos – RV: Registro/Documento vivo – ID: Identidad digital soberana – DyL: Distribución y logística – CS: Cadena de suministro – L: Licitaciones – CO: Certificaciones de origen – PC: Pólizas o seguro de caución –C: Criptomoneda – IoT: Internet of Things.

6 Conclusión

A lo largo del trabajo se introdujeron los conceptos básicos de blockchain con sus principales características, las cuales aportan un conjunto de importantes ventajas como la seguridad, confiabilidad y transparencia de los datos. En función de esas características se analizó qué tipo de aplicaciones se verían beneficiadas y se las clasificó en: Registros públicos, Trazabilidad, Financieras e IoT. Luego, se relacionó el uso de blockchain con las diferentes áreas de un gobierno. Para esto se propuso una estructura de gobierno de seis áreas y se identificaron las posibles aplicaciones que puedan generar beneficios en las mismas. Se presentó la situación de blockchain en Argentina, en particular la propuesta que presenta la BFA y, por último, fueron relevados 70 casos de uso de blockchain relacionados con áreas de gobierno, concluyendo que la más avanzada es el área de producción y economía debido al crecimiento de las criptomonedas, y la menos avanzada es el área de seguridad. Por el lado de las características el porcentaje más alto lo obtuvieron las relacionadas con la utilización de registro público, seguidas por las financieras.

7 Bibliografía

- 1- PREUKSCHAT, A. et al. (2017). Blockchain: la revolución industrial de internet. Gestión 2000.
- 2- Buterin, V., (2015). On public and private blockchains, Ethereum Blog 7.
- 3- Tapscott, D., Tapscott, A., (2017). How blockchain will change organizations. MIT Sloan Manage. Rev. 58 (2), 10–13.
- 4- Secretaría de Gobierno de Modernización. Disponible en: <https://www.argentina.gob.ar/modernizacion>. Visitado: 5/07/2019.
- 5- Blockchain Federal Argentina. Disponible en: <https://www.bfa.ar/>. Visitado: 5/07/2019.
- 6- Szabo, N., (1994). Smart contracts.
- 7- Casino, F. et al. (2019). A systematic literature review of blockchain-based applications: Current status, classification and open issues.
- 8- Zheng, Z. et al. (2016). Blockchain challenges and opportunities: a survey. Work Pap.
- 9- Nakamoto, Satoshi. "Bitcoin: A peer-to-peer electronic cash system." (2008).
- 10- Blockchain Services. Disponible en: <http://www.blockchainservices.es/category/casos-exito-blockchain/>. Visitado: 5/07/2019.

Desarrollo de Ciudades Inteligentes Sostenibles – Un Caso de Estudio Implementado por el Municipio de Vicente López

Sonia Rueda, Elsa Estevez, Pablo Fillottrani
Departamento de Ciencias e Ingeniería de la Computación
Universidad Nacional del Sur, Bahía Blanca, Argentina
{svr, ece, prf}@cs.uns.edu.ar

Resumen. Este trabajo presenta un caso de estudio de una iniciativa de ciudad inteligente desarrollada por el Municipio de Vicente López. La documentación del caso de estudio se realizó en el marco del Proyecto CAP4CITY que busca desarrollar capacidades de gobernanza para ciudades inteligentes sostenibles en América Latina. Como marco conceptual para la investigación se presentan definiciones de ciudad inteligente y las dimensiones a analizar. Se resumen las características más importantes del proyecto y se explican detalles del caso de estudio. La contribución de este trabajo se focaliza en la documentación de un caso de estudio relacionado a una iniciativa de ciudad inteligente en Argentina y en resumir algunas lecciones aprendidas del caso analizado.

Palabras clave: Ciudades Inteligentes Sostenibles; Gobernanza Digital, Construcción de Capacidades Humanas; Programas de Educación

1 Introducción

El proceso de crecimiento urbano, en detrimento del desarrollo de las zonas rurales, es una tendencia global que está provocando transformaciones profundas en las ciudades. En la actualidad, de acuerdo a información publicada por el Banco Mundial, el 55% de la población mundial es urbana [1]. En 2050, este indicador será del 68% [2].

Esta realidad impone a las ciudades, en particular a las grandes urbes, la necesidad de planificar e implementar procesos de adaptación que garanticen la sostenibilidad [1]. Las tecnologías de la información y las comunicaciones (TIC) adquieren en este contexto un rol relevante, hasta el punto de inducir un nuevo concepto de ciudad. Bajo el término, Ciudad Inteligente se constituye un paradigma urbano hacia el que converge la transformación de las ciudades actuales [3].

Las TIC permiten mejorar la eficiencia y la eficacia en los procesos y servicios urbanos, pero son las personas, los gobernantes con los ciudadanos, quienes deben desarrollar estrategias y planes para responder a los desafíos que derivan del crecimiento de la población, los efectos del cambio climático y el agotamiento de los recursos naturales.

Una ciudad inteligente se transforma en forma continua para mejorar la calidad de vida de los ciudadanos, de manera eficiente y sostenible en base a la innovación. Así, más

que la tecnología digital, es la innovación aplicada a la gestión de la ciudad, a sus procesos y a su infraestructura, la que caracteriza a una ciudad inteligente. En particular, los proyectos de transformación de una ciudad en base a un modelo inteligente se orientan fundamentalmente a mejorar la eficiencia energética y del consumo de recursos naturales, mejorar la movilidad, fomentar el desarrollo de actividades económicas “verdes”, mejorar la calidad de vida de los residentes, y desplegar mecanismos para la participación ciudadana y la gobernanza.

Este trabajo presenta un caso de estudio de una iniciativa de ciudad inteligente desarrollada por el Municipio de Vicente López. La iniciativa fue documentada para el Proyecto CAP4CITY (ver Sección 3). A fin de documentar el caso, se buscó información disponible en Internet en el mes de marzo de 2019. Las iniciativas implementadas por el Municipio de Vicente López fueron identificadas a partir de búsquedas no estructuradas realizadas en el dominio público durante el mismo mes, cumpliendo con actividades previstas del Proyecto CAP4CITY. La contribución de este trabajo se focaliza en documentar un caso de estudio de una iniciativa de ciudad inteligente y en resumir algunas lecciones aprendidas del caso analizado.

El resto de este documento se estructura de la siguiente forma. La Sección 2 explica el concepto de ciudades inteligentes y sus dimensiones. La Sección 3 introduce brevemente el proyecto de investigación y desarrollo que enmarca este trabajo, mientras que la Sección 4 resume el enfoque seguido por el Gobierno Nacional en Argentina para estas iniciativas. La Sección 5 explica el caso de estudio de Vicente López y la Sección 6, lo discute. Por último, la Sección 7 resume conclusiones y el trabajo futuro.

2 Ciudades Inteligentes y sus Dimensiones

Como se explicó en la sección previa, debido a la relevancia del desarrollo urbano y a las posibilidades de utilizar las TIC como herramienta estratégica para alcanzar objetivos de desarrollo, como el Objetivo de Desarrollo Sostenible #11 de Naciones Unidas – “Lograr que las ciudades y los asentamientos humanos sean inclusivos, seguros, resilientes y sostenibles” [4], el tema de ciudades inteligentes ha despertado mucho interés en los últimos años, tanto de los académicos, como gobiernos e industria.

Trabajos científicos han aportado varias definiciones, algunas se muestran en la Tabla 1. De todas, consideramos la más completa u holística a la primera, la provista por Caragliu et.at [5]. Esta definición es la única, a nuestro entender, que habla de las tres dimensiones más importantes del desarrollo sostenible; a saber, el desarrollo económico, la inclusión social y la protección del medio ambiente; y menciona un aspecto que es clave para poder administrar los conflictos de intereses entre los diferentes actores que contribuyen al desarrollo de ciudades inteligentes, esto es, la gobernanza participativa.

Un estudio realizado para la Unión Europea [6], reconoce seis dimensiones para el desarrollo de una ciudad inteligente: 1) Economía Inteligente, 2) Movilidad Inteligente,

3) Medio Ambiente Inteligente, 4) Personas Inteligentes, 5) Condiciones de Vida Inteligente, y 6) Gobernanza Inteligente. De acuerdo dicho modelo, la economía inteligente puede medirse en términos de espíritu innovador, emprendedurismo, imagen económica y marcas, productividad, flexibilidad del mercado laboral, internacionalización y habilidad para la transformación. La movilidad inteligente en función de accesibilidad local, accesibilidad nacional e internacional, disponibilidad de infraestructura TIC, y sistemas de transportes seguros, innovadores y sostenibles. El Medio Ambiente Inteligente puede medirse con los siguientes indicadores, atractivo de las condiciones naturales, polución, protección ambiental, y administración de recursos sostenibles. La dimensión de Personas Inteligentes puede evaluarse en función de nivel de calificación, afinidad con la educación continua, pluralidad social y étnica, flexibilidad, creatividad, apertura de criterios y visión cosmopolita, y participación en asuntos públicos; mientras que para la dimensión de Condiciones de Vida Inteligente se definen los indicadores de facilidades culturales, condiciones de salud, seguridad individual, calidad de viviendas, facilidades de educación, atractivos turísticos, y cohesión social. Por último, la dimensión de gobernanza se mide con el nivel de participación en toma de decisiones, servicios sociales y públicos, gobernanza transparente y estrategias y perspectivas políticas.

Tabla 1. Algunas Definiciones de Ciudad Inteligente

NO.	DEFINICIÓN	REFERENCIA
1	Las ciudades inteligentes se refieren a las inversiones en capital humano y social y en infraestructura de comunicaciones tradicional (transporte) y moderna (TIC) para impulsar el crecimiento económico sostenible y una alta calidad de vida, con un manejo racional de los recursos naturales, a través de la gobernanza participativa.	[5]
2	Una fusión de ideas sobre cómo las TIC podría mejorar el funcionamiento de las ciudades, su eficiencia, su competitividad, y proporcionar nuevas formas en que los problemas de la pobreza, la privación social y el medio ambiente pobre podrían abordarse.	[7]
3	Entorno urbano que, con el apoyo de sistemas de TIC generalizados, es capaz de ofrecer servicios avanzados e innovadores a los ciudadanos para mejorar la calidad general de su vida	[8]
4	Una alta capacidad de aprendizaje e innovación, que está integrada en la creatividad de su población, sus instituciones de creación de conocimiento, y su infraestructura digital para la comunicación y la gestión del conocimiento "	[9]

3 Proyecto CAP4CITY

El Proyecto CAP4CITY (<https://www.cap4city.eu/>), parte del Programa Erasmus+ sobre construcción de capacidades financiado por la Unión Europea, está dedicado a fortalecer las capacidades de gobernanza para el desarrollo de ciudades inteligentes sostenibles en América Latina, a través del diseño de curricular de cursos de grado y posgrado, un programa de Especialización y Maestría, y cursos en línea gratuitos y masivos (Massive Online Open Courses, MOOCs) y con la innovación en los modelos pedagógicos. El Proyecto es ejecutado por un consorcio de 12 universidades, cuatro europeas, y ocho latinoamericanas, dos de Argentina, de Brasil, de Chile y de Colombia. Las universidades argentinas integrantes del consorcio son la Universidad Nacional de La Plata y la Universidad Nacional del Sur.

El Proyecto tiene una duración de tres años y la ejecución se divide en nueve paquetes de trabajo: 1) Estudio del Arte – investigación sobre programas de estudio en la temática y relevamiento de iniciativas de ciudades inteligentes sostenibles, 2) Desarrollo de los módulos de entrenamiento, 3) Desarrollo de MOOCs, 4) Capacitar a los Capacitadores, 5) Implementar cursos – cursos de grado y electivos, 6) Implementar cursos de posgrado, 7) Administración del Proyecto, 8) Gestión de la Calidad, y 9) Diseminación.

4 Ciudades Inteligentes y Sostenibles en Argentina

La Secretaría de Modernización de la Nación Argentina promueve el desarrollo de ciudades inteligentes y sostenibles en el marco de su paradigma de País Digital [6]. La intención es definir un modelo inclusivo, innovador, eficiente y transparente que sirva de horizonte para los municipios de todo el territorio argentino. El modelo ofrece una visión global y aplica las TIC para:

- Planificar el tendido de la infraestructura, su ordenamiento territorial y financiamiento.
- Diseñar una estrategia integral para brindar y comunicar servicios.
- Dinamizar la economía, facilitando el acceso al capital humano y a otros insumos y productos.
- Preservar el medio ambiente.
- Promover la participación ciudadana.
- Resolver los problemas de forma innovadora, transformando las dificultades en oportunidades.
- Anticipar y superar las expectativas de sus habitantes y usuarios.

En este contexto, la Secretaría de Modernización ofrece un programa de capacitación para Municipios cuya realización es requisito para tener acceso al Premio Nacional a la Calidad Municipal. En diciembre de 2018, eran 113 las ciudades que habían culminado el proceso de evaluación que permite acceder a este premio. El Municipio de Vicente López fue uno de los ganadores del premio a partir de sus acciones orientadas a mejorar

la calidad de los servicios, promover la transparencia y la inclusión digital de los ciudadanos.

En el marco del segundo paquete de actividades del proyecto CAP4CITY - Fortalecimiento de Capacidades de Gobernanza para Ciudades Inteligentes Sostenibles, propusimos al Municipio de Vicente López, entre otros, para relevar iniciativas, experiencias y acciones concretas que utilicen tecnología para mejorar la calidad de vida de los ciudadanos de manera sostenible, promover la equidad de oportunidades y construir ciudadanía. La sección siguiente describe una de sus iniciativas.

5 La Experiencia de Vicente López

Vicente López forma parte del aglomerado urbano del Gran Buenos Aires. Es un municipio urbano en el cual todo el partido es una ciudad, formada por nueve barrios. La Municipalidad se ubica en el barrio de Olivos. En el censo de 2010 se registraron 269.420 habitantes. Se ubica a 9 km del centro de la ciudad de Buenos Aires. Cuenta con tres centros hospitalarios y 20 unidades de atención para la salud. Está conectada con la ciudad de Buenos Aires por el sistema Metrobus y además, por dos avenidas, una autopista y cinco ramales de ferrocarril.

En el marco del proyecto CAP4CITY se relevó la implementación de tres proyectos: 1) Fortalecimiento del Sistema de Monitoreo Urbano, 2) Seguimiento de los ODS, y 3) Plataforma de Datos Abiertos. Para cada uno de los proyectos, siguiendo el patrón de análisis propuesto en la actividad de relevamiento del estado del arte del Proyecto CAP4CITY, se plantea estudiar, el qué, el cómo, el por qué, y quienes diseñan e implementan las iniciativas. A continuación presentamos los detalles relevados de la primera iniciativa.

5.1 Sistema de Monitoreo Urbano

En el marco del Plan Integral de Seguridad del Municipio de Vicente López, desde 2015 se fortalece en forma continua el Sistema de Monitoreo Urbano destinado a prevenir y predecir accidentes de tránsito, delitos, derrames de residuos peligrosos en la vía pública y otros siniestros.

La evaluación y mejora continua del sistema permite:

- Identificar oportunidades de mejora del sistema de monitoreo para mejorar la prevención de accidentes y delitos
- Visibilizar las características y funcionalidades del sistema en el marco de la política de gobierno abierto
- Innovar de forma permanente en la apropiación, uso y difusión de las TIC como herramienta para mejorar la calidad de vida de los ciudadanos

Probablemente el mayor desafío es sostener la inversión que requieren mantener el equipamiento actualizado y el personal capacitado. Otro aspecto esencial es garantizar el cumplimiento del protocolo de actuación del Centro asegurando la privacidad de la información monitoreada y almacenada a través de imágenes digitales.

5.2 ¿Qué?

El Centro de Monitoreo Urbano es un elemento esencial del sistema. El plan de mejora incluye además:

- capacitación del personal del Centro de Monitoreo Urbano, en particular para la atención telefónica de emergencias;
- incorporación de equipamiento, en especial el incremento de cámaras instalación de espacios públicos con zonas wifi y “puntos seguros” formados por totén con un botón de alarma y una cámara
- instalación de fibra óptica para mejorar la conectividad y permitir la visualización en tiempo real
- incorporación de software para procesamiento de imágenes
- archivo digital de imágenes encriptadas

El Centro de Operación de Red (NOC) monitorea el estado de salud de toda la red municipal que está montada sobre fibra óptica. Se encuentra funcionando en el Centro universitario de Vicente López y brinda conectividad a más de mil cámaras de monitoreo urbano y más de 80 dependencias municipales. Desde NOC se identifican alertas, fallas en áreas o equipos, se realizan estadísticas de tráfico de red en tiempo real, se calcula el consumo del servicio wifi, El funcionamiento del NOC es monitoreado constantemente por cámaras, asegurando la calidad de los recursos y servicios provistos a los sistemas municipales.

5.3 ¿Cómo?

La Sala de Control cuenta con un video wall central donde los operadores realizan la visualización de las imágenes captadas por las cámaras ubicadas en puntos estratégicos y es la base operativa de los móviles, la central de alerta de botones de pánico y la central de atención de los números municipales para denuncias y emergencias de hechos de inseguridad.

Los vecinos pueden comunicarse con el Centro de Monitoreo Urbano para reportar robos, accidentes u otros siniestros. El sistema se completa con otras iniciativas mediadas por tecnología como chalecos inteligentes con GPS, botones antipánico y líneas telefónicas para reportar y atender emergencias las 24 h todos los días del año.

El personal técnico del Centro de Monitoreo Urbano concentra todas las comunicaciones relacionadas con emergencias y delitos, clasifica la información que ingresa por la línea gratuita o captan de las cámaras, y deriva hacia las áreas competentes para dar una respuesta inmediata. El operador identifica quién llama,

desde dónde lo hace y clasifica el grado de la emergencia. El coordinador general tiene a su cargo la logística de trabajo durante una crisis urbana. El coordinador operativo a cargo del turno posee terminales que permiten acceder a los datos, con el objetivo de enviar personal a los distintos eventos observados en la central. Por ejemplo destinar policías ante la comisión de un delito; dar aviso al área de emergencias, ya sea por árboles caídos, cables cortados o anegamiento de calles y a los servicios de salud por personas accidentadas

Para poder ver las imágenes en tiempo real se utiliza una red de fibra óptica de última generación que se extiende por distintas zonas del Municipio. El parque de cámaras está compuesto por cámaras fijas (permanecen siempre apuntadas a un lugar estratégico y no pueden moverse en forma remota), y las cámaras domos, que pueden ser manejadas desde el Centro con un radio de giro de 360 grados y acercamiento de varios cientos de metros. Todo lo que las cámaras captan es grabado en una sala de control y almacenamiento de última generación. Funciona en una sala anexa que aloja de modo restringido el sistema de archivo digital de imágenes protegido con doble encriptación.

5.4 ¿Por Qué?

El Sistema de Monitoreo Urbano tiene objetivos económicos, sociales y de gobernanza. Aspira y de hecho ha reducido el número de accidentes y contribuyendo a justificar el costo que provoca su atención. Ha permitido desarticular bandas delictivas y resolver casos de abuso y acoso como así también accidentes de tránsito. La expectativa es consolidar un sistema que mejore la seguridad y la convivencia urbana y pueda ser utilizado como modelo por otros municipios.

5.5 ¿Quiénes?

La seguridad es uno de los principales reclamos que los ciudadanos manifiestan a los órganos de gobierno, de modo que aunque el fortalecimiento del Sistema de Monitoreo Urbano es liderado por la Secretaría de Modernización y Gobierno Digital dentro del ámbito de la Municipalidad de Vicente López, de alguna manera son los ciudadanos quienes han impulsado esta iniciativa. La Secretaría de Seguridad cuenta con una plataforma de Big Data que permite evaluar el impacto que del sistema de monitoreo y planificar su crecimiento.

Los operarios del Centro de Monitoreo Urbano están especializados para detectar y recibir avisos de emergencia y cubren todas las posiciones y turnos para cumplir con el servicio de 24 horas los 365 días del año. Las acciones que derivan de la atención de una emergencia se coordinan con organismos externos, como Policía Federal, Prefectura Naval, Policía Aeroportuaria e incluso con otros de la provincia de Buenos Aires.

6 Discusión

El desarrollo de ciudades inteligentes presenta importantes desafíos. Usualmente, los más importantes para los países en vías de desarrollo, como son los países de la región de América Latina, es la falta de capacidades, desde humanas – recursos humanos calificados para planificar, dirigir, desarrollar y operar soluciones para ciudades inteligentes; pasando por capacidades organizacionales – herramientas para la dirección y estandarización de procesos, manuales de procedimientos, y estándares de calidad, entre otras; y capacidades institucionales – acuerdos de colaboración con actores públicos y no públicos, plataformas de participación ciudadana, y marcos regulatorios y legales, por mencionar algunos [3].

Del caso de estudio de la ciudad de Vicente López podemos resumir algunas lecciones aprendidas: 1) la necesidad y la relevancia de la participación ciudadana para el diseño de la solución más eficiente para el problema local; 2) la necesidad de fortalecer las capacidades de gobernanza para diseñar y utilizar el sistema, incluyendo la adopción de enfoques para la gobernanza participativa; 3) la relevancia del problema resuelto, como es el monitoreo y la detección y prevención de delitos en un contexto urbano; 4) la repetición del problema para otros entornos urbanos en Argentina; y 5) la posibilidad de contar con una solución que pueda replicarse y adoptarse en otros municipios. Esto último resalta la importancia de documentar los casos de estudio existentes en el país y crear un repositorio de acceso abierto, que pueda ser accedido por otros municipios interesados en implementar soluciones, que representan buenas prácticas implementadas por otros municipios para problemas comunes.

7 Conclusiones y Trabajo Futuro

Este trabajo presenta un caso de estudio de una iniciativa relacionada al desarrollo de ciudades inteligentes en Argentina. El objetivo del mismo es poder analizar los diferentes esfuerzos que se están realizando en esta área y estudiar desde el mundo académico cómo contribuir con la formación de los recursos humanos. El estudio se realizó en el marco de un Proyecto CAP4CITY, ejecutado con el propósito de desarrollar un diseño curricular e innovar en los modelos pedagógicos para desarrollar capacidades de gobernanza para ciudades inteligentes sostenibles.

Nuestro trabajo futuro consiste en documentar diferentes casos de estudio para poder evaluar el estado de la práctica en ciudades inteligentes sostenibles. En función de esta evaluación, posteriormente definir competencias necesarias para fortalecer las capacidades de gobernanza para estos modelos de ciudades.

Agradecimientos

Los autores agradecen a las autoridades de la Municipalidad de Vicente López, Argentina, por la colaboración en la documentación del caso de estudio. Esta

publicación fue realizada en el contexto del Proyecto CAP4CITY – “Strengthening Governance Capacity for Smart Sustainable Cities” (www.cap4city.eu) co-financiado por el Programa Erasmus+ de la Unión Europea. Acuerdo Número 598273-EPP-1-2018-1-AT-EPPKA2-CBHE-JP. Número de proyecto: 598273



Referencias

- [1] The World Bank, “Urban Population,” *Urban Population Growth*, 2019. [Online]. Available: <https://data.worldbank.org/indicator/sp.urb.totl.in.zs>. [Accessed: 02-Aug-2019].
- [2] United Nations Department of Economic and Social Affairs (UNDESA), *World Urbanization Prospects 2018*. 2018.
- [3] E. Estevez, N. Vasco Lopes, and T. Janowski, *Smart Sustainable Cities Reconnaissance Study*. 2016.
- [4] Naciones Unidas, “Objetivo de Desarrollo Sostenible,” 2015. [Online]. Available: <https://www.un.org/sustainabledevelopment/es/>. [Accessed: 01-Aug-2019].
- [5] A. Caragliu, C. Del Bo, and P. Nijkamp., “Smart Cities in Europe,” *J. Urban Technol.*, vol. 18, no. 2, pp. 65-82+, 2011.
- [6] R. Giffinger, C. Fertner, H. Kramar, N. Pichler-Milanovic, and E. Meijers, “Smart City Model,” 2010.
- [7] M. Batty and et.al., “Smart Cities of the Future,” *Eur. Phys. J. Spec. Top.*, vol. 214, no. 1, pp. 481–518, 2015.
- [8] G. I. Piro, L. A. Cianci, G. Grieco, Boggia, and P. Camarda, “Information Centric Services in Smart Cities,” *J. Syst. Softw.*, vol. 88, no. 1, pp. 169–188.
- [9] E. Tranos and D. Gertner, “Smart Networked Cities?,” *Innov. Eur. J. Soc. Sci. Res.*, vol. 25, no. 2, pp. 175–190, 2015.

Evaluando la Calidad de Uso en aplicaciones web de gobierno electrónico y ciencia ciudadana con estándares ISO/IEC

Marcelo Castro¹, Víctor Sánchez Rivero¹, Luis Vargas¹,
José Farfán, Elizabeth Reinoso¹, María Aparicio¹,
Fabiana Aragón¹, José Zapana¹, Andrea Cándido², Liliana Cazón³
Ingeniería + Software (I+S)
¹Facultad de Ingeniería- Universidad Nacional de Jujuy
{mcastro.fi.unju, ivansrivero, fabiana.aragon.unju, jose.zapana.ar}@gmail.com
{jhfafan, eliro20}@hotmail.com
{alevar98, mariaparicio}@yahoo.com
²Facultad de Humanidades- Universidad Nacional de Jujuy
agcandido@fhyics.unju.edu.ar
³Facultad de Ciencias Económicas- Universidad Nacional de Jujuy
lcazon@fce.unju.edu.ar

Abstract. El presente trabajo tiene por objeto exponer la utilización de un modelo web para la evaluación de la usabilidad, específicamente la Calidad en Uso, en tres casos de estudio pertenecientes a aplicaciones de gobierno electrónico y ciencia ciudadana. El modelo se encuentra basado en el estándar ISO/IEC 25010:2011 [1], el cual detalla las sub-características y los atributos a evaluar. Para el proceso de planificación y evaluación, se utilizó el estándar ISO/IEC 25062:2006 [2], que establece el formato CIF, Common Industry Format, para confeccionar informes de pruebas de usabilidad. Este trabajo contiene una breve introducción al concepto de usabilidad, para luego exponer el modelo de usabilidad web utilizado y los casos de estudio elegidos. En las pruebas de usabilidad realizadas oportunamente, por el equipo de investigación, participaron docentes investigadores y alumnos colaboradores del grupo de trabajo. Además se describen los procesos de planificación y evaluación, cuyos resultados se presentan a través de un ejemplo de los casos de estudio elegidos a través del Informe de las pruebas de usabilidad.

Keywords: Usabilidad, Experiencia de Usuario, Aseguramiento de la calidad del Software, Estándares ISO

1. Introducción

En el presente trabajo se expondrán los resultados obtenidos al evaluar la usabilidad de sitios web y aplicaciones pertenecientes a los dominios de gobierno electrónico y juegos científicos. Es importante señalar que nuestro equipo de investigación viene trabajando con distintos estándares y modelos basados particularmente en la norma ISO Square 25000 [3]. Se ha aplicado el estándar ISO 25062: 2006 [2] para a evaluar la usabilidad del software Galaxy Conqueror, un juego destinado a la ciencia ciudadana, que fue desarrollado por el equipo del Laboratorio de Investigación y Formación en Informática Avanzada (LIFIA), perteneciente a la Facultad de Informática de la Universidad Nacional de La Plata[4]. También se evaluaron algunos servicios de gobierno electrónico ofrecidos por la Dirección Provincial de Rentas de la Provincia de Jujuy (DPRJ) y la Administración Federal de Ingresos Públicos (AFIP).

Aunque existen diversas definiciones respecto a la usabilidad y en general todas comparten los mismos aspectos conceptuales, tienen ciertas diferencias en el alcance de las mismas. Según Janice Redish y Joseph Dumas [5], usabilidad significa que la gente que utilice un producto pueda realizar rápida y fácilmente sus tareas. Para ellos la usabilidad representa centrarse en el usuario, que la gente que utilice un producto lo haga de forma productiva y establecen que los usuarios son personas ocupadas que tienen que hacer cosas; y ellos son los que deciden cuando un producto es fácil de usar.

Para Steve Krug[6] la usabilidad realmente significa tener la certeza de que algo funciona bien, es decir, que cualquier persona con conocimientos o habilidades medias, e inclusive por debajo de la media, puede navegar por un sitio web, pilotear un avión o abrir una puerta, sin que cualquier labor que lleve a cabo conlleve su frustración.

Sin embargo Jakob Nielsen[7], establece que es importante tener presente que la usabilidad no es algo simple o una propiedad que ofrezca una única cara relacionada con la interfaz de usuario ofrecida. La usabilidad tiene múltiples componentes e involucra cinco atributos: aprendizaje, eficiencia, memorización, tratamiento de errores y satisfacción.

Asociados a la usabilidad, existen otros conceptos, en tal sentido Ben Shneiderman [8], define a la usabilidad universal como la posibilidad y grado con el que, usuarios diversos con tecnologías diferentes e intereses también distintos, pueden acceder a la información y a los servicios. Para la W3C [9] la accesibilidad web significa que personas, con algún tipo de discapacidad, van a poder hacer uso de la Web. De acuerdo a lo que establece Donald Norman [10], la experiencia de usuario es la respuesta a cómo se sienten los usuarios después de interactuar con un sistema.

De lo expuesto precedentemente se puede observar que existen múltiples conceptos relacionados con la usabilidad.

En el presente trabajo se utilizará un modelo de usabilidad para productos software genérico, extendido y adaptado a productos orientados a la Web y a la norma ISO/IEC 25010 [1].

El proceso de planificación y evaluación del caso de estudio elegido, se llevará a cabo tomando como base lo propuesto por el estándar ISO/IEC 25062 [2].

2. Modelo de Usabilidad Web

En las evaluaciones realizadas, se utilizó el modelo de usabilidad Web propuesto en Abrahão e Insfran [11] para productos software genéricos. Este modelo se ha extendido y adaptado a productos orientados a la Web y a la norma ISO/IEC 25000 SQuaRE y se encuentra detallado en WUEP: Un Proceso de Evaluación de Usabilidad Web Integrado en el Desarrollo de Software Dirigido por Modelos [12].

El Modelo de Usabilidad Web utilizado tiene en cuenta la división de sub-características que se realiza en la ISO/IEC 25010 [1] y que hace referencia al modelo de calidad propuesto por SQuaRE. Sin embargo, dichas sub-características son muy genéricas y se encuentran en un alto nivel de abstracción. En consecuencia el Modelo de Usabilidad Web propuesto para el presente trabajo, descompone dichas sub-características en atributos cercanos al dominio Web, intentando que las definiciones de los atributos se solapen lo mínimo posible entre ellas, obteniendo un modelo de usabilidad muy amplio.

Para lograr adaptar el modelo de calidad propuesto por la norma ISO/IEC 25000 SQuaRE [3], se hace necesario contemplar dos vistas del modelo de usabilidad: una que contemple la usabilidad de la aplicación Web vista desde la perspectiva de un producto software, y otra que contemple la usabilidad de la aplicación Web desde la perspectiva de los usuarios en un determinado contexto, denominada usabilidad en uso o comúnmente señalada como perspectiva de Calidad en Uso.

Esta última perspectiva de Calidad en Uso, es la que se utilizó para evaluar los casos de estudio elegidos. Cada métrica fue asociada a un único atributo, con el objetivo de descubrir problemas de usabilidad en base a los valores obtenidos, pudiendo además cuantificar cómo dicho atributo afecta al nivel de usabilidad de la aplicación Web. En la Tabla 1 se presenta una descripción detallada de la vista Calidad en Uso perteneciente al Modelo de Usabilidad Web, describiendo cada sub-característica, atributos y métricas en los que la misma se descompone.

Tabla 1. Vista Calidad en Uso.

Sub-característica	Atributo	Métrica		
1. Efectividad en uso	1.1. Facilidad de ayuda	1.1.1 Efectividad de la ayuda online	Legibilidad del tutorial	
			Efectividad del sistema de ayuda	
			Facilidad de uso del sistema de ayuda	
	1.2 Rendimiento de las tareas del usuario	1.1.2 Completitud de la ayuda online	Proporción de funcionalidades no cubiertas en las consultas del usuario	
		1.1.3 Frecuencia de consulta de ayuda	Número de veces que el usuario accede a la ayuda por tarea	
		1.2.1 Completado de las tareas	Número de tareas completadas	
2. Eficiencia en uso	2.1 Eficiencia de las tareas del usuario	1.2.2 Exactitud de las tareas	Número de tareas completadas de forma adecuada	
		2.1.1 Tiempo para completar las tareas	Tiempo medio necesario para cumplir una tarea	
	2.2 Esfuerzo cognitivo	2.1.2 Carga de la tarea	Índice User Task Load (UTLIndex)	
		2.2.1 Esfuerzo mental subjetivo	Tasa Subjective Mental Effort (SME ratio)	
		2.2.2 Facilidad de recordar (memorabilidad)	Facilidad de la función de aprendizaje	
	2.3 Limitaciones del contexto		Facilidad de realizar tareas de aprendizaje	
		2.3.1 Carga del sistema	Memoria consumida durante el uso de la aplicación Web	
	3. Satisfacción en Uso	3.1 Satisfacción cognitiva	2.3.2 Adaptación a las habilidades del usuario	Número de perfiles de usuario contemplados
				Número de incidencias en la tarea
3.2 Satisfacción emocional		3.1.1 Utilidad percibida	Número de funcionalidades que el usuario encuentra útiles	
		3.1.2 Calidad de los resultados	Número de funcionalidades que el usuario esperaba encontrar	
		3.2.1 Atracción subjetiva percibida	Número de comentarios positivos del usuario	
3.3 Satisfacción física		3.2.2 Frustración percibida	Número de interrupciones en una tarea	
		3.2.3 Riesgos del contenido	Número de comentarios negativos acerca del contenido	
			Número de comentarios positivos del usuario	
3.4 Confianza		3.4.1 Aparición de errores	Número de errores entre operaciones.	
		3.4.2 Credibilidad del sitio	Calidad de las impresiones del usuario	
	3.4.3 Riesgos económicos	Número de incidentes involucrando pérdidas económicas		
4. Usabilidad en uso Adherida a normas o convenciones	4.1 Grado de conformidad a la ISO/IEC 25000 SQuaRE	Ratio de conformidad cubierto		
	4.2 Grado de conformidad a criterios ergonómicos	Ratio de conformidad cubierto		
	4.3 Grado de conformidad con cuestionarios SUMI, SUS y QUIS	Ratio de conformidad cubierto		

3. Casos de Estudio Elegidos

Los casos de estudios elegidos para efectuar las evaluaciones fueron:

- El sitio web de la Dirección Provincial de Rentas de la Provincia de Jujuy (DPRJ) y, específicamente, los Servicios Web, sin clave fiscal, que ofrece la citada repartición gubernamental y que se encuentra disponible en el sitio web <https://www.rentasjujuyonline.gob.ar/cedulavirtual/ServiciosWeb.aspx>.
- El sitio web de la Administración Federal de Ingresos Públicos de la República Argentina (AFIP) y específicamente los Servicios Web con clave fiscal en los cuales se encuentra la opción Declaración Jurada y Pagos, que ofrece la citada repartición gubernamental y que se encuentra disponible en el sitio web https://auth.afip.gob.ar/contribuyente_/loginClave.xhtml.
- Galaxy Conqueror, un juego destinado a la ciencia ciudadana, disponible en <https://galaxyconqueror.cientopolis.org/>

En la Figura 1 puede apreciarse la pantalla correspondiente a los Servicios Web, pertenecientes a la DPRJ.

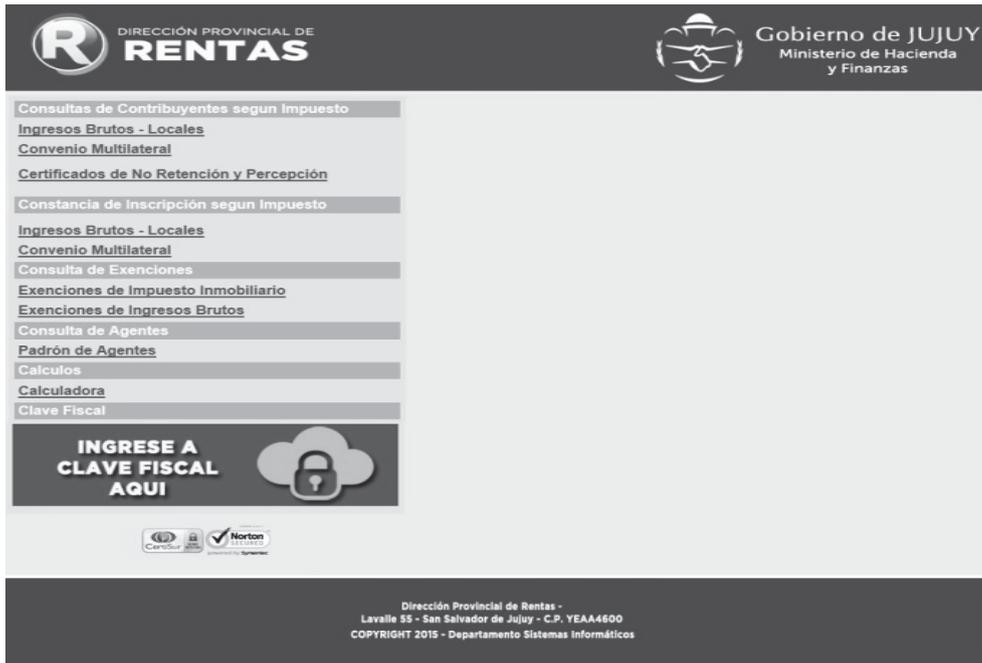


Figura 1. Pantalla de Servicios Web.

Como puede observarse en la Figura 1, los servicios web ofrecidos son mayormente consultas y el uso de una Calculadora de intereses sobre un impuesto determinado, como por ejemplo Ingresos Brutos.

En el presente trabajo se evaluaron los siguientes Servicios Web:

- Consultas de Contribuyentes según impuestos.
- Constancias de inscripción según impuestos.
- Consultas de Agentes.
- Calculadora.

En la Figura 2 puede apreciarse la pantalla correspondiente al menú de declaraciones juradas y pagos pertenecientes a la AFIP.

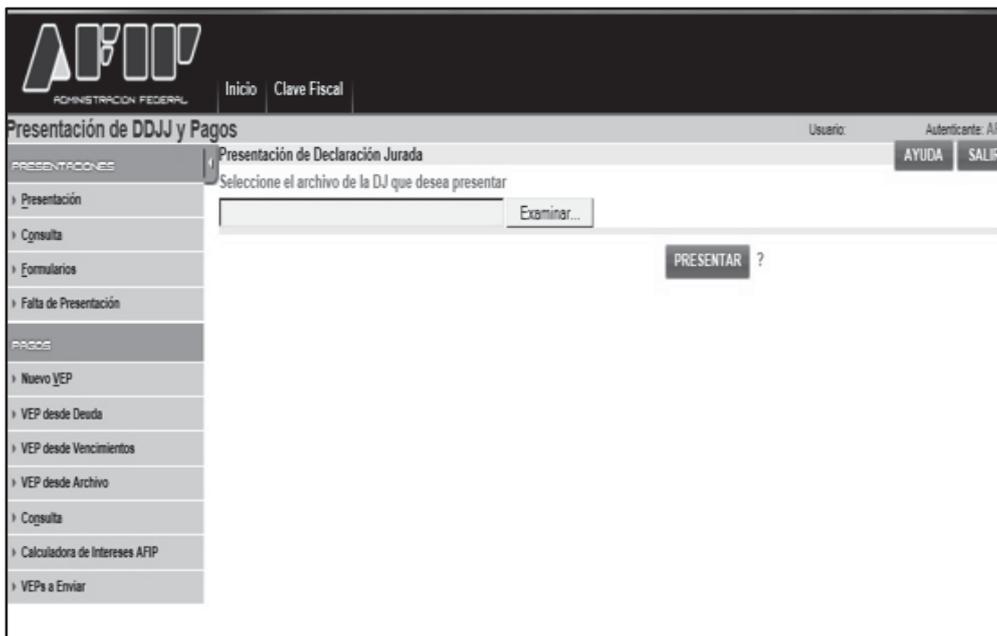


Figura 2. Pantalla de Declaraciones Juradas y Pagos.

Se evaluaron los siguientes servicios:

- Consultas de DDJJ presentadas en el sistema Contribuyentes según impuestos.
- Consulta Falta de Presentación.
- Nuevo VEP (Generación de un Volante Electrónico de Pagos).

En la figura 3 puede observarse la pantalla correspondiente al juego Galaxy Conqueror, el cual motiva a los voluntarios, a marcar potenciales galaxias en una imagen del cielo nocturno. Un breve tutorial les enseña las características básicas. Luego, un equipo de astrónomos del Nuevo Observatorio Virtual Argentino evalúa las galaxias marcadas por los voluntarios.



Figura 3. Pantalla de inicio del juego Galaxy Conqueror.

4. Proceso de Planificación de la Evaluación

Para la evaluación de la Calidad de uso de las aplicaciones detalladas precedentemente y teniendo en cuenta lo propuesto por la norma ISO/IEC 25062 [2], se desarrollaron las siguientes actividades teniendo en cuenta el tiempo estimado para cada etapa:

a. Etapa: Diseño y especificación de la prueba.

Tiempo estimado: 1 semana.

Actividades:

- Definir el equipo que tendrá a cargo la evaluación de usabilidad (*).
- Definir y describir claramente los productos a evaluar.
- Establecer objetivos de la prueba de usabilidad.
- Definir características de los participantes (**).
- Seleccionar participantes.
- Definir y describir las tareas a ejecutar durante la prueba.
- Definir el equipamiento y software a utilizar.
- Determinar las variables a registrar y métricas a utilizar.
- Definir las fechas y preparar el lugar de realización de las pruebas (**).
- Explicitar, documentalmente, la descripción de la prueba: consignas y cuestionarios.
- Efectuar una prueba piloto, para determinar desviaciones a lo previsto (**).
- Efectuar las correcciones necesarias.
- Convocar formalmente a los participantes seleccionados, teniendo en cuenta lo definido.

(*) El equipo encargado de realizar la evaluación estará conformado por los integrantes del equipo de investigación I+S (Ingeniería y Software) perteneciente a la Facultad de Ingeniería de la Universidad Nacional de Jujuy. La coordinación general estará a cargo del Director del equipo de investigación.

(**) Los participantes serán docentes y alumnos integrantes del equipo de investigación. Los perfiles, en general, de los usuarios son usuarios expertos y usuarios visitantes.

(***) El Lugar de realización de las pruebas será el Centro de Cómputos de la Facultad de Ingeniería.

(****) Para la prueba piloto, se designarán dos docentes investigadores y dos alumnos, pertenecientes al equipo de investigación.

b. Etapa: Desarrollo de la prueba.

Tiempo estimado: 1 semana.

Actividades:

- Poner en funcionamiento y controlar equipamiento y software.
- Recibir a los participantes y explicar lo que se desarrollará durante la prueba.
- Ejecutar las tareas, según lo definido.
- Realizar cuestionarios definidos.
- Efectuar una reunión con el equipo encargado de realizar la evaluación, para evaluar el proceso desarrollado.

c. Etapa: Análisis y presentación de resultados.

Tiempo estimado: 1 semana.

Actividades:

- Analizar datos obtenidos en la evaluación (registro, clasificación y estadísticas).
- Elaborar y presentar resultados obtenidos.

5. Proceso de Ejecución de la Evaluación

A continuación se puede visualizar el informe del CIF (Common Industry Format), especificado por el estándar ISO/IEC 25062 [2], para confeccionar informes de pruebas de usabilidad, para el proceso de evaluación de los productos software de los casos de estudio elegidos.

A modo de ejemplo se detallan los ítems A,B,C y D para los servicios Web pertenecientes a la Dirección Provincial de Rentas de la Provincia de Jujuy.

A - ISO/IEC 25062 Common Industry Format for Usability Test Reports “Servicios Web DPRJ”.

Evaluado por: I+S (Ingeniería y Software) para la Dirección Provincial de Rentas de la Provincia de Jujuy, Lavalle 63, San Salvador de Jujuy, Jujuy.

B - RESUMEN EJECUTIVO: los servicios web ofrecidos por la DPRJ a través de su sitio web, permiten, a los contribuyentes y ciudadanos en general, realizar:

1. Consultas de contribuyentes según impuesto.
2. Obtener constancia de inscripción según impuesto.
3. Consulta de Exenciones.
4. Consulta de Agentes.
5. Cálculos.

Se realizó un test de usabilidad, evaluando la Calidad de Uso, cuya tarea principal fue interactuar con el software desarrollado por la DPRJ y particularmente con las opciones 1,2,4 y 5, expuestas precedentemente.

En el test de usabilidad participaron 10 docentes investigadores y 4 alumnos. Cada participante ingresó al software bajo estudio y realizó las tareas de verificación que se explicitan en la Tabla 1:

C- OBJETIVOS DEL TEST:

El test fue realizado para evaluar la Calidad de Uso de los Servicios Web de la DPRJ. Los atributos evaluados se encuentran detallados en el ítem B del presente informe.

D- MÉTODO:

D.1- Participantes.

En la prueba de usabilidad participaron 10 docentes investigadores en el área de la ingeniería de software y 4 alumnos colaboradores, con el equipo de investigación.

La experiencia de los usuarios en la utilización de aplicaciones de gobierno electrónico, se puede resumir en que los docentes poseen una alta experiencia en el uso de aplicaciones web para gobierno electrónico y los alumnos colaboradores no poseen ninguna experiencia en el uso de las citadas aplicaciones. En la Tabla 2 se puede visualizar el nivel de experiencia en la utilización de aplicaciones web para gobierno electrónico, utilizando una escala del 1 (más baja) al 7 (más alta).

Es importante señalar que los valores correspondientes al Nivel de experiencia consignados en la Tabla 2, a pesar de ser subjetivos, los mismos fueron obtenidos teniendo en cuenta la experiencia de los participantes en el dominio de gobierno electrónico, en algunos casos más de quince años, como así también en la funcionalidad específica de la aplicación web bajo estudio debido a que mensualmente el servicio web es utilizado por los participantes.

Tabla 2. Clasificación de los participantes por Nivel de experiencia- Escala 1 a 7.

Participante	Perfil	Nivel de experiencia	Participante	Perfil	Nivel de Exp.
1	Docente –Investigador	7	8	Docente –Investigador	7
2	Docente –Investigador	7	9	Docente –Investigador	7
3	Docente –Investigador	7	10	Docente –Investigador	7
4	Docente –Investigador	7	11	Alumno Colaborador	1
5	Docente –Investigador	7	12	Alumno Colaborador	1
6	Docente –Investigador	7	13	Alumno Colaborador	1
7	Docente –Investigador	7	14	Alumno Colaborador	1

D.2 - Contexto de uso del producto en el test.

D.2.1 - Tareas.

D.2.1.1 - Descripción de los escenarios de tareas.

En primer lugar se realizó una explicación verbal del escenario de la prueba a los participantes. La explicación se centró en tres tareas: a) Ingresar al sitio de los servicios Web de la DPRJ, b) Efectuar una sesión de interacción de las opciones: Consultas de contribuyentes según impuesto, Obtener constancia de inscripción según impuesto, Consulta de Exenciones, Consulta de Agentes y Cálculos y c) Salir de la Aplicación

D.2.1.2 - Entorno computacional de los participantes.

Para la prueba de usabilidad se utilizaron 10 computadoras, ubicadas en el Centro de Cómputos de la Facultad de Ingeniería, con las siguientes características:

- Procesador: Pc Intel Core I7 – 8Mb Cache 3.40 Ghz.
- MotherBoard (Plataforma B75 con USB3, UEFI Bios, PCIe 3.0, Sata3, Virtu MVP, Intel B75, 8 x USB 2.0/4 x USB 3.0, 8 Channel High Definition Audio Realtek ALC 889, Gigabit LAN, 4 x SATA 3 Gb/s ports, 1 xPCIe 4.0 x 16 1xPCI- E 1x, 1x PCI 32, 2 x DDR sockets supporting up 16 Gb, Dual Channel).
- Disco Rígido (HD): 1TB SATA3 WD 64 MB.
- Memoria: 8gb DDR3 1600 MHZ.
- Monitor: Led 19” (1368 x 768).

D.3 - Diseño experimental y procedimiento.

Se dividieron dos grupos de usuarios, los docentes investigadores con experiencia en aplicaciones web para gobierno electrónico y aquellos con escasa experiencia en el uso de las mismas. Se les indicó que realizarán una sesión interactiva con las tareas detalladas en el punto D.2.1.1.

Posteriormente se les proporcionó un cuestionario para que respondieran preguntas, o también, se cronometró la tarea realizada. Es importante señalar que debido a la extensión del cuestionario, el mismo no se presenta en este trabajo.

Luego se agruparon las preguntas realizadas, en el cuestionario, en los atributos especificados en el punto B.

Finalmente se procedió al análisis de las variables de performance obtenidas, clasificándolas y elaborando los informes finales.

E- PRESENTACIÓN DE LOS RESULTADOS.

A continuación en la Tabla 3, se describen los resultados obtenidos en base a lo especificado en el inciso D, teniendo en cuenta la estructura de sub-características/atributos/métricas, expuesta en la Tabla 1.

Tabla 3. Resultados obtenidos

Sub-característica:	Atributo	Métrica	Escala	Valor medio obtenido
Efectividad en uso/Facilidad de Ayuda	Efectividad de la ayuda On-Line	Legibilidad del tutorial	Valor real entre 0 y 7	6,43
		Facilidad de uso del sistema de ayuda	Valor real entre 0 y 7	6,71
	Compleitud de la ayuda On-Line	Proporción de funcionalidades no cubiertas en las consultas del usuario	Valor real entre 0 y 7	0,36
	Frecuencia de consulta de ayuda	Número de veces que el usuario accede a la ayuda por tarea	Valor real entre 0 y 7	2,36
Efectividad en uso/Rendimiento de las tareas del usuario	Completado de tareas	Número de tareas completadas	Valor real entre 0 y 7	6,14
	Exactitud de las tareas	Número de tareas completadas de forma adecuada	Valor real entre 0 y 7	4
Eficiencia en uso /Eficiencia de las tareas del Usuario	Tiempo para completar las tareas	Tiempo medio necesario para cumplir una tarea	Valor real entre 0 y 7	4,92
	Carga de la Tarea	Index User Task Load (UTLindex)	Valor real entre 0 y 7	3,57
Eficiencia en uso/Esfuerzo Cognitivo	Esfuerzo mental subjetivo	Tasa Subjective Mental Effort (SME ratio)	Valor real entre 0 y 7	4,57
	Facilidad de recordar (memorización)	Facilidad de la función de aprendizaje	Valor real entre 0 y 7	2,57
		Facilidad de realizar tareas de aprendizaje	Valor real entre 0 y 7	2,42
Eficiencia en uso/Limitaciones del contexto	Carga del sistema	Memoria consumida durante el uso de la aplicación web	Valor real entre 0 y 7	6,50
	Adaptación a las habilidades del usuario	Número de perfiles de usuarios contemplados	Valor real entre 0 y 7	7
		Número de incidencias en las tareas	Valor real entre 0 y 7	6,35
Satisfacción en Uso/Satisfacción cognitiva	Utilidad Percibida	Número de funcionalidades que el usuario encuentra útiles	Entero mayor o igual que 0	2,071
	Calidad de los Resultados	Número de funcionalidades que el usuario esperaba encontrar	Entero mayor o igual que 0	4
Satisfacción en Uso/Satisfacción Emocional	Atracción Subjetiva Percibida	Número de comentarios positivos del usuario	Entero mayor o igual que 0	3
	Frustración Percibida	Número de interrupciones en una tarea	Entero mayor o igual que 0	8,92
	Riesgos del Contenido	Número de comentarios negativos acerca del contenido	Entero mayor o igual que 0	5,84
Satisfacción en Uso/Satisfacción Física	Satisfacción Física	Número de comentarios positivos del usuario	Entero mayor o igual que 0	3
Satisfacción en Uso/Confianza	Aparición de Errores	Número de errores entre operaciones	Entero mayor o igual que 0	6
	Credibilidad del Sitio	Calidad de las impresiones del usuario	Entero mayor o igual que 0	5
	Riesgos Económicos	Número de incidentes involucrando pérdidas económicas.	Entero mayor o igual que 0	9

E.1 Gráficas de Resultados Obtenidos por Sub-Características

De acuerdo a los valores medios obtenidos, en la Tabla 3 se elaboraron gráficos (Figura 4) que resumen los resultados para las distintas sub-características evaluadas.

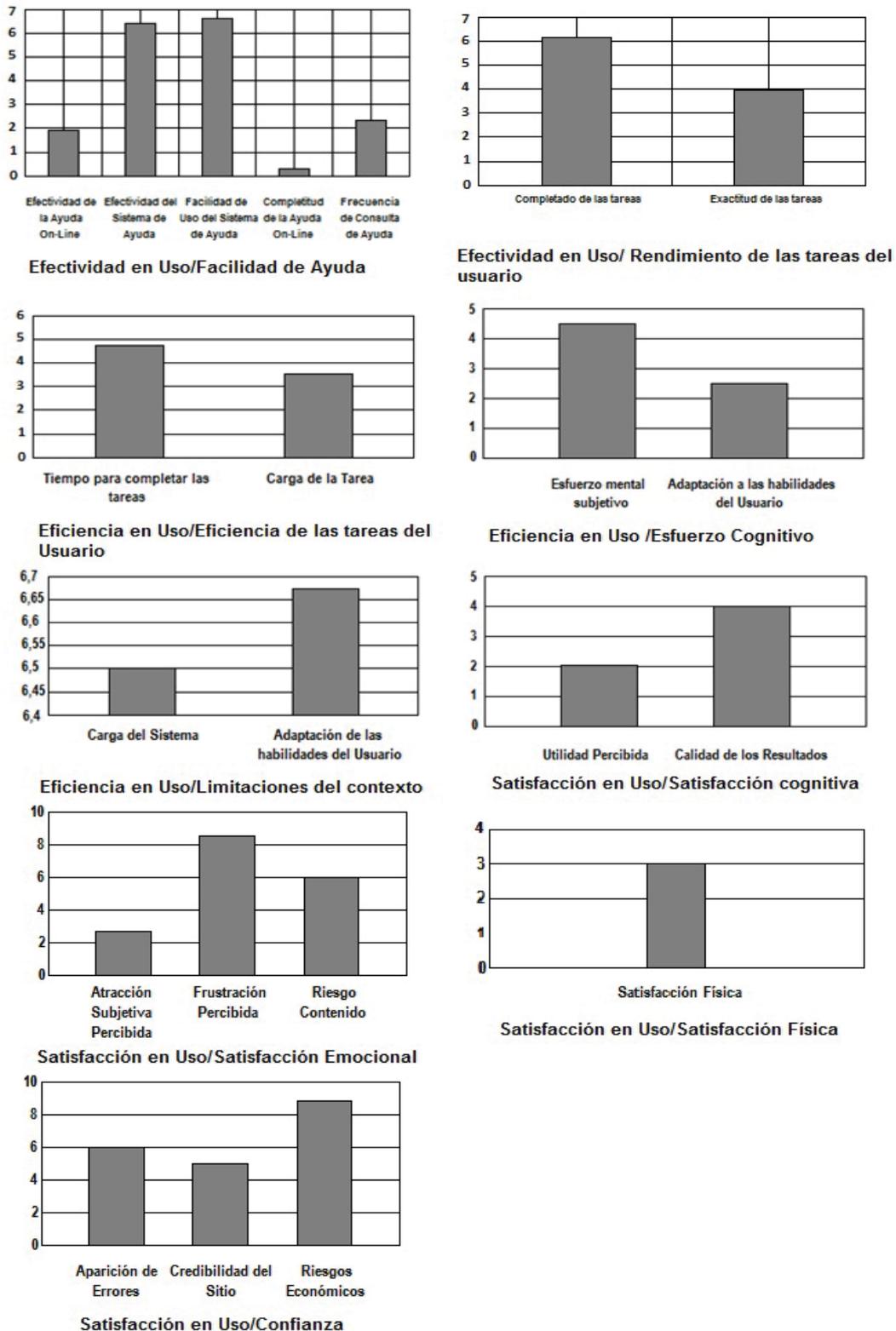


Figura 4. Gráficas de Resultados Obtenidos por Sub-Características

6. Conclusiones

Teniendo en cuenta las tres pruebas de usabilidad realizadas, se puede observar que independientemente del dominio de la aplicación, la Calidad en Uso es escasa por no concluir que es inexistente, lo cual representa un aspecto que debe revisarse por parte de los desarrolladores del producto software; especialmente en el caso de las aplicaciones de gobierno electrónico.

En el caso de la prueba de usabilidad de Galaxy Conqueror, al tratarse de la evaluación de un juego serio, no se intentó probar y medir la facilidad de uso, porque una de las principales motivaciones de los juegos es precisamente el reto a la habilidad de un jugador, sino verificar la adecuación al nivel de dificultad requerido en cada fase del juego. En otras palabras, la usabilidad de un juego en línea es alta cuando su nivel de dificultad es el adecuado, particularmente para el caso de estudio elegido el nivel de frustración es bajo, lo cual demuestra que se trata de un juego serio, que no necesita demasiada habilidad para lograr cumplir con los objetivos que demanda.

En referencia al objetivo planteado por el presente trabajo sobre la utilización del estándar ISO 25062, se puede concluir que el mismo resulta adecuado y perfectamente adaptable a cualquier tipo de producto software, inclusive en el caso de un juego en línea, cuyos objetivos de usabilidad son muy diferentes a aplicaciones funcionales, donde se trata de evaluar eficacia, eficiencia y grado de satisfacción de los usuarios

Finalmente y en relación al uso de los estándares de la familia ISO/IEC 25000 SQuaRE, se puede decir que el ISO/IEC 25010 resultó muy útil y de fácil aplicación para el diseño de modelos Web, como el utilizado en el presente trabajo y se complementó correctamente con el estándar ISO/IEC 25062.

7. Referencias

1. ISO/IEC 25010, Systems and software engineering -Systems and software Quality - Requirements and Evaluation (SQuaRE) -System and software quality models (2011).
2. ISO/IEC 25062, Software engineering -Software product Quality Requirements and Evaluation (SQuaRE) - Common Industry Format(CIF) for usability test reports (2006).
3. ISO/IEC 25000, Software Engineering - Software Product Quality Requirements and Evaluation (SQuaRE),Guide to SQuaRE (2005).
4. Castro, Marcelo A., Sánchez Rivero, Víctor D., Farfán, José H., Vargas, Luis A., Reinoso, Elizabeth, Aparicio, María C., Aragón, Fabiana R., Zapana, José V., Cándido, Andrea G. y Cazón, Liliana E. "Evaluando la usabilidad de Galaxy Conqueror con ISO 25062", Actas de las XI Jornadas de Ciencia y Tecnología de Facultades de Ingeniería del NOA, San Salvador de Jujuy, Argentina (2016).
5. Dumas, J. S. and J. C. Redish, A Practical Guide to Usability Testing, Rev. Edition, Intellect Books (1999).
6. Krug, S., Don't Make Me Think: A Common Sense Approach to the Web, 2nd Edition, New Riders Publishing (2005).
7. Nielsen, J., The Usability Engineering Life Cycle, IEEE Computer 25(3), Pág. 12-22 (1992).
8. Shneiderman, B., Universal Usability, ACM 43(5), Pág. 84-91(2000).
9. Accesibilidad, WorldWide Web Consortium (W3C). <http://www.w3c.es>, [Último acceso: abril de 2016].
10. Norman, D., The Design of Everyday Things, Basic Books (2002).
11. Allen, M.; Currie, L.; Bakken, S.; Patel, V. and Cimino, J. "Heuristic evaluation of paperbased Web pages: A simplified inspection usability methodology", (2006).
12. Fernández Martínez, A., "WUEP: Un Proceso de Evaluación de Usabilidad Web Integrado en el Desarrollo de Software Dirigido por Modelos", Tesina de Máster en Ingeniería del Software, Métodos Formales y Sistemas de Información, Universidad Politécnica de Valencia, (2009).

Framework IoT aplicado a Monitoreo de Precipitaciones de la Provincia del Chaco

Federico Aguirre, Lucas Ibañez, Claudio Basilio, Sergio Gramajo

Universidad Tecnológica Nacional, Ecom, Argentina
{federodani, lucas.sebib, basilio.claudio}@gmail.com, sergio@fre.utn.edu.ar

Resumen. En los últimos años, en la provincia del Chaco, se generaron abundantes precipitaciones provocando inundaciones y generando pérdidas para el sector agrícola, ganadero y la sociedad en general. A su vez, la aparición de enfermedades debido a la saturación que sufren las napas por los anegamientos de agua por largos períodos, se proliferaron. Esta problemática exige que se tomen medidas para tener datos confiables y disponibles, esencialmente de precipitaciones, para tomar decisiones. En este sentido, nuestro trabajo de investigación y desarrollo se basa la necesidad de contar con una fuente de datos precisos y en tiempo real que contribuyan a la activación de los distintos planes de contingencias que se encuentran establecidos y dar soporte a la toma de decisiones. Así, este trabajo se centra, específicamente, en la contribución tecnológica para afrontar estos cambios climáticos y se propone un Framework IoT aplicado al monitoreo de precipitaciones en toda la provincia del Chaco, que va desde la captación de las precipitaciones mediante pluviómetros, la comunicación por fibra óptica, procesamiento, generación de alertas y estadísticas en tiempo real.

Palabras Claves: Framework IoT; Monitoreo Web en Tiempo Real; Cambio Climático.

1 Introducción

Las intensas y prolongadas precipitaciones que afectan a la provincia del Chaco [1], generaron pérdidas millonarias para el sector ganadero y agrícola (producción de girasol, algodón, soja, maíz y sorgo). Los anegamientos de agua provocado por los niveles altos de las napas, trae como consecuencia diferente tipos de enfermedades, combinadas con las altas temperaturas en la región, da la aparición de dengue o presencia de gastroenteritis aguda, etc. Estos son problemas que se enmarcan en diferentes estratos del Estado. Sin embargo, nuestro trabajo se centra en proponer, desde el punto de vista tecnológico y de telecomunicaciones, un framework innovador para la automatización de la captación de los datos que ayude a tomar decisiones.

Hoy una solución de esta naturaleza es posible debido a los avances en el campo de Internet de las Cosas (IoT) y las telecomunicaciones que están transformando las maneras en que las personas interactúan con el entorno [3][4][5]. Estos progresos buscan optimizar tareas rutinarias u operativas, los recursos y además incrementar el cuidado del medio ambiente. La accesibilidad de sensores o actuadores, placas electrónicas y

los avances en la optimización de sistemas de software han llevado a la creación de dispositivos económicos que pueden ser usados en la implementación de sistemas de IoT hogareños o de menor escala que los industriales. Sin embargo, el correcto desarrollo de un sistema de IoT podría significar un gran desafío como el correcto diseño de una arquitectura que favorezca la escalabilidad en cuanto a cantidad de clientes del sistema, sensores, actuadores y ubicación geográfica de los mismos.

La utilización de un sistema de monitoreo y censado de diversas variables climáticas desplegado en una zona geográfica extensa y utilizando las tecnologías y avances de IoT permitirá disponer de información actualizada y en tiempo real para poder tomar los diferentes cursos de acción que se dispongan para estas situaciones. Además, permitirá optimizar las tareas y procedimientos actuales de captación de datos en pos de garantizar la generación y entrega de valor de mejores resultados para la toma de decisiones más eficientes.

Teniendo en cuenta esta problemática y la disponibilidad tecnológica, en este trabajo proponemos un desarrollo innovador que permite modernizar el proceso de análisis de variables climáticas como pluviométrica. Así, el desarrollo de nuestra propuesta está focalizada particularmente para su utilización por la sociedad y mediante los organismos oficiales de manejo de este tipo de datos como son (en la provincia del Chaco) la Administración Pública del Agua y el gobierno Provincial.

Este artículo se estructura de la siguiente manera. En la sección 2, se presenta un framework de IoT genérico que brinda un conjunto de servicios para la gestión y administración de sistemas de IoT de cualquier escala. En la sección 3 presentamos un caso de uso de dicho framework con mediciones ambientales. Para finalizar enunciamos proyectos futuros y concluimos resaltando los aportes más significativos de nuestra investigación.

2 Framework IoT

El framework propuesto, de ahora en más SiMo (Sistema de Monitoreo), nace con el objetivo de aglomerar tecnologías e infraestructura para hacer frente a soluciones para IoT, haciendo más simple y transparente su aprendizaje. Esto permite la creación de un ecosistema interconectado en donde el uso de recursos convencionales aumenta en eficiencia gracias a la aplicabilidad de la tecnología. SiMo permite la integración de múltiples dispositivos hardware, entre ellos sensores y actuadores para la captación de datos de acuerdo a los parámetros que mejor se ajusten a las necesidades de los usuarios y escenarios de trabajo. Además, permite personalizar distintas configuraciones para cada uno de ellos y posteriormente obtener los datos para almacenarlos y a su vez controlar objetos (como sensores, actuadores, equipos, etc.) de manera simple y amigable para que puedan cooperar entre ellos para alcanzar objetivos en común.

SiMo, se encuentra dividido en dos grandes módulos: SiMoW (SiMo Web) y SiMoRa (SiMo en Raspberry), dedicados a tareas puntuales que hacen en su totalidad a la solución final.

Ambos módulos están diseñados para trabajar en conjunto, como se observa en la Fig. 1. Todos los parámetros de configuración y funcionamiento de SiMoRa pueden ser seteados y consultados desde SiMoW, mediante el uso de REST API [6].

El desarrollo del Framework se basa en un diseño de desarrollo de software siguiendo una arquitectura de capas [11][12], permitiendo añadir nuevas funcionalidades según se necesite, para trabajar sobre diversos escenarios, siendo totalmente transparente al usuario. Esto facilita la escalabilidad y adaptabilidad a las necesidades de cualquier proyecto que se plantee como objetivo.

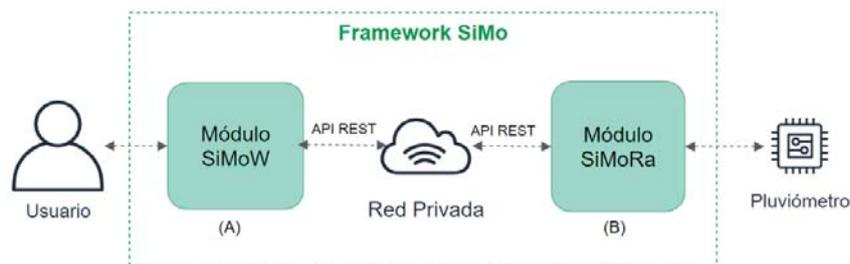


Fig. 1. Framework SiMo con módulos (A) SiMoW y (B) SiMoRa

2.1 Módulo SiMoW

El módulo SiMoW (Fig.1.A), es una plataforma web, el cual presenta una interfaz amigable para la visualización de datos (tanto en tiempo real como históricos), monitoreo de estado de dispositivos y configuración remota de los mismos. Por otro lado, permite configurar alertas y notificaciones.

SiMoW está desarrollado usando el framework Django de Python en conjunto con otras tecnologías como Nginx, Gunicorn, Redis, Supervisor, Bootstrap y MySQL como motor principal de Base de Datos [15][16][17].

A su vez, SiMoW consta de una serie de módulos, que se puede observar en la Fig. 2, como la administración de perfiles permitiendo gestionar permisos y accesos a las distintas funcionalidades que presenta el sistema, personalizando para cada tipo de usuario las acciones que puede realizar.

La gestión de Equipos y Pluviómetros permite organizar equipos y sensores en lugares geocalizados permitiendo visualizar en un mapa la ubicación y el estado actual de los mismos.

Actualmente se dispone de varios tipos de pluviómetros admisibles y disponible para su utilización. Además, SiMoW, en conjunto con SiMoRa, están desarrollados de tal manera que a medida que se necesite añadir nuevos sensores no soportados por el sistema, se incorporen fácilmente debido a su modularidad.

Para la visualización de información se dispone de un dashboard en tiempo real [10] y además cuenta con paneles en donde se podrá visualizar información estadística histórica, ya sea de manera gráfica para una mejor comprensión, o bien en formatos tabulares.

La información recibida por los equipos y sensores es almacenada cronológicamente según el momento en que fue obtenido dicho valor, para una posterior consulta o generación de reportes o a fines de cálculos estadísticos.

Por último, SiMoW cuenta con la posibilidad de gestionar alertas y notificaciones. A cada equipo o pluviómetro se puede le puede asociar alertas, permitiendo tomar cursos de acción ante la activación de dicha alerta, como ser notificaciones en diferentes medios de comunicación. Actualmente, se cuenta con varios sistemas de mensajería como son Telegram, Email, Slack y SMS para el envío de notificaciones.

Cabe destacar, que SiMoW brinda soporte y control sobre SiMoRa, mediante la utilización del módulo de API REST, lo que facilita tanto la obtención como manipulación en forma remota del sistema.



Fig. 2. Módulos de SiMoW

2.2 Módulo SiMoRa

El módulo SiMoRa (Fig.1.B), se encarga de interactuar directamente con los pluviómetros, tomando control del censado de los datos, el almacenamiento temporal, la posterior comunicación y sincronización de los mismos.

El módulo SiMoRa está desarrollado sobre una placa Raspberry Pi, sobre el cual se ejecutan un conjunto de servicios. La misma tiene como Sistema Operativo base un Raspbian [13].

La comunicación entre los pluviómetros y la Raspberry Pi puede ser por una conexión directa y cableada a los pines de la GPIO o a través de comunicación inalámbrica como wifi, módulos transceiver NRF24L01, etc. [13][14]. SiMoRa centraliza la captura, recolección y envío de datos, aunque existe la posibilidad de manejo de varios tipos de comunicación con otros dispositivos como placas Arduinos gracias el soporte de múltiples protocolos de mensajería como MQTT [8][9].

Los Servicios se desarrollaron utilizando Python como lenguaje de desarrollo. Para el resguardo temporal de la información se utilizó SQLite como base de datos [15].

SiMoRa consta de 3 módulos básicos. Un módulo de monitoreo, encargado de la recopilación, almacenamiento y envío de datos censados por cada pluviómetro conectado al mismo. Desarrollado como una aplicación multithreading, lo que favorece la independencia en la captura de datos. Cada pluviómetro activo se ejecuta sobre un hilo y se encarga de la recopilación de los datos de acuerdo a los parámetros con los que se configuró previamente. Los datos capturados son enviados a un servicio REST establecido con posterioridad (vía JSON), y para el cual, SiMoW cuenta con soporte a dicho

servicio [6]. SiMoRa envía los datos a SiMoW asimismo posee la posibilidad de establecer otros servicios secundarios adonde reportar. En el caso de que no se cuente con la comunicación con este servicio debido alguna falla de conexión en la red, los datos se almacenarán localmente hasta que la comunicación se restablezca y se puedan sincronizar los mismos.

El segundo módulo es una API REST que se desarrolló utilizando el microframework de Python llamado Flask [7], para la comunicación con SiMoW. Este permite tener el control remoto de SiMoRa gracias a la amplia variedad de microservicios desarrollados como la sincronización de información de sensores (alta, baja, edición y eliminación), inicio y detención de módulos y del mismo equipo Raspberry Pi, lectura on demand de datos y la configuración de múltiples parámetros operacionales.

Por último, un módulo encargado de manejar un servidor MQTT para la comunicación M2M con otros dispositivos [8][9].

En la Fig.3 podemos ver la interacción entre cada uno de estos módulos. Cada aplicación se presenta como servicios independientes que cooperan y se comunican entre sí. Todos los servicios se autoinician con el encendido de la Raspberry Pi y cuentan con un log donde se deja registro de cada acción que se lleva adelante.

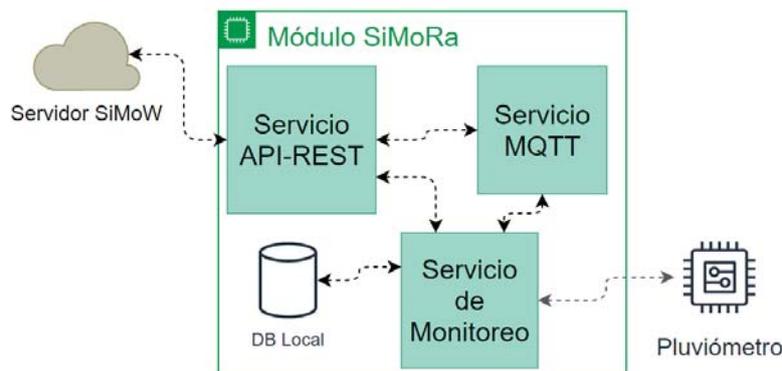


Fig. 3. Módulos de SiMoRa

3 Caso de aplicación: Monitoreo de Precipitaciones

La APA (Administración Provincial del Agua) es el organismo oficial y constitucional que se encarga de regular, proyectar, ejecutar planes generales de obras hidráulicas, riego, canalización y defensa, entre otras labores. Para llevar a cabo esta labor se requiere, entre otras cosas, contar con información de las precipitaciones de toda la provincia del Chaco. Actualmente, dicha entidad, cuenta con 35 equipos pluviómetros aproximadamente, distribuidos por toda la provincia como se puede observar en la Figura 4, donde cada uno dispone de un equipo contador de pulsos y una pantalla por la cual se puede visualizar la cantidad de pulsos por milímetros caídos. Este mismo es operado por la policía ya que los mismos se encuentran ubicados en destacamentos policiales. El proceso es manual y cada determinada hora preestablecida se debe

efectuar la lectura observando dicha pantalla y comunicándose con la entidad del APA para registrar dicho valor, luego reiniciar manualmente el equipo para que esté listo para otra toma de datos.

Este tipo de operatoria trae diversos problemas ya que se debe tener que contar de una persona para que realice dicha labor y, a su vez, disponer de un medio de comunicación de alta disponibilidad para enviar el dato. Esto puede generar un problema extra en diversas épocas del año (especialmente en verano) ya que es difícil contar debido a que hay muchos cortes de energía o a causa de los temporales se queden incomunicados.



Fig. 4. Ubicación de pluviómetros del APA

Es, por tanto, que SiMo viene a brindar solución a las problemáticas mencionadas anteriormente, permitiendo a su vez poder reutilizar equipos pluviómetros y adaptarlos para trabajar con la plataforma mediante la utilización del módulo SiMoRa. Esto permite ahorrar costos de adquisición de nuevos equipos y reutilizar los existentes.

Por otro lado, la empresa ECOM chaco cuenta con sitios de fibra óptica distribuidos en toda la provincia, muy cercanos a los lugares donde se encuentran dichos equipos pluviómetros, por lo cual puede garantizar la alta disponibilidad y seguridad para el sistema.

Cada equipo pluviómetro con su módulo SiMoRa, se encargará de reportar los datos al módulo SiMoW, donde cada usuario previamente registrados a dicha plataforma, podrá visualizarlo y acceder a las funcionalidades que tengan disponibles según el perfil tenga establecido. Dentro de las funcionalidades básicas, y a modo ejemplificativo

breve de las diferentes funcionalidades, podemos visualizar estadísticas de un pluviómetro, observando los datos registrados dentro de un período o fecha en particular, pudiendo ver acumulados por 5 minutos, hora o en días, entre otras cosas (ver Figuras 5 y 6).

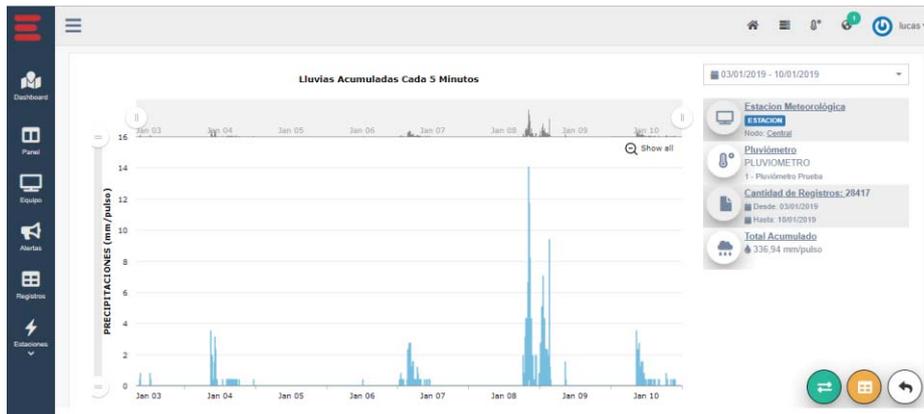


Fig. 5. Visualización de Estadísticas de registros de precipitaciones cada 5 minutos



Fig. 6. Visualización de Estadísticas de registros de precipitaciones por día

4 Conclusión y Trabajos a futuros

Actualmente, el Framework SiMo se encuentra en fase de pruebas y trabajando con varios tipos de pluviómetros. Es importante mencionar que no solamente da solución a la problemática antes mencionada, sino que puede ser un sistema escalable donde se monitoreen variables climáticas en tiempo real, como estaciones meteorológicas portables, permitiendo garantizar la máxima disponibilidad del servicio de comunicación por medio de fibra óptica. Otra de las ventajas es que su implementación ayuda a disminuir

costos, tanto de equipos como de tiempo y servicios. A su vez, este tipo de solución permitirá ayudar a tomar decisiones basadas en información precisa y tener una herramienta ante los problemas de las grandes inundaciones, principalmente en las ciudades, como mecanismos de alertas o nuevos dispositivos o sensores. Disponer de información en tiempo real, dará lugar a mejorar los planes de contingencias, alertas y en otros casos a nuevos proyectos hídricos. En este sentido los avisos se pueden plasmar por una diversidad de medios de comunicación existentes.

Como extensión al proyecto se está trabajando en incorporar nuevas tecnologías de comunicación LoRa y Sigfox, permitiendo darle a la plataforma mayor flexibilidad a la hora de comunicarse, logrando poder llegar a alcanzar lugares remotos dentro de las ciudades. Asimismo, nuestro equipo se encuentra trabajando en un nuevo prototipo que permite ser totalmente autónomo, utilizando como fuente alternativa de alimentación, la energía solar y pudiendo almacenarse en baterías que permitirían ser utilizadas como fuente primaria de alimentación. Finalmente, la segunda fase del proyecto consiste en dotar a la plataforma con la capacidad de procesamiento de datos y analítica predictiva para obtener conocimiento a partir de los mismos.

Reconocimientos

Este trabajo es el resultado de un convenio de transferencia tecnológica entre la UTN FRRe y la empresa Ecom Chaco SA en colaboración la Administración Provincial del Agua con el objeto de crear una plataforma moderna para la toma de decisiones ambientales (pluviométricas) de la provincia del Chaco.

Referencias

1. APA Anuario de precipitaciones Provincia del chaco 1956 - 2017, <http://apachaco.gob.ar/site/images/anuarios/ANUARIO%20PLUVIOMETRICO.pdf>, último acceso 29/07/2019
2. Administración Provincial del Agua (APA), <http://apachaco.gob.ar/site/index.php/a-p-a/que-es-la-a-p-a>, último acceso 29/07/2019
3. Vermesan, O., Friess, P., Guillemin, P., Giaffreda, R., Grindvoll, H., Eisenhauer, M., et al. (2015). Internet of things beyond the hype: Research innovation and deployment. IERC Cluster SRIA
4. Advancing the IoT for Global Commerce, <https://autoidlabs.org/>, último acceso 29/07/2019.
5. Gubbi, J., Buyya, R., Marusic, S., & Palaniswami, M. (2013). Internet of Things (IoT): A vision, architectural elements, and future directions. *Future Generation Computer Systems*, 29, 1645–1660.
6. J. Mueller, SOAP and REST Basics and Differences <https://smartbear.com/blog/test-and-monitor/understanding-soap-and-rest-basics>, último acceso 29/07/2019.
7. Flask, último acceso 29/07/2019.
8. OASIS, MQTT Version 3.1.1 Plus Errata 01, <http://docs.oasis-open.org/mqtt/mqtt/v3.1.1/mqtt-v3.1.1.html>, último acceso 29/07/2019.
9. IBM, Why MQTT is good for IoT? <https://www.ibm.com/developerworks/ssa/library/iot-mqtt-why-good-for-iot/index.html>, último acceso 29/07/2019.

10. Seppo J. Ovaska, Real-Time Systems Design and Analysis, 4^o Edición, Wiley, 2012.
11. L. Tan, N. Wang, Future Internet: The Internet of Things, in 3rd International Conference on Advanced Computer Theory and Engineering (ICACTE), 2010.
12. Atzori L., Iera A., Morabito G.: "The Internet of Things: A survey". ELSEVIER Journal Computer Networks. Volume 54, Issue 15. 2010.
13. Schmidt, M. Raspberry Pi: a quick-start guide. Pragmatic Bookshelf. (2014).
14. Richardson, M., Wallace, S, Getting started with raspberry PI., O'Reilly Media, Inc., 2012.
15. SQLite, <https://www.sqlite.org/docs.html>, último acceso 29/07/2019.
16. Nginx, <https://docs.nginx.com>, último acceso 29/07/2019.
17. Django, <https://docs.djangoproject.com/en/2.2/>, último acceso 29/07/2019.
18. Oracle, Why MySQL? <https://www.mysql.com/why-mysql/>, último acceso 29/07/2019.

Marco de trabajo para la aplicación de línea de producto en el desarrollo de software para gobierno digital

Mauro Cambarieri, Luis Vivas, Nicolás Garcia Martinez

Laboratorio de Informática Aplicada, Universidad Nacional de Río Negro, Argentina

{mcambarieri, lvivas, ngarciam}@unrn.edu.ar

Abstract. Este artículo presenta un marco de trabajo para el desarrollo de software en gobierno digital, utilizando un enfoque basado en el modelo extractivo, el cual permite que parte de una o varias aplicaciones (productos) ya existentes puedan ser generalizados para que sean objeto de una Línea de Producto de Software (LPS). Este paradigma se centra en gestionar aspectos comunes de los productos, y su complementario, lo variable. La reutilización en el entorno LPS es planificada, y la incorporación de nuevas variantes del producto se realiza de forma sistemática y controlada; esto agiliza el desarrollo, la puesta en marcha y también el mantenimiento. La contribución es un marco de trabajo que permitirá la adopción del paradigma de LPS para el desarrollo de software en gobierno digital bajo el proceso de diseño “bottom up” extractivo que permite identificar componentes en común y una arquitectura de referencia. El marco de trabajo propuesto se valida mediante un caso de estudio referido a sistemas de información de la Universidad Nacional de Río Negro.

Keywords: Línea de productos de software, gobierno digital, reutilización, arquitectura de software.

1 Introducción

En 1968, en el Congreso sobre Ingeniería del Software que organizó el Comité de Ciencia de la OTAN, se introdujo por primera vez la idea de la reutilización. En ese mismo congreso el ingeniero de Bell Laboratories, M.D. McIlroy, afirmó que “La industria del software se asienta sobre una base débil, y un aspecto importante de esa debilidad es la ausencia de una subindustria de componentes” [1].

En el contexto actual de la Ingeniería del Software, se entiende por componente, un fragmento reemplazable de un sistema, dentro del cual se encuentran implementadas un conjunto de funcionalidades. Un componente se puede desarrollar aislado del resto del mundo.

Uno de los desafíos centrales en el desarrollo de software, justamente es la reutilización, lo que permite utilizar nuevamente uno o más artefactos realizados como parte del desarrollo de un nuevo producto o sistema. Existen diferentes técnicas

que han facilitado de alguna manera el uso de artefactos de desarrollo de granularidad cada vez mayor. La reutilización ocurra muchas veces de manera oportunista, esto es utilizar algún artefacto ya desarrollado en el nuevo sistema, de esta manera al no disponer de contextos suficientemente amplios como para detectar los elementos reutilizables y donde puedan utilizarse se desemboca en este tipo de reutilización.

Desde la perspectiva de una institución, en el contexto de gobierno digital, es deseable lograr un enfoque de reutilización sistemática con el fin de lograr beneficios asociados a utilizar los artefactos construidos previamente en cada desarrollo nuevo que se realiza. *Para que la reutilización del software fuera sistemática, los procesos de desarrollo deberían abordar la construcción colectiva de familias de productos relacionados por un dominio.*

Este trabajo explora el paradigma LPS exponiendo las diferentes estrategias para su implementación. El marco propuesto se valida mediante un caso de estudio.

El trabajo está estructurado de la siguiente manera: La Sección 2 presenta los conceptos relacionados, incluyendo Desarrollo Basado en Componentes y Línea de Producto de Software. La Sección 3 explica a través de un caso de estudio y muestra las estrategias de aplicar el enfoque extractivo en LPS. La Sección 4 presenta otros aportes científicos en esta línea de investigación y discute la contribución de este trabajo. Por último, la Sección 5 brinda conclusiones y explica los trabajos futuros.

2 Conceptos Utilizados

Las siguientes secciones presentan los conceptos básicos utilizados en este trabajo, incluyendo: Ingeniería del Software Basada en Componentes y Reutilización (Sección 2.1), Línea de Producto de Software (Sección 2.2) y Arquitectura de Software (Sección 2.3).

2.1. Ingeniería del Software Basada en Componentes y Reutilización

La Ingeniería del Software basada en componentes (CBSE: *component-based software engineering*) se ocupa del desarrollo de sistemas a partir de componentes reutilizables, para favorecer la reutilización y, de este modo, evitar la duplicación de componentes, impidiendo que el análisis, el diseño, la implementación y la prueba de la misma solución, se desarrolle una y otra vez en muchos de los productos, a diferencia del enfoque tradicional de desarrollo de software que involucra generalmente la construcción de cada una de las piezas desde cero, cada una de ellas implementada a propósito para el producto en curso. Algunas o estas piezas o elementos provienen de productos anteriores y son reutilizados con ciertos cambios [1]. CBSE, tiende a concentrar el esfuerzo en la composición de componentes de software desarrollados (que por lo general poseen una funcionalidad concreta), trata el desarrollo de sistemas a partir de la composición e integración de componentes reutilizables. Algunos de ellos, pueden requerir una vinculación con otros componentes para proveer la funcionalidad completa. En cuanto a la comunicación como la integración de los componentes con los elementos del nuevo sistema que se encuentra desarrollando se realiza entendiendo los componentes como cajas negras,

esto permite que se utilicen tal y como están disponibles y sin considerar ninguna modificación interna. [1]

A diferencia del enfoque tradicional, CBSE incluye las siguientes actividades: 1) Selección y evaluación de componentes, 2) Adaptación, 3) Composición, y 4) Evolución [1].

Según [2] existen dos factores principales cuando el enfoque que se utiliza está basado en componentes: el primero, la reutilización – *la facilidad para reutilizar componentes existentes en la creación de sistemas más complejos* – y el segundo, la evolución – *crear un sistema altamente formado por componentes facilita su mantenimiento, ya que si el diseño es correcto los cambios estarán muy localizados y los efectos tendrán poca o ninguna propagación en el resto de los componentes del sistema*. Estos factores se aplican bajo las siguientes condiciones: en primer lugar, deben existir componentes para su reutilización. Como segundo, debe existir un conjunto de componentes bien diseñados y listos para su utilización. Por otro lado, debe existir un modelo de componentes que soporte el enlazado y la interacción de componentes, esto es, debe existir un marco de referencia estándar en el que los componentes puedan existir y comunicarse. Finalmente, es necesario un proceso y la definición y diseño de arquitecturas que den soporte al CBD.

La reutilización favorece el desarrollo de herramientas de software con un alto grado de flexibilidad, haciendo uso de componentes en diversos proyectos, ampliando las capacidades, mejoras, funcionalidades y mantenimiento del software.

Hay varias definiciones del término *Reutilización de Software*. Algunas de estas definiciones son las siguientes:

“La reutilización de software es el proceso de implementar o actualizar sistemas de software usando activos de software existentes” [3]

“Reutilización de software es el proceso de crear sistemas de software a partir de software existente, en lugar de desarrollarlo desde el comienzo” [4].

Existen varias modalidades de reutilización utilizadas, con ello beneficios y costos según el enfoque que se aplique; el oportunista, se beneficia de componentes de software construidos en productos anteriores, pero que no fueron especialmente desarrollados para ser reutilizados; en el enfoque más planificado o sistemático, los componentes se construyen pensando en que serán reutilizados, lo que significa que debe plantearse el componente con mayor generalidad, y por ello en el momento de su desarrollo, mayor inversión de tiempo y recursos.

2.2. Línea de Producto de Software

La definición más comúnmente aceptada de una LPS procede del autor Clements donde “se definen las líneas del producto de software como un conjunto de sistemas software, que comparten un conjunto común de características (features), las cuales satisfacen las necesidades específicas de un dominio o segmento particular de mercado, y que se desarrollan a partir de un sistema común de activos base (core assets) de una manera preestablecida” [5].

En este contexto, una LPS permite acotar el tiempo de producción de componentes software, con altos niveles de calidad y una alta capacidad de reutilización.

En cuanto a los beneficios y costos, se puede mostrar que existe una tendencia según diferentes casos de estudios realizados, un aumento en la productividad que duplica o triplica la producción de software con respecto a los enfoques tradicionales, por lo tanto, las LPS pueden incrementar significativamente la productividad del software entendida como una reducción del esfuerzo y el costo necesario para desarrollar, mantener y poner en marcha el conjunto de productos software similares [6].

En una LPS, los activos base se benefician también de cada producto desarrollado en la línea dado que toman ventaja del análisis, diseño, implementación, planificación, prueba, estos no son únicamente una mera reutilización de código. Las líneas de productos potencian la reutilización estratégica [2], dado que en este contexto es más planificado que oportunista, [7]: en desarrollo de software tradicional, se aprecia la posibilidad de reutilizar un componente después de haberlo desarrollado; en LPS, la reutilización es planificada, de manera que se reutilizan la mayor parte de los activos base en todos los productos de la línea [6].

Como se muestra en la siguiente figura, tomada de [6], se puede observar que existe un ahorro en cuanto al desarrollo en un entorno LPS con respecto a un entorno convencional: mayor número de productos, mayor esfuerzo. Según [6], en el enfoque tradicional es muy común realizar una copia de una aplicación ya desarrollada que se parece a la se desea desarrollar, y a partir de esa copia esta evoluciona de forma independiente. Este parecido que es en cuanto a la arquitectura, tecnologías, frameworks, requisitos de análisis y diseño, etc. sirve para agilizar el desarrollo y la puesta en marcha de la aplicación en cuestión, pero no en el caso de mantenimiento.

En este contexto ambas aplicaciones se deben mantener de manera independiente, aun cuando el mantenimiento es igual. Existen 2 tipos de mantenimiento: 1- el *correctivo*, hay que depurar el código en ambas aplicaciones; 2- el *perfectivo*, realiza mejoras en las dos aplicaciones. El resultado de realizar esta copia es que los costos de mantenimiento no se reducen, si en cuanto a la reutilización de código y otros elementos, lo que significa claramente una reutilización oportunista.

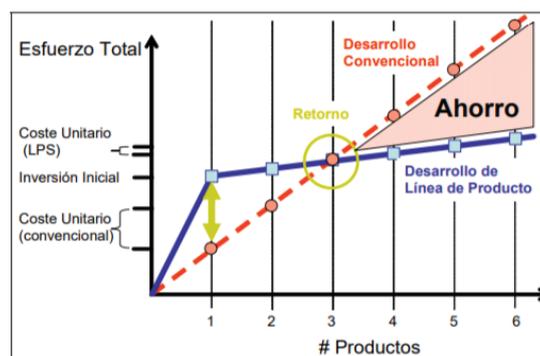


Fig. 1. Desarrollo Convencional vs. Línea de Producto [6].

Por otro lado, en un entorno de LPS, se centra en gestionar aspectos comunes de los productos, y su complementario, lo variable. La reutilización en el entorno LPS es planificada, y la incorporación de nuevas variantes del producto se realiza de forma

sistemática y controlada; esto agiliza el desarrollo, la puesta en marcha y también el mantenimiento. En cuanto al mantenimiento hay una reducción de los costos dado que son capitalizados por todos los productos, cuando se detecta un error o bug en un producto, la corrección del mismo impacta en todos los productos de la línea gracias a la existencia de este marco común que ofrece la LPS.

Podemos concluir en este punto que en un entorno convencional se centra en un único producto y cada uno tiene su mantenimiento y equipo de desarrollo, por el contrario, en LPS se aprovecha los recursos dada la semejanza entre productos, y el número de productos diferentes que podemos gestionar eficazmente.

2.2.1 Aspectos metodológicos - Estrategias

Uno de los factores para aplicar el proceso de desarrollo de la LPS depende del ámbito de la LPS. Es fundamental saber identificar y acotar la familia de productos que serán objeto de la línea, es decir, ¿Cuál es el grado de re-uso que van a soportar los activos base? Para esto se definen 3 enfoques [6]:

- **Modelo proactivo.** El enfoque a largo plazo recibe el nombre de proactivo ya que intenta adelantarse a las necesidades existentes por ejemplo dentro de cinco años, metodológicamente, es equivalente a los métodos en cascada.
- **Modelo reactivo:** Este enfoque se basa en la agilidad, va introduciendo *features* conforme las necesidades del negocio van cambiando. Metodológicamente, se equipara con los métodos ágiles.
- **Modelo extractivo.** Este modelo parte de una o varias aplicaciones ya existentes para generalizarlas hacia una LPS. Tal vez sea éste el enfoque más realista en la medida en que la mayoría de las organizaciones raramente disponen de los recursos para acometer desde cero la LPS, y, sin embargo, sí disponen de aplicaciones cuyo mantenimiento les impulsa a embarcarse en las LPS.

2.2.2 Aspectos metodológicos - Procesos

Una LPS comprende esencialmente 2 etapas:

La ingeniería de dominio- ID (core asset development, denominada por Clements [5]) es la encargada de diseñar los componentes de software que, de acuerdo a su trazabilidad, hacen parte del dominio común y que son sujetos a reutilización. En esta etapa se definen las similitudes y variantes del modelo de dominio en cada una de las fases. Como producto final en esta etapa se considera un componente altamente reusable y flexible que permita a partir de su definición, la consecución de nuevos modelos y artefactos.

La Ingeniería de Aplicación - IA (product development, denominada por Clements [6]) se encarga de articular los artefactos definidos en la etapa de dominio a partir de una plataforma específica. El objetivo de esta etapa consiste en lograr la integración de artefactos bajo uno o varios sistemas, logrando con ello un alto grado de

reusabilidad de sus componentes a partir de las especificaciones, la definición y desarrollo, documentación e interoperabilidad entre componentes.

2.2.3 Gestión en LPS

La gestión juega un papel fundamental para el establecimiento de una línea de productos. La división formal en dos etapas: Ingeniería de Dominio e Ingeniería de Aplicación es complementada con una parte dedicada a la gestión, por algunos autores como Clements. Esta actividad juega un papel fundamental en el éxito de la línea de producto, ya que no sólo es la encargada de asignar recursos, sino de coordinarlos y supervisarlos. Esta gestión se realiza tanto en el nivel técnico (gestión de productos concretos) como en el nivel organizativo. Los artefactos de gestión creados como planificaciones también forman parte de los elementos comunes (*core assets*) [6].

2.3. Arquitectura de Software

La arquitectura de software y entornos de Trabajo (*Framework*) utilizados por los productos (aplicaciones) que se definen en la sección 3 se muestra en la figura 2. En el caso de una aplicación empresarial puede dividirse en tres capas lógicas bien definidas [9]: 1) la capa de presentación, 2) la capa de negocio y 3) la capa de persistencia. El principio para la separación en capas es que cada una esconde su lógica al resto y solo brinda puntos de acceso a dicha lógica. En la capa de presentación los objetos trabajan directamente con las interfaces de negocios, implementando el patrón arquitectónico *Model-View-Controller* [9]. En este, el modelo (*Model*) es modificable por las funciones de negocio, siendo estas solicitadas por el usuario, mediante el uso de un conjunto de vistas (*View*) que solicitan dichas funciones de negocio a través de un controlador (*Controller*), que es quien recibe las peticiones de las vistas y las procesa. La capa de negocio está formada por servicios implementados por objetos de negocio. Estos delegan gran parte de su lógica en los modelos del dominio que se intercambian entre todas las capas. Finalmente, la capa de persistencia facilita el acceso a los datos y su almacenamiento en una base de datos.

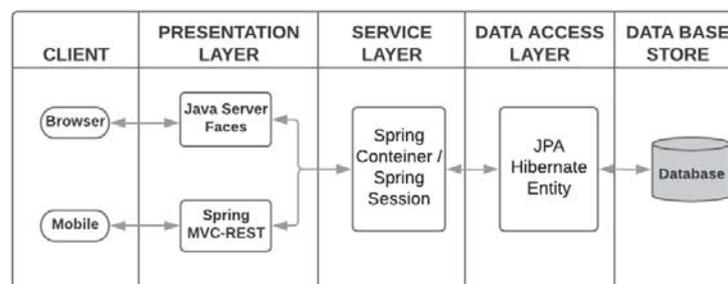


Fig.2 Arquitectura en capas.

3 Caso de estudio

El Departamento de Soluciones Informáticas de la UNRN, se encuentra actualmente avanzando en pos de mejorar las gestiones de los agentes Docentes y No Docentes incrementando el número de procedimientos digitalizados, para ellos se propuso inicialmente desarrollar en etapas las siguientes aplicaciones:

Aplicación	Descripción
Legajos Digitales (1er etapa)	Es un sistema web que permite generar, administrar, mantener y consultar legajos digitales (legajos de empleados, alumnos, pacientes en los hospitales escuela de Odontología y Veterinaria, etc.) definiendo cualquier estructura que se disponga de acuerdo a las características que tengan los tipos de legajos a administrar. El sistema permite incorporar, buscar o consultar legajos y la documentación incluida en los mismos. Además, se implementan servicios web para consultas de información de legajos de agentes.
Estructura Organigrama Funcional (2da etapa)	Es un sistema de gestión web que ayuda a crear y definir con claridad las funciones de las diferentes unidades administrativas de la UNRN. Esta herramienta permite facilitar la coordinación institucional, dotando a la institución de un instrumento administrativo que determine la forma de la estructura orgánica básica, la misión y función de cada una de las unidades administrativas, sus relaciones de dependencia, supervisión y coordinación; así como, el detalle de las principales funciones de las unidades.
Licencias Digitales (3er etapa)	Este sistema web permite que los agentes docentes y no docentes de la UNRN puedan ingresar para generar sus licencias y avisos de ausencia a través de internet pensado como una herramienta para agilizar y tener la información disponible para la dirección de RRHH.

El desarrollo de software bajo el paradigma LPS está centrado en la reutilización o derivación de activos base. Un activo base de software puede estar compuesto por; arquitecturas, patrones de diseño, componentes, documentación, especificaciones y requisitos del sistema, planes y pruebas entre otras [5], lo que permite hacer uso de cada uno de estos aspectos para desarrollar la familia o conjunto de productos.

El concepto de línea de productos que utiliza una arquitectura base y un conjunto de elementos reutilizables es directamente aplicable a las organizaciones o instituciones, que desarrollan productos o sistemas. También es posible obtener ventajas de la puesta en marcha de una línea de productos en organizaciones más orientadas a proyectos o incluso en los departamentos de informática que dan soporte a los sistemas informáticos como es el caso de estudio presentado [2].

Según [8] el interés por las LPS surge en el campo de la reutilización del software, cuando se observa que resulta mucho más provechoso reutilizar diseños arquitectónicos completos en lugar de componentes software individuales. Es posible que pueda aplicarse un enfoque LPS e identificar los activos base, que estarán presentes en todos o en la gran mayoría de todos los productos (aplicaciones) que se

implementen, a modo de ejemplo se puede mencionar algunos de los activos base comunes a los tres productos como muestra la figura 3:

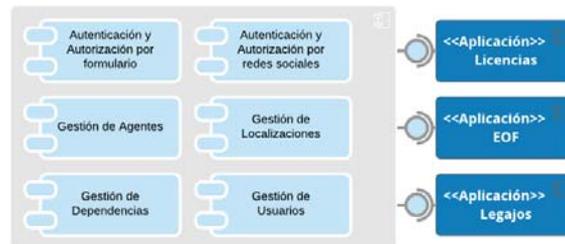


Fig.3 Activos base para los tres productos.

La etapa de Ingeniería de Dominio abarca el análisis y la identificación del ámbito de aplicación de la LPS, que incluye la captura de todo el conocimiento sobre el dominio y el negocio, a través del modelado de partes comunes y variables, que están presentes en una arquitectura de referencia, la cual es una arquitectura de software de alto nivel de abstracción, genérica y basada en un esquema instanciable o modelo de variabilidad, que dirige la derivación de productos concretos de la familia LPS.

Dada la arquitectura propuesta en la sección 2.3 y los componentes comunes que se plantearon anteriormente, es posible, como estrategia, aplicar el enfoque LPS en el caso de estudio, utilizando el modelo extractivo definido en la sección 2.2.1. El enfoque extractivo comienza con un conjunto de productos o aplicaciones ya desarrolladas y de manera iterativa e incremental se refactorizan los elementos arquitectónicos. En este caso se realizó un análisis de las aplicaciones existentes, a partir de los cuales, mediante un análisis de similitud semántica de componentes, se redefinen las arquitecturas de las aplicaciones o productos a partir de la información o documentación disponible, determinándose componentes comunes y variantes.

Es importante determinar la aplicabilidad de la LPS en este contexto, bajo el proceso de diseño “bottom up” (de abajo hacia arriba), extractivo, repetible y sistemático, para construir una arquitectura LPS de referencia. [10]. La arquitectura de una LPS es una arquitectura de software genérica, representa la estructura de toda la familia de productos y no solamente la de un producto en particular. Describe los aspectos comunes y variables de una familia de productos de software.

La arquitectura LPS debe ser instanciada cada vez que se desarrolla un producto de la línea.

Una contribución importante del trabajo es definir un proceso de Ingeniería del Software hacia una LPS en el dominio del conjunto de aplicaciones en el contexto de gobierno digital con el fin de: 1- Implementar un repositorio LPS de “core assets” con componentes comunes y variables especificados a un alto nivel de abstracción; 2- diseñar la arquitectura LPS para los sub-dominios, mediante un enfoque ascendente o “bottom-up” basado en el estudio de aplicaciones existentes ya desarrolladas, y 3- definir estrategias para instanciar la arquitectura obtenida para cada sub-dominio, a partir de los “core assets” identificados como reutilizables.

4 Trabajos Relacionados

Este trabajo está basado en tareas de investigación que analizaron los aportes científicos que se encuentran en esta misma línea. En particular, se investigaron trabajos relacionados con la implementación de LPS en gobierno digital. Algunos de los trabajos más importantes se discuten a continuación.

Fajar, AN y sus colegas [11] realizaron una investigación sobre la situación del cambio dinámico en el gobierno indonesio, esta condición afecta al desarrollo de software lo que causa reemplazo, modificación y mejora. Existen algunas aspectos comunes y variables del software en el gobierno indonesio, por tanto, para gestionarlo, presentan un marco de trabajo denominado ZumaFramework (ZEF) para reducir la complejidad del mismo. Por otra parte, el autor indica que el desarrollo de sistemas en las instituciones gubernamentales es para la mejora de los procesos de negocio que pueden ayudar a hacer una buena gobernanza. Las características de similitud y variabilidad del software podrían lograrse a partir de los procesos de negocio. Esto puede relacionarse con el costo y el tiempo asignado para el desarrollo del sistema. Por esto, propusieron un framework SPLGS para desarrollar sistemas bajo el paradigma LPS en el gobierno de Indonesia [12]. En [13] la investigación muestra que las organizaciones están interesadas en la reducción de costos asociados al desarrollo de software, y es esencial utilizar herramientas automatizadas y un proceso sistemático de reutilización. El trabajo presenta un entorno de generación de CMS, en el contexto de LPS automatizada a través de frameworks, generadores de aplicaciones y repositorios de reutilización. Finalmente, [14] presenta el enfoque de Ingeniería de la Línea de Producción de Software (en inglés, Software Production Line Engineering- SPLE) en el que las organizaciones gubernamentales o universitarias sin fines de lucro y con escasos recursos, son cada vez más solicitadas para desarrollar aplicaciones personalizadas. Estas aplicaciones a menudo tienen arquitecturas y cumplen requisitos muy similares. Dadas estas características, proponen una metodología SPLE a una serie de aplicaciones.

5 Conclusiones y Trabajo Futuro

Este trabajo presentó un marco de trabajo para el desarrollo de software basado en LPS. La ventaja radica en potenciar la reutilización estratégica en el desarrollo de software en gobierno digital. La adopción de LPS implica un cambio de mentalidad en el desarrollo de software porque permite centrarnos en construir y gestionar los aspectos comunes de los productos, identificando los componentes y definiendo arquitecturas de referencia de alto nivel de abstracción. En este entorno es posible lograr una reutilización más planificada, dado que las incorporaciones de nuevas variantes de los productos se realizan de forma sistemática y controlada respecto al enfoque de desarrollo tradicional que plantea la reutilización más oportunista.

Desde la perspectiva de la Universidad que se encuentra en un proceso de transformación digital y despapelización de los procesos administrativos, es deseable contar con técnicas y metodologías de desarrollo donde pueda ser posible aplicar un enfoque de reutilización sistemática con el fin de lograr los siguientes beneficios:

utilizar los artefactos construidos previamente en cada desarrollo nuevo, aumentar la productividad de los recursos dada la semejanza entre productos y gestionar eficazmente el conjunto de productos diferentes. Por otro lado, lograr una reducción de costos en cuanto a pruebas y mantenimiento que será capitalizado por todos los productos de la línea gracias a la existencia de este marco común que ofrece la LPS.

Trabajos a futuro incluyen extender el marco de trabajo aplicando otras técnicas y metodologías de desarrollo en las que la reutilización juega un papel esencial, como la Ingeniería del Software Dirigida por Modelos (*Model-Driven Engineering- MDE*), este paradigma dirigido por los modelos, es decir, el sistema de modelos tiene suficiente detalle para permitir la generación de código de un sistema completo a partir de los modelos propios, esto es, construir modelos que puedan ser directamente compilados y ejecutados, “el modelo es el código” [15]. Por otro lado, explorar los Lenguajes Específicos de Dominio (*Domain Specific Language- DSL*) y la Programación Generativa que de manera similar al desarrollo de LPS ofrece la construcción, con un enfoque de reutilización proactivo, de familias de productos relacionados con algún dominio particular, en lugar de productos independientes.

Referencias

1. Usaola, M.: Desarrollo de software basado en reutilización. UOC. Formación de Postgrado. 2013
2. Peñalvo, F., et al.: Líneas de Productos, Componentes, Frameworks y Mecanos. Informe Técnico, Departamento de Informática y Automática - Universidad de Salamanca (2002)
3. Sodhi, J., Sodhi,P: Software reuse: Domain analysis and design process. McGraw-Hill. (1999)
4. Sametinger, J.: Software engineering with reusable components. Springer Verlag, (1997).
5. Clements,P., Northrop,L.: Software Product Lines: Practices and Patterns. Addison-Wesley (2001).
6. Díaz O., Trujillo, S.: Fábricas de Software: experiencias, tecnologías y organización” (2ª edición). Editorial Ra-Ma, (2010).
7. Trujillo, S., Batory, D. y Díaz, O: Feature Oriented Model Driven Development: A Case Study of Portlets. International Conference on Software Engineering (ICSE) (2007).
8. Gomaa, H.: Designing Software Product Lines with UML. From use cases to pattern based software architectures. Addison-Wesley (2005).
9. Fowler, M. “Patterns of Enterprise Application Architecture”, Addison-Wesley, (2002).
10. Losavio F., Ordaz O., Esteller V: Refactoring-Based Design of Reference Architecture, RACCIS, Vol 5, No 1, pp. 32-48, ISSN 2248-7441. (2015)
11. Fajar A.N., Shofi I.M., "Reduced software complexity for E-Government applications with ZEF framework". TELKOMNIKA, Vol.15, No.1, pp. 415~420 ISSN: 1693-6930 (2017).
12. Fajar A.N., Shofi I.M.: Development of SPL Government System with Ontology Web Language, 4th Conference International Conference on Cyber and IT Service Management. (2016).
13. Lima V.M.A.D., Marcacini R.M., Lima M.H.P., Cagnin M.I., Turine M.A.S.: A generation environment for front-end layer in e-government content management systems. 9th Conference Latin American Web Congress. (2014).
14. Diwan P., et al: Applying software product line engineering in building Web portals for supercomputing services. Proceedings of the 28th Conference Annual ACM Symposium on Applied Computing. (2013).
15. Oquendo, F.: p-Method: A Model-Driven Formal Method for Architecture-Centric Software Engineering. ACM SIGSOFT Software Engineering Notes. Volume 31 Number 3. (2006).

Minería de Emociones y Análisis Visual Aplicado a la Red Social Twitter

Florencia Marrocchi, Carolina Rapetti, Ana Maguitman, Elsa Estevez

Departamento de Ciencias e Ingeniería de la Computación
Universidad Nacional del Sur, Bahía Blanca, Argentina
{[flor.marro_rapeticarolina](mailto:flor.marro_rapeticarolina@gmail.com)}@gmail.com, {agm, ece}@cs.uns.edu.ar

Resumen. Las emociones forman un aspecto muy importante en nuestras vidas. Lo que hagamos o digamos reflejan de algún modo ciertas emociones. Analizar las emociones de un ser humano es fundamental para comprender su comportamiento, y puede lograrse a través de datos como texto, voz, expresiones faciales, entre otras. El cálculo de emociones consiste en la tarea de analizar texto para obtener emociones implícitas en él. Con la llegada de las redes sociales, analizar el texto disponible en estos sitios es una tarea muy atractiva. Analizar estos datos a través de Internet significa que estamos abarcando opiniones recientes de personas en distintos lugares del mundo sobre cualquier tópico deseado. Este documento resume un trabajo de investigación y desarrollo relacionado con la minería de emociones basado en el modelo emocional de Plutchik y enfocado en un escenario de gobierno digital. La contribución del mismo es presentar una extensión del modelo de Plutchik y mostrar una aplicación de software que lo aplica.

Palabras clave: Minería de Emociones; Análisis de Sentimientos; Rueda de Emociones de Plutchik; Gobierno Digital; Participación Ciudadana

1 Introducción

Una de las redes sociales con más usuarios en el mundo hoy en día es Twitter, un servicio de microblogging que permite a sus usuarios enviar y publicar mensajes breves, generalmente solo de texto. Twitter es una fuente de información al instante, cualquier evento adquiere relevancia en cuestión de minutos, incluso las noticias se difunden antes que en cualquier medio periodístico. Según el estudio “Digital 2019: Q2 Global Digital Statshot” [1], en abril de 2019 esta red social contaba con unos 330 millones de usuarios activos. Su fortaleza se basa en la difusión de información en tiempo real, y en que es mayoritariamente pública, lo que la convierte en una fuente importante de datos de gran variedad. A partir de estos grandes volúmenes de datos, muchas organizaciones realizan escuchas sociales y monitoreos que si son procesados y transformados de manera correcta, pueden convertirse en una fuente de información útil para las mismas. El análisis de emociones es una de las opciones elegidas al momento de obtener una visión pública sobre un tema, producto o servicio en

particular, a partir de personas reales que están hablando del mismo de manera activa en línea.

El objetivo de este trabajo es extender el modelo de Plutchik y aplicarlo en una aplicación concreta. Para ello, se agregaron palabras en castellano a las ya provistas por el modelo y se desarrolló una aplicación web que permite generar una consulta online a Twitter según distintos tópicos de especial interés. A partir de la consulta, se obtienen los tweets en tiempo real, se realiza un análisis y clasificación de las emociones de los usuarios, y se las visualiza gráficamente. Finalmente, se utiliza la herramienta desarrollada para crear un escenario específico sobre gobernanza digital en el cual podamos visualizar y analizar los resultados obtenidos para un tema de debate en auge. Asimismo se incorporan características adicionales para incentivar al usuario, como la sugerencia de temas populares en Twitter en ese momento y una muestra de los tweets con los que se realizó la consulta.

El resto de este documento se estructura de la siguiente forma. La Sección 2 explica conceptos relacionados a minería de sentimientos, mientras que la Sección 3 presenta el modelo de Plutchik. La Sección 4 explica el diseño de la herramienta, y la Sección 5 las herramientas utilizadas para el desarrollo. La Sección 6 discute el uso de la herramienta en el contexto de las elecciones. Por último, la Sección 7 resume conclusiones y el trabajo futuro.

2 Conceptos Preliminares – Análisis de Sentimientos y Emociones

Para poder comprender lo narrado en este documento es necesario, primero, analizar temas de interés al trabajo realizado. En esta sección se tratará el tema “Minería de Emociones” aplicado al modelo de Plutchik [2], el cual sostiene que hay ocho emociones humanas primarias: alegría, confianza, miedo, sorpresa, tristeza, repugnancia, enojo y anticipación.

El procesamiento de lenguaje natural se utiliza entre otras cosas para el análisis de emociones y el análisis de sentimientos. Aunque el propósito de estos análisis sea similar, se diferencian en la identificación de los estados, por un lado el análisis de sentimiento detecta la polaridad del usuario, mientras que el análisis de emociones identifica específicamente las emociones que expresa un usuario según un modelo de emociones seleccionado.

Para realizar el análisis de emociones a través de texto, es necesario realizar un procesamiento de la información del mismo. Las técnicas que se emplean para diseñar e implementar un clasificador emocional se pueden clasificar en dos enfoques más una combinación de ellos. El primer enfoque es el basado en un léxico. Un léxico emocional es un depósito de conocimiento que contiene unidades textuales anotadas con etiquetas emocionales. Se basan en recursos léxicos como léxicos, bolsas de palabras u ontología. El segundo enfoque es el basado en Machine Learning. Este enfoque utiliza algoritmos de aprendizaje automático para entrenar el sistema y mapear una función para la clasificación futura de emociones. Se basa en las características lingüísticas que elegimos para entrenar la máquina [3]. Este enfoque se

adapta a los cambios de dominio ya que aprende rápidamente nuevas características a partir del corpus (entrada de texto). El tercer enfoque es el híbrido. Este realiza una combinación de los dos primeros enfoques para obtener resultados de mayor precisión.

Para poder analizar las emociones en un texto, se debe hacer un preprocesamiento del mismo. El procesamiento previo reduce la dimensionalidad al eliminar los datos no deseados (ruido). El preprocesamiento en el texto se puede dividir en tres submódulos: tokenización, eliminación de ruido y normalización del texto, como se explica debajo.

En la *Tokenización* el texto se divide en tokens que representan palabras, oraciones o párrafos. La tokenización se realiza de acuerdo al limitador que separa una palabra, oración o párrafo de sí mismo. La división que genera la tokenización elimina las interrelaciones y los símbolos que le dan sentido al texto. En Twitter, por ejemplo, se consideran algunas situaciones especiales, ya que los tweets suelen contener menciones (nombre de usuario precedido por el símbolo '@') o links URL, los cuales deben ser considerados como tokens que serán removidos. Los tweets suelen contener también hashtags, pero a diferencia de los casos anteriores, éstos representan información importante a ser considerada, por lo que se conservarán como tokens y, por último, se hace la eliminación de *Stopwords*.

3 Modelo de Plutchik

Al momento de investigar sobre análisis de emociones en Twitter y herramientas públicas que permitan hacerlo, hemos encontrado que hay pocos trabajos que utilicen el idioma español. Por este motivo, se vio como una gran oportunidad crear una aplicación pública, que haciendo uso de tweets únicamente en español realice análisis de emociones para cualquier tema de interés que se mencione en dicha red social.

Por otra parte, este idioma generó algunas limitaciones. Fue necesario descartar el uso de varias herramientas para el procesamiento de lenguaje que no proporcionaban la capacidad de realizar operaciones sobre este idioma. También fueron descartadas fuentes de información que podrían servir para el análisis de emociones o para utilizar con alguna técnica de machine learning que únicamente hallamos en idioma inglés.

3.1 Extensión al Modelo

En función de la metodología elegida para el análisis (basada en lexicón), se requiere que los datos etiquetados estén en el idioma elegido. La fuente utilizada como lexicón originalmente se encontraba en idioma inglés, con una traducción automática al español, donde muchas de las palabras no se habían traducido correctamente.

El lexicón original (NRC Affect Intensity Lexicon) [4] se trata de una tabla excel con unas 14182 palabras, traducidas en 105 idiomas, donde cada una tiene diez atributos correspondientes a las ocho emociones de Plutchik más dos de polaridad. A

partir del lexicón inicial, se realizó el trabajo de traducir correctamente al español el conjunto dado de palabras, quitando las traducciones en los demás idiomas y dejando únicamente la lista de palabras con sus correspondientes 10 emociones. El lexicón final es entonces una traducción al español del NRC Affect Intensity Lexicon que quedará disponible como herramienta para futuros trabajos. Adicionalmente, se extendió el modelo con la clasificación de emojis, como se explica en la Sección 4.2.

4 Diseño de la Herramienta

El desarrollo de la aplicación web y el proceso de minería de texto para el análisis de emociones se realizaron utilizando los frameworks Angular [5] y Flask [6] respectivamente, siguiendo las siguientes etapas.

4.1 Búsqueda y Obtención de Tweets

La fuente de información que se utiliza en el proyecto son los tweets obtenidos de la base de datos de Twitter. Utilizando la librería Tweepy [7] se obtuvieron los tweets filtrando por idioma, palabras claves, fechas límite y cantidad de tweets. Además se incorporó la restricción de no obtener retweets, ya que estos implicarían información repetida que ya fue analizada. Una limitación de la versión gratuita de la API de Twitter [8] es que solo permite obtener tweets de hasta una semana atrás y limita la cantidad de tweets que se pueden obtener.

4.2 Incorporación de Emojis al Lexicón

Los emojis son símbolos muy utilizados en mensajes electrónicos y sitios web. Se trata de una representación de una emoción utilizando caracteres. Estos emojis representan una emoción en el tweet y es por eso que se decidió incluirlos en el análisis. Para ello se buscó el significado de cada emoji [9] y se incorporó al lexicón respetando el formato original. Finalmente, el lexicón usado quedó compuesto por todas las palabras iniciales y la unión de todos los emojis con sus respectivas emociones.

4.3 Pre-procesamiento de Texto

Con la lista de tweets obtenidos, se procedió a eliminar información que no fuera relevante para simplificar el análisis de emociones a realizarse por palabra, realizando:

Tokenización utilizando la librería NLTK [10], teniendo en cuenta los casos especiales como las expresiones propias de Twitter (hashtags, menciones y links URL).

Procesos no incluidos. Es una práctica común en las redes sociales utilizar mayúsculas cuando se quiere dar ímpetu a una expresión, como si se tratara de “aumentar” el tono. Sin embargo, se decidió ignorar esto y solo tener en cuenta el significado de la palabra para asignar la emoción correspondiente.

Otra situación común en los textos extraídos de redes sociales es que no respeta las reglas ortográficas del lenguaje. Sin embargo, hacer esta corrección podría desvirtuar la información obtenida transformando palabras que no están mal escritas, sino que simplemente no forman parte de la lengua española, lo que daría como resultado una emoción que no se corresponde con la palabra original.

Por último, la cantidad de retweets que tiene un tweet podría brindar información de cuántas personas están de acuerdo con ese contenido. Pero a su vez, Twitter permite escribir al retweetear un tweet (“citar tweet”) por lo que es muy usual que los usuarios citen un tweet para explicar por qué no están de acuerdo con el mismo. Por este motivo, se decidió no obtener información a partir de la cantidad de retweets.

4.4 Análisis de Emociones

Habiendo finalizado el pre-procesamiento de los tweets, se tienen las listas de tokens para cada tweet, las cuales son procesadas utilizando el modelo de Plutchik mencionado anteriormente. La respuesta del procesamiento del texto (análisis de emociones) da como resultado entonces las emociones promedio para el conjunto de tweets obtenidos, las emociones para cada tweet, y los tweets utilizados para ese análisis. Con esta información se realiza la representación gráfica, como se explica en la siguiente sección.

4.5 Análisis Visual

La principal información a representar en la aplicación web consiste en las ocho emociones de Plutchik con sus respectivos promedios de apariciones encontrados en la búsqueda, para esto se utiliza un gráfico de radar, conformado por ocho emociones para los ejes, cada una representada con el color que le corresponde según Plutchik.

Adicionalmente se creó un gráfico de barras horizontal para representar la polaridad de la información obtenida, calculada de la misma forma que las emociones.

4.6 Aplicación Web

La aplicación realizada consiste de una app web single-page que contiene un formulario de búsqueda en el cual es posible ingresar los *hashtags a buscar*, donde el usuario tiene la posibilidad de ingresar hasta cinco hashtags de búsqueda. Luego de ingresar el primero, si así lo desea, tiene la posibilidad de adjuntar otro. La siguiente opción es la posibilidad de elegir si la búsqueda con más de un hashtag será realizada con un conjunto de tweets que contengan todos estos (conjunción) o un conjunto de tweets que contengan al menos uno (disyunción). En tercer lugar, dispone de la

posibilidad de limitar la fecha límite de búsqueda. Por último, debido a límites establecidos por la herramienta utilizada, hay un máximo de 3000 tweets que se pueden obtener en una consulta. La búsqueda se puede personalizar entonces para obtener resultados de entre 1 y 3000 tweets.

Además del formulario de búsqueda, se pueden ver dos gráficos de representación de información. El primero es un gráfico de tipo radar el cual muestra el porcentaje total de emociones obtenidas a partir de la búsqueda y, el segundo es un gráfico de barras horizontal que muestra la polaridad de los tweets obtenidos.

Una vez realizada la búsqueda, junto a los resultados de la misma se muestran los tweets, con el color de la emoción predominante en el mismo, a partir de los cuales se realizó el análisis.

Como característica adicional, se puede visualizar una lista con los 5 trending topics del momento, con la finalidad de sugerir al usuario temas de búsqueda.

Finalmente, ubicada en el menú principal de la aplicación, se encuentra la sección de ayuda, donde se explica brevemente cómo realizar una búsqueda.

4.7 Ejemplo de procesamiento de un tweet

Para comprender el proceso del análisis de emociones tomamos un tweet de ejemplo y analizamos cómo fue el procesamiento hasta llegar al resultado final.

Sea un tweet obtenido durante una búsqueda : “#CIOSummit2019 | Líderes de TI y de Transformación Digital compartirán experiencias de buenas prácticas e implementación de acciones que permiten una interacción entre los ciudadanos y el Estado más eficiente. ¡Estamos construyendo el legado! <https://t.co/TS6MtcPayg> #Colombia40 <https://t.co/JBCUR8xSNp>”

Los pasos realizados sobre el texto son:

1. Tokenización. El resultado obtenido es: “[‘líderes’, ‘de’, ‘ti’, ‘y’, ‘de’, ‘transformación’, ‘digital’, ‘compartirán’, ‘experiencias’, ‘de’, ‘buenas’, ‘prácticas’, ‘e’, ‘implementación’, ‘de’, ‘acciones’, ‘que’, ‘permiten’, ‘una’, ‘interacción’, ‘entre’, ‘los’, ‘ciudadanos’, ‘y’, ‘el’, ‘estado’, ‘más’, ‘eficiente’, ‘estamos’, ‘construyendo’, ‘el’, ‘legado’]”
2. Búsqueda de cada token en el lexicón y obtención de clasificación.
3. Se realiza una sumatoria de resultados encontrados para cada emoción, cada token puede representar cero o más emociones: {'Positive': 2.0, 'Negative': 0.0, 'Anger': 0.0, 'Anticipation': 1.0, 'Disgust': 0.0, 'Fear': 0.0, 'Joy': 0.0, 'Sadness': 0.0, 'Surprise': 0.0, 'Trust': 1.0}
4. El mismo proceso se repite para todos los tweets obtenidos en la búsqueda. Finalmente se realiza un promedio de todos los valores de emociones encontradas según la cantidad de tweets analizados. Esta información es luego representada en un gráfico de radar.

5 Uso de la Aplicación en Gobierno Digital

La minería de emociones aplicada a una red social proporciona una gran oportunidad para analizar opiniones de usuarios con respecto a diferentes temas. El auge de Twitter en la actualidad y los diferentes debates de opinión que se generan en torno al contexto político del país, proporcionan una gran oportunidad para estudiar la influencia de la opinión de los usuarios sobre la imagen de los candidatos, las propuestas y el desarrollo de las campañas durante el período previo a las elecciones.

Entonces, realizar un análisis de emociones sobre las opiniones y comentarios que los usuarios de redes sociales expresan diariamente en relación a los hechos políticos que acontecen previo a las elecciones presidenciales, podría ser de gran utilidad para evaluar, en principio, de qué manera éstos influyen o afectan en la intención de voto o en el resultado de encuestas electorales oficiales, y posteriormente, determinar si se condicen con los resultados de las elecciones.

Se generaron diferentes escenarios relacionados al contexto político actual en Argentina cambiando los parámetros de entrada al sistema. Se mantuvo siempre el mismo rango de fechas para comparar en el mismo período de tiempo. Este rango se estableció desde una semana atrás a la fecha actual, esto es desde 10/07/2019 al 17/07/2019. También se mantuvo en 1000 la cantidad de tweets obtenidos.

5.1 Búsqueda de un Hashtag

#Elecciones2019. Luego de generar la búsqueda obtenemos como resultado el gráfico en la Figura 1 donde la emoción predominante es confianza, seguido de una mezcla de tristeza y anticipación.

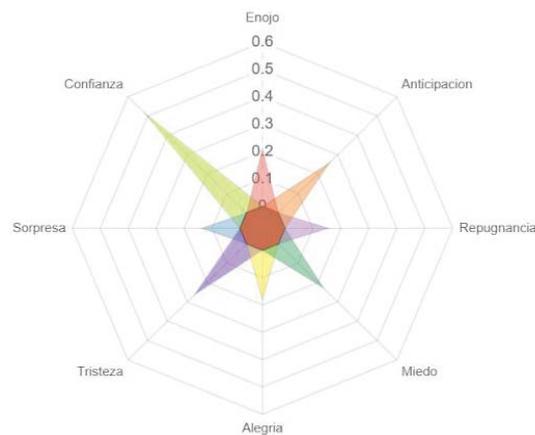


Fig. 1. Visualización de emociones con la búsqueda “#Elecciones2019”.

5.2 Búsqueda de Hashtags en Disyunción (OR)

Mediante la función de agregar hashtags y la opción de disyunción, se realizaron búsquedas de dos o más hashtags donde los resultados obtenidos contengan alguno de los hashtags buscados. Los resultados dependen de qué tan relacionados están estos hashtags ya que son un promedio total de las emociones de los tweets obtenidos y si corresponden a temáticas no relacionadas es probable que se obtenga un resultado distorsionado. La búsqueda consistió de los hashtags “#cambiamos” o “#elecciones”, donde se puede visualizar que la emoción predominante sigue siendo Confianza, pero crece el valor para Tristeza y disminuye para Anticipación de 0,3 a 0,2.

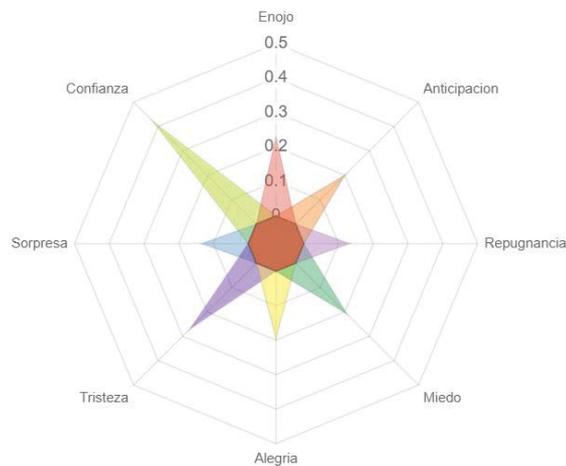


Fig. 2. Escenario búsqueda “#cambiamos” o “#elecciones”.

En la Figura 2 se pueden visualizar ejemplos de diferentes tweets con su respectiva emoción predominante representada por el color del texto. En el primer texto la emoción que predomina es Enojo (color rojo del texto) y al leerlo se puede comprobar su concordancia. En el segundo caso el que el usuario que escribe utiliza la ironía, no contemplada en el análisis de emociones realizado, por lo cual la emoción predominante (confianza) no concuerda con la intención del escritor.

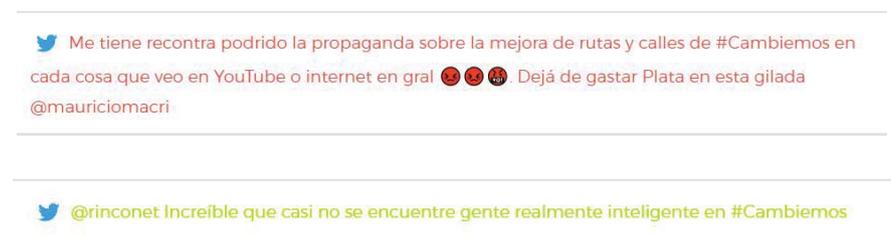


Fig. 3. Ejemplo de tweets con colores de emoción predominante.

5.3 Búsqueda de Hashtags en Conjunción (AND)

Se utilizó la función de agregar hashtags para realizar una búsqueda compuesta de hashtags con la opción conjunción. En la búsqueda realizada, “#cambiamos” y “#elecciones”, se observa que la emoción predominante con gran diferencia sigue siendo Confianza, pero se suma la emoción Miedo, que se ubica apenas por debajo de Tristeza. Al analizar una conjunción, se puede afirmar que el resultado corresponde al conjunto, a diferencia de la disyunción, donde el resultado puede verse afectado por opiniones diferentes sobre cada uno de los temas individualmente.

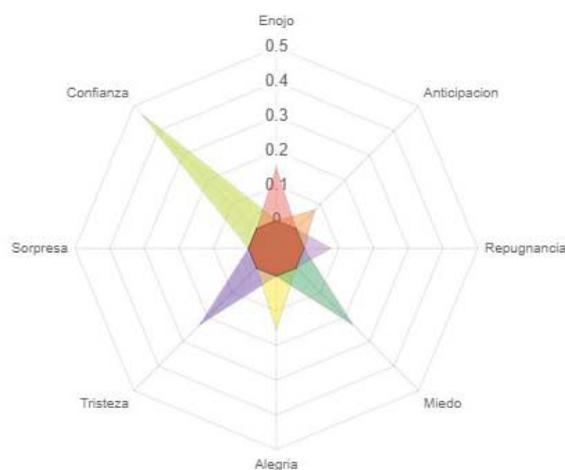


Fig. 4. Escenario de búsqueda “#cambiamos” y “#elecciones”.

6 Conclusiones y Trabajo Futuro

Luego de haber finalizado las tareas propuestas, concluimos que se lograron los resultados esperados teniendo en cuenta que el análisis se hace sobre un lenguaje informal, con muchos errores de ortografía y abreviaciones que dificultan el procesamiento. Estos resultados se pueden mejorar implementando herramientas de lenguaje natural más complejas que profundicen el análisis. Por último, si bien se utilizó la aplicación considerando el escenario actual de las elecciones, la herramienta desarrollada permite analizar las emociones de los usuarios de Twitter sobre cualquier tema de interés para la sociedad.

Nuestro trabajo futuro se focaliza en las siguientes líneas. Una mejora es trabajar con la API paga de Twitter la cual permite acceso a tweets más antiguos y a mayor cantidad de los mismos. Así se podría realizar análisis en distintos períodos de tiempo y, por lo tanto, comparar un tópico en la actualidad con el mismo tópico en cierto período de tiempo atrás y analizar resultados. Otra mejora interesante a explorar es la

de incorporar técnicas de Machine Learning o aprendizaje supervisado. Durante el desarrollo del proyecto se buscó la forma de incorporar alguna técnica pero esto no fue posible ya que para aprendizaje automático, la técnica que más se adecuaba al proyecto, necesitaba un corpus de información etiquetada en español. Si bien existen corpus en idioma inglés, no fue posible encontrar un corpus en español para el enfoque de análisis de emociones adoptado en este trabajo. Esto representa una importante limitación a la hora de desarrollar métodos de aprendizaje supervisado en este idioma debido a que se trata de conjuntos de entrenamiento muy grandes que requieren ser traducidos “a mano” para no perder la esencia de las expresiones que pueden variar de un idioma a otro.

Referencias

- [1] Kemp, S.: Digital 2019: Q2 Global Digital Statshot. DataReportal (2019). <https://datareportal.com/reports/digital-2019-q2-global-digital-statshot> [Accessed 17 Jul. 2019].
- [2] R. Plutchik. The Nature of Emotions: Human emotions have deep evolutionary roots, a fact that may explain their complexity and provide tools for clinical practice. *Am. Sci.*, vol. 89, no. 4, pp. 344–350 (2001).
- [3] Moine, J., Haedo, A., Gordillo, S.: Estudio comparativo de metodologías para minería de datos. WICC (2011). https://digital.cic.gba.gob.ar/bitstream/handle/11746/3525/11746_3525.pdf-PDFA.pdf?sequence=1&isAllowed=y.
- [4] Mohammad, S., Turney, P.: Emotions evoked by common words and phrases: using mechanical turk to create an emotion lexicon. *Proceedings of the NAACL HLT 2010 Workshop on Computational Approaches to Analysis and Generation of Emotion in Text*, 26–34 (2010).
- [5] Angular, <https://angular.io/>.
- [6] Flask, <https://palletsprojects.com/p/flask/>.
- [7] Tweepy, <https://www.tweepy.org/>.
- [8] Twitter API, <https://developer.twitter.com/en/products/tweets>.
- [9] Wood, I., Ruder, S.: Emoji as emotion tags for tweets. (2016). <https://www.insight-centre.org/sites/default/files/publications/ianwood-emotionworkshop.pdf>.
- [10] NLTK documentation, <https://www.nltk.org/>.

Servicio para la gestión inteligente del consumo de energía eléctrica hogareña. Una aplicación móvil para su monitoreo

Lautaro Antonio López, Héctor Luis Vivas, Nicolás García Martínez, Mauro Germán Cambarieri

Laboratorio de Informática Aplicada. Licenciatura en Sistemas.
Universidad Nacional de Río Negro.
Viedma, Río Negro.
{llopez, lvivas, ngarciam, mcambarieri}@unrn.edu.ar

Resumen. El objetivo del presente trabajo es el desarrollo de una solución basada en tecnologías libres para monitorear el consumo de electricidad residencial, el cual constituye el 43% del consumo eléctrico de la República Argentina. El sistema propone una arquitectura de tres componentes, un sensor recolector de datos hogareño, un servidor para alojar los datos y las funciones de procesamiento en la nube y una aplicación móvil para acceder a la información. El producto obtenido es viable desde el punto de vista económico por los materiales y tecnologías utilizadas. El análisis de los resultados obtenidos permite aseverar que la solución funciona de manera satisfactoria y adecuada a los usuarios.

Keywords. Internet de las cosas. Soluciones móviles. Ciudades inteligentes. Accesibilidad.

1 Introducción

El crecimiento acelerado de la residencia en zonas urbanas y periurbanas del planeta está dando lugar a un replanteo general de la manera en que se las habita. Se estima que en 2030 al 60% de la población mundial habitará en ciudades y que en 2050 podría alcanzar los dos tercios [1] La Agenda 2030 para el Desarrollo Sostenible de las Naciones Unidas destaca, como su Objetivo 11. Lograr que las ciudades y los asentamientos humanos sean inclusivos, seguros, resilientes y sostenibles [2]. Por otro lado, el Mecanismo de Facilitación de la Tecnología que se estableció en la Agenda de Acción de Addis Abeba con el fin de apoyar el logro de los Objetivos de Desarrollo Sostenible hace especial hincapié en “la creación, el desarrollo y la difusión de últimas innovaciones y tecnologías y conocimientos asociados, así como la transferencia de tecnología en condiciones mutuamente convenidas, son potentes motores del crecimiento económico y el desarrollo sostenible” [3]. En mayo de 2016, la Comisión de Ciencia y Tecnología para el Desarrollo del Consejo Económico Social de las Naciones Unidas, en su 19º período de sesiones, debatió sobre las principales tendencias en materia de urbanización y desarrollo sostenible, destacando el rol fundamental de las tecnologías de la información y las comunicaciones en el diseño, desarrollo y gestión de las ciudades e infraestructuras inteligentes [4]. Por otro lado, en el punto 11 del mismo documento, se cita como un componente de una infraestructura

inteligente, la inteligencia energética y destaca el ejemplo de la ciudad de Kashiwa-no-ha (Japón) y la aplicación de sistemas domésticos de gestión de la energía a escala del término municipal.

Con estos antecedentes se plantea la necesidad de desarrollar un sistema de monitoreo de consumo eléctrico domiciliario para el uso eficiente y sustentable del citado recurso, con el objetivo de comprometer al propio usuario en acciones de ahorro energético a partir del conocimiento de su consumo diario, permitiéndole el acceso en línea a través de una aplicación móvil de los datos que se generan en su propio hogar.

Tecnologías disponibles como internet de las cosas [5], dispositivos móviles, sensores, procesamiento en la nube y dispositivos de hardware libre son integrados en esta solución para generar datos que pueden utilizarse inteligentemente y de esta forma optimizar recursos energéticos y mejorar su rendimiento.

El presente trabajo aborda, en el punto 2, la problemática actual del consumo de energía eléctrica y la necesidad de hacer un uso responsable de la misma. El punto 3 describe la solución desarrollada y en el punto 4. Se analizan los resultados, se plantean las conclusiones obtenidas y se formulan los trabajos futuros.

2 Trabajos Relacionados

En 2014, Betancor Caro [6] desarrolló un monitor de electricidad integrado por similares componentes de hardware que el que se plantea en este trabajo, el software desarrollado se basa en una arquitectura cliente servidor, siguiendo el patrón MVC¹, utilizando PHP para el servidor y el Framework AngularJS para las interfaces de usuario. Para móviles cuenta con una interfaz adaptativa con un reducido nivel de usabilidad. La tecnología y el patrón de diseño utilizado no garantizan la escalabilidad en un escenario de múltiples monitores. En 2016, Guacaneme Valbuena y Pardo Agudelo [7], diseñaron una solución con la finalidad de censar el consumo residencial de electricidad y agua. La arquitectura es similar al anterior y al presente trabajo. El software se implementó usando NodeJS para el Back-end y para la comunicación entre los monitores y el servidor se utilizó Websocket. Se desarrolló una sola aplicación web. Su arquitectura permite una rápida escalabilidad. En 2018, Peláez Aucay y Jimenez Guamán [8], plantearon un sistema de monitoreo basado en PLC sobre la red eléctrica. Para esto, integraron dispositivos capaces de separar el canal de comunicación de la electricidad que viaja por la red. Este tipo de comunicaciones es interesante en un esquema de redes de electricidad inteligentes, pero en la actualidad, las instalaciones de la mayoría de los hogares y redes de distribución urbanas no están preparadas para soportar que este protocolo de comunicación, funcione correctamente, viéndose la señal de datos afectada por interferencias electromagnéticas. El software se desarrolló utilizando un servidor web LAMP².

¹ MVC: Modelo-Vista-Controlador.

² LAMP: Linux-Apache-MySQL-PHP.

3 Situación/Problemática

En la actualidad, el consumo desmedido de recursos no renovables está impactando en nuestro entorno de diferentes formas, cambios en el clima, catástrofes naturales, el efecto invernadero, etc. Para lidiar con esto, debemos realizar grandes cambios y muchos de estos deben ser culturales.

La electricidad, es generada en su mayoría en base a recursos no renovables. Por lo que es importante realizar un uso responsable de la misma. En Argentina, los sectores residenciales, acumulan un porcentaje de consumo de electricidad del 43%, mientras que el restante se divide entre el sector industrial y el comercial [9]

Por ende, resulta de gran relevancia generar conciencia en las personas para que tengan un estilo de vida más sustentable. Para esto se plantea una solución que sea útil a la hora de controlar y, en lo posible, reducir el consumo de electricidad.

La mayoría de los ciudadanos no son conscientes de toda la energía que usan a diario en sus hogares. Esto, en parte, se debe a todos los dispositivos tecnológicos con los que convivimos día a día; las heladeras, lavarropas, televisores, routers, dispositivos móviles, iluminación, calefacción, etc. Son utilizados de forma casi permanente, sin tener en cuenta el consumo de electricidad que representan.

De acuerdo con diferentes estudios consultados, este consumo desmedido se produce, en gran parte, porque el usuario no tiene un conocimiento del consumo de electricidad en tiempo real. [10] [11]. Esto también sucede en Argentina, donde la información sobre gasto/consumo sólo se conoce cuando llega la factura al final del periodo, que puede ser mensual o bimestral. Durante esta ventana de tiempo, las personas no son conscientes de la electricidad que están utilizando.

4 Solución propuesta

Teniendo en claro la problemática que nace del uso desmedido de los recursos energéticos y la importancia de generar conciencia para que esto no suceda, se plantea una alternativa moderna, sencilla y al alcance del ciudadano para poder monitorear el consumo de electricidad de una residencia.

Para poder enfrentarse al desafío que implica reducir el consumo, es primordial tener información del mismo. Algo que parece casi obvio, pero que en Argentina no está disponible para los ciudadanos. El consumo de electricidad no se puede conocer hasta el momento que llega la factura de la empresa distribuidora de electricidad al finalizar el periodo de facturación.

Por ello, se propone una solución con el objetivo de reducir esa ventana de tiempo, y brindar a los ciudadanos información útil sobre el consumo en tiempo real.

De esta manera, la electricidad dejaría de ser un recurso “invisible” del que no se tiene conocimiento, para transformarla en algo que está monitoreada las veinticuatro horas del día. Esto no es algo menor, llenar el vacío de información que se genera entre los periodos de facturación, ayuda a que las personas sean conscientes y estén atentos a acciones o hábitos que antes no tenían en cuenta pero que pueden modificar, para reducir el consumo de electricidad, significando un ahorro para la persona. [10]

4.1 Planificación.

Para llevar a cabo las tareas de planificación, gestión y desarrollo de la solución, se utilizó la metodología de trabajo SCRUM encuadrada en el marco de desarrollo ágil. Luego de que fueron llevadas a cabo las tareas de relevamiento correspondientes, la información obtenida fue volcada en historias de usuario (HU) que conformaron la “pila del producto”, tal como se muestran en la Tabla 1.

Historias de usuario	
HU 1	Administración de los usuarios
HU 2	Administración de las estaciones de monitoreo
HU 3	Cargar facturas de electricidad al sistema
HU 4	Ver consumo actual de electricidad
HU 5	Ver consumo de electricidad mensual
HU 6	Alertas de consumo excesivo
HU 7	Monitoreo del consumo eléctrico
HU 8	Consulta de las estaciones de monitoreo de un usuario

Tabla 1. Historias de usuario definidas.

Para llevar a cabo el desarrollo de las diferentes piezas de software de forma más controlada y eficiente, se planificaron un conjunto de iteraciones. En total fueron cuatro y cada una duró 30 días. Las iteraciones quedaron definidas como se muestran en la Tabla 2.

Sprint 1	HU 1. Administración de los usuarios
	HU 2. Administración de las estaciones de monitoreo
Sprint 2	HU 7. Monitoreo del consumo eléctrico
	HU 3. Cargar facturas de electricidad al sistema
Sprint 3	HU 8. Consulta de las estaciones de monitoreo de un usuario
	HU 4. Ver consumo actual de electricidad
Sprint 4	HU 6. Alertas de consumo excesivo
	HU 5. Ver consumo de electricidad mensual

Tabla 2. Planificación de las iteraciones.

4.2 Solución desarrollada

Se basó sobre una arquitectura integrada por tres componentes: el servidor de datos, el monitor de consumo eléctrico y una aplicación móvil. Tal como se exponen en la Figura 1.

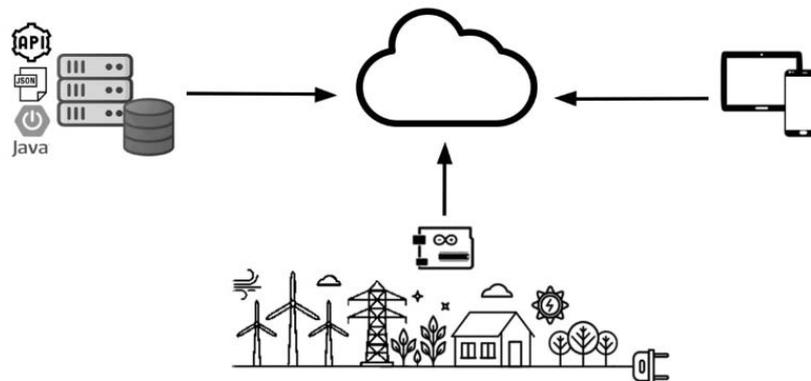


Figura 1. Arquitectura de la solución

El servidor de datos, se integró para procesar la información enviada por los monitores de consumo y brindar los servicios necesarios para la aplicación móvil.

Fue implementado siguiendo los lineamientos típicos para una arquitectura del estilo REST. Los cuales plantean: el uso de mensajes HTTP sin estado, la definición de una interfaz genérica para las operaciones y, por último, propone la utilización de hipermedios tanto para el envío de información como para indicar el estado de la misma. [12]

El framework utilizado en la implementación fue Spring Boot que permite agilizar el desarrollo de aplicaciones JAVA.

Para garantizar que la información del servidor no sea accedida por agentes externos, este se protegió utilizando un mecanismo de autenticación basado en el estándar JSON web tokens. [13]

Los servicios que fueron desarrollados se describen en la Tabla 3.

Gestión de usuarios	Consta de los servicios necesarios para llevar a cabo la administración de los usuarios del sistema.
Consulta de monitores de consumo	Permite al usuario acceder al listado de estaciones que posee junto al estado de cada uno
Consultas sobre el consumo eléctrico	Permite filtrar entre fechas determinadas y la unidad para ponderar el consumo
Gestión de las facturas de electricidad	Almacena el importe en pesos argentinos, la cantidad de kWh consumidos, el año y el periodo al que pertenece.
Alertas	Para variaciones abruptas en sus patrones de consumo de electricidad, y el estado de conexión del monitor

Tabla 3. Servicios desarrollados

El monitor de consumo eléctrico, fue desarrollado integrando tecnologías libres: un microcontrolador Arduino tipo Uno R3, un sensor de corriente no invasivo tipo SCT 013 y un módulo de conectividad WiFi tipo ESP 8266, tal como se muestra en la Figura 2.

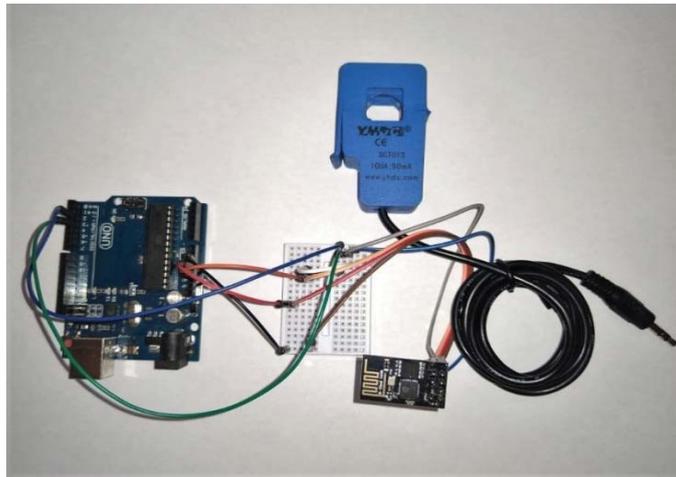


Figura 2. Prototipo de monitor de electricidad.

La aplicación móvil, fue desarrollada para teléfonos con sistema operativo Android 7, pero es compatible con versiones anteriores.

Entre sus funcionalidades son de destacar: la pantalla de ingreso, el menú de la aplicación (Figura 3), la carga de datos de las facturas de electricidad emitidas por la empresa proveedora del servicio (Figura 4), el acceso a las estaciones de consumo, donde se exhibe el nombre de cada una, el domicilio de la misma y el estado en el que se encuentra, que puede variar de acuerdo a su funcionamiento: un icono de color verde junto al mensaje “Conectado”, si la estación se encuentra censando datos y enviándolos al servidor sin ningún inconveniente. Si el monitor no envió ningún dato en un período de tiempo mayor a las dos horas, se mostrará un icono de color amarillo, junto al mensaje: “Última conexión hace XXX minutos” con el registro de la hora en que se envió el último dato al servidor. Por último, en el caso de que la estación no envíe datos por un periodo mayor a 8 (ocho) horas, se muestra un icono rojo junto a un mensaje “Desconectado” (Figura 5).



Figura 3. Menú

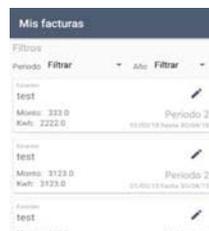


Figura 4. Facturas



Figura 5. Estaciones

La información consolidada se observa en la Figura 6, donde se despliega el consumo de la estación en tiempo real en un reloj, el cual posee distintos intervalos definidos por colores, donde puede distinguirse el consumo normal en color verde, un aumento anormal del gasto energético en color amarillo y una situación extrema en color rojo. Estos intervalos se definen en forma automática en base a los patrones de consumo históricos del usuario. Debajo de este gráfico, se expone el consumo en kilowatts por hora (kWh) de la última hora y a la par de este valor el correspondiente gasto monetizado y, a continuación, un gráfico que muestra el consumo del día anterior en formato de líneas, donde: sobre el eje Y, se exhibe la cantidad de kWh que fueron utilizados; y sobre el eje X, se indican las horas del día a la que se corresponden las mediciones.



Figura 6. Pantalla de monitoreo

Por último, la aplicación tiene la capacidad de recibir alertas en formato de notificación “on push” sobre: desvíos en los patrones de consumo y eventuales situaciones que pongan fuera de línea el monitor de consumo.

5 Verificación y análisis de los resultados.

Se llevaron a cabo un conjunto de pruebas sobre las diferentes partes de la solución. En principio se enfocaron en el monitor de consumo, con la finalidad de que el error máximo en las mediciones del sensor no supere el 10%.

Estas tuvieron lugar en el Laboratorio de Informática Aplicada de la UNRN³. Se construyó un circuito eléctrico donde se conectaron 2(dos) resistencias eléctricas, cada una con un consumo de 2000 watts. Además, se acopló un tester digital de corriente alterna para tener conocimiento de la potencia dentro del circuito. Durante el proceso se registraron las mediciones del monitor y las de tester en periodos de una hora, probando diferentes escenarios con consumo intermitente y sin consumo. En base a los valores obtenidos se calculó el porcentaje de error en cada etapa para establecer una constante de ajuste que permitió reducir el error en cada medición. Luego, se repitió la prueba, aplicando la corrección obtenida. Los resultados de ambas etapas se visualizan en la Tabla 4.

		Resultados de las mediciones		
		Monitor	Tester	Porcentaje de error
Primera etapa	Prueba intermitente	3524w	2680w	31%
	Consumo máximo	6048w	4360w	38%
	Sin consumo	0.04w	0.03w	33%
Segunda etapa	Prueba intermitente	1581w	1740w	9%
	Consumo máximo	4162w	4529w	8%
	Sin consumo	0.0261w	0.03w	13%

Tabla 4. Resultados de la prueba

La otra prueba, consistió en instalar el monitor de consumo eléctrico en un domicilio residencial por un periodo de 30 días con el objeto de coleccionar los datos de consumo eléctrico. La fecha de inicio fue el 11 de febrero del 2019 a las 00:00:00 y finalizó el 12 de marzo de 2019 a las 23:59:59. Los resultados se muestran en el Gráfico 1.

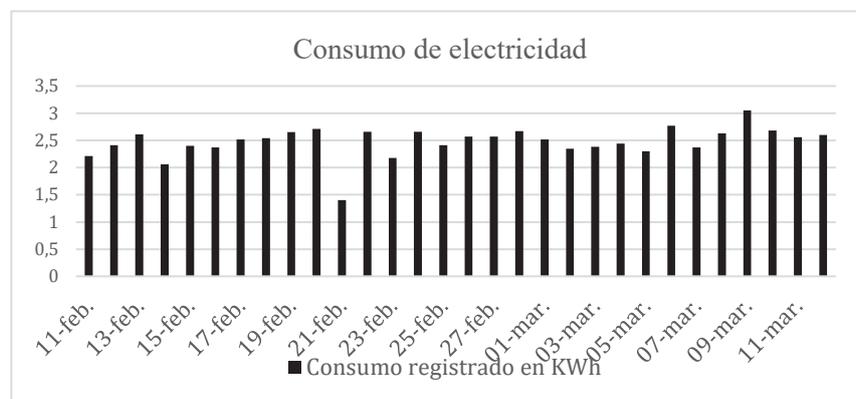


Gráfico 1. Resultados obtenidos de la prueba en una residencia

En el periodo definido, se registraron 49452 (cincuenta y ocho mil cuatrocientas cincuenta y dos) mediciones. El consumo acumulado, fue de 74.24 kilowatts por hora.

En comparación con la factura de electricidad del periodo noviembre-diciembre del 2018 que indica un consumo de 70 kWh y la de diciembre-enero del 2019, cuyo consumo es de 63 kWh del domicilio donde se realizaron las pruebas. Encontramos que

el valor obtenido por el monitor de electricidad, se aproxima al consumo estimado dentro del periodo y al registrado por la empresa proveedora del servicio eléctrico, tal cual se aprecia en el facsímil de la factura siguiente (Figura 7)

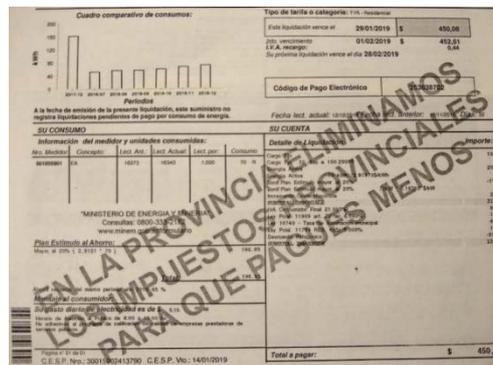


Figura 7. Factura de consumo durante el período estudiado

6 Conclusiones y trabajos futuros

A la luz de las pruebas ejecutadas se observó el correcto funcionamiento de la solución implementada. Los componentes utilizados en el monitor de consumo no presentaron problemas a la hora de realizar las tareas de censado de la electricidad.

Es importante destacar, el bajo costo de los componentes utilizados, lo que favorece a su fabricación y distribución. Teniendo en cuenta el panorama de ciudades y servicios inteligentes, este aspecto de la arquitectura facilita su integración a los hogares.

Mediante la utilización de monitores de consumo de forma masiva, es propicio el escenario de servicios de electricidad inteligentes y redes energéticas capaces de auto regularse basándose en la información obtenida por estos dispositivos.

Se estima continuar avanzando y mejorando la solución, a través de la integración de componentes de hardware que permitan la conectividad a redes más adecuadas para internet de las cosas, como LoRa o LoRaWAN; investigar el uso de componentes alternativos para mejorar la precisión en las mediciones.

Por otro lado, se buscará profundizar en el análisis de los datos obtenidos, contrastando los datos suministrados por el monitor de consumo y los brindados por la empresa proveedora del servicio de manera de lograr un mayor y mejor impacto sobre el hábitat ciudadano, haciéndolo más sustentable y resiliente.

7 Referencias

- [1] United Nations, Department of Economic and Social Affairs, Population Division (2015). World Urbanization Prospects: The 2014 Revision, (ST/ESA/SER.A/366). Disponible en <https://population.un.org/wup/Publications/Files/WUP2014-Report.pdf>

- [2] Naciones Unidas. Resolución aprobada por la Asamblea General el 25 de septiembre de 2015. 70/1. Transformar nuestro mundo: la Agenda 2030 para el Desarrollo Sostenible Disponible en: <https://undocs.org/es/A/RES/70/1>
- [3] Naciones Unidas. Resolución aprobada por la Asamblea General el 27 de julio de 2015 69/313. Agenda de Acción de Addis Abeba de la Tercera Conferencia Internacional sobre la Financiación para el Desarrollo (Agenda de Acción de Addis Abeba) Disponible en: https://unctad.org/meetings/es/SessionalDocuments/ares69d313_es.pdf
- [4] Naciones Unidas. Consejo Económico y Social. Consejo de Ciencia y Tecnología para el Desarrollo. Ciudades e infraestructuras inteligentes. Informe del Secretario General E/CN.16/2016/2. Mayo de 2016. Disponible en https://unctad.org/meetings/es/SessionalDocuments/ecn162016d2_es.pdf
- [5] Dave Evans. (2011, abril) Internet de las cosas como la próxima evolución de Internet lo cambia todo. Cisco Internet Business Solutions Group. Disponible en: https://www.cisco.com/c/dam/global/es_mx/solutions/executive/assets/pdf/internet-of-things-iot-ibsg.pdf
- [6] Y. Betancor Caro. Sistema doméstico de monitorización del consumo eléctrico. Trabajo de fin de grado. Universidad de la Laguna. Tenerife. España. 2014. Disponible en: <https://riull.ull.es/xmlui/bitstream/handle/915/628/Sistema%20domestico%20de%20monitorizacion%20del%20consumo%20electrico.pdf?sequence=1&isAllowed=y>
- [7] G. Guacaneme Valbuena y D. Pardo Agudelo. Diseño e implementación de un sistema de medición de consumo de energía eléctrica y agua potable remoto con interacción al usuario basado en el concepto “internet de las cosas”. Trabajo de fin de grado. Universidad Distrital San Francisco José de Caldas. 2016. Bogotá. Colombia. Disponible en: <http://repository.udistrital.edu.co/handle/11349/4315>
- [8] E. Pelaez Aucay y P. Gimenez Guaman. Diseño de un sistema de medición y monitoreo del consumo de energía por circuitos en el hogar, mediante tecnología de comunicación por línea de potencia. Trabajo de graduación. Universidad del Azuay. Cuenca. Ecuador. 2018. Disponible en: <http://dspace.uazuay.edu.ec/bitstream/datos/7930/1/13668.pdf>
- [9] CAMMESA. (2017). Informe anual 2017. Disponible en: <http://ageera.com.ar/wp-content/uploads/2018/09/informe-anual-CAMMESA-2017.pdf>
- [10] Hargreaves, Tom & Nye, Michael & Burgess, Jacquelin. (2010). Making energy visible: A qualitative field study of how householders interact with feedback from smart energy monitors. Energy Policy. 38. 6111-6119. 10.1016/j.enpol.2010.05.068. Disponible en: https://www.researchgate.net/publication/222295056_Making_energy_visible_A_qualitative_field_study_of_how_householders_interact_with_feedback_from_smart_energy_monitors
- [11] L. Putra, M. Yudishtira y B. Kanigoro. Design and Implementation of Web Based Home Electrical Appliance Monitoring, Diagnosing, and Controlling System. En International Conference on Computer Science and Computational Intelligence (ICCCSCI 2015). 2015
- [12] Roy Fielding. (2000). Architectural Styles and the Design of Network-based Software Architectures. Universidad de California, Irvine. Disponible en: https://www.ics.uci.edu/~fielding/pubs/dissertation/fielding_dissertation.pdf
- [13] Internet Engineering Task Force (IETF). (2015). JSON Web Token (JWT). Disponible en: <https://tools.ietf.org/html/rfc7519> [2019, enero]

Taxonomía para aplicaciones móviles de servicios de gobierno en municipios de Argentina

Rocío Muñoz ¹, Juan Santiago Preisegger ¹, Ariel Pasini , Patricia Pesado 

Instituto de Investigación en Informática LIDI (III-LIDI)*
Facultad de Informática – Universidad Nacional de La Plata 50 y 120 La Plata Buenos Aires

*Centro Asociado Comisión de Investigaciones Científicas de la Pcia. de Bs. As. (CIC)

¹ Becario Postgrado UNLP

{rmunoz, jspreisegger, apasini, ppesado}@lidi.info.unlp.edu.ar

Abstract: Se presenta una taxonomía para la clasificación de aplicaciones móviles de servicios de gobierno en municipios de Argentina. La misma posee un conjunto de categorías y subcategorías que permiten clasificar aplicaciones móviles según la interacción de los ciudadanos con estas. Para el desarrollo de dicha taxonomía, se seleccionaron 53 municipios de Argentina y se analizaron 89 aplicaciones móviles ofrecidas y desarrolladas por dichos municipios. Finalmente, todas las aplicaciones móviles analizadas fueron clasificadas según la taxonomía definida.

Keywords: Aplicaciones móviles, Taxonomía, Gobiernos municipales, Interacción ciudadana.

1 Introducción

En las últimas décadas se han producido avances extraordinarios en tecnologías. La innovación tecnológica ha estado avanzando rápidamente, alcanzando niveles admirables de sofisticación, particularmente con respecto a las tecnologías móviles y las comunicaciones. La aparición de la *tecnología móvil* ha creado enormes oportunidades para mejorar el papel de la buena gobernanza. Históricamente, los ciudadanos han interactuado con agencias gubernamentales locales de manera off-line, pero debido a los avances tecnológicos, muchas de estas actividades ahora pueden realizarse mediante el uso de dispositivos móviles.

Una taxonomía es una agrupación de objetos similares bajo ciertas dimensiones y características. Permite definir categorías y subcategorías para realizar una clasificación de los objetos. Para que la taxonomía desarrollada se considere adecuada tiene que cumplir con ciertos atributos: ser concisa, inclusiva, exhaustiva y extensible.

En base a estos atributos, surgió la necesidad de analizar diversas aplicaciones móviles en 53 municipios seleccionados de Argentina, con el fin de clasificarlas y crear así una taxonomía que tenga en cuenta la interacción del ciudadano con estas

aplicaciones. Dicha taxonomía posee cuatro dimensiones: Informativa, Interactiva, Contributiva y Recreativa, cada una de las cuales con un conjunto de características relacionadas.

Para comprobar que la taxonomía cumpla con todos los atributos previamente mencionados, se clasificaron las 89 aplicaciones relevadas en cada una de las dimensiones y características propuestas.

En la sección dos se describen ventajas del uso de dispositivos móviles en la actualidad. En la tercera sección se presentan definiciones, características y enfoques para la creación de taxonomías. Luego, se presenta un relevamiento de municipios de Argentina y las aplicaciones oficiales que estos ofrecen, seguido por la creación de una taxonomía acorde al propósito planteado. Finalmente, se presentan las conclusiones de este artículo.

2 Uso de dispositivos móviles en la actualidad

En las últimas dos décadas se han producido avances extraordinarios en tecnologías. La innovación tecnológica ha estado ocurriendo a un ritmo rápido, alcanzando niveles admirables de sofisticación, particularmente con respecto a las **tecnologías móviles** y las comunicaciones. Actualmente, los teléfonos inteligentes tienen numerosas funciones que permiten diversas formas de comunicación continua. También permiten a los usuarios tener acceso en tiempo real a las aplicaciones de acuerdo con sus necesidades concretas y específicas, en función de su ubicación física, formando así un concepto de *dispositivo personalizable* [1].

El modelo de Ciudad Inteligente ha emergido con fuerza a nivel mundial como una nueva forma de repensar la gestión urbana. Este concepto entiende que integrar el uso de las TIC en la evolución de una ciudad no solo supone mejoras notables en la provisión de los servicios, sino que constituye en sí misma una vía sostenible para el desarrollo económico y social. Una *Ciudad con Inteligencia* se caracteriza por el uso cotidiano de la tecnología para mejorar la eficiencia en la respuesta a las demandas ciudadanas [2].

La aparición de la tecnología móvil ha creado enormes oportunidades para mejorar el papel de la buena gobernanza, específicamente en cuanto al incremento de información, la interacción con el público y medios más efectivos y eficientes para realizar transacciones públicas. Se define como aplicación móvil al uso de esta tecnología por parte de un usuario final para un propósito particular [3].

Históricamente, los ciudadanos han interactuado con agencias gubernamentales locales de manera off-line, pero debido a los avances tecnológicos, muchas de estas actividades ahora pueden realizarse mediante el uso de dispositivos móviles. Las diferentes formas que tienen los ciudadanos de interactuar con el municipio son:

- Mediante *interacciones receptivas*: transacciones unidireccionales de información desde la agencia gubernamental al ciudadano, o viceversa. Los ciudadanos, por lo general, buscan conocer ordenanzas gubernamentales, planes de crecimiento ciudadano, etc. Por otra parte, las agencias gubernamentales buscan

comprender las características demográficas de un determinado tramo del censo, comprender las características dimensionales de las parcelas, etc.

- Mediante *herramientas interactivas*: interacciones entre la agencia gubernamental y los ciudadanos. Los intercambios interactivos "off-line" implican una transacción bidireccional de información, ya que los ciudadanos a menudo quieren compartir sus pensamientos sobre cómo se están haciendo las cosas en la comunidad y lo que se planifica en el futuro. Antes, el ciudadano tenía opciones limitadas: asistir a una audiencia pública, visitar la oficina de planificación en persona, llamar a los administradores de la ciudad, etc. Las aplicaciones móviles han proporcionado opciones adicionales que hacen que la interactividad sea más accesible.

- Mediante *herramientas transaccionales*: intercambio entre agencias gubernamentales y ciudadanos. Muchas de estas actividades habrían requerido, anteriormente, que un ciudadano visite las oficinas del gobierno local, pero ahora pueden realizarse de forma on-line. Algunas de estas son: la presentación de permisos, variaciones y apelaciones, el pago de tarifas por permisos y otras solicitudes.

3 Taxonomías

3.1 Descripción de una taxonomía

Una taxonomía es una clasificación u ordenación en grupos de objetos que tienen características en común. Según Nickerson [4], una de las principales ventajas que ofrecen las taxonomías es la reducción de la complejidad y la identificación de similitudes y diferencias entre dichos objetos. Las taxonomías adecuadas desempeñan un papel importante en la investigación ya que la clasificación de los objetos ayuda a los investigadores y profesionales a comprender y analizar dominios complejos, permitiendo formular hipótesis sobre estas relaciones.

Existe una diferencia entre los conceptos de tipología y taxonomía [5]: las tipologías se derivan de manera conceptual o deductiva y las taxonomías de manera empírica. En el caso de las tipologías, el investigador puede proponer una categorización basada en un ideal o modelo teórico. Por otro lado, para las taxonomías, la clasificación propuesta se basa en datos empíricos, comenzando con ciertos datos y derivando una clasificación mediante un análisis de estos. El objetivo es encontrar similitudes entre los datos y clasificar objetos similares en las categorías.

Para la taxonomía, se definen dimensiones y características. Las dimensiones brindan una primera clasificación y las características se agrupan dentro de las dimensiones definiendo una clasificación secundaria. Se dice que una taxonomía será útil si tiene los siguientes atributos deseables:

- Ser *concisa*: la taxonomía debe contener un número limitado de dimensiones y características, que sean fáciles de comprender y de aplicar.
- Ser lo suficientemente *inclusiva*: la taxonomía debe contener suficientes dimensiones y características que resulten de interés.
- Ser *exhaustiva*: la taxonomía debe proporcionar la clasificación de todos los objetos actuales dentro del dominio en consideración.

- Ser *extensible*: la taxonomía debe permitir dimensiones adicionales y nuevas características cuando aparecen nuevos tipos de objetos.

3.2 Creación de una taxonomía

El desarrollo de una taxonomía es un proceso complejo. Antes de comenzar, el investigador debe decidir sobre las características o metacaracterísticas más completas que sirvan de base para la clasificación. La elección de la metacaracterística debe basarse en el propósito de la taxonomía.

Bailey [6] describe los enfoques conceptuales y empíricos de un modelo de dos niveles. Aunque los investigadores pueden abordar la clasificación a través de cualquiera de los niveles, sugiere que un enfoque común y, a menudo más útil, es utilizar un modelo de tres niveles que incluya niveles conceptuales, empíricos e indicadores/operacionales. En este método, el investigador tiene dos opciones: comenzar con el enfoque deductivo y luego examinar los casos empíricos (deductivo a empírico) para ver cómo encajan con la conceptualización, o comenzar con grupos de datos empíricos y luego conceptualizar deductivamente la naturaleza de cada grupo (de empírico a deductivo).

En el campo de los sistemas de información, se encontró que el modelo indicador de tres niveles de Bailey es el más atractivo para el desarrollo de taxonomías. No tiene un enfoque único, sino que se basa tanto en la deducción como en el empirismo: el investigador comienza examinando un subconjunto de objetos que desea clasificar e identifica las características generales de estos objetos. La identificación de estas características conduce al primer esfuerzo en una taxonomía, siendo agrupadas en dimensiones que forman la *taxonomía inicial*. Cada dimensión contiene características que son mutuamente excluyentes y colectivamente exhaustivas. Este proceso se basa en los datos empíricos (limitados) que se han recopilado sobre los objetos y la conceptualización deductiva del investigador. El investigador revisa, entonces, la primera taxonomía para buscar conceptualizaciones adicionales que podrían no haber sido identificadas en los datos empíricos originales, pudiendo deducir nuevas características que se ajustan a las dimensiones existentes o conceptualizando nuevas dimensiones. Se examinan los casos empíricos utilizando las nuevas características y dimensiones para determinar su utilidad en la clasificación de objetos, surgiendo así una *taxonomía revisada*.

4 Aplicaciones móviles en municipios de Argentina

4.1 Municipios seleccionados

Para el desarrollo de este artículo se realizó un análisis de diversos municipios de Argentina, buscando incluir ciudades con distintos tamaños y realidades.

Se tuvieron en cuenta, por un lado, los 50 municipios miembros de RECIA (Red de Ciudades Inteligentes de Argentina), el cual es un espacio de intercambio de experiencias y conocimientos que busca promover las ciudades inteligentes y la mejora

de las gestiones de gobierno. La Red está conformada por referentes de municipios inteligentes de todo el país y trabaja de manera horizontal con gobiernos locales y otros actores en temas como el diseño de políticas, herramientas y mejores prácticas [7].

Además, era deseable incluir en el análisis a los 8 municipios más grandes de Argentina según las proyecciones del INDEC para el año 2018 [8]. De estos municipios se encontró que 3 de ellos no pertenecían a la Red, por lo que fueron agregados, completando así una totalidad de 53 municipios seleccionados.

Finalmente, se logró tener un abanico de municipios del país con diferentes características y tamaños, teniendo así un 5,66% de municipios con más de 1 millón de habitantes, un 9,43% con entre 500 mil y 1 millón de habitantes, un 15,09% con entre 250 mil y 500 mil habitantes, un 28,30% con entre 100 mil y 250 mil habitantes, un 9,43% con entre 50 mil y 100 mil habitantes y un 32,09% con menos de 50 mil habitantes. El Gráfico 1 muestra la cantidad de municipios para cada clasificación de tamaños.

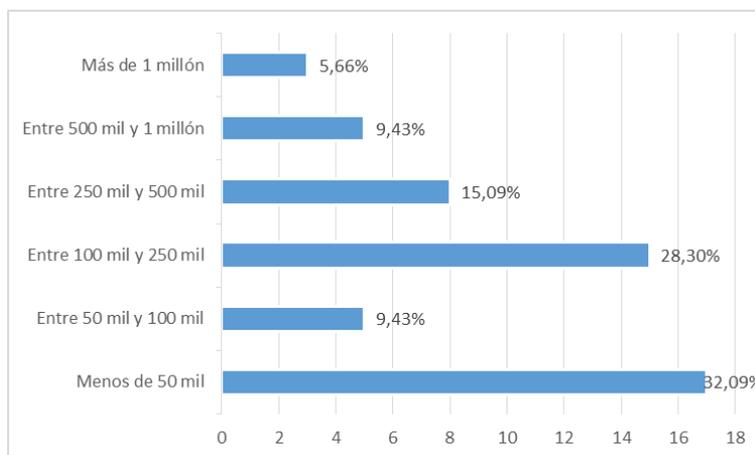


Gráfico 1: Población de los municipios según proyección del INDEC (2018)

Los 53 municipios seleccionados son: en la provincia de Buenos Aires, los municipios de 3 de Febrero, Adolfo Alsina, Bahía Blanca, Bolívar, Campana, Castelli, Chivilcoy, Coronel Pringles, Escobar, Exaltación de la Cruz, General San Martín, Junín, La Plata, Lincoln, Luján, Mar Chiquita, Mar del Plata, Mercedes, Necochea, Pergamino, Pilar, Pinamar, Puan, Ramallo, San Antonio de Areco, San Isidro, San Nicolás, San Pedro, Suipacha, Tandil, Tigre, Vicente López, Villarino y Zárate; en la provincia de Chubut, los municipios de Esquel, Gaiman, Puerto Madryn y Rawson; en la provincia de Córdoba, los municipios de Córdoba y Villa María; en la provincia de Mendoza, los municipios de Godoy Cruz y Luján de Cuyo; en la provincia de Neuquén, los municipios de Neuquén y Villa La Angostura; en la provincia de Tucumán, el municipio de San Miguel de Tucumán; en la provincia de Salta, el municipio de Salta; en la provincia de San Luis, los municipios de Juana Koslay y Tilisarao; en la provincia de Santa Fé, los municipios de Rafaela, Reconquista, Rosario y Santa Fé de la Vera Cruz; y por último la Ciudad Autónoma de Buenos Aires.

4.2 Relevamiento de aplicaciones oficiales en los municipios seleccionados

Una vez seleccionados los municipios, se realizó la búsqueda de aplicaciones móviles ofrecidas por cada uno de ellos, utilizando para la búsqueda la tienda de la plataforma Google Play.

En cada municipio se buscaron aplicaciones ofrecidas y desarrolladas por el gobierno correspondiente, a las cuales denominaremos “aplicaciones oficiales”.

De los 53 municipios relevados, se encontró que 26 de ellos (49%) no ofrecían ninguna aplicación oficial (incluso se encontraron varios que tampoco ofrecían aplicaciones de terceros relacionadas con el municipio), 9 de los municipios (17%) solo ofrecían una aplicación oficial (la cual, en algunos casos, abarcaba varias áreas del municipio y funcionaba como una aplicación general), 11 municipios (22%) brindaban dos o tres aplicaciones, 3 municipios (5%) ofrecían entre cuatro y siete aplicaciones, otros 3 municipios (5%) entre ocho y diez y solo 1 municipio (2%) tenía una cartera de más de 10 aplicaciones móviles para ofrecer a sus ciudadanos. Esto puede observarse en el Gráfico 2.

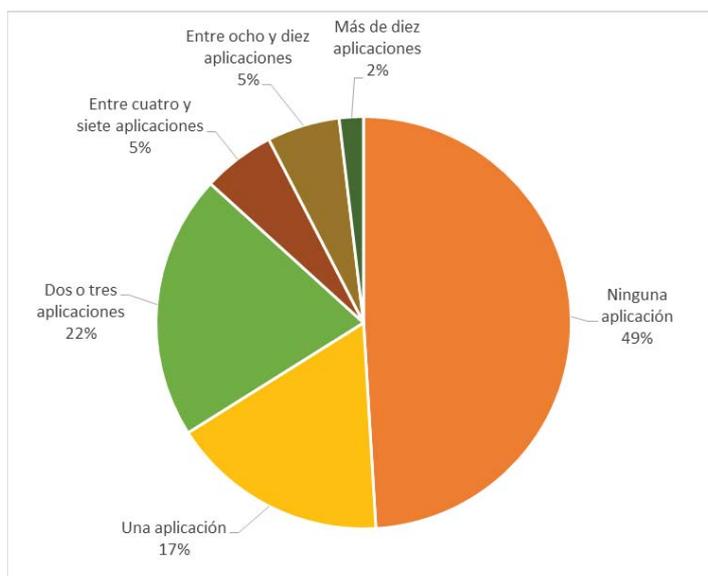


Gráfico 2: Cantidad de aplicaciones oficiales ofrecidas por los municipios

Dentro de las 89 aplicaciones móviles encontradas para los distintos municipios, podemos distinguir aplicaciones de varios tipos y con diferentes propósitos, y clasificarlas en los siguientes grupos: Agendas culturales/turismo, Alertas ciudadanas, Transporte público, Juegos educativos, Radios, Pago de servicios, Farmacias de turno, Reclamos, Mapas, Estacionamiento medido, Emergencia, etc.

5 Taxonomía para aplicaciones móviles

5.1 Creación de una taxonomía mediante el análisis de aplicaciones móviles

La taxonomía presentada tiene como propósito clasificar las aplicaciones móviles desde un punto de vista comunicacional, por lo que se define como metacaracterística a la interacción entre los ciudadanos y las aplicaciones.

Para lograr una taxonomía que cumpla con los atributos deseables presentados (concisa, inclusiva, exhaustiva y extensible) se decidió utilizar el enfoque empírico: se analizaron las aplicaciones móviles ofrecidas por los municipios, relevadas en el punto 4.2, y se buscaron similitudes en las formas de interactuar con estas para crear diferentes categorías. De esta forma, se identificaron cuatro dimensiones:

- **Informativa:** aplicaciones que brindan información al ciudadano. La información es enviada unidireccionalmente de la aplicación al ciudadano.
- **Interactiva:** aplicaciones que permiten que el ciudadano intercambie información con la aplicación, compartiendo su opinión o participando en gestiones de gobierno. La información tiene un flujo bidireccional entre el ciudadano y la aplicación.
- **Contributiva:** aplicaciones que permiten que el ciudadano brinde información de interés para el gobierno. La información se envía unidireccionalmente del ciudadano a la aplicación.
- **Recreativa:** aplicaciones que permiten al ciudadano interactuar de forma lúdica con ellas. No existe el intercambio de información.

Una vez definidas las cuatro categorías iniciales, se decidió incluir, además, un enfoque deductivo para establecer subcategorías/características que ayuden a brindar una mejor clasificación de las aplicaciones para cumplir con el propósito.

En la categoría **Informativa** se definen cuatro subcategorías: *Alerta*, *Estática*, *Dinámica* y *Basada en la ubicación*. La subcategoría *Alerta* hace referencia a aplicaciones que interactúan con el ciudadano en tiempo real, enviándole información de manera automática, por ejemplo, ante emergencias. Las aplicaciones categorizadas como *Estáticas* son aquellas que brindan información a los ciudadanos sin depender de la interacción con los mismos. Por otro lado, cuando se habla de aplicaciones con categoría *Dinámica*, se hace referencia a aquellas que esperan un accionar del ciudadano para brindarles información. Por último, la subcategoría *Basada en la ubicación*, como su nombre lo indica, hace referencia a las aplicaciones que brindan a los ciudadanos información personalizada, en base a la geolocalización de los mismos.

Para la categoría **Interactiva** se establecen dos subcategorías: *Atención al ciudadano* y *Servicio de transporte*. En el caso de la subcategoría *Atención al ciudadano*, se decide incluir en esta a aquellas aplicaciones que permiten al ciudadano informar diversos problemas en el municipio o realizar solicitudes, con la posibilidad de realizar un seguimiento de sus incumbencias a través de la aplicación. Por otro lado, la subcategoría *Servicio de transporte* hace referencia a las aplicaciones que permiten que el ciudadano solicite un transporte público a través de estas e interactúen realizando diversas consultas sobre el estado del mismo.

En la categoría **Contributiva** se incluyen tres subcategorías: *Recopilación de datos*, *Pedido de ayuda* y *Recaudación*. La subcategoría *Recopilación de datos* hace referencia a aquellas aplicaciones que se nutren con la información que los ciudadanos les brindan completando formularios o interactuando de otra manera. Por su parte, las aplicaciones categorizadas como *Pedido de ayuda* son aquellas donde el ciudadano puede generar una alerta para pedir ayuda mediante, por ejemplo, un botón de pánico enviando su geolocalización a quien corresponda. Al hablar de *Recaudación*, se toma como aplicaciones de esta subcategoría a aquellas que le permiten al ciudadano realizar pagos al municipio.

Por último, para la categoría **Recreativa**, se definen dos subcategorías: *Juegos* y *Realidad Virtual*. Estas subcategorías no precisan demasiada explicación, sus nombres son lo suficientemente autoexplicativos. Las aplicaciones de la subcategoría *Juegos* son las que permiten a los ciudadanos conocer y/o aprender, mediante algún tipo de juego educativo, sobre el municipio u otras cuestiones de interés, como puede ser la salud, leyes, etc. Por otro lado, la subcategoría *Realidad Virtual* engloba a aquellas aplicaciones que, mediante el uso de tecnologías como la realidad aumentada, permiten a los ciudadanos conocer más sobre el municipio de manera innovadora y entretenida.

La taxonomía final desarrollada se puede observar en la Figura 1.



Figura 1: Taxonomía para aplicaciones móviles

5.2 Clasificación de las aplicaciones según la taxonomía

Para comprobar si la taxonomía desarrollada es correcta, es decir si cumple con los atributos requeridos, se decidió clasificar cada una de las 89 aplicaciones móviles relevadas en las distintas categorías y subcategorías definidas. El Gráfico 3 muestra el resultado de esta clasificación.

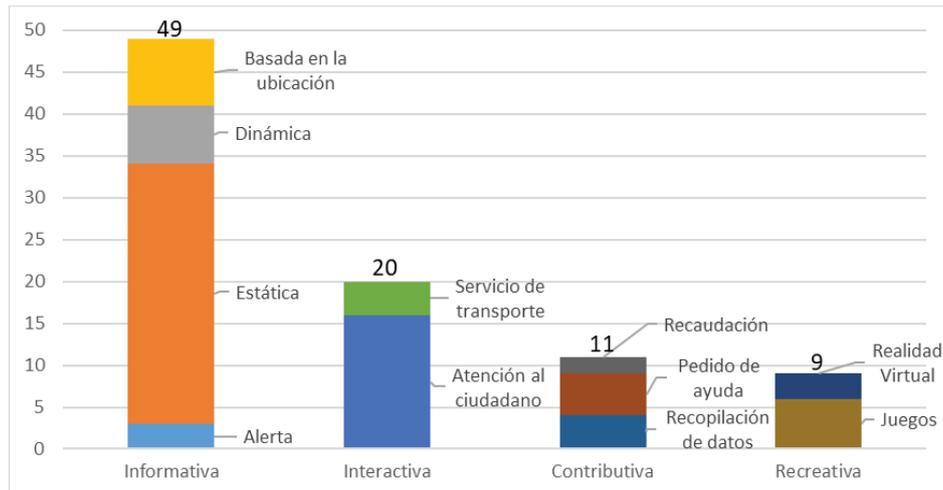


Gráfico 3: Aplicaciones móviles clasificadas según la taxonomía desarrollada

Como se puede observar, gran parte de las aplicaciones relevadas corresponden a la categoría *Informativa*. Hay 49 aplicaciones que pertenecen a esta categoría, siendo el 63% de la subcategoría *Estática*, como por ejemplo aplicaciones de noticias, novedades, agendas, radios, etc.; el 16% de la subcategoría *Basada en la ubicación*, como aplicaciones que indican cuándo llega un micro, qué cosas hay para hacer en las cercanías del ciudadano; el 14% de la subcategoría *Dinámica*, por ejemplo aplicaciones para consultar multas, deudas, escanear códigos QR; y el 7% de la subcategoría *Alerta*, como por ejemplo aplicaciones que advierten de fenómenos climáticos, crecidas de ríos, etc.

De las aplicaciones relevadas correspondientes a la categoría *Interactiva*, se encontró que de las 20 aplicaciones el 80% pertenecen a la subcategoría *Atención al ciudadano*, como es el caso de aplicaciones para realizar reclamos al municipio, mientras que de la subcategoría *Servicio de transporte* solo hay un 20%, como por ejemplo aplicaciones para solicitar taxis, las cuales contribuyen a la seguridad de los ciudadanos.

Siguiendo con la siguiente categoría, *Contributiva*, 11 de las aplicaciones relevadas fueron categorizadas como tal, entre las cuales el 45% pertenece a la subcategoría *Pedido de ayuda*, por ejemplo aplicaciones con botones de pánico para alertar a la policía, bomberos o en casos de violencia de género; el 37% a la subcategoría *Recopilación de datos*, con aplicaciones que permiten, por ejemplo, dar aviso de mascotas perdidas, o aplicaciones para brindar datos sobre infracciones o mal estado de las calles; y el 18% a la subcategoría *Recaudación*, teniendo en cuenta aquellas

aplicaciones de estacionamiento medido o que permiten realizar pagos de servicios/trámites.

En último lugar se encuentran las aplicaciones que pertenecen a la categoría *Recreativa*. De las 9 identificadas se puede observar que el 67% son de la subcategoría *Juegos*, entre las cuales se incluyen aquellas con juegos educativos, y el 33% de la subcategoría *Realidad Virtual*, que incluye las aplicaciones que permiten a los ciudadanos interactuar con diferentes partes del municipio a través del uso de tecnologías como, por ejemplo, la realidad aumentada.

6 Conclusiones

Se realizó un análisis de la importancia del uso de dispositivos móviles en la actualidad y de la forma en que los ciudadanos utilizan las tecnologías móviles para la comunicación con los municipios. Además, se investigaron las diferentes definiciones, características y tipos de enfoques para la creación de taxonomías.

Para el presente artículo, 53 municipios de Argentina fueron seleccionados y se llevó a cabo un relevamiento de las aplicaciones móviles ofrecidas y desarrolladas por el gobierno correspondiente, las cuales fueron denominadas “aplicaciones oficiales”. Se encontró que casi la mitad de los municipios (el 49%) no ofrecía ninguna aplicación oficial. Observando los municipios restantes, se hallaron 89 aplicaciones móviles. Cada una de estas aplicaciones fue analizada desde el punto de vista de la interacción de los ciudadanos con estas, con el fin de crear una taxonomía que permita obtener una clasificación de las mismas.

En este trabajo se desarrolló una taxonomía para aplicaciones móviles de servicios de gobierno en municipios de Argentina, utilizando tanto un enfoque empírico como deductivo, y se obtuvo como resultado una taxonomía que cumple con los todos los atributos deseados: es concisa (contiene 4 dimensiones y 11 características, fáciles de entender y aplicar), es inclusiva (se cubren todos los aspectos necesarios para la categorización de las aplicaciones móviles), es exhaustiva (todas las aplicaciones móviles relevadas pudieron ser clasificadas) y es extensible (es posible que la taxonomía crezca en el futuro, en caso de ser necesario).

7 Agradecimientos

Esta publicación fue realizada en el contexto del Proyecto CAP4CITY – “Strengthening Governance Capacity for Smart Sustainable Cities” (www.cap4city.eu) co-financiado por el Programa Erasmus+ de la Unión Europea. Acuerdo Número 598273-EPP-1- 2018-1-AT-EPPKA2-CBHE-JP. Número de proyecto: 598273

CAP4CITY

Co-funded by the
Erasmus+ Programme
of the European Union



8 Referencias

1. D. Dutra, D. Soares (2019). "Mobile Applications in Central Government of Brazil and Portugal". ICEGOV2019.
2. L. Jolíás, A. Prince (2016). "Definiendo un modelo de Smart Cities para el contexto argentino" en "Ciudades Inteligentes, el aporte de las TIC a la comunidad", pp. 15-30.
3. W. Riggs, K. Gordon (2017). "How is Mobile Technology Changing City Planning? Developing a Taxonomy for the Future" en "Environment and Planning B: Urban Analytics and City Science" vol. 44, pp. 100-119.
4. R. Nickerson, J. Muntermann, U. Varshney, H. Isaac (2009). "Taxonomy development in information systems: Developing a taxonomy of mobile applications".
5. K. D. Bailey (1994). "Typologies and Taxonomies - An Introduction to Classification Techniques". Sage, Thousand Oaks, California.
6. K. D. Bailey (1984). "A Three-Level Measurement Model" en "Quality and Quantity" vol. 18, pp. 225-245.
7. "Municipios miembros de la Red de Ciudades Inteligentes de Argentina". RECIA. Disponible en el sitio web: <https://recia.com.ar/miembros/>. Visitado en: Julio de 2019.
8. "Proyecciones por departamento". INDEC. Disponible en el sitio web: <https://www.indec.gob.ar/indec/web/Nivel4-Tema-2-24-119>. Visitado en: Julio de 2019.



CACIC 2019

Congreso Argentino de Ciencias de la Computación

ACTAS

XXV Congreso Argentino de Ciencias de la Computación CACIC 2019

Patricia Pesado y Marcelo Arroyo

Compiladores

El Congreso Argentino de Ciencias de la Computación (CACIC) es organizado por la Red de Universidades Nacionales con carreras de Informática. La edición número XXV, correspondiente al año 2019, fue organizada por el Departamento de Computación de la Facultad de Ciencias Exactas, Físico-Químicas y Naturales de la Universidad Nacional de Río Cuarto.

La presente publicación digital reúne los artículos aceptados en los diferentes *workshops* del mencionado congreso, los que describen resultados de investigaciones y/o desarrollos sobre problemas concretos de la ciencia de la computación.

En este sentido, el libro de actas que aquí se presenta se constituye en una de las principales fuentes de información sobre temas de investigación de las diferentes áreas de la disciplina, por lo que reviste de interés tanto en el ámbito académico como en el de las industrias de software y hardware.

ISBN 978-987-688-377-1



9 789876 883771

e-book

UniRío
editora



Universidad Nacional
de Río Cuarto