

# **Advances in Computing Science and Applied Algorithms**

---

---

# Research in Computing Science

---

## Series Editorial Board

### Editors-in-Chief:

*Grigori Sidorov (Mexico)*  
*Gerhard Ritter (USA)*  
*Jean Serra (France)*  
*Ulises Cortés (Spain)*

### Associate Editors:

*Jesús Angulo (France)*  
*Jihad El-Sana (Israel)*  
*Alexander Gelbukh (Mexico)*  
*Ioannis Kakadiaris (USA)*  
*Petros Maragos (Greece)*  
*Julian Padget (UK)*  
*Mateo Valero (Spain)*

### Editorial Coordination:

*Alejandra Ramos Porras*

*Research in Computing Science* es una publicación trimestral, de circulación internacional, editada por el Centro de Investigación en Computación del IPN, para dar a conocer los avances de investigación científica y desarrollo tecnológico de la comunidad científica internacional. **Volumen 137**, noviembre 2017. Tiraje: 500 ejemplares. *Certificado de Reserva de Derechos al Uso Exclusivo del Título* No. : 04-2005-121611550100-102, expedido por el Instituto Nacional de Derecho de Autor. *Certificado de Licitud de Título* No. 12897, *Certificado de licitud de Contenido* No. 10470, expedidos por la Comisión Calificadora de Publicaciones y Revistas Ilustradas. El contenido de los artículos es responsabilidad exclusiva de sus respectivos autores. Queda prohibida la reproducción total o parcial, por cualquier medio, sin el permiso expreso del editor, excepto para uso personal o de estudio haciendo cita explícita en la primera página de cada documento. Impreso en la Ciudad de México, en los Talleres Gráficos del IPN – Dirección de Publicaciones, Tres Guerras 27, Centro Histórico, México, D.F. Distribuida por el Centro de Investigación en Computación, Av. Juan de Dios Bátiz S/N, Esq. Av. Miguel Othón de Mendizábal, Col. Nueva Industrial Vallejo, C.P. 07738, México, D.F. Tel. 57 29 60 00, ext. 56571.

**Editor responsable:** *Grigori Sidorov, RFC SIGR651028L69*

**Research in Computing Science** is published by the Center for Computing Research of IPN. **Volume 137**, November 2017. Printing 500. The authors are responsible for the contents of their articles. All rights reserved. No part of this publication may be reproduced, stored in a retrieval system, or transmitted, in any form or by any means, electronic, mechanical, photocopying, recording or otherwise, without prior permission of Centre for Computing Research. Printed in Mexico City, in the IPN Graphic Workshop – Publication Office.

# **Advances in Computing Science and Applied Algorithms**

**Juan Carlos Chimal Eguía  
Horacio Rodríguez Bazán  
Christian E. Maldonado (eds.)**



Instituto Politécnico Nacional  
"La Técnica al Servicio de la Patria"



Instituto Politécnico Nacional, Centro de Investigación en Computación  
México 2017

**ISSN: 1870-4069**

---

Copyright © Instituto Politécnico Nacional 2017

Instituto Politécnico Nacional (IPN)  
Centro de Investigación en Computación (CIC)  
Av. Juan de Dios Bátiz s/n esq. M. Othón de Mendizábal  
Unidad Profesional “Adolfo López Mateos”, Zacatenco  
07738, México D.F., México

<http://www.rcs.cic.ipn.mx>

<http://www.ipn.mx>

<http://www.cic.ipn.mx>

The editors and the publisher of this journal have made their best effort in preparing this special issue, but make no warranty of any kind, expressed or implied, with regard to the information contained in this volume.

All rights reserved. No part of this publication may be reproduced, stored on a retrieval system or transmitted, in any form or by any means, including electronic, mechanical, photocopying, recording, or otherwise, without prior permission of the Instituto Politécnico Nacional, except for personal or classroom use provided that copies bear the full citation notice provided on the first page of each paper.

Indexed in LATINDEX, DBLP and Periodica

Printing: 500

Printed in Mexico

## Editorial

Computer Science is pervasive to the scientific endeavor. There is no area or field of study which, nowadays, can thrive without the use of computational resources, models and algorithms to analyze ever growing amounts of data in increasingly complex ways. This trend is clearly reflected in the selection of papers included in this volume of *Research in Computing Science*.

This issue presents both new and improved algorithms and methods as well as applications to other fields of study. Accepted works range from improved algorithms for solving classical computational problems, to the application of known models and techniques on biomedical challenges.

All submitted papers were reviewed by two or more independent members of the editorial committee. This volume contains revised and corrected versions of the 15 accepted papers.

Our deepest gratitude goes to all the parties involved in the creation of this volume: Foremost to the authors of the articles for their dedication to the excellence of the works presented. We are also grateful for the hard labor the members of the editorial board invested in the evaluation and selection of the highest quality papers amongst many others of high value. We are also indebted to the *Sociedad Mexicana de Inteligencia Artificial* (SMIA) for their collaboration towards the completion of this journal. Our special and profound gratefulness to the *Centro de Investigación en Computación* of the *Instituto Politecnico Nacional* (CIC-IPN) for their invaluable collaboration in the publishing of this issue. The submission, review, and selection processes were enabled by the widely adopted tool *EasyChair*, ([www.easychair.org](http://www.easychair.org)) free of charge.

*Juan Carlos Chimal Eguía*  
*Horacio Rodríguez Bazán*  
*Christian E. Maldonado*  
Guest Editors

November 2017



## Table of Contents

	Page
<b>Procesador soft-core de una única instrucción</b> .....	9
<i>Fernando Olivera Domingo, Felipe Iturriaga Cortés, Christian Abraham Almaraz de Horta</i>	
<b>Lagarto I – Una plataforma hardware/software de arquitectura de computadoras para la academia e investigación</b> .....	19
<i>Cristóbal Ramírez, César Hernández, Carlos Rojas Morales, Gustavo Mondragón García, Luis A. Villa, Marco A. Ramírez</i>	
<b>Reconstrucción de profundidad a partir de una sola imagen con perspectiva mediante redes neuronales completamente convolucionales</b> .....	29
<i>José E. Valdez Rodríguez, Hiram Calvo, Edgardo M. Felipe Riverón</i>	
<b>Sistema de monitoreo remoto y detección de anomalías cardiacas en pacientes ambulatorios</b> .....	39
<i>César Enrique Rose Gómez, Luz María Márquez Agúndez, Juan Carlos Hernández Cruz, María Trinidad Serna Encinas</i>	
<b>Implementación de un cancelador activo de ruido en un ducto usando el algoritmo FX-ECLMS</b> .....	51
<i>E. Pichardo, X. Maya, A. A. Vázquez, J. G. Avalos, J. C. Sánchez</i>	
<b>Identificación del ratón en base a su huella dactilar utilizando características Ordinales Lineales Ortogonales (OLOF)</b> .....	63
<i>Jorge Mateo Juárez, Aldrin Barreto Flores, Leopoldo Altamirano Robles, Verónica Edith Bautista López, Salvador Eugenio Ayala Raggi, Francisco Ramos Collazo</i>	
<b>Clasificación de hiperintensidades presentes en la sustancia blanca en imágenes de resonancia magnética usando extracción de características y redes neuronales</b> .....	75
<i>Miguel de Jesús Martínez Felipe, Oleksiy Pogrebnyak, Jesús Alberto Martínez Castro, Edgardo Manuel Felipe Riverón</i>	
<b>Aplicación del método de bisección para la mejora automática del comportamiento a-todo-momento del algoritmo ASO</b> .....	85
<i>Juan C. Moreno-Torres, Marco A. Moreno-Armendáriz</i>	
<b>Uso de tecnologías web para la entrega de datos de sensado de una red de distribución de agua usando la placa de desarrollo Intel Galileo</b> .....	97
<i>Juan Antonio Alvarado Elías, Roberto Solís, Oscar Osvaldo Ordaz García, José Guadalupe Arceo Olague</i>	

<b>Aplicación del análisis sintáctico automático en la atribución de autoría de mensajes en redes sociales .....</b>	<b>109</b>
<i>Francisco Antonio Castillo Velásquez, Jonny Paul Zavala De Paz, Mayra Sánchez Castillo, Adrián Márquez Escandón, Ismael Morales Hernández, Jonathan Flores Guzmán</i>	
<b>Implementación de algoritmos genéticos para el diseño, optimización y selección de vigas .....</b>	<b>121</b>
<i>Valentín Vázquez Castillo, Derlis Hernández Lara, Emmanuel Alejandro Merchán Cruz, Ricardo Gustavo Rodríguez Cañizo, Edgar Alfredo Portilla Flores</i>	
<b>Sistema para la medición, visualización y registro de la radiación solar .....</b>	<b>135</b>
<i>Ismael Díaz Rangel, María del Consuelo Escoto Ponce de León, Enrique José Tinajero Pérez, Luis Francisco Hernández Villalobos, Gustavo Angel Pioquinto Pérez</i>	
<b>WannaCry: Análisis del movimiento de recursos financieros en el blockchain de bitcoin .....</b>	<b>147</b>
<i>Víctor Gabriel Reyes Macedo, Moisés Salinas Rosales</i>	
<b>Algoritmo genético para el problema logístico de asignación de la mochila (Knapsack Problem) .....</b>	<b>157</b>
<i>Gladys Bonilla Enríquez, Diana Sánchez Partida, Santiago Omar Caballer Morales</i>	
<b>Clasificación de imágenes de redes mitocondriales asociadas al estudio del cáncer de mama .....</b>	<b>169</b>
<i>F.J. Vázquez-Vázquez, E. Castillo-Montiel, J.C. Chimal-Eguía, P. Maycotte-González</i>	

# Procesador *soft-core* de una única instrucción

Fernando Olivera Domingo, Felipe Iturriaga Cortés,  
Christian Abraham Almaraz de Horta

Instituto Politécnico Nacional,  
Unidad Profesional Interdisciplinaria de Ingeniería Campus Zacatecas (UPIIZ),  
Zacatecas, Mexico

{foliverad, fiturriaga}@ipn.mx, crisalmaraz@outlook.com

**Resumen.** En este trabajo se presenta un procesador *soft-core*, implementado sobre un dispositivo lógico programable FPGA, de tipo OISC (*One Instruction Set Computer*), es decir que cuenta con una única instrucción y que esta es completa Turing. El procesador desarrollado es de 8 bits (aunque se puede modificar fácilmente) y se compara con el procesador de 8 bits de Xilinx (Picoblaze) con el fin de presentar una propuesta sobre sus posibles aplicaciones.

**Palabras clave:** arquitectura de computadoras, lógica programable, sistemas embebidos.

## Single-Instruction Soft-Core Processor

**Abstract.** This paper presents a processor *soft-core*, implemented on a programmable logic device FPGA, of type OISC (*One Instruction Set Computer*), that is to say that it has a 'unique instruction' and that this is complete Turing. The developed processor is 8 bits (although it can be easily modified) and compared with the 8-bit Xilinx processor (Picoblaze) in order to present a proposal about its possible applications.

**Keywords:** Computer Architecture, Programmable Logic, Embedded Systems.

## 1. Introducción

Generalmente cuando se utiliza un procesador, este ya está implementado con circuitos integrados, siendo lo que se denomina un procesador *hard-core*. Sin embargo existen circunstancias donde se necesita mayor flexibilidad para modificar su set de instrucciones o simplemente no existe una arquitectura

comercial adecuada, en esos casos se puede hacer uso de un dispositivo lógico programable, como son las FPGA (*Field Programmable Gate Array*), para sintetizar procesadores mediante lenguaje de descripción de hardware (HDL). A estos dispositivos se les conoce como procesadores *soft-core*. Estos procesadores no son implementados directamente en el silicio por lo que presentan mayores capacidades de configuración a costa de perder velocidad de procesamiento [14]. Debido a los elevados costos de los procesadores *hard-core*, los procesadores *soft-core* se convierten en una alternativa excelente para la investigación, desarrollo y validación de nuevas arquitecturas no comerciales.

## 2. OISC

En este trabajo se presenta un procesador OISC, es decir, de una sola instrucción. Parecería inútil realizar un procesador así si no consideramos que esta instrucción concreta sea completa Turing [13] (suponiendo un almacenamiento ilimitado y una fiabilidad absoluta), lo que significa que una máquina con solo esta instrucción puede simular una máquina universal de Turing, lo que de manera más coloquial implica que la máquina puede, en principio, hacer cualquier cálculo que cualquier otra computadora es capaz de hacer (en otras palabras, es programable). Obsérvese, sin embargo, que no dice nada sobre el esfuerzo de escribir un programa para la máquina o sobre el tiempo que puede tomar el cálculo.

Los procesadores OISC son también llamados URISC (*Ultimate Reduced Instruction Set Computer*) [8], por llevar al extremo el principio de los procesadores RISC (*Reduced Instruction Set Computer*) de utilizar un set de instrucciones reducido, en este caso compuesto por una sola instrucción. Sin embargo es discutible que los procesadores OISC se puedan incluir dentro de los procesadores RISC o sean una nueva clase con características diferenciadas respecto a los procesadores RISC y CISC [6].

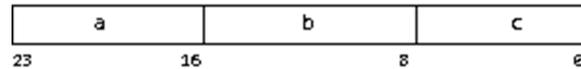
Se pueden separar los procesadores OISC en tres categorías [10]:

- *Transport Triggered Architecture Machines*: Máquinas en las que la computación es un efecto secundario del movimiento de datos (cuentan solo con una instrucción de tipo MOVE)[6].
- *Bit Manipulating Machines*: Máquinas que manipulan bits (o bytes) y saltan incondicionalmente a una dirección especificada por uno de los operandos[9].
- *Arithmetic Based Turing Complete-Machine*: Máquinas que unen la realización de una operación aritmética junto con un salto condicional [6].

La ventaja que presenta este tipo de computadoras es que no requieren un código de operación porque solo tiene una instrucción, lo que simplifica considerablemente el diseño de la CPU.

## 3. Desarrollo de un procesador OISC en FPGA

En este trabajo se ha desarrollado un procesador OISC del último tipo presentado en la sección anterior, con la instrucción SUBLEQ A B C (SUBtract



**Fig. 1.** Formato de la única instrucción del procesador.

and branch if Less or Equal), que significa “resta el valor almacenado en la posición A de la memoria al valor almacenado en la posición B de la memoria, almacena el resultado en la posición B de la memoria y si es menor o igual que cero salta a la dirección de memoria correspondiente al valor almacenado en la posición C de la memoria”. Se puede demostrar que esta instrucción es Turing completa basándose en el Teorema de Bhm-Jacopini [1] siguiendo la misma línea presentada por Gilreath y Laplante [6].

La máquina presentada en cuestión tiene un tamaño de datos de 8 bits, pero las instrucciones son de 24 bits, estando el formato de la instrucción en términos de las posiciones de memoria que direcciona la instrucción (ver figura 1). En este caso se eligió utilizar una arquitectura tipo Harvard con una memoria ROM (asíncrona o síncrona) con un tamaño de palabra de 24 bits, que almacena el programa, y una memoria RAM de doble puerto síncrona con un tamaño de palabra de 8 bits, que le permite leer dos datos a la vez y escribir uno cuando la señal de habilitación esté activa. La CPU cuenta solo con dos estados pues no necesita decodificar, por lo que la máquina de estados solo tiene un estado donde carga la nueva instrucción y otro donde la ejecuta. No se ha incluido segmentación (*pipeline*) porque la idea principal es realizar un procesador lo más sencillo posible (con un área mínima). Se podría implementar una arquitectura von Neumann, considerando que cada instrucción serían tres posiciones de la memoria RAM (y que por tanto el contador de programa debe ir avanzando 3 posiciones de memoria cada nueva instrucción), sin embargo, aunque se ahorra mucha área en memoria se pierde velocidad del procesador por los accesos a memoria RAM. El procesador no necesita contar con registros, sin embargo es posible tener registros de propósito general o de entrada/salida (I/O) mapeados en memoria. La figura 2 muestra un esquemático de los componentes del procesador.

#### 4. Trabajos relacionados

Desde que fue propuesto el primer procesadores OISC (o URISC) [12] se han desarrollado distintos procesadores OISC como prueba de concepto, pero en muchos casos no son más que simulaciones [3]. También se ha planteado el uso de procesadores OISC en aplicaciones como algoritmos genéticos [5], procesamiento de imágenes [7], computación encriptada [2] [11] o incluso como multi-procesador [10].

Los procesadores OISC, por su sencillez, puede ser útiles en aplicaciones donde se necesita un microcontrolador sencillo integrado con otros circuitos lógicos

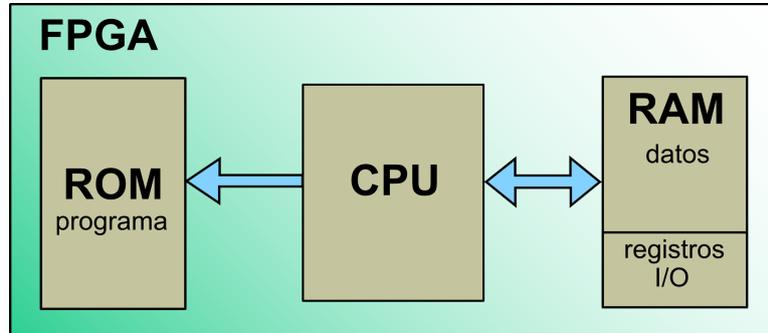


Fig. 2. Esquemático de la arquitectura del procesador.

dentro de una FPGA. Es por ello que el objetivo del trabajo ha sido comparar un procesador OISC basado en la instrucción SUBLEQ con el procesador *soft-core* de 8-bits de Xilinx: Picoblaze [15] para poder encontrar posibles aplicaciones. Creemos que los procesadores OISC pueden adaptarse bien en aplicaciones donde se necesite utilizar pocos recursos.

## 5. Metodología

Para el desarrollo del procesador se utilizó una metodología Top-Down, donde se realiza una descripción inicial del sistema sin conocer todos los detalles de la implementación y dicho diseño se divide en bloques más sencillos y se va refinando el diseño a medida que se baja en el nivel de diseño, llegando hasta un nivel en que el que el sintetizador mapea el lenguaje de descripción de hardware (HDL) en bloques lógicos. Como lenguaje de descripción de hardware se utilizó VHDL y el procesador se implementó y simuló teniendo como objetivo una tarjeta FPGA Spartan3E XC3S500E de Xilinx.

## 6. Resultados

Se obtuvo como producto el procesador SUBLEQ diseñado. En la tabla 1 se ofrece una comparativa entre el microprocesador *soft-core* SUBLEQ desarrollado y el microprocesador *soft-core* Picoblaze de Xilinx, que es un procesador también de 8 bits, con 16 registros de propósito general y un set de instrucciones (ISA) compuesto por 57 instrucciones.

Se puede observar que la máxima frecuencia de reloj es similar, más aún si consideramos que Picoblaze generalmente no puede trabajar a su frecuencia máxima sin tener algunos problemas de latencia, consiguiéndose en la tarjeta objetivo frecuencias máximas de reloj de entorno a 87 MHz. Sin embargo el área ocupada en la FPGA en el procesador SUBLEQ es de solo de 15 *slices*, 6 veces

**Tabla 1.** Comparativa entre el procesador desarrollado y el Picoblaze de Xilinx.

Procesador	Bits datos	Bits instrucciones	Máx. frec.	Slices	BRAM	Ciclos/instr.	ISA
Picoblaze	8	18	125.3 MHz	96	1	2	57
SUBLEQ	8	24	101.5 MHz	15	2	2	1

menor que las 96 que requiere Picoblaze. Aunque ambos procesadores necesitan 2 ciclos de reloj por instrucción, hay que recordar que el procesador desarrollado solo tiene una instrucción, por lo que para realizar la misma operación que Picoblaze necesitará realizar dicha instrucción varias veces. Por ejemplo para realizar la instrucción LOAD sX,sY de Picoblaze se necesita realizar la instrucción SUBLEQ varias veces:

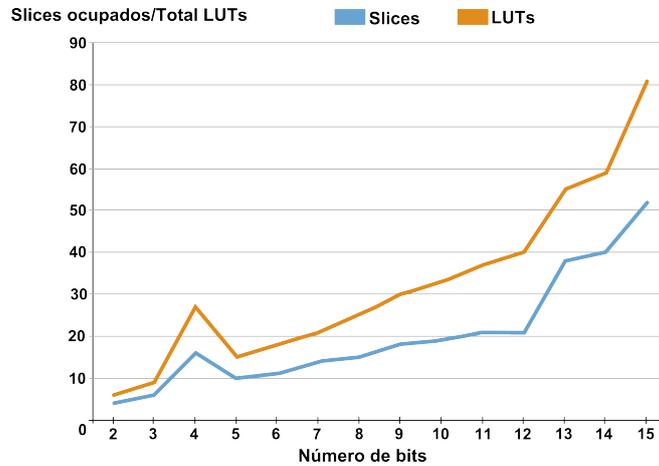
```
SUBLEQ sX sX
SUBLEQ sY Z
SUBLEQ Z sX
SUBLEQ Z Z
```

Siendo sX la dirección de memoria (que puede ser un registro mapeado en memoria) que recibe el dato que hay en la dirección de memoria sY y Z una dirección de memoria que comienza y termina con un valor nulo. Se omite el tercer operando de la instrucción SUBLEQ porque sería una dirección de memoria con el valor de la posición de la siguiente instrucción, de modo que independientemente del resultado de la substracción la ejecución siga un orden secuencial. De modo que dicha instrucción por ejemplo se ejecutará en 8 ciclos de reloj en el procesador desarrollado, frente a los 2 ciclos de reloj en Picoblaze.

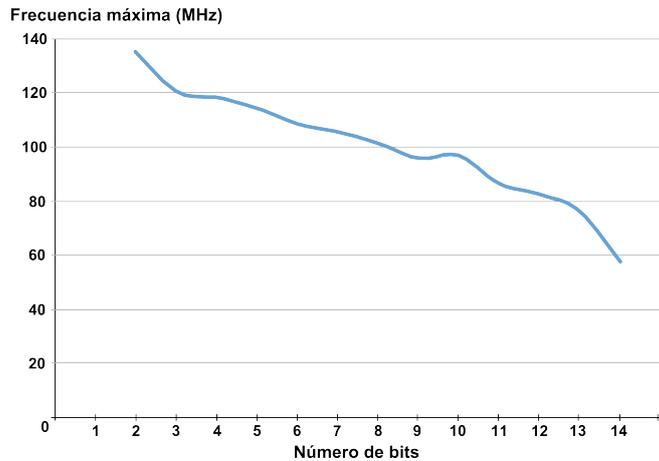
Sin embargo hay que considerar que la sencillez del procesador desarrollado permite implementar 6 procesadores en paralelo en el mismo área de silicio que un único Picoblaze. Es cierto que cada procesador SUBLEQ utiliza dos bloques BRAM (memoria embebida en las FPGA de Xilinx) en vez de uno. Sin embargo utilizando memoria distribuida (memoria implementada en LUTs de la FPGA) para realizar la memoria ROM, todavía se obtiene un procesador de 30 *slices*, 3 veces menos que Picoblaze.

Viendo el pequeño tamaño del procesador desarrollado se decidió probar a desarrollar versiones con un tamaño de palabra de datos distintos. Inicialmente se trató de dotar a dichos procesadores de una memoria RAM lo más grande posible de acuerdo al tamaño de palabra ( $n$ ):  $n \times 2^n - 1$ . Con memorias ROM de  $3n \times 2^n - 1$ . Sin embargo los 20 BRAM con los que cuenta la tarjeta objetivo ya no eran suficientes para implementar un procesador con un tamaño de palabra de 15 bits. En las figuras 3 y 4 se muestra la evolución del área utilizada así como de la frecuencia máxima de reloj.

Posteriormente se decidió limitar el tamaño de la memoria RAM a  $(n \times 2^8 - 1)$ , así como de la memoria ROM a  $(3n \times 2^8 - 1)$ . En las figuras 5 y 6 se muestra la evolución del área utilizada, así como de la frecuencia máxima de reloj.

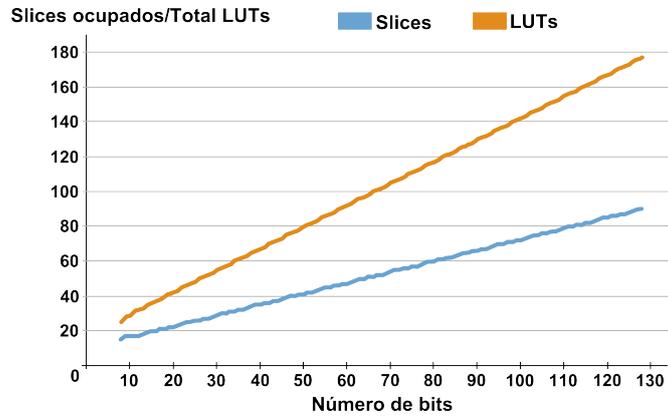


**Fig. 3.** Evolución del área (en LUTs y *slices*) en función del tamaño de palabra del procesador.

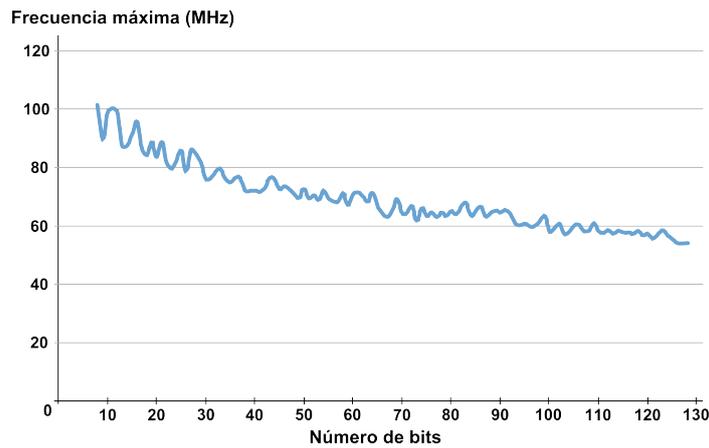


**Fig. 4.** Evolución de la frecuencia máxima de reloj en función del tamaño de palabra del procesador.

Se puede observar que incluso con un tamaño de palabra de datos de 128 bits se consigue un procesador de solo 90 *slices* (más pequeño que picoblaze aunque no en bloques BRAM) y una frecuencia máxima de reloj superior a 50 MHz.



**Fig. 5.** Evolución del área (en LUTs y *slices*) en función del tamaño de palabra del procesador limitando la memoria.



**Fig. 6.** Evolución de la frecuencia máxima de reloj en función del tamaño de palabra del procesador limitando la memoria.

## 7. Conclusiones y trabajo futuro

Los resultados apuntan a que pueden existir algunas aplicaciones donde el procesador desarrollado pueda tener un buen desempeño. Por un lado parece que en aplicaciones donde limitar el área sea importante, un procesador de este tipo podría funcionar bien; es necesario sin embargo profundizar en el desempeño del procesador SUBLEQ desarrollado, para lo que sería necesario comparar la velocidad de resolución de algunos algoritmos tanto en Picoblaze como en el

procesador SUBLEQ. Para ello es necesario contar con herramientas de ensamblado y/o compilación (como las utilizadas en [10]) adaptadas al procesador.

Cabe destacar que limitando el tamaño de la memoria se consiguen tamaños de palabra muy altos con pocos recursos. Atendiendo a la taxonomía de Flynn [4] para las arquitecturas paralelas se puede sugerir que una arquitectura de tipo MIMD (en realidad SIMD porque solo existe una instrucción en el procesador desarrollado) como las implementadas en las computadoras VLIW (*Very Long Instruction Word*), también llamadas últimamente EPIC (*Explicitly Parallel Instruction Computers*), podría tener un desarrollo y resultados interesantes en el caso de una computadora SUBLEQ. Aunque una aplicación de ese estilo requeriría esfuerzos adicionales en el compilador para evitar dependencias.

Por otro lado, la sencillez de personalización del procesador desarrollado deja entrever la posibilidad de desarrollar aplicaciones reconfigurables, donde el propio compilador decida en función del software a ejecutar, cuales serán las características del procesador (tamaño de la memoria, tamaño de palabra ...) y este se configure antes de cargar el software.

Por último decir que es necesario comparar Picoblaze y el procesador SUBLEQ desarrollado en FPGA más potentes como Artix o Kintex.

**Agradecimientos.** Esta investigación ha sido financiada con el proyecto SIP 20171871 de la Secretaría de Investigación y Posgrado del Instituto Politécnico Nacional.

## Referencias

1. Böhm, C., Jacopini, G.: Flow diagrams, turing machines and languages with only two formation rules. *Communications of the ACM* 9(5), 366–371 (1966)
2. Chatterjee, A., Sengupta, I.: Furisc: Fhe encrypted urisc design. *IACR Cryptology ePrint Archive* 2015, 699 (2015)
3. Davidar: <https://github.com/davidar/subleq> (2009), <https://github.com/davidar/subleq>
4. Flynn, M.J., Rudd, K.W.: Parallel architectures. *ACM Computing Surveys (CSUR)* 28(1), 67–70 (1996)
5. Gilreath, W., Laplante, P.: A one instruction set architecture for genetic algorithms. In: *Biocomputing*, pp. 91–113 (2004)
6. Gilreath, W.F., Laplante, P.A.: *Computer architecture: A minimalist perspective*, vol. 730. Springer Science & Business Media (2003)
7. Laplante, P., Gilreath, W.: One instruction set computers for image processing. *Journal of VLSI signal processing systems for signal, image and video technology* 38(1), 45–61 (2004)
8. Mavaddat, F., Parhami, B.: Urisc: the ultimate reduced instruction set computer. *International Journal of Electrical Engineering Education* 25(4), 327–334 (1988)
9. Mazonka, O.: Bit copying: The ultimate computational simplicity. *Complex Systems Journal* 19, N3, 263–285 (2011)
10. Mazonka, O., Kolodin, A.: A simple multi-processor computer based on subleq. *arXiv preprint arXiv:1106.2593* (2011)

11. Mazonka, O., Tsoutsos, N.G., Maniatakos, M.: Cryptoleq: A heterogeneous abstract machine for encrypted and unencrypted computation. *IEEE Transactions on Information Forensics and Security* 11(9), 2123–2138 (2016)
12. van der Poel, W.L.: The logical principles of some simple computers. Ph.D. thesis, University of Amsterdam (1956)
13. Turing, A.M.: On computable numbers, with an application to the entscheidungsproblem. *Proceedings of the London mathematical society* 2(1), 230–265 (1937)
14. Xilinx: Microblaze Processor Reference Guide (2006), [https://www.xilinx.com/support/documentation/sw\\_manuals/mb\\_ref\\_guide.pdf](https://www.xilinx.com/support/documentation/sw_manuals/mb_ref_guide.pdf)
15. Xilinx: Picoblaze 8-bit Embedded Microcontroller User Guide (2011), [https://www.xilinx.com/support/documentation/ip\\_documentationug129.pdf](https://www.xilinx.com/support/documentation/ip_documentationug129.pdf)



## **Lagarto I – Una plataforma hardware/software de arquitectura de computadoras para la academia e investigación**

Cristóbal Ramírez, César Hernández, Carlos Rojas Morales,  
Gustavo Mondragón García, Luis A. Villa, Marco A. Ramírez

Instituto Politécnico Nacional, Centro de investigación en Computación,  
México

ralc\_@hotmail.com, {hdzces, carlosrojo3000}@gmail.com,  
gustavomog@hotmail.com, {lvilla, mars}@cic.ipn.mx

**Resumen.** En la actualidad, la enseñanza del diseño de arquitectura de procesadores sigue siendo una tarea compleja, debido a la cantidad de información que se maneja y a la falta de herramientas enfocadas a condensar esta información, haciéndola más digerible para el estudiante. En este sentido, el grupo de arquitectura de computadoras del centro de Investigación en Computación del Instituto Politécnico Nacional trabaja en un proyecto llamado "Lagarto" con el objetivo de generar una plataforma de cómputo abierta para la academia e investigación, que facilite la comprensión de conceptos fundamentales de Arquitectura de Computadoras y Sistemas Operativos. Este trabajo presenta la arquitectura de un núcleo llamada Lagarto I, el cual es un procesador escalar segmentado de 32-bits que ejecuta el set de instrucciones MIPS32 R6 para operaciones de tipo entero y punto flotante. La arquitectura está descrita utilizando Verilog HDL y es sintetizable en un FPGA. Así mismo, se abordan las diferentes maneras en que es posible probar la arquitectura con códigos escritos en lenguaje C o ensamblador.

**Palabras clave:** arquitectura de computadoras, diseño de procesadores, compiladores, lenguaje ensamblador, FPGA.

### **Lagarto I - A Computer Architecture Hardware/Software Platform for Academia and Research**

**Abstract:** Nowadays, the teaching of processor architecture design continues to be a complex task, due to the amount of information that is handled and the lack of tools focused on condensing this information, making it easier to understand for the student. In this sense, the computer architecture research team at the Centro de Investigación en Computación of the Instituto Politécnico Nacional is working on a project called "Lagarto" with the aim of generating an open computing platform for academia and research to ease the understanding of fundamental concepts of Computer Architecture and Operating Systems. This

paper presents the architecture of the Lagarto I core, a 32-bit pipelined scalar processor that executes the MIPS32 R6 instruction set for integer and floating-point operations. The architecture is described using Verilog HDL and is synthesizable in an FPGA. Likewise, the different ways in which it is possible to test the architecture, with codes written in C or assembly language, are addressed.

**Keywords:** computer architecture, processor design, compilers, assembly language, FPGA.

## 1. Introducción

Hoy en día, el curso de Arquitectura de Computadoras sigue siendo uno de los cursos fundamentales en la formación de estudiantes de Ciencias de la Computación o Ingeniería de Cómputo. Comprender el comportamiento de un procesador durante la ejecución de un programa no es una tarea fácil, desafortunadamente, a lo largo del curso, los estudiantes se enfrentan con la necesidad de invertir una gran cantidad de tiempo intentando comprender conceptos abstractos, que pueden ir desde la codificación de un programa en alto nivel, pasando por las etapas de compilación hasta llegar a código máquina, terminando con la ejecución de instrucciones en el *pipeline* de un procesador, entorpeciendo el proceso de asimilación de los contenidos del curso, y por consiguiente, no llegar a cumplir con los objetivos planteados. Una solución a esta problemática, es la de facilitar al estudiante una herramienta que permita visualizar el flujo de información a lo largo de las etapas del *pipeline* del procesador: señales de control decodificadas, error en la predicción de saltos, errores de cache, entre otros.

En este sentido, se han propuesto diferentes metodologías para la enseñanza de Arquitectura de Computadoras, partiendo del hecho que la enseñanza se vuelve ineficiente si los métodos son enfocados únicamente en aspectos teóricos. Algunas propuestas plantean el uso de simuladores de arquitecturas de computadoras con el fin de comprender los conocimientos teóricos con una experiencia práctica. Sin embargo, por experiencia sabemos que esta práctica no cumple con el objetivo fundamental del curso: el diseño de procesadores; ya que, desde la perspectiva del estudiante, la brecha que existe entre lo visto en clase y un procesador real puede ser muy grande.

Para afrontar este problema, universidades como UC Berkeley y el MIT han desarrollado una plataforma abierta llamada RISC-V [3], que originalmente surge para fortalecer los conceptos fundamentales de arquitectura de computadoras tanto en la enseñanza como en la investigación, y que ha llevado al desarrollo de diferentes arquitecturas [4]. Actualmente, departamentos de arquitectura de computadores de diferentes universidades como lo son la Universidad de Manchester y la Universidad Politécnica de Cataluña, han hecho uso de esta plataforma para la impartición de cursos y realizar investigación sobre arquitecturas multi-núcleos, NoC, etc. LEON es otra propuesta de un procesador de 32-bit que ejecuta el ISA SPARK V8, desarrollada en VHDL. El modelo es altamente reconfigurable y enfocado para diseños de *System-on-*

Chip (SoC). El código fuente de este procesador se encuentra disponible y con uso ilimitado para investigación y educación [5, 6].

En las referencias antes mencionadas se observa que, el proceso pedagógico en la enseñanza de Arquitectura de Computadoras parte desde un supuesto en donde el alcance del curso no es la implementación física del diseño del procesador. Se omite todo el conocimiento que se genera al llevar cualquier modelo teórico-simulado al mundo real. En un mundo en donde el desarrollo tecnológico y la innovación marcan la diferencia en el desarrollo de la sociedad, llegar hasta la última milla, en donde la idea se convierte en un producto tangible, proporciona el conocimiento que conduce hacia una sociedad emprendedora.

En este trabajo se describe el núcleo de un SoC de código abierto llamado Lagarto I [7], desarrollado para facilitar la comprensión del diseño arquitectural y el comportamiento de un procesador RISC de 32-bit segmentado desde diferentes perspectivas: Diseño, Implementación, Simulación y Validación.

El resto del documento está organizado de la siguiente manera. En la Sección 2 se describe la microarquitectura del procesador Lagarto I. La Sección 3 discute las diferentes maneras en las que se puede usar y probar la arquitectura. La Sección 4 muestra los resultados obtenidos al ejecutar diferentes programas en el procesador. Finalmente, las conclusiones y trabajo futuro son presentados en la Sección 5.

## 2. Microarquitectura del procesador Lagarto I

Desde una perspectiva general, Lagarto I puede ser visto como un procesador de 5 etapas (Fig. 1):

- Búsqueda y extracción de instrucciones,
- Decodificación,
- Lectura de registros/Emisión,
- Ejecución,
- Escritura de Retorno.

**Tabla 1.** Unidades de ejecución / etapas.

Unidad de ejecución	Etapas
Enteros	1
Load/Store	2
Branch	1
FP – Operaciones Simples	1
FP – Operaciones Complejas	4
FP – Operaciones Complejas	12

Sin embargo, a partir de la etapa de Lectura de registros la instrucción es emitida a su unidad de ejecución correspondiente, siempre y cuando ésta cumpla con los

requerimientos para ser emitida. Se tienen diferentes unidades de ejecución con diferente latencia dentro de la arquitectura, tal y como se muestra en la Tabla 1.

El diseño contempla unidades de ejecución de Punto Flotante de alto rendimiento [8] que cumplen con el estándar IEEE 754 [9].

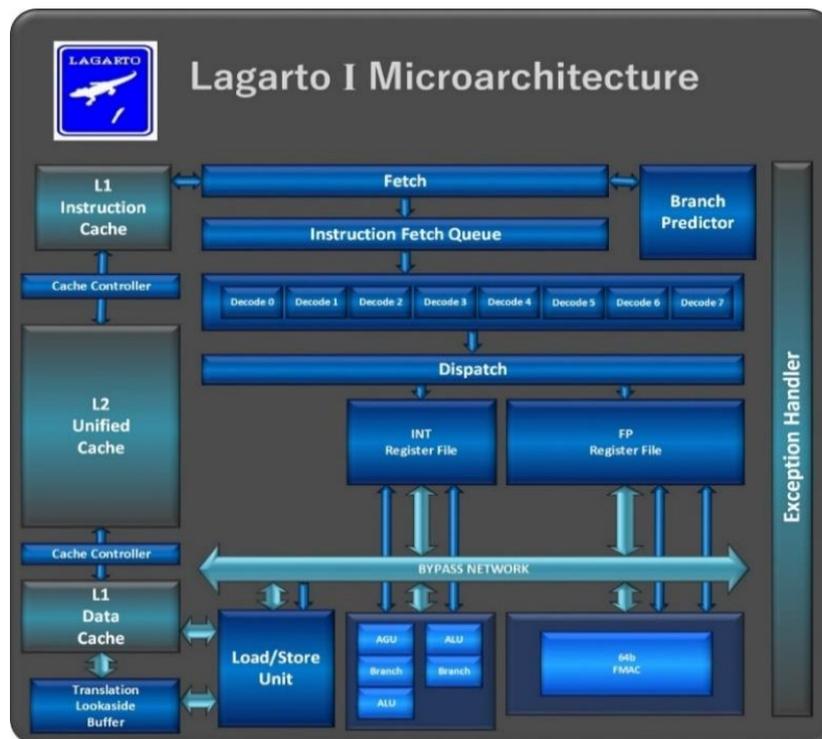


Fig. 1. Microarquitectura del procesador Lagarto I.

Lagarto I hace uso de un predictor dinámico y una red de *bypass* completa para adelantar datos entre diferentes unidades funcionales: unidad de enteros, unidad de punto flotante, unidad de load/store, entre otras, mejorando el rendimiento del procesador.

El diseño modular del procesador Lagarto I, permite al estudiante intercambiar diferentes bloques, pudiendo haber sido desarrollados por ellos mismos, como lo pueden ser diferentes predictores de saltos para evaluar rendimiento de cada uno de ellos, aumentar o disminuir los tamaños de las memorias caché.

### 3. Formas de usar y probar Lagarto I

Existen diferentes maneras de usar y probar Lagarto I. La Fig. 2 muestra en la parte izquierda las formas en las que se puede escribir código, ya sea usando un compilador de "C", o bien directamente en código ensamblador. Ambos códigos son traducidos a

código máquina, siendo éste el que es interpretado por el procesador (unos y ceros), y dichos códigos son cargados a las memorias caché del procesador para así poder ser ejecutados; del lado derecho se muestran dos interfaces en las cuales se pueden ver los resultados que arroja el procesador.

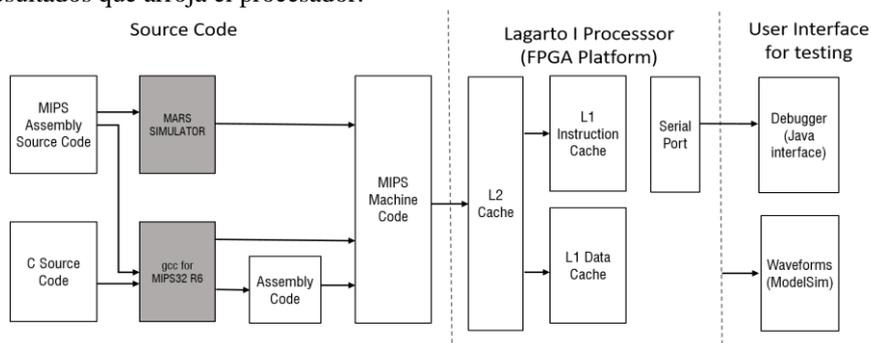


Fig. 2. Entorno de uso de Lagarto I.

### 3.1 Programando en lenguaje Ensamblador

Para escribir código en MIPS ensamblador para Lagarto I, es utilizado el simulador para Lagarto I, es utilizado el simulador MARS, desarrollado en la Universidad Estatal de Missouri; este es un entorno interactivo de desarrollo (IDE) para programación en lenguaje ensamblador MIPS y enfocado para la educación [10].

### 3.2 Programando en lenguaje C

Una segunda opción de generar código máquina para Lagarto I, es escribir código en lenguaje C y compilarlo para arquitecturas MIPS. Lagarto I utiliza el ISA MIPS32 R6, por lo que el compilador debe soportar dicho ISA que corresponde al más actualizado a la fecha. Codescape MIPS SDK [11] provee un compilador completo y un *toolchain* para desarrollo MIPS.

### 3.3 Core Simulation

Es posible utilizar el *soft-core* Lagarto I sin necesidad de ser sintetizado en un FPGA al simularlo en ModelSim, un entorno de simulación multi-lenguaje HDL. Dentro de este entorno es posible correr un banco de pruebas y utilizar una plantilla personalizada (archivo *.do*) para visualizar cada una de las etapas del procesador.

### 3.4 Depurador Lagarto I

Depurador Lagarto I es una interfaz gráfica para el usuario, desarrollada en el Centro de Investigación en Computación del Instituto Politécnico Nacional, la cual permite visualizar los datos computados dentro del procesador Lagarto I cuando se encuentra

sintetizado en un FPGA. Esta aplicación cuenta con diferentes características como lo son: control de ejecución paso a paso, control para disminuir o aumentar la frecuencia de operación del procesador, permite modificar el contenido de las memorias cache mediante la carga de nuevos programas o datos, entre otras.

#### 4. Pruebas y resultados

Para verificar el correcto funcionamiento del procesador, se codifica un programa en lenguaje ensamblador que calcula el valor de PI mediante el algoritmo de Chudnovsky, el cual es un método rápido para calcular los dígitos de PI. En 2011 se calcularon los primeros 10 billones de dígitos haciendo uso de este algoritmo. Dicho algoritmo es elegido debido a la variedad de las operaciones aritméticas de las que hace uso, así como de operaciones de conversión de entero a punto flotante, movimientos entre bancos de registros, gestionando una gran cantidad de dependencias de datos entre instrucciones, lo que permite comprobar el correcto funcionamiento de distintos módulos de la arquitectura. El programa principal hace llamadas a otras funciones como potencia, raíz cuadrada y factorial, pudiéndose validar el correcto funcionamiento del predictor de saltos y el control implementado en el pipeline del procesador.

Dicho programa fue compilado y ejecutado en el software MARS y el resultado final es mostrado en la Fig. 3, en el registro \$f8.

Name	Float	Double
\$f0	0x00000000	0x0000000000000000
\$f1	0x00000000	0x0000000000000000
\$f2	0x4df452e1	0x400000004df452e1
\$f3	0x40000000	0x4000000000000000
\$f4	0x3a83126f	0x5c692e193a83126f
\$f5	0x5c692e19	0x5c692e1900000000
\$f6	0x41400000	0x3f80000041400000
\$f7	0x3f800000	0x3f80000000000000
\$f8	0x40490fda	0x0000000040490fda
\$f9	0x00000000	0x0000000000000000
\$f10	0x00000000	0x0000000000000000
\$f11	0x00000000	0x0000000000000000
\$f12	0x00000000	0x0000000000000000
\$f13	0x00000000	0x0000000000000000
\$f14	0x00000000	0x0000000000000000
\$f15	0x00000000	0x0000000000000000
\$f16	0x00000000	0x0000000000000000
\$f17	0x00000000	0x0000000000000000
\$f18	0x00000000	0x0000000000000000
\$f19	0x00000000	0x0000000000000000
\$f20	0x3f800000	0x3f8000003f800000
\$f21	0x3f800000	0x3f80000000000000
\$f22	0x4b4f6371	0x3f8000004b4f6371
\$f23	0x3f800000	0x3f80000000000000
\$f24	0x3f800000	0x4df452e13f800000
\$f25	0x4df452e1	0x4df452e100000000
\$f26	0x00000000	0x0000000000000000
\$f27	0x00000000	0x0000000000000000
\$f28	0x42000000	0x5080000042000000
\$f29	0x50800000	0x5080000000000000
\$f30	0x4e7452e2	0x4df452e14e7452e2
\$f31	0x4df452e1	0x4df452e100000000

Fig. 3. Banco de registros de punto flotante en el software MARS.

En la Fig. 4 se aprecia la conversión del valor de PI tomando en cuenta los primeros 25 decimales al formato de Punto Flotante precisión simple (*float* en lenguaje C), que corresponde al obtenido con el simulador MARS bit a bit (**40490FDA**).

### Conversión Punto Flotante IEEE-754

De Punto Flotante a su representación en 32-bit y 64-bit Hexadecimal y sus binarios equivalentes.

Ingresar un número Real aquí, luego haga click en "Con Redondeo" o en "Sin Redondear" para ver el resultado de la conversión.

Número decimal:

---

**Precisión simple (32 bits):**

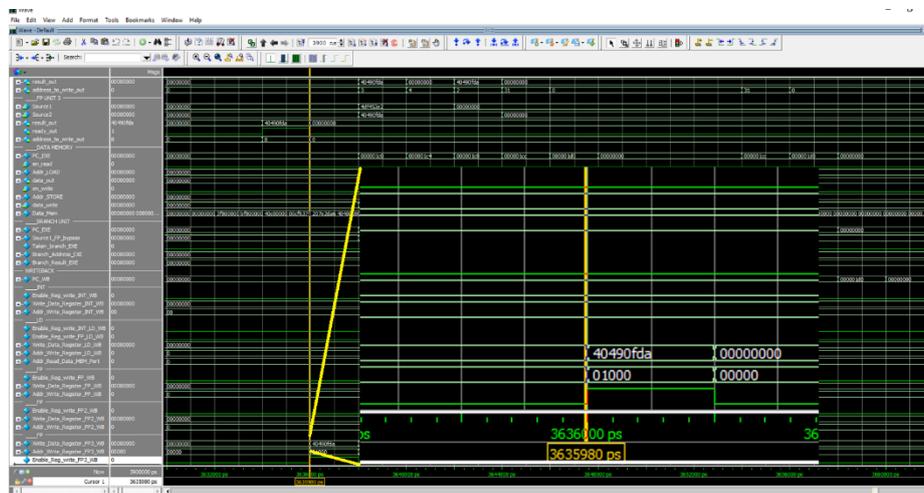
Binario:

Bit 31 Bit de Signo 0 0: + 1: -	Bits 30 - 23 Exponente 10000000 Valor decimal del campo exponente y exponente 128 - 127 = 1	Bits 22 - 0 Significado 1.10010010000111111011010 Valor decimal del significado 1.5707963
---	---	---

Hexadecimal:  Decimal:

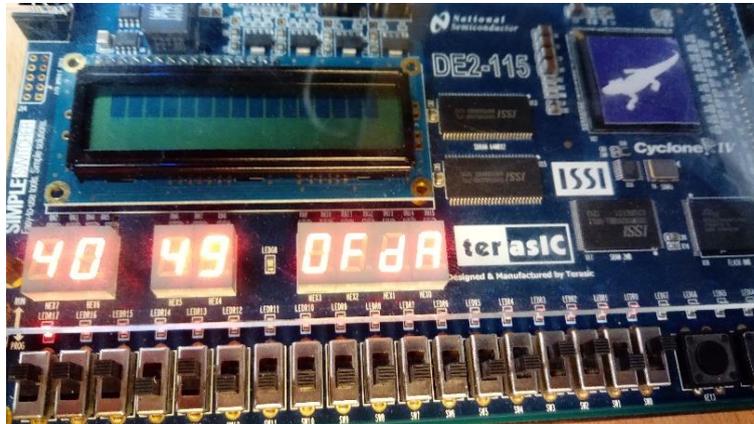
**Fig. 4.** Conversión del valor de PI a formato de Punto Flotante.

En la simulación de la ejecución de este programa sobre Lagarto I se obtienen los resultados mostrados en la Fig. 5, validando el correcto funcionamiento del diseño.



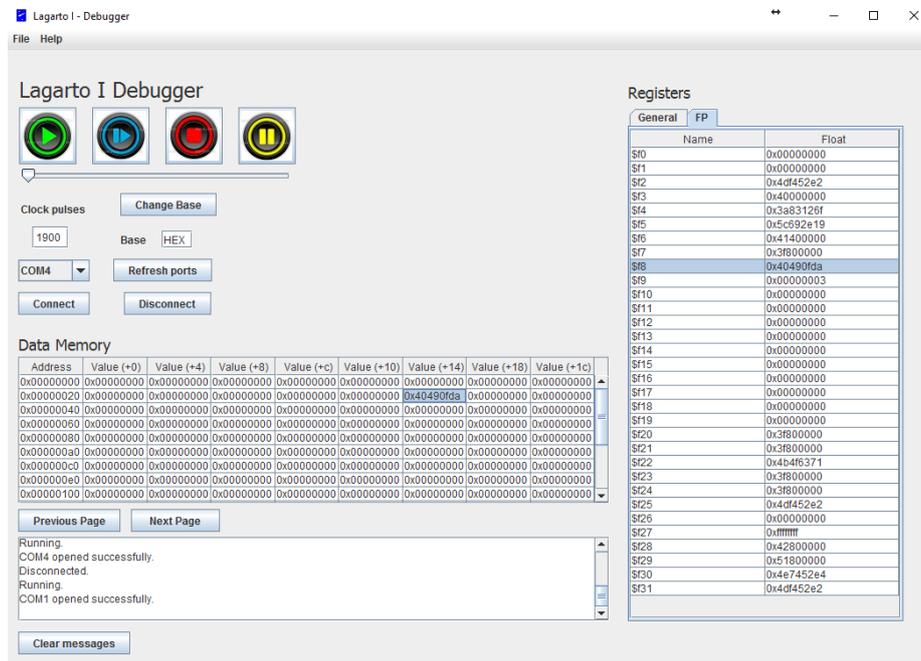
**Fig. 5.** Resultado de la ejecución del programa que realiza el cálculo de PI utilizando el Simulador de Altera ModelSim.

Adicionalmente, el proyecto es sintetizado usando una tarjeta de desarrollo DE2-115 de Altera, e inicializando las memorias de datos e instrucciones con el programa a ejecutar. Los resultados se muestran en la Fig. 6.



**Fig. 6.** Resultado de la ejecución del programa que realiza el cálculo de PI sobre el procesador Lagarto I sintetizado en el FPGA.

Posteriormente, se visualiza el cómputo del programa paso a paso para monitorear los datos almacenados en el banco de registros y la memoria de datos haciendo uso del Depurador diseñado para la arquitectura Lagarto. El resultado calculado en el FPGA se puede visualizar en la interfaz gráfica, tal y como se muestra en la Fig. 7.



**Fig. 7.** Resultado de la ejecución del programa que realiza el cálculo de PI sobre el procesador Lagarto I sintetizado en el FPGA conectado a la interfaz gráfica.

Por otro lado, el algoritmo es codificado en lenguaje C, y compilado para una arquitectura MIPS, como se muestra en la Fig. 8.

```
cristobal@RALC: ~/Documents/MIPS/PI
cristobal@RALC:~/Documents/MIPS/PI$ $MIPS_ELF_ROOT/bin/mips-imp-elf-gcc -mabi=32 -mips32r6 -EB -lm -c PI.c -o PI.o
cristobal@RALC:~/Documents/MIPS/PI$ $MIPS_ELF_ROOT/bin/mips-imp-elf-gcc -mabi=32 -EB -T uhi32.ld PI.o -o PI.elf -lm
cristobal@RALC:~/Documents/MIPS/PI$ $MIPS_QEMU/qemu-system-mips64 -nographic -no-shutdown -semihosting -cpu I6400 -k
ernel PI.elf -singlestep
La aproximación de PI es: 3.141592741012573242187500000000
cristobal@RALC:~/Documents/MIPS/PI$
```

Fig. 8. Proceso de Compilación para una arquitectura MIPS.

Se puede apreciar que un procesador de 32 bits logra obtener los primeros 6 dígitos exactos del valor de pi, mientras el resto es una aproximación al valor verdadero.

En la Fig. 9, se muestran los archivos creados en el proceso de compilación en donde primero se creó el archivo objeto (.o), posteriormente el archivo (.elf) que es un ejecutable para arquitecturas MIPS tal y como se muestra en la Fig. 10. El archivo (.s) contiene el programa en ensamblador correspondiente al código escrito en C.

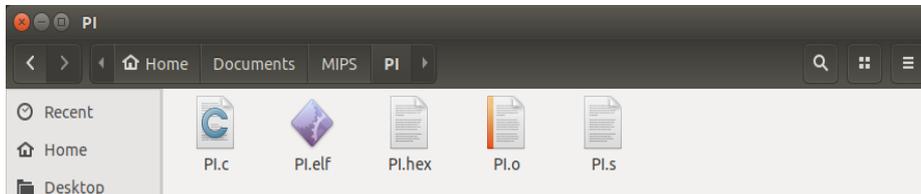


Fig. 9. Archivos creados en la compilación para una arquitectura MIPS.

```
cristobal@RALC: ~/Documents/MIPS/PI
cristobal@RALC:~/Documents/MIPS/PI$ file PI.elf
PI.elf: ELF 32-bit MSB executable, MIPS, MIPS32 rel6
version 1 (SYSV), statically linked, not stripped
cristobal@RALC:~/Documents/MIPS/PI$
```

Fig. 10. Archivo ejecutable para una arquitectura MIPS.

Finalmente, haciendo uso del comando *hexdump* sobre el ejecutable obtenido al realizar la compilación para MIPS, se obtienen los archivos para inicializar a las memorias caché del procesador y ejecutar el programa en Lagarto I.

## **5. Conclusiones y trabajo futuro**

En este trabajo es presentada la microarquitectura del procesador Lagarto I, una plataforma abierta enfocada en facilitar la comprensión y el uso de procesadores para investigación y la academia, aprovechando los beneficios de los FPGA y los lenguajes HDL. Lagarto I es parte de un *System-on-Chip* (SoC) que se encuentra bajo desarrollo en el centro de Investigación en Computación, y el cual ha abierto las puertas al desarrollo de diferentes proyectos como redes de interconexión, extensiones multimedia, controladores de periféricos, entre otros. Finalmente, Lagarto I como herramienta educativa, está siendo introducida impartiendo talleres en diferentes congresos y universidades en México, despertando el interés por parte de profesores y estudiantes.

## **Referencias**

1. M. b. Imagination, MIPS Architecture for Programers. Volume I-A: Introduction to the MIPS32 Architecture, Revision 6.01 (2014)
2. M. b. Imagination, MIPS Architecture for Programers. Volume II-A: The MIPS32 Instruction Set, Revision 6.01 (2014)
3. RISC-V Project Homepage, <https://riscv.org/>, last accessed 2017/07/21
4. Raveendran, A., Baramu, V.: A RISC-V Instruction Set Processor-Micro-architecture Design and Analysis. International Conference on VLSI Systems, Architectures, Technology and Applications (VLSI-SATA), Bangalore (2016)
5. LEON3 Processor. <http://www.gaisler.com/index.php/products/processors/leon3>, last accessed 2017/07/25
6. LEON4 Processor, <http://www.gaisler.com/index.php/products/processors/leon4ft>
7. Lagarto Project Homepage, <http://www.lagarto.org>, last accessed 2017/07/21
8. Ramírez, C.: Design and implementation of an out of order execution engine of floating point arithmetic operations. Master Thesis, Universidad Politécnica de Cataluña, Barcelona, Spain (2015)
9. IEEE Standard for Floating-Point Arithmetic. IEEE Std 754(2008), 1–70 (2008)
10. MARS, MIPS assembler and Runtime Simulator. <http://courses.missouristate.edu/KenVollmar/mars/> last accessed 2017/07/25
11. Codescape MIPS SDK, <https://community.imgtec.com/developers/mips/tools/codescape-mips-sdk/>, last accessed 2017/07/25

# Reconstrucción de profundidad a partir de una sola imagen con perspectiva mediante redes neuronales completamente convolucionales

José E. Valdez Rodríguez, Hiram Calvo, Edgardo M. Felipe Riverón

Instituto Politécnico Nacional, Centro de Investigación en Computación,  
Ciudad de México, México

jvaldezr1000@alumno.ipn.mx, {hcalvo, edgardo}@cic.ipn.mx

**Resumen.** La reconstrucción de la profundidad a partir de una sola imagen ha sido una tarea difícil debido a la complejidad y la cantidad de indicios de profundidad que puede contener una sola imagen. Las Redes Neuronales Convolucionales (RNC) se han utilizado con éxito para reconstruir la profundidad de objetos en escenas generales; sin embargo, estos trabajos no se han adaptado para el problema particular de la reconstrucción de la profundidad a partir de la perspectiva. Nuestra propuesta se basa en construir una RNC computacional eficiente; nos centramos en Redes Neuronales Completamente Convolucionales (RNCC), cuyo entrenamiento requiere de una sola etapa. Comparamos nuestra RNC con el estado del arte actual en reconstrucción de profundidad, obteniendo mejoras en niveles globales para imágenes viales con perspectiva presente.

**Palabras clave:** Reconstrucción de profundidad, redes neuronales completamente convolucionales, RNCC, emparejamiento estereoscópico.

## Reconstruction of Depth from a Single Perspective Image through Completely Convolutional Neural Networks

**Abstract.** The reconstruction of the depth from a single image has been a difficult task due to the complexity and the amount of depth indications that a single image can contain. Convolutional Neural Networks (RNC) have been used successfully to reconstruct the depth of objects in general scenes; however, these works have not been adapted for the particular problem of depth reconstruction from perspective. Our proposal is based on building an efficient computational RNC; we focus on Completely Convolutional Neural Networks (RNCC), whose training requires a single stage. We compared our RNC with the current state of the art in depth reconstruction, obtaining improvements in global levels for road images with present perspective.

**Keywords:** Depth Reconstruction, Completely Convolutional Neural Networks, RNCC, Stereoscopic Pairing.

## 1. Introducción

La reconstrucción de la profundidad a partir de una sola imagen (opuesto a las imágenes estereoscópicas) es una tarea difícil debido a la cantidad de señales de profundidad que una sola imagen puede contener, tales como sombras, perspectiva, imagen borrosa por el movimiento [12]. En este trabajo nos enfocamos en reconstruir la profundidad en imágenes cuya perspectiva es dominante; un ejemplo pueden ser imágenes de autopistas o calles. Este tipo de imágenes necesitan ser analizadas por sistemas autónomos tales como robots o automóviles los cuales solamente cuentan con una representación 2D para calcular su desplazamiento [4,14,10]. Diversos trabajos previos en reconstrucción de profundidad han abordado el problema general de estimar la componente de profundidad en una amplia gama de imágenes, centrándose en objetos contra un fondo sólido o complejo, o edificios de ciudades y personas.

Hasta donde sabemos, no hay trabajos específicos dedicados al problema de la reconstrucción de la profundidad utilizando RNCC. Varias arquitecturas de RNC en el estado del arte proponen redes de múltiples etapas que requieren entrenamiento separado para cada una de ellas, lo cual hace que el proceso de entrenamiento tome más tiempo. En este trabajo proponemos una arquitectura de Red Neuronal Completamente Convolutiva (RNCC) capaz de reconstruir la profundidad a partir de una sola imagen, utilizando únicamente capas convolucionales, y por tanto requiere sólo una etapa de entrenamiento. La arquitectura propuesta en este trabajo es capaz de reconstruir la profundidad a nivel global y local. La organización de este trabajo se muestra a continuación: en la sección 2 se describe el trabajo relacionado en el cual está basada esta investigación, en la sección 3 se describe el método propuesto, en la sección 4 se muestran nuestros resultados, así como la comparación con el estado del arte, y finalmente en la sección 5 se muestran las conclusiones de este trabajo.

## 2. Trabajo relacionado

El problema de reconstruir profundidad a partir de una sola imagen ha sido atacado a través de diferentes métodos. Aunque el utilizar Redes Neuronales Convolutivas (RNC) se ha convertido en una de las mejores técnicas para resolver este problema, esta técnica sigue en etapa de desarrollo debido principalmente a la complejidad en el diseño de este tipo de redes neuronales. Uno de los primeros trabajos en utilizar esta técnica se puede ver en Eigen, Puhersch y Fergus [6]. Ellos proponen el uso de dos RNC: La primera reconstruye la profundidad a nivel global y la segunda refina los detalles locales. Ellos establecen una RNC que

puede ser aplicada a la reconstrucción de profundidad y proponen una función de error.

Eigen y Fergus [5] utilizan tres RNCs entrenadas por separado. La primera reconstruye la profundidad a nivel global; la segunda trata de reconstruir la profundidad a nivel global a la mitad de resolución espacial de la imagen de entrada y la tercera refina los detalles a nivel local; además, proponen una nueva función de error para entrenar sus RNC.

Liu, Shen y Lin [18] presentan una RNC que toma como entrada una imagen pre-procesada basada en superpíxeles. Ellos complementan con un método probabilístico, en el cual tratan de mejorar los resultados. Después, del mismo equipo, Liu, Shen, Lin y Reid [19] cambia la arquitectura de su RNC manteniendo la etapa de mejoramiento de la imagen.

Finalmente, Afifi y Hellwich [2] utilizan una sola RNC configurada con regresión para reconstruir la profundidad con su propia función de error.

En general, las arquitecturas discutidas anteriormente son similares, los cambios entre ellas están basados en la función de error, las funciones de activación entre las capas de las RNCs, el número de filtros por capa y el tamaño de los filtros. Dado que los autores de los trabajos relacionados utilizan diferentes conjuntos de datos para entrenar y probar sus RNC, se dificulta la comparación directa entre el desempeño nuestra propuesta y el trabajo relacionado no dejándonos otra opción que reimplementar alguno de las RNC para poder realizar una comparación adecuada. Dentro de los trabajos que se consideran, el trabajo que obtiene los mejores resultados en el estado del arte es el de Afifi y Hellwich [2]. Además, su arquitectura está basada en una RNC con un solo paso de entrenamiento.

A pesar de que su RNC es llamada completamente convolucional (fully convolutional), al utilizar una capa *upsample* y bloques residuales, consideramos que su RNC no es puramente convolucional. Los autores presentan resultados de reconstrucción de profundidad sobre imágenes de sillas con diferente fondo y perspectiva. Dado que nos interesa evaluar el desempeño de nuestra RNCC con un mismo conjunto de imágenes en las cuales la perspectiva es un factor importante, re-implementamos la RNC de Afifi y Hellwich [2] como se ve en la Figura 1.

En la Región A se puede observar la operación de reducción, la cual es realizada a través de cuatro bloques residuales y una capa convolucional. En la región B se observa la operación *upsample* la cual trata de recuperar el tamaño original de la imagen de entrada y ésta se realiza a través de dos bloques residuales y una capa convolucional. En la sección 3 se darán más detalles acerca de estas operaciones.

En la siguiente sección presentamos nuestra propuesta. Al igual que la RNC de Afifi y Hellwich, nuestra RNCC está basada en una sola RNC, pero la diferencia principal es que nuestra RNCC está compuesta solamente de capas convolucionales y no recupera el tamaño de la imagen original (*upsampling*) sobre la misma arquitectura de la RNCC. Nuestra arquitectura tampoco emplea bloques residuales.

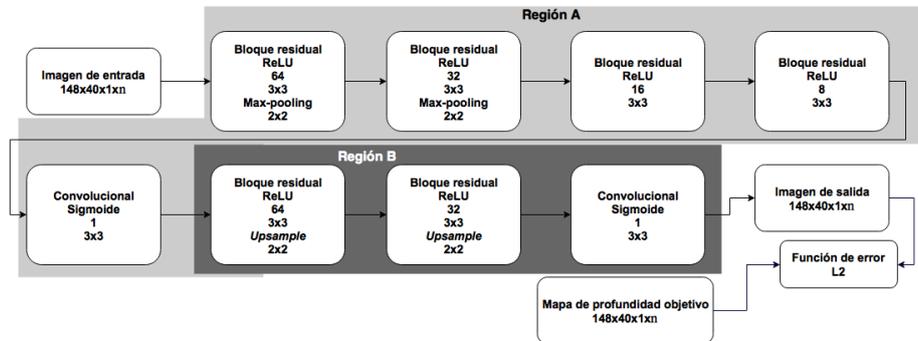


Fig. 1. Arquitectura de RNC de Affi y Hellwich [2].

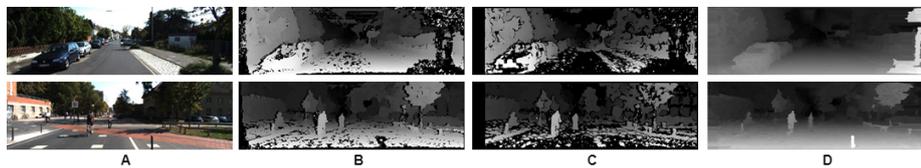


Fig. 2. Comparación entre los mapas de profundidad obtenidos con los diferentes algoritmos de emparejamiento estereoscópico, (A) Imágenes estereoscópicas, (B) *Semiglobal Matching*, (C) *Blockmatching method*, (D) *Efficient large-scale stereo matching method*.

### 3. Método propuesto

En esta sección presentamos la arquitectura de nuestra Red Neuronal Completamente Convolutiva (RNCC). Hasta donde sabemos, no hay conjuntos de datos públicos disponibles para entrenar nuestra RNCC por lo que se decidió crear un conjunto de datos a partir de la sección *Object tracking* de *The KITTI Vision Benchmark Suite* [7]. Este conjunto de datos contiene 15,000 escenas al aire libre brindadas como imágenes estereoscópicas. Este conjunto de datos no contiene los mapas de profundidad objetivo, por lo que hemos construido dicho mapa utilizando diferentes algoritmos de emparejamiento estereoscópico, tales como *Semiglobal stereo matching* [11] y *Blockmatching* [13].

En la Figura 2 podemos observar una breve comparación entre los algoritmos de emparejamiento estereoscópico probados para obtener el mapa de profundidad objetivo. Por último, hemos decidido utilizar el algoritmo *Efficient large-scale stereo matching* [8], debido a la calidad y la evaluación recibida por Menze y Geiger [20]. Este algoritmo recibe pares de imágenes estereoscópicas, como se describe en la Figura 3, y su resultado es el mapa de profundidad objetivo.

Nuestra RNCC fue construida a partir de capas Convolutivas [15] y *Max-pooling* [21]. Adicionalmente, agregamos un sesgo a cada capa de la RNCC. En la Figura 4 se puede observar la representación de cada capa en la RNCC.

Reconstrucción de profundidad a partir de una sola imagen con perspectiva mediante redes ...

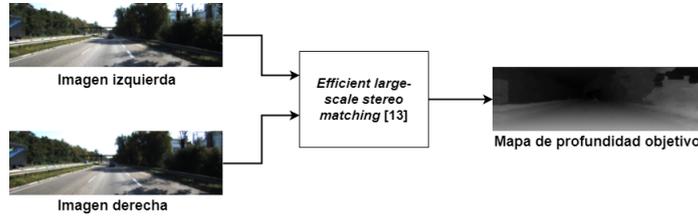


Fig. 3. Creación del mapa de profundidad objetivo.



Fig. 4. Representación de una capa de la RNC.

**RNC propuesta.** El diagrama a bloques de esta RNC se presenta en la Figura 5. Esta RNC está compuesta de cinco capas, de las cuales cuatro son capas convolucionales con función de activación *Rectified Linear Unit* (ReLU) [3] y la salida de la RNC tenemos una capa convolucional con función de activación sigmoide para limitar la salida de la RNC entre valores de 0 y 1. Esta red neuronal recibe como entrada la imagen izquierda de tamaño 147x37 pixeles en niveles de gris y como mapa de profundidad objetivo recibe el mapa de profundidad correspondiente a la entrada obtenido con el algoritmo de emparejamiento estereoscópico de tamaño 37x10 pixeles en niveles de gris. A la salida la RNC nos entrega el mapa de profundidad estimado con un tamaño de 37x10 pixeles en niveles de gris. Este tamaño se debe a que se están utilizando capas max-pooling en las dos primeras capas convolucionales.

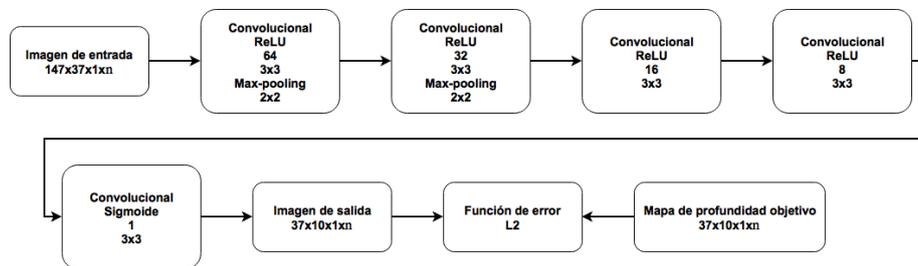


Fig. 5. Arquitectura de RNC propuesta.

**Función objetivo.** Para entrenar nuestra RNCC necesitamos minimizar el error. Esta función mide la diferencia global entre el mapa de profundidad objetivo y la reconstrucción de profundidad dada por la RNCC. Utilizamos la norma  $L2$  como la función objetivo. La norma  $L2$  se puede calcular como se muestra en la ecuación 1.

$$L2 = \frac{1}{2n} \sum_{i=1}^n \|y(i) - y'(i)\|_2^2, \quad (1)$$

donde:

$y'(i)$  = Mapa de profundidad reconstruido,  
 $y(i)$  = Mapa de profundidad objetivo,  
 $n$  = Número de imágenes por lote.

**Medidas de error.** Al evaluar nuestro método utilizamos varias medidas de error para poder compararnos con el trabajo relacionado:

*Raíz del Error Cuadrático Medio:*

$$RECM = \sqrt{\frac{1}{|T|} \sum_{y' \in |T|} (y - y')^2}. \quad (2)$$

*Error Cuadrático Medio:*

$$EMC = \frac{1}{|T|} \sum_{y' \in |T|} (y - y')^2. \quad (3)$$

*Raíz del Error Cuadrático Medio Logarítmico:*

$$RECMLOG = \sqrt{\frac{1}{|T|} \sum_{y' \in |T|} (\log(y) - \log(y'))^2}. \quad (4)$$

*Raíz del Error Cuadrático Medio Logarítmico Invariante a la Escala:*

$$RECMLOGSI = \frac{1}{|T|} \sum_{y' \in |T|} (\log(y) - \log(y'))^2. \quad (5)$$

*Diferencia Absoluta Relativa:*

$$DAR = \frac{1}{|T|} \sum_{y' \in |T|} \frac{|y - y'|}{y'}. \quad (6)$$

*Diferencia Cuadrada Relativa:*

$$DCR = \frac{1}{|T|} \sum_{y' \in |T|} \frac{\|y - y'\|^2}{y'}. \quad (7)$$

En las fórmulas:

$y'$  = Mapa de profundidad reconstruido,  
 $y$  = Mapa de profundidad objetivo,  
 $T$  = Número de píxeles en la imagen.

#### 4. Experimentos y resultados

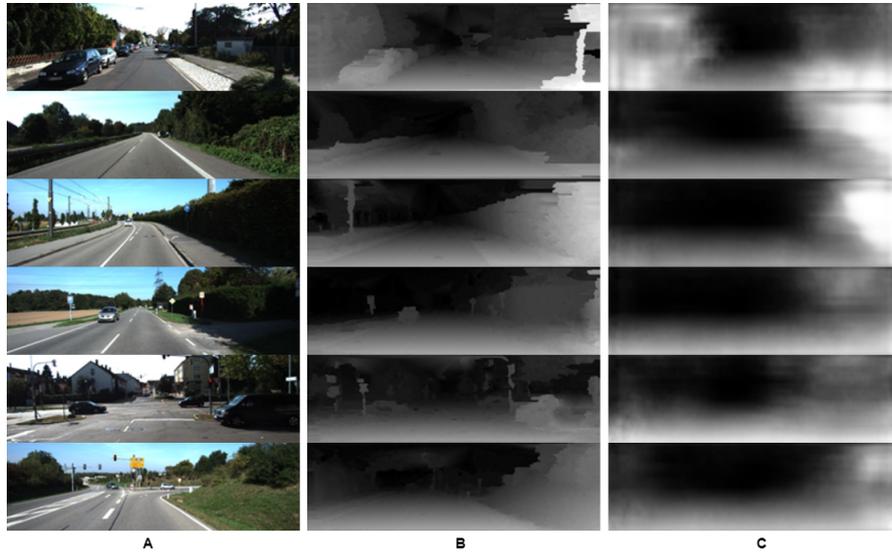
En esta sección se describirán los resultados de nuestra propuesta. Nuestra RNCC fue entrenada a través del algoritmo de Retropropagación [17] y Descenso de Gradiente Estocástico [16]. Utilizamos 1,000 épocas y un tamaño de lote de 40. Del conjunto de datos completo tomamos 12,482 imágenes, solamente del lado izquierdo con su respectivo mapa de profundidad objetivo para entrenar la RNCC, y 2,517 imágenes respectivamente para prueba.

El tamaño de las imágenes de entrada a la RNCC es de 147x37 píxeles y el tamaño de la imagen de salida es de 37x10 píxeles. Para recuperar el tamaño original de las imágenes y poder compararnos con resultados obtenidos en trabajos del estado del arte utilizamos Interpolación Bilineal [9]; con lo anterior tenemos  $T = 370$  (sin recuperar el tamaño original) y  $n = 40$  dado el tamaño del lote. Nuestra RNCC fue implementada en un entorno de trabajo de Python, *Tensorflow* [1] en el cual las RNCC pueden ser entrenadas en GPU para obtener un mejor desempeño.

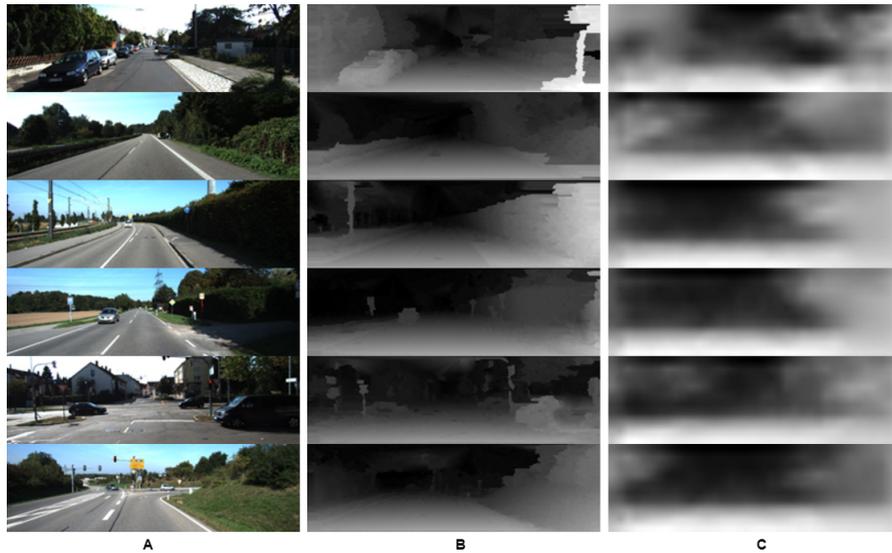
Ambas arquitecturas de RNC se entrenaron en una GPU NVIDIA GTX 960; tomó dos días realizar el entrenamiento y menos de un segundo en reconstruir la profundidad cuando la RNC recibe una sola imagen. Como se explicó en la sección 2, se implementó la RNC propuesta por Afifi y Hellwich [2] dado que ellos solamente entrenan su RNC una vez; esta RNC también puede ser entrenada con la norma  $L2$  y aun así ellos obtienen mejores resultados que el resto del trabajo relacionado sin utilizar una RNC adicional. En la Figura 6 se puede observar una muestra de los resultados obtenidos con la implementación de [2].

En la Figura 7 se presenta una muestra de los resultados obtenidos con nuestra RNCC.

Comparando cualitativamente nuestros resultados mostrados en la figura anterior, se puede observar que nuestra RNC es capaz de reconstruir la profundidad a nivel global. A nivel local, se puede observar que la RNC es capaz de mostrar más detalles que en propuestas anteriores. Por nivel global nos referimos a la imagen sin detalles pequeños y a nivel local nos referimos a los pequeños detalles que pueda contener la imagen, tales como automóviles, peatones, postes, etc. Para un análisis cuantitativo, en la Tabla 1 presentamos las medidas de error de nuestra propuesta y la RNC implementada mencionado anteriormente. Los errores REMC, EMC y DAR miden el error a nivel global de las imágenes, mientras que los errores REMCLOG, REMCLOGSI y DCR miden los errores a nivel local. Comparando nuestro método con el estado del arte, obtenemos mejores resultados en la mayoría de medidas del error a nivel global pero a nivel local se necesita mejorar.



**Fig. 6.** Resultados de la RNC de Afifi and Hellwich [2] con nuestro conjunto de datos. (A) Imagen de entrada, (B) Mapa de profundidad objetivo  $y$ , (C) Salida de la RNC  $y'$ .



**Fig. 7.** Muestra de resultados obtenidos con nuestra RNCC. (A) Imagen de entrada, (B) Mapa de profundidad objetivo  $y$ , (C) Salida de la RNCC  $y'$ .

**Tabla 1.** Comparación de resultados entre nuestra propuesta y el estado del arte.

	RECM	ECM	RECMLOG	RECMLOGSI	DAR	DCR
Estado del arte [2]	0.1496	0.0260	0.3618	<b>0.2068</b>	<b>9.8658</b>	<b>45.1089</b>
RNC propuesta	<b>0.1301</b>	<b>0.0193</b>	<b>0.3317</b>	0.2269	11.1982	58.5867

## 5. Conclusiones y trabajo futuro

Hemos presentado una arquitectura de RNC capaz de reconstruir la profundidad utilizando solamente capas convolucionales, y por tanto, que requiere únicamente de entrenamiento en una sola etapa. Hemos adaptado un conjunto de datos existente de imágenes estereoscópicas para desarrollar un recurso orientado a probar la reconstrucción en profundidad de imágenes con perspectiva. Utilizamos este conjunto de datos para probar nuestra RNCC y comparamos su desempeño con el estado del arte actual en reconstrucción de profundidad. Encontramos que el uso de capas convolucionales ayuda a mejorar los resultados a nivel global. La perspectiva de la imagen toma un papel importante en la reconstrucción de la profundidad. Cuantitativamente, la profundidad local se puede estimar con nuestra RNCC, pero esto todavía necesita ser validado con otros conjuntos de datos.

Como trabajo futuro planeamos experimentar con diferentes tamaños de filtro, diferentes funciones de error y funciones de activación para mejorar aún más nuestra RNCC propuesta, así como el uso de una etapa de refinamiento de la imagen.

**Agradecimientos.** Agradecemos al Instituto Politécnico Nacional (SIP, COFAA, EDD, EDI and BEIFI), y CONACyT por su apoyo a esta investigación.

## Referencias

1. Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., Corrado, G.S., Davis, A., Dean, J., Devin, M., Ghemawat, S., Goodfellow, I., Harp, A., Irving, G., Isard, M., Jia, Y., Jozefowicz, R., Kaiser, L., Kudlur, M., Levenberg, J., Mané, D., Monga, R., Moore, S., Murray, D., Olah, C., Schuster, M., Shlens, J., Steiner, B., Sutskever, I., Talwar, K., Tucker, P., Vanhoucke, V., Vasudevan, V., Viégas, F., Vinyals, O., Warden, P., Wattenberg, M., Wicke, M., Yu, Y., Zheng, X.: TensorFlow: Large-scale machine learning on heterogeneous systems (2015), <http://tensorflow.org/>, software disponible en tensorflow.org
2. Affi, A.J., Hellwich, O.: Object depth estimation from a single image using fully convolutional neural network. In: Digital Image Computing: Techniques and Applications (DICTA), 2016 International Conference on. pp. 1–7. IEEE (2016)
3. Arora, R., Basu, A., Mianjy, P., Mukherjee, A.: Understanding deep neural networks with rectified linear units. arXiv preprint arXiv:1611.01491 (2016)
4. Bills, C., Chen, J., Saxena, A.: Autonomous MAV flight in indoor environments using single image perspective cues. In: Robotics and automation (ICRA), 2011 IEEE international conference on. pp. 5776–5783. IEEE (2011)
5. Eigen, D., Fergus, R.: Predicting depth, surface normals and semantic labels with a common multi-scale convolutional architecture. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 2650–2658 (2015)
6. Eigen, D., Puhrsch, C., Fergus, R.: Depth map prediction from a single image using a multi-scale deep network. In: Advances in neural information processing systems. pp. 2366–2374 (2014)

7. Geiger, A., Lenz, P., Urtasun, R.: Are we ready for autonomous driving? the kitti vision benchmark suite. In: Conference on Computer Vision and Pattern Recognition (CVPR) (2012)
8. Geiger, A., Roser, M., Urtasun, R.: Efficient large-scale stereo matching. In: Asian conference on computer vision. pp. 25–38. Springer (2010)
9. Gonzalez, W., Woods, R.E.: Eddins, digital image processing using MATLAB. Third New Jersey: Prentice Hall (2004)
10. Häne, C., Sattler, T., Pollefeys, M.: Obstacle detection for self-driving cars using only monocular cameras and wheel odometry. In: Intelligent Robots and Systems (IROS), 2015 IEEE/RSJ International Conference on. pp. 5101–5108. IEEE (2015)
11. Hirschmuller, H.: Stereo processing by semiglobal matching and mutual information. IEEE Transactions on pattern analysis and machine intelligence 30(2), 328–341 (2008)
12. Howard, I.P.: Perceiving in depth, volume 1: basic mechanisms. Oxford University Press (2012)
13. Konolige, K.: Small vision systems: Hardware and implementation. In: Robotics research, pp. 203–212. Springer (1998)
14. Kundu, A., Li, Y., Dellaert, F., Li, F., Rehg, J.M.: Joint semantic segmentation and 3D reconstruction from monocular video. Georgia Institute of Technology (2014)
15. LeCun, Y., Boser, B.E., Denker, J.S., Henderson, D., Howard, R.E., Hubbard, W.E., Jackel, L.D.: Handwritten digit recognition with a back-propagation network. In: Advances in neural information processing systems. pp. 396–404 (1990)
16. LeCun, Y., Bottou, L., Bengio, Y., Haffner, P.: Gradient-based learning applied to document recognition. Proceedings of the IEEE 86(11), 2278–2324 (1998)
17. LeCun, Y.A., Bottou, L., Orr, G.B., Müller, K.R.: Efficient backprop. In: Neural networks: Tricks of the trade, pp. 9–48. Springer (2012)
18. Liu, F., Shen, C., Lin, G.: Deep convolutional neural fields for depth estimation from a single image. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 5162–5170 (2015)
19. Liu, F., Shen, C., Lin, G., Reid, I.: Learning depth from single monocular images using deep convolutional neural fields. IEEE transactions on pattern analysis and machine intelligence 38(10), 2024–2039 (2016)
20. Menze, M., Geiger, A.: Object scene flow for autonomous vehicles. In: Conference on Computer Vision and Pattern Recognition (CVPR) (2015)
21. Zeiler, M.D., Fergus, R.: Visualizing and understanding convolutional networks. In: European conference on computer vision. pp. 818–833. Springer (2014)

## **Sistema de monitoreo remoto y detección de anomalías cardiacas en pacientes ambulatorios**

César Enrique Rose Gómez, Luz María Márquez Agúndez,  
Juan Carlos Hernández Cruz, María Trinidad Serna Encinas

Tecnológico Nacional de México, Instituto Tecnológico de Hermosillo,  
División de Estudios de Posgrado e Investigación,  
México

crose@ith.mx, lucynitzy@hotmail.com, jc23@live.com.mx, tserna@ith.mx

**Resumen.** En el sector salud de nuestro país, como en algunos otros, se tienen diversas problemáticas debido al incremento de la población, lo cual impacta en la demanda de recursos como servicios de atención a la salud, de centros hospitalarios, de médicos, entre otros. Aunado a lo anterior, se tiene la carencia de especialistas o la centralización de los médicos y hospitales en las principales ciudades del país, con lo cual un gran sector de la población no tiene una atención suficiente y de calidad. Actualmente, el incremento de las enfermedades cardiovasculares en el mundo constituye la primera causa de muerte, en nuestro país también se encuentran estas enfermedades en los primeros lugares de la mortalidad. En este artículo se presenta una propuesta de una arquitectura para el monitoreo remoto de pacientes ambulatorios, específicamente aquellos con alguna anomalía cardíaca.

**Palabras clave:** telemedicina, procesamiento ECG, alerta para anomalías cardiacas, redes neuronales.

### **System of Remote Monitoring and Detection of Cardiac Anomalies in Outpatients**

**Abstract.** In the health sector of our country, as in some others, there are various problems due to the increase in population, which impacts on the demand for resources such as health care services, hospitals, physicians, etc. In addition to the above, there is a lack of specialists or the centralization of physicians and hospitals in the main cities of the country, which means that a large sector of the population does not have sufficient quality care. Currently, the increase in cardiovascular diseases in the world is the leading cause of death; in our country these diseases are also in the first places of mortality. In this article we present a proposal of architecture for the remote monitoring of outpatients, specifically those with some cardiac anomaly.

**Keywords:** telemedicine, ECG processing, cardiac anomalies alert, neural networks.

## **1. Introducción**

En México al igual que en otros países en el mundo se ha iniciado la Telemedicina para encontrar diversas soluciones a los problemas inherentes al sector salud. Como se ha definido en [1], la Telemedicina es “El suministro de servicios de atención sanitaria en los que la distancia constituye un factor crítico, por profesionales que apelan a tecnologías de la información y de la comunicación con objeto de intercambiar datos para hacer diagnósticos, preconizar tratamientos y prevenir enfermedades y heridas, así como para la formación permanente de los profesionales de atención de salud y en actividades de investigación y de evaluación, con el fin de mejorar la salud de las personas y de las comunidades en que viven”.

La Telemedicina puede prestar diversos servicios como: Tele-Consulta, Tele-Asistencia médica y sanitaria y Tele-Educación [1]. Entre estos servicios se encuentra la Tele-Cardiología, que desde hace tiempo ha estado presente en nuestro país de diversas maneras. La Tele-Cardiología es la aplicación de la Telemedicina a la prevención, diagnóstico y tratamiento de las enfermedades cardiovasculares. Una de las aplicaciones de la Tele-Cardiología es el monitoreo remoto de pacientes que tengan alguna anomalía cardíaca, lo cual permite verificar el estado del paciente usando su electrocardiograma (ECG), o la generación de una alerta si se detecta automáticamente una anomalía.

Actualmente, el incremento de las enfermedades cardiovasculares constituyen la primera causa de muerte en el mundo, en nuestro país se encuentran estas enfermedades en los primeros lugares en la estadística de mortalidad [2].

En este artículo se describen los módulos para un sistema de Tele-Cardiología, con los cuales se puede adquirir y pre-procesar la señal ECG, así como la detección de eventos y características que permitan un procesamiento de tipo inteligente y la comunicación de los datos en el sistema.

## **2. Desarrollo**

### **2.1 Antecedentes**

La cantidad de personas con enfermedades cardiovasculares se va incrementando por diversos factores como se indica en varios reportes [3][4], uno de ellos es el estilo de vida de las personas. Por ejemplo, en lugares donde la alimentación está basada en la alta ingesta de carnes rojas y existe una baja actividad física, estos son indicadores que están asociados con las enfermedades cardiovasculares, como se muestra en [5].

El diagnóstico, el seguimiento de la terapia y el pronóstico de las enfermedades cardiovasculares, se suele basar en un examen rápido y confiable llamado electrocardiograma (ECG). El electrocardiograma es un registro gráfico de la actividad eléctrica del corazón y proporciona abundante información que puede ser almacenada, transmitida y procesada de diversas maneras. La interpretación de señales electrocardiográficas es una de las muchas ramas de la ciencia médica, que es estudiada por un médico especializado en cardiología, para hacer un diagnóstico de enfermedades

cardíacas o reconocer alguna cardiopatía. El estudio de las características del ECG ayuda al cardiólogo a diagnosticar anomalías; por ejemplo, ataques al corazón, problemas de conducción, embolias, arritmias, hipertrofias, taquicardias y bradicardias, entre otras.

## 2.2 Arquitectura del Sistema de Monitoreo Remoto

El proyecto que actualmente se está desarrollando en el TecNM / Instituto Tecnológico de Hermosillo, considera el uso de dispositivos móviles que permitan la adquisición y procesamiento de la señal ECG y de otros datos biomédicos y de contexto. El resultado de este procesamiento es la generación de una alerta, así como de distintos datos. Estos datos deben ser enviados a una central hospitalaria o a un consultorio médico para su análisis. Asimismo, desde la central hospitalaria o desde el consultorio médico, se requieren datos a los dispositivos móviles para el monitoreo de los pacientes. En la Fig. 1 se muestra un diagrama a bloques del sistema, posteriormente se describen algunos de los componentes del mismo.

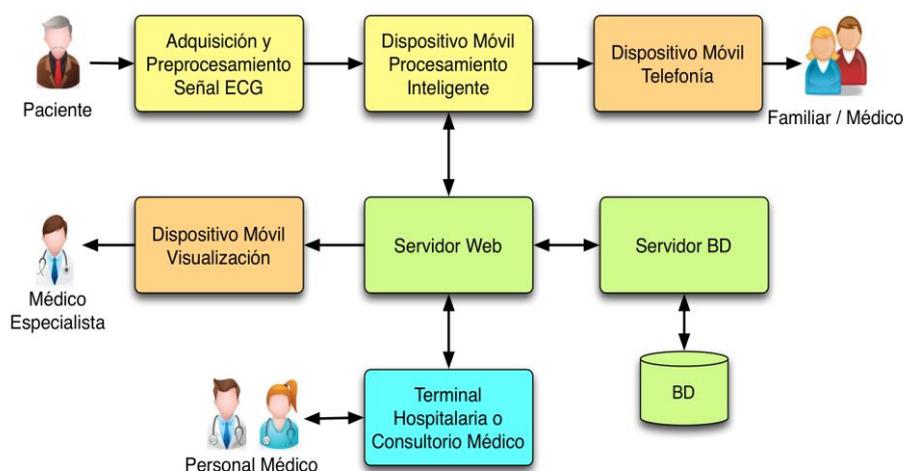


Fig. 1. Diagrama a bloques del sistema.

La arquitectura propuesta permite la comunicación entre los diversos dispositivos para el control de la adquisición de los datos, la transferencia de datos, el almacenamiento de los datos, el requerimiento de tareas y la solicitud de la atención del personal médico o de la propia familia o cuidadores del paciente. En nuestro proyecto se usa una arquitectura SOA (Arquitectura Orientada a Servicios) [6], buscando un mejor rendimiento en cuanto a la comunicación en la estructura que conforma este sistema, cuyas interfaces son realizadas en diversas plataformas y/o lenguajes. En la Fig. 2 se muestran los componentes de esta arquitectura.

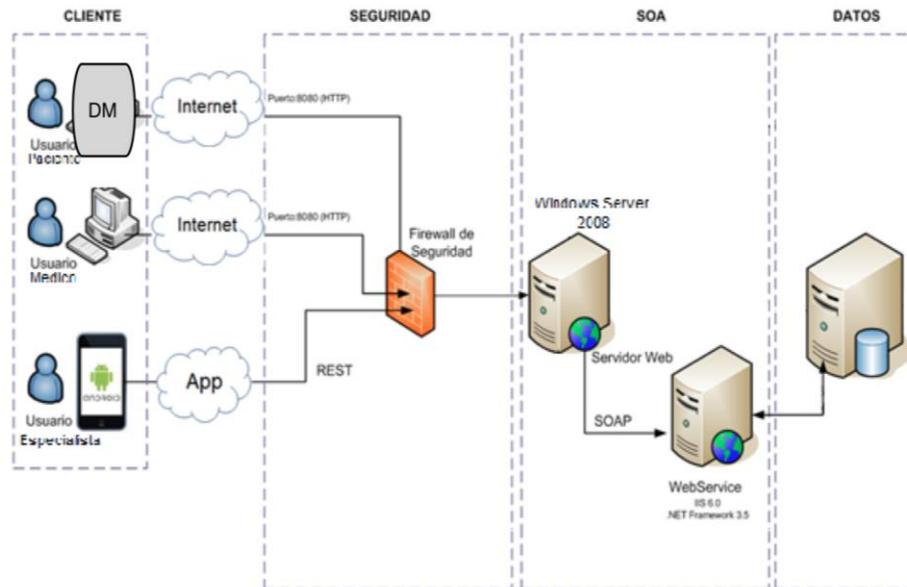


Fig. 2. Arquitectura SOA del sistema.

### Dispositivo médico

Cada paciente tiene un dispositivo médico (DM en la Fig. 2), éste consta de dos componentes, uno de ellos es la tarjeta de adquisición de datos y el otro es una computadora Raspberry para efectuar el procesamiento inteligente. Se diseñó un circuito electrónico para obtener la señal ECG, para ello se ha considerado usar solo la derivación unipolar DI para obtener la señal ECG, este circuito tiene una etapa de amplificación, una etapa de filtrado, una etapa de aislamiento y una etapa de conversión analógica a digital. Una vez que se obtiene la señal de manera digital se envía por comunicación Bluetooth a la computadora Raspberry que es nuestro dispositivo móvil como se muestra en la Fig. 3.



Fig. 3. Adquisición y procesamiento de la señal.

El procesamiento inteligente se lleva a cabo en el dispositivo móvil, para el cual se ha diseñado un algoritmo para controlar los procesos a ejecutar en el dispositivo móvil como se muestra en la Fig. 4 como un diagrama de bloques.

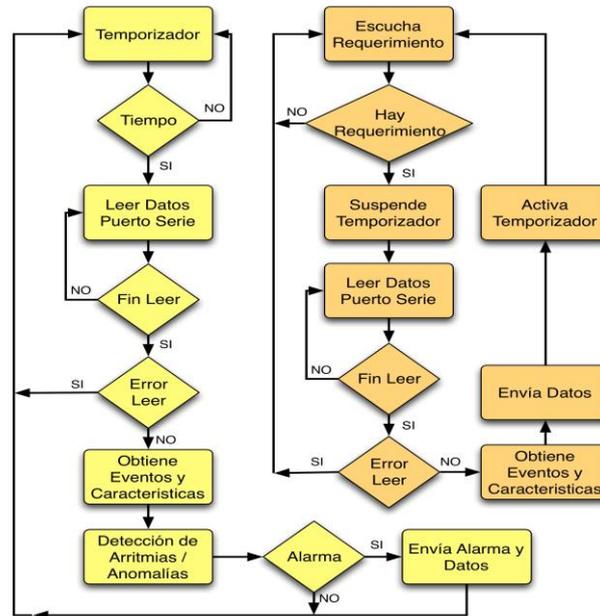


Fig. 4. Algoritmo para el control de procesos, detección de eventos y detección de anomalías cardíacas.

Se han considerado dos funcionalidades generales:

a) Un proceso que cada cierto tiempo adquiera la señal ECG, la procese y en caso de detectar alguna anomalía, genere la alerta correspondiente, envíe el mensaje SMS y envíe a la central hospitalaria los datos de la señal y los datos de ritmo, frecuencia cardíaca, intervalos, entre otros.

b) Un proceso que permita un requerimiento por parte de la central hospitalaria para obtener los datos de la señal y los datos de ritmo, frecuencia cardíaca, intervalos, etc. y los envíe a la central hospitalaria.

Como se puede apreciar en la Fig. 4, una parte fundamental para el procesamiento inteligente es la obtención de los eventos y características de la señal ECG, un latido normal en un ECG tiene la forma de onda que muestra en la Fig. 5, la cual nos permite apreciar las ondas y eventos que la constituyen.

El algoritmo que se ha desarrollado para obtener las anteriores características usa el algoritmo Pan-Tompkins [7] para la identificación de la magnitud de los picos de la onda R, de la onda Q y de la onda S. Posteriormente la detección del complejo QRS, esto es, el inicio del complejo QRS y el final del complejo QRS, así como la onda P y la onda T son detectados usando la transformada Wavelet. Ya que el objetivo de este artículo es presentar sólo la arquitectura del sistema, en [8] se puede encontrar el algoritmo realizado para obtener las características del ECG.

Asimismo, el algoritmo de nuestra propuesta para la detección de anomalías cardíacas está basado en la sugerencia presentada en [9], está compuesto por un conjunto de reglas, las cuales usan diversos eventos extraídos de la señal ECG, tales

como el intervalo PQ, el intervalo QT, intervalo PR, etc. y de un clasificador implementado con una red neuronal.

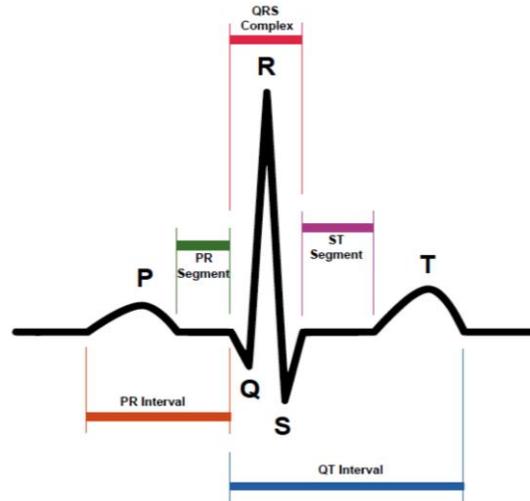


Fig. 5. Ondas y eventos en una señal ECG.

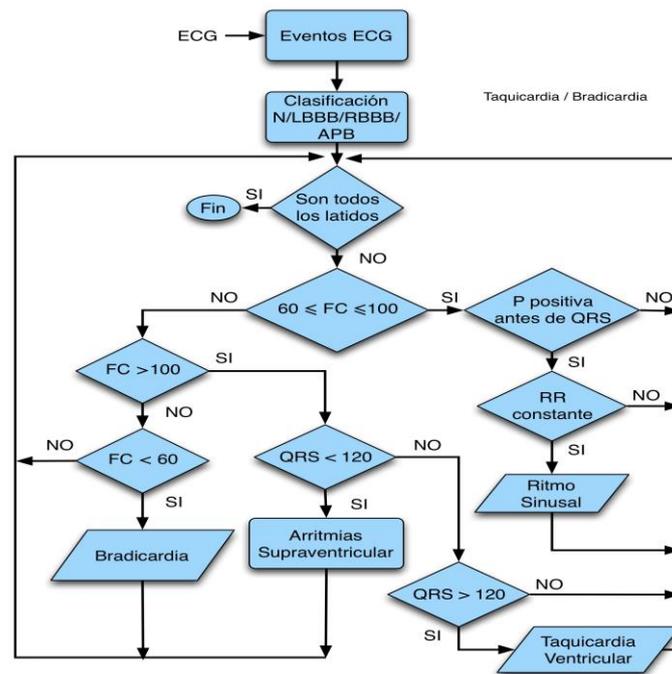


Fig. 6. Algoritmo para la detección de arritmias, bradicardia y taquicardia.

En este artículo para tener una comprensión general del procesamiento inteligente sólo se presenta en la Fig. 6 el algoritmo que permite la detección de bloqueos (LBBB, RBBB, AV) y de detección de bradicardia o taquicardia. En [10] se puede encontrar a mayor detalle los otros algoritmos implementados y los resultados de su experimentación.

### **2.3 Implementación del sistema**

Los escenarios que maneja nuestro sistema como se muestra en la Fig. 2 son los siguientes:

1. Programa desarrollado en el lenguaje ANSI C (bajo el paradigma de programación estructurada) en la computadora Raspberry para las siguientes tareas:
  - a. Control de procesos
  - b. Detección de eventos y características de la señal ECG.
  - c. Detección de anomalías cardíacas con la implementación de los algoritmos:
    - i. Detección de arritmias usando una red neuronal de retropropagación con una capa oculta.
    - ii. Detección de bradicardia y taquicardia.
    - iii. Detección de intervalos de conducción.
    - iv. Detección de arritmias supraventriculares.
2. Programa en lenguaje C para la conversión de la señal ECG a digital y su envío usando comunicación serial Bluetooth en la tarjeta de adquisición de datos.
3. Aplicación Móvil (Médico o Especialista) para visualización de la señal de alarma en Android.
4. Aplicación en C# para PC (Interfaz Central Hospitalaria).
5. Web Service en C# (Servidor Web).
6. Servidor para Base de Datos (Windows Server).

Estos son inicializados cuando el usuario paciente se conecta con el dispositivo médico y comienza el envío de la señal para su procesamiento en una computadora Raspberry, el cual realiza la comunicación con el Servidor y el Web Service por medio del protocolo SOAP el cual por seguridad está respaldado por un Firewall y un Proxy que nos ayuda a mantener la seguridad de nuestro servidor. La aplicación móvil será ejecutada por el usuario especialista cuando requiera visualizar la señal-alarma, así realiza la comunicación con el Web Service por medio del protocolo REST con la misma seguridad. La terminal hospitalaria tendrá el mismo protocolo SOAP y la misma seguridad para el acceso correcto hacia el Web Service.

## **3. Experimentación y resultados**

Por cuestiones de espacio se describe en esta sección la experimentación de la comunicación de los datos desde el dispositivo médico hacia el servidor, se generaron

de manera aleatoria los datos de entrada con respecto a las variables; presión sistólica, presión diastólica, spo2 y caída, entre otros, las cuales se están considerando a ser incluidas en desarrollos futuros, asimismo se usaron los datos correspondientes a la señal ECG para generar la alerta de anomalía cardíaca, se uso la base de datos de arritmias MIT-BIH [11], la cual es un conjunto de archivos que contienen diversas señales ECG, esta base de datos ha sido la usada en la experimentación de los algoritmos presentados en [8, 10].

Para generar las alertas se usan reglas, en la Fig. 7 se muestra un diagrama del flujo donde se tienen los procesos que se realizan para generar las alertas codificadas. La alerta general que incluye las alertas de los sensores biomédicos, así como las variables de contexto. En el bloque denominado Codificación Alerta se codifican las alertas derivadas de los sensores biomédicos; oximetría, presión arterial y ECG. Asimismo, se incluye la variable de contexto de caída.

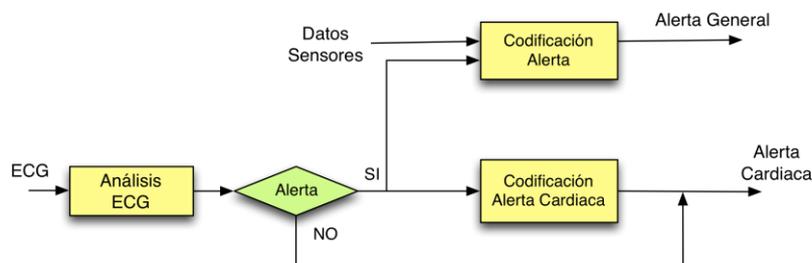


Fig. 7. Diagrama de flujo para la codificación de alertas.

En la Tabla 1 se muestran algunos códigos para las alertas de anomalías cardíacas consideradas en los algoritmos previamente mencionados, los cuales se usan en el proceso Codificación Alerta Cardíaca.

Tabla 1. Códigos para algunas alertas de anomalía cardíaca.

Tipo alerta	Código Binario	Código Decimal
Bloqueo Rama Izquierda	000000001	1
Bloqueo Rama Derecha	000000010	2
Bloqueo APB	000000100	4
Bradicardia	000001000	8
Bradicardia Extrema	0000010000	16

En la tabla 2 se muestran los códigos para la existencia de alerta de los diversos datos. Por ejemplo si hay alertas por anomalía cardíaca, caída e hipoxia, entonces se genera el valor 28, de tal manera que en el código de la alerta se puede saber que alertas sucedieron.

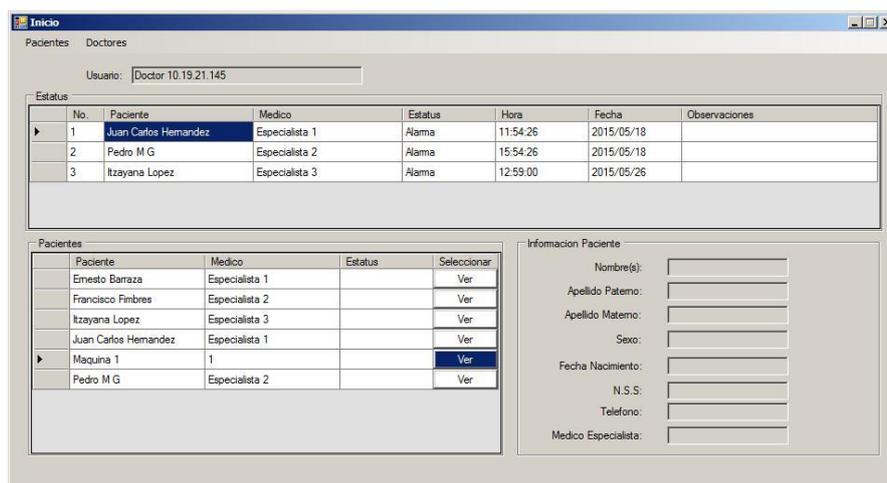
**Tabla 2.** Códigos para alerta general.

Tipo alerta	Código Binario	Código Decimal
Hipertensión	00001	1
Hipotensión	00010	2
Hipoxia	00100	4
Caída	01000	8
Anomalia Cardíaca	10000	16

En la tabla 3 se muestran algunos valores generados aleatoriamente para probar la comunicación de los datos, asimismo se muestra el registro y tipo de latido de la señal ECG, la cual se selecciona también aleatoriamente de la base de datos MIT-BIH para generar la alerta en el caso que se encuentre una anomalía, asimismo, se puede apreciar el resultado de la codificación en la columna denominada Alerta General.

**Tabla 3.** Datos generados aleatoriamente y la alerta general codificada.

Registro	Tipo Latido	Presión sistólica	Presión diastólica	Oximetría	Caída	Alerta cardíaca	Alerta General
109	LBBB	103	66	12	1	1	28
111	LBBB	166	129	34	0	1	20
101	NOR	71	32	5	0	0	4
122	NOR	157	121	80	0	0	4
115	NOR	166	131	83	0	0	12
100	NOR	237	201	76	0	0	5



**Fig. 8.** Interfaz para la central hospitalaria o consultorio médico.

Cuando sucede una alerta, se realiza el envío de los datos en una estructura. Ya que la arquitectura usada para el sistema es orientada a servicios, se ha construido un servidor de servicios bajo el protocolo SOAP, el cual usa documentos XML para el envío de los datos.

Una vez recibidos los datos se procesan en el servidor. En la Fig. 8 se muestra la interfaz para el equipo en la central hospitalaria o consultorio médico. En la Fig. 9 se muestra la señal ECG en el dispositivo móvil del médico o especialista.

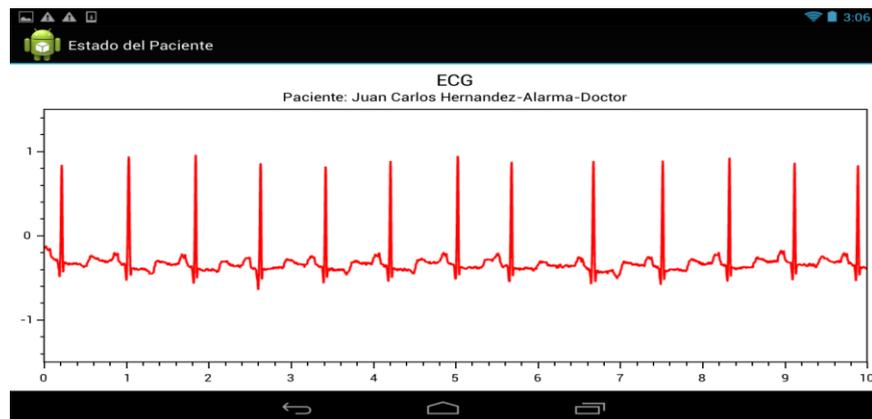


Fig. 9. Visualización de la señal ECG en el dispositivo móvil del médico o especialista.

#### 4. Conclusiones

En este artículo se ha presentado una propuesta para un sistema de Tele-Cardiología, donde los pacientes ambulatorios pueden ser monitoreados en cualquier momento por el personal médico sin realizar el traslado de las personas hacia la unidad hospitalaria, asimismo, se puede tener la generación de alertas cuando se tenga una detección automática de una cardiopatía, con lo cual el paciente puede ser auxiliado por el personal médico o por sus familiares y/o cuidadores.

El artículo muestra solo una parte del proyecto que actualmente se desarrolla, ya que como se ha descrito previamente, se usan tanto variables biomédicas, como variables de contexto, dentro de las primeras se ha mostrado que el ECG es una señal que permite procesarla para detectar las anomalías y se puede visualizar la señal por el médico de manera remota para realizar un diagnóstico, las variables de contexto permiten obtener información del paciente tales como la postura del cuerpo, si sufrió una caída, en donde se encuentra, información que resulta de gran valor para auxiliar en caso necesario al paciente.

Aún se sigue trabajando en este proyecto, donde se están analizando otros tipos de sensores para las variables biomédicas y de contexto, también se están diseñando algoritmos para detectar arritmias mas peligrosas como las fibrilaciones y un algoritmo para predecir infartos y asistolias.

Como trabajo futuro se considera que esta arquitectura permite almacenar una gran cantidad de información que puede analizarse con tecnología “Big Data” para obtener información o conocimiento relevante a las enfermedades cardíacas.

## **Referencias**

1. Secretaría de Salud, Subsecretaría de Innovación y Calidad, Centro Nacional de Excelencia Tecnológica en Salud, Telemedicina, Serie Tecnologías en Salud, Volumen 3 (2007)
2. Instituto Nacional de Salud Pública, Encuesta Nacional de Salud y Nutrición. Resultados nacionales (2012)
3. INEGI, Estadísticas a Propósito del Día Mundial del Corazón, Instituto Nacional de Estadística y Geografía (2007)
4. SSA, Sistema Epidemiológico y Estadístico de las Defunciones, Secretaría de Salud, (2009)
5. ISSSTE, Sonora (2014)
6. Erl, T.: *Service-Oriented Architecture: Concepts, Technology, and Design*. 1st edn. Pearson, USA (2009)
7. Pan, J., Tompkins, W.J.: A Real Time QRS Detection Algorithm. *IEEE Transactions on Biomedical Engineering*, 32(3), 230–236 (1985)
8. Rose, C., Serna, M.: *Procesamiento del Electrocardiograma para la Detección de Cardiopatías*. Encuentro Nacional de Computación, Ocotlán, Oaxaca (2014)
9. The ECG Course. <http://en.ecgpedia.org>
10. Rose, C., Serna, M.: Generación de Alertas de Anomalías Cardíacas usando un Algoritmo Inteligente Híbrido. *Electro* 37, 274–279 (2015)
11. Moody G.B., Mark, R.G.: The MIT-BIH Arrhythmia Database on CD- ROM and software for use with it. *Computers in Cardiology* 17, 185–188 (1990)



## Implementación de un cancelador activo de ruido en un ducto usando el algoritmo FX-ECLMS

E. Pichardo, X. Maya, A. A. Vázquez, J. G. Avalos, J. C. Sánchez

Instituto Politécnico Nacional,  
Sección de Estudios de Posgrado e Investigación,  
ESIME CULHUACAN,  
México

edua\_pim95@hotmail.es, xmayar1300@alumno.ipn.mx,  
avazquezp1000@alumno.ipn.mx, javaloso@ipn.mx, jcsanchezgarcia@gmail.com

**Resumen.** En este artículo se presenta la implementación de un sistema de cancelación activa de ruido en un ducto usando el algoritmo de mínimos cuadrados promediados con error codificado (Error Coded Least Mean Square-ECLMS). Para reducir los efectos producidos por la trayectoria secundaria se utilizó la estructura de filtrado-X. El sistema fue implementado en el procesador digital de señales TMS320C6713 de Texas Instruments. Los resultados experimentales demuestran que al usar el algoritmo ECLMS se logra cancelar de manera adecuada señales de baja frecuencia, además, la carga computacional se reduce considerablemente.

**Palabras clave:** control activo de ruido, filtrado adaptativo, algoritmo ECLMS.

### Implementation of Active Noise Control in a Duct Using the FX-ECLMS Algorithm

**Abstract.** This paper deals with the implementation of an active noise control system in a duct using the error coded least mean square (ECLMS) algorithm. The effects caused by the secondary path were reduced using a filter-x structure. The system was implemented in the digital signal processor TMS320C6713 from Texas Instruments. The experimental results show that using the ECLMS algorithm the low frequency signals are effectively cancelled, moreover, the computational cost is considerably reduced.

**Keywords.** Active noise control, adaptive filter, ECLMS algorithm.

#### 1. Introducción

La Cancelación Activa de Ruido (CAR) se basa en el principio de superposición, el cual consiste en utilizar una fuente canceladora de ruido que se encarga de atenuar la

fuente primaria de ruido, esto se logra produciendo una señal anti-ruido que es igual en amplitud y frecuencia a la señal de ruido, pero con fase contraria. Para generar la señal anti-ruido es necesario utilizar técnicas de filtrado adaptativo debido a que las características de la señal como el contenido espectral, la amplitud y la fase, así como el medio y la velocidad de propagación no son estacionarios [1].

La Fig. 1 muestra la estructura general de los filtros adaptativos, en donde  $x(n)$  es una señal de entrada,  $d(n)$  es la señal deseada,  $y(n)$  es la señal de salida del filtro adaptativo y  $e(n)$  es la señal de error necesaria para adaptar los parámetros del filtro, la cual se calcula mediante la diferencia entre  $y(n)$  y  $d(n)$  [2].

La principal característica de los filtros adaptativos es que son variantes en el tiempo debido a que es necesario compensar los cambios de la señal de entrada para obtener la señal de salida deseada. Para poder compensar estos cambios se utilizan algoritmos adaptativos, los cuales se encargan de actualizar los coeficientes del filtro adaptativo.

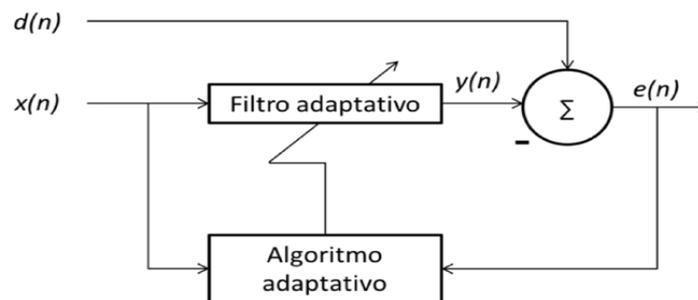


Fig. 1. Estructura general de filtro adaptativo.

Existe una gran variedad de algoritmos que se pueden clasificar en algoritmos de baja complejidad computacional con baja velocidad de convergencia y algoritmos de alta velocidad de convergencia con un alto coste computacional. Algunos de los algoritmos más utilizados por su baja complejidad computacional son el algoritmo de mínimos cuadrados promediados (Least Mean Square-LMS) y su versión normalizada (Normalized Least Mean Square-NLMS). Dichos algoritmos han sido implementados con éxito en diferentes trabajos de sistemas CAR [3-6], sin embargo, como se mencionó anteriormente la velocidad de convergencia de los algoritmos utilizados suele ser lenta. En [7] se presenta un sistema CAR en el que utiliza el algoritmo de proyecciones afines (Affine Projection-AP), el cual presenta una velocidad de convergencia más rápida que la de los algoritmos LMS, sin embargo, su complejidad computacional es bastante elevada ya que para el cálculo de coeficientes se requiere realizar diversas operaciones de inversión de matrices, esto provoca que su implementación en hardware sea muy complicada. Por lo antes mencionado, es evidente que el reto en el desarrollo de los sistemas CAR se encuentra en el uso de un algoritmo que pueda ser de alta velocidad de convergencia y que su complejidad computacional sea baja.

En [8] se presentó el algoritmo de mínimos cuadrados promediados con error codificado (ECLMS), el cual es una variante que presenta una alta velocidad de convergencia y una baja complejidad computacional con respecto al algoritmo LMS, sin embargo,

aún no se han reportado trabajos en los cuales se aplique a sistemas CAR. En este trabajo presentamos una versión del algoritmo ECLMS con filtrado-X (FX-ECLMS) para sistemas CAR, con la finalidad de realizar un sistema que mantenga un equilibrio entre la velocidad de convergencia y su complejidad computacional. Los resultados experimentales demuestran que el algoritmo cancela de manera adecuada el ruido de bajas frecuencias dentro de un ducto, y lo más importante es que la carga computacional se reduce considerablemente.

## 2. Algoritmo ECLMS

El algoritmo ECLMS reduce las operaciones de punto flotante mediante el proceso de codificación del error, de esta forma se reduce la complejidad computacional y se modifica el factor de convergencia con la finalidad de aumentar la velocidad de convergencia.

La ecuación de actualización de coeficientes  $w(n)$  del algoritmo se muestra en (1), donde se observa que el producto entre el error codificado,  $C[e(n)]$ , y la muestra de la señal de entrada,  $x(n)$ , es una operación entre valores enteros, por lo tanto, el número de operaciones de punto flotante se reduce:

$$w(n + 1) = w(n) + \mu C[e(n)]x(n). \quad (1)$$

La codificación del error se realiza efectuando un proceso de cuantización en la señal de error y posteriormente asignando un código a la muestra cuantizada. Este proceso se realiza con la ecuación (2):

$$C[e(n)] = \text{round} \left( \frac{Q[e(n)]}{\text{Res}} \right). \quad (2)$$

Siendo  $Q[e(n)]$  la cuantización del error,  $\text{Res}$  la resolución del codificador y  $\text{round}$  una función que redondea el resultado obtenido de la división entre ambos, la resolución es calculada con (3):

$$\text{Res} = \frac{e_{\max}}{2^b - 1}, \quad (3)$$

donde  $e_{\max}$  es el error máximo probable y  $b$  el número de bits utilizados para codificar la señal. Debido a que se desconoce el error máximo, se hace una estimación a partir de la amplitud máxima de la señal deseada o de la señal de entrada del filtro adaptativo [8]. En el algoritmo ECLMS, conforme el proceso de adaptación avanza la señal de error es minimizada hasta que la función  $\text{round}$  ajusta los valores a cero o uno comportándose como los algoritmos de signo, por lo tanto, si la señal de error se mantiene en esos valores no es necesario actualizar los coeficientes del filtro, dicho proceso se muestra en (4) [9].

$$w(n + 1) = \begin{cases} w(n) & \text{si } C[e(n)] = 0 \text{ o } 1 \\ w(n) + \mu C[e(n)]x(n) & \text{si } C[e(n)] \neq 0 \text{ o } 1 \end{cases} \quad (4)$$

### 3. Estructura FX-ECLMS

El concepto básico de la cancelación activa de ruido consiste en procesar dos señales provenientes una de un sensor de referencia ubicado cerca de la fuente de ruido y otra de un sensor primario ubicado cerca de la fuente secundaria, es decir la fuente que produce la señal anti-ruido. Posteriormente estas señales son procesadas mediante un algoritmo adaptativo el cual usa la señal del micrófono de referencia para estimar la señal anti ruido.

En un escenario ideal, la Cancelación Activa de Ruido se puede considerar un problema de identificación de sistemas. El sistema a identificar es llamado trayectoria primaria  $P(z)$ , el cual está conformado por la respuesta acústica entre el micrófono de referencia y el micrófono de error. Sin embargo, un sistema CAR está conformado por varios sistemas cuya función de transferencia debe ser tomada en cuenta.

Estos sistemas incluyen micrófonos y sus preamplificadores, bocinas, amplificadores, filtros pasa bajas, convertidores A/D y D/A y la trayectoria entre la bocina que produce el anti-ruido y el micrófono que se encarga de monitorear la señal de error. El efecto acumulado de todos estos sistemas es denominado trayectoria secundaria  $S(z)$ .

La trayectoria secundaria provoca que la señal de error no está alineada en tiempo con la señal de referencia, lo que produce inestabilidad en el sistema. Para evitar este efecto se usa la estructura de filtrado-X. El método consiste en ubicar un sistema con una respuesta idéntica a la de  $S(z)$ , el cual es denominado  $S'(z)$ , de manera que la señal de referencia se filtrará en este sistema y su señal de salida servirá para realizar el cálculo de los coeficientes del algoritmo como se muestra en la Fig. 2 [1].

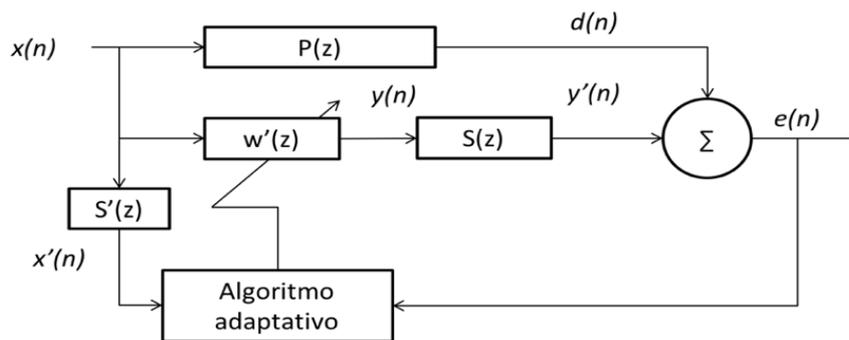


Fig. 2. Estructura de Filtrado-x.

De esta manera la señal de entrada al algoritmo adaptativo  $x'(n)$  difiere de la señal de ruido de referencia  $x(n)$ , de la misma manera que la señal  $y'(n)$  difiere de la señal de salida del filtro adaptativo  $y(n)$  debido a los efectos de  $S(z)$ .

El rendimiento del sistema CAR depende de la identificación de  $S(z)$ , mientras más precisa sea la estimación de  $S(z)$  mayor será la velocidad de convergencia.

### 3.1 Modelado del camino secundario

El modelado del camino secundario  $S(z)$  se muestra en la Fig. 3. Para obtener la estimación  $S'(z)$ , es necesario excitar ambos modelos con la misma señal de entrada, posteriormente se ajustan los coeficientes del filtro  $S'(z)$  mediante un algoritmo adaptativo para disminuir la diferencia entre ambas señales, en este caso es posible usar el algoritmo LMS si el proceso se hace fuera de línea, ya que no sería de gran relevancia la velocidad de convergencia del algoritmo.

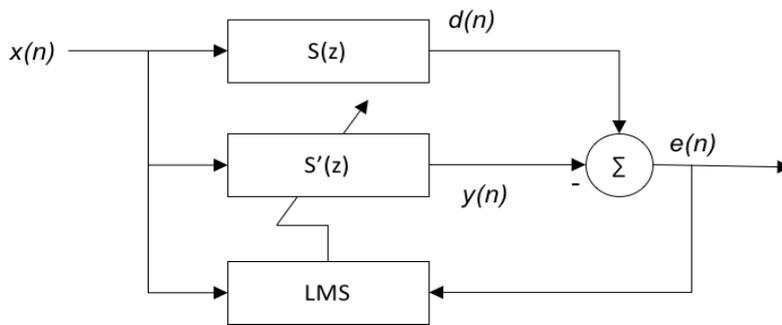


Fig. 3. Modelado de la trayectoria secundaria  $S(z)$ .

Para obtener los coeficientes del filtro  $S'(z)$  se hace uso de la ecuación (5)

$$s'(n + 1) = s'(n) + \mu x(n)e(n), \quad (5)$$

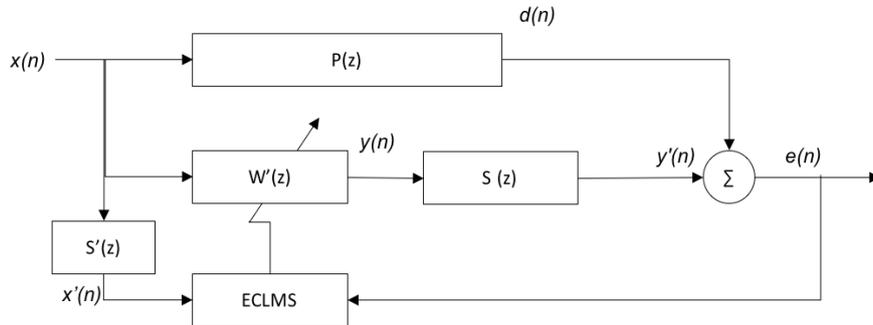
donde  $s'(n)$  es el vector de coeficientes de  $S'(z)$ ,  $\mu$  es el factor que determina la velocidad de convergencia, y  $x(n)$  es la señal de entrada que excita a los sistemas.

La señal de salida del filtro se obtiene mediante (6):

$$y(n) = x(n) * s'(n). \quad (6)$$

### 3.2 Modelado de la trayectoria primaria y atenuación de la señal de ruido

Una vez obtenidos los coeficientes que modelan la trayectoria secundaria se procede a la etapa de cancelación la cual consiste en filtrar la señal de referencia  $x(n)$  mediante  $S'(z)$  y usar la señal resultante  $x'(n)$  para calcular los coeficientes del algoritmo ECLMS como se muestra en la Fig. 4.



**Fig. 4.** Modelado de la trayectoria primaria y atenuación de la señal de ruido.

En el esquema de la Fig. 4. se observa que la señal deseada  $d(n)$  es la señal de referencia  $x(n)$  modificada debido a la propagación de esta por el camino físico primario.

Filtrando la señal de referencia mediante  $W'(z)$ , se obtiene  $y(n)$  la cual es propagada por la bocina canceladora, después la trayectoria  $S(z)$  la altera produciendo así la señal  $y'(n)$ . La señal de error se obtiene mediante su respectivo micrófono y esta representa la suma acústica entre las señales  $d(n)$  y  $y'(n)$ .

De esta forma el algoritmo FX-ECLMS se establece mediante (7):

$$w'(n + 1) = w'(n) + \mu x'(n)C[e(n)]. \quad (7)$$

$w'(n)$  es el vector de coeficientes que modelan la trayectoria primaria,  $\mu$  es el factor de convergencia y  $C[e(n)]$  es la función que codifica el error.

La señal anti ruido se describe mediante (8):

$$y(n) = x(n) * p'(n). \quad (8)$$

#### 4. Escenario de pruebas

El escenario utilizado para las pruebas del algoritmo fue un ducto de madera, Fig. 5 el cual tiene una sección transversal cuadrada con una dimensión aproximada de 9cm y una longitud de 121cm.

Se construyeron filtros analógicos pasa bajas de 500 Hz para evitar que señales con frecuencias altas modificaran la señal de referencia. El controlador fue implementado en el DSP TMS320C6713 el cual fue programado con la herramienta Code Composer Studio. Los micrófonos utilizados son de tipo MEMS, modelo ADMP401-1, el cual tiene una respuesta omnidireccional y una sensibilidad de -37 dBV, y como amplificador de potencia para la señal anti-ruido se utilizó la tarjeta TPA3110 Clase-D de Texas Instruments. Todos los componentes necesarios para implementar el sistema de control de ruido activo fueron posicionados como se muestra en el diagrama de la Fig. 6.



Fig. 5. Escenario de pruebas.

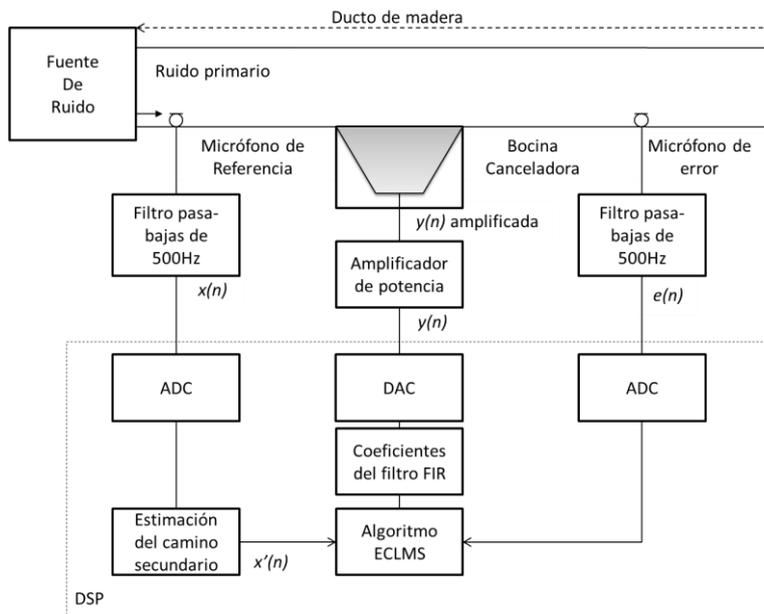


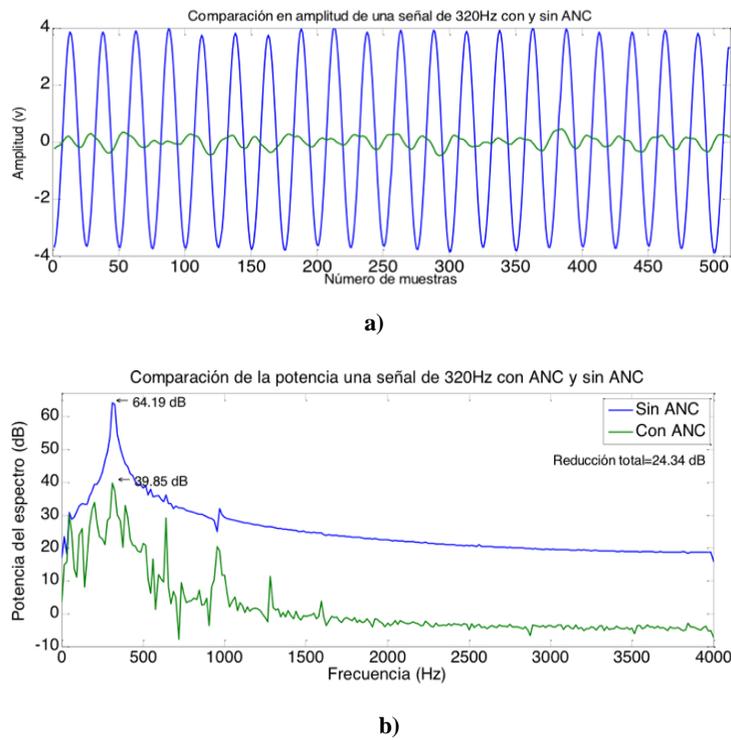
Fig. 6. Diagrama de bloques del sistema CAR.

## 5. Resultados

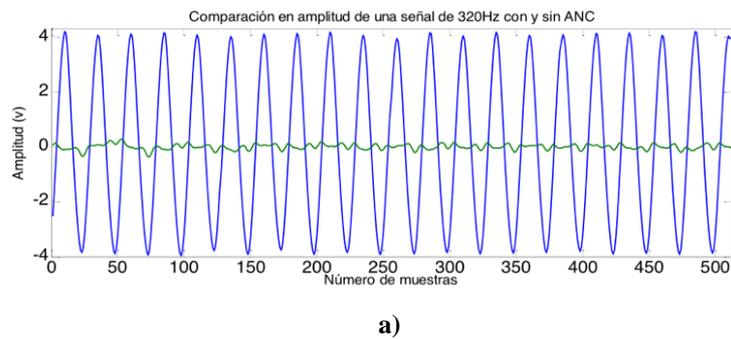
Con el propósito de evaluar el algoritmo FX-ECLMS, se realizaron diversas pruebas y los resultados fueron comparados con los del algoritmo FX-LMS. Los experimentos se realizaron utilizando como señal de referencia un tono sinusoidal con frecuencias desde los 280 Hz hasta los 580 Hz y la suma de diferentes tonos sinusoidales con frecuencias fundamentales de 300, 360 y 420 Hz. No se realizó con frecuencias más altas, debido a que es bien conocido que este tipo de sistemas tienen un buen funcionamiento para señales de baja frecuencia.

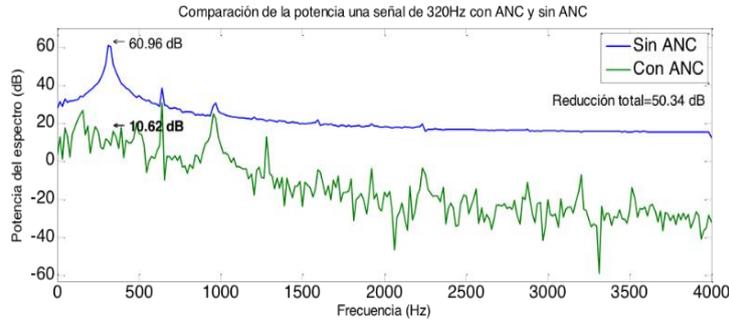
### 5.1 Resultados de la implementación

Ejecutando el algoritmo propuesto se analizó la señal que recibe el micrófono de error cuando el sistema CAR se encuentra activo e inactivo, dichas señales fueron almacenadas para ser comparadas en amplitud (V) y en potencia (dB). Los resultados para un tono de 320 Hz utilizando el algoritmo FX-LMS y el algoritmo FX-ECLMS, respectivamente, se muestran en la Fig. 7 y en la Fig. 8.



**Fig. 7.** Señal obtenida en el micrófono de error utilizando el algoritmo FX-LMS y una señal de referencia de 320 Hz. a) Comparación en amplitud. b) Comparación del espectro de frecuencia.





b)

**Fig. 8.** Señal obtenida en el micrófono de error utilizando el algoritmo FX-ECLMS y una señal de referencia de 320 Hz. a) Comparación en amplitud. b) Comparación del espectro de frecuencia.

La Tabla 1 muestra los resultados obtenidos de las diversas pruebas, las cuales demuestran que se reduce la potencia de la señal entre 24 dB y 41dB, usando el algoritmo FX-LMS, y de 38 dB hasta 50 dB cuando se utilizó el algoritmo FX-ECLMS. También se observa la diferencia del número de actualizaciones hechas por cada uno de los algoritmos. Como se mencionó anteriormente el algoritmo FX-ECLMS no necesita actualizar sus coeficientes en cada iteración, como lo haría la versión original. De esta manera, para una señal de ruido con frecuencia fundamental de 320Hz el algoritmo FX-ECLMS logró ahorrar un total de 28.31% de actualizaciones, en el caso de una señal de ruido con frecuencia fundamental de 350 Hz llevo a cabo un total de 58,024 actualizaciones, lo cual representó una reducción del 36.31% del total de actualizaciones.

**Tabla 1.** Comparativa del número de actualizaciones realizadas por los algoritmos FX-LMS y FX-ECLMS, y nivel de potencia reducida.

Frecuencia fundamental de la señal (Hz)	Actualizaciones hechas por el algoritmo FX-LMS	Actualizaciones hechas por el algoritmo FX-ECLMS	Potencia reducida FX-LMS (dB)	Potencia reducida FX-ECLMS (dB)
300	101,418 (100%)	79,497 (78.38%)	41.39	38.08
320	100,674 (100%)	72,177 (71.69%)	24.34	50.34
350	91,095 (100%)	58,024 (63.69%)	31.41	44.54

Para comprobar la eficiencia del algoritmo se hizo una prueba más robusta, la cual consistió en usar como fuente de ruido una señal multitono con frecuencias de 300, 360 y 420 Hz.

Para este experimento la potencia de los tonos se redujo en diferente nivel para cada una de las frecuencias. La Tabla 2 muestra las reducciones en potencia y las actualizaciones realizadas por cada uno de los algoritmos utilizados. En este caso el algoritmo

propuesto no reduce en forma considerable los niveles de potencia con respecto a los obtenidos con el algoritmo FX-LMS, sin embargo, el número de actualizaciones se reduce.

**Tabla 2.** Comparativa del número de actualizaciones realizadas y potencia reducida por los algoritmos Fx-LMS y Fx-ECLMS para una señal multitono de 300, 360 y 420 Hz.

Frecuencias fundamentales del multitono (Hz)	Actualizaciones FX-LMS	Actualizaciones FX-ECLMS	Potencia reducida Fx-LMS (dB)	Potencia reducida Fx-ECLMS (dB)
300	408,071 (100%)	388,558 (95.21%)	31.23	21.66
360			13.77	9.66
420			5.85	2.82

## 6. Conclusiones

Los resultados obtenidos demuestran que el algoritmo FX-ECLMS implementado en un sistema CAR reduce de manera adecuada las señales de ruido de baja frecuencia. Cabe mencionar, que en algunos casos la reducción del ruido es mayor que la obtenida con el algoritmo FX-LMS. Además, se comprobó que el algoritmo FX-ECLMS no requiere actualizar sus coeficientes en cada iteración, reduciendo así la carga computacional, lo cual es muy benéfico ya que se puede liberar recursos del procesador y así utilizarlos para realizar otras tareas. También es importante considerar que la codificación del error disminuye el número de operaciones de punto flotante, de esta forma al realizar un menor número de operaciones y en menos ciclos de reloj, el procesador reducirá el consumo de potencia. Por lo tanto, queda demostrado que este algoritmo puede ser una gran alternativa para la implementación de sistemas CAR.

## Referencias

1. Kuo, S. M., Morgan, D. R.: Active Noise Control System; Algorithms and DSP implementations. 1st edn. John Wiley & Sons, Inc., New York (1996)
2. Diniz, P. S. R.: Adaptive Filtering: Algorithms and practical implementation. 4th edn. Springer, Boston (2013)
3. Tokatli, A.: Design and implementation of a DSP based active noise controller for headsets. Dissertation, Middle East Technical University, Turkey (2004)
4. Moazzam, M., Rabbani, M. S.: Performance Evaluation of Different Active Noise Control (ANC) Algorithms for Attenuating Noise in a Duct. Dissertation, Blekinge Institute of Technology, Sweden (2013)
5. Garcia, M., Diego, P., Quintana, R.: DSP implementation of the FxLMS algorithm for active noise control: Texas instruments TSM320C6713DSK. In: Automatic Control (CCAC), IEEE 2nd Colombian Conference, pp. 1–6 (2015)
6. Sidhu, S.: Implementation of Active Noise Cancellation in a Duct. Dissertation, Simon Fraser University, Canada (2013)

7. Ferrer, M., De Diego, M., González, A., Pinero, G.: Efficient implementation of the affine projection algorithm for active noise control applications. In: Signal Processing Conference, 12th European, IEEE (2004)
8. Velázquez, J., Ávalos, J. G., Sánchez, J. C.: Comparación e Implementación de los Algoritmos ECLMS y OAECLMS en un DSP. *Computación y Sistemas*, vol. 14, no. 2, p. 175–185 (2010)
9. Martínez, H. M., Avalos, J. G., Sánchez, J. C., Velázquez, J.: Mejora de la velocidad de procesamiento del algoritmo ECLMS e implementación en el DSP C6713. In: Congreso Internacional en Ingeniería Electrónica, pp. 306–311 (2014)



# Identificación del ratón en base a su huella dactilar utilizando características Ordinales Lineales Ortogonales (OLOF)

Jorge Mateo Juárez<sup>1</sup>, Aldrin Barreto Flores<sup>1</sup>, Leopoldo Altamirano Robles<sup>4</sup>,  
Verónica Edith Bautista López<sup>2</sup>, Salvador Eugenio Ayala Raggi<sup>1</sup>,  
Francisco Ramos Collazo<sup>3</sup>

<sup>1</sup> Facultad de Ciencias de la Electrónica,

<sup>2</sup> Facultad de Ciencias de la Computación,

<sup>3</sup> Bioterio Claude Bernard,  
Benemérita Universidad Autónoma de Puebla,  
Puebla, México

<sup>4</sup> Instituto Nacional de Astrofísica, Óptica y Electrónica,  
Puebla, México

jorge.mateo@alumno.buap.mx, aldrin.barreto@correo.buap.mx,  
altamirano@inaoep.mx, vbautista@cs.buap.mx, saraggi@ece.buap.mx,  
francisco.collazo@correo.buap.mx

**Resumen.** Este trabajo presenta un método para el reconocimiento de la huella del ratón según la región de interés delimitada por cinco puntos de referencia, utilizando características ortogonales lineales (OLOF) y algoritmos tales como: mejora de contraste, filtros Gabor, análisis de Procrustes, Warping y Matching. Es un método no invasivo que no causa ningún daño al animal. Se evaluó un conjunto de 14 tipos de 4 cepas diferentes, considerando 742 imágenes de huellas de la extremidad posterior izquierda, donde se realiza el procesamiento, la alineación y la normalización de estas imágenes. Nuestros resultados muestran que nuestra propuesta se obtiene del 92.86 % de reconocimiento.

**Palabras clave:** OLOF, procrustes, filtro Gabor, warping.

## Identification of the Mouse Based on its Footprint Using the Ordinal Characteristics of the Orthogonal Line (OLOF)

**Abstract.** This work presents a method for the recognition of mouse footprint according to the region of interest bounded by five reference points, using linear orthogonal features (OLOF) and algorithms such as:

contrast enhancement, Gabor filters , analysis of Procrustes, Warping and Matching. It is a non-invasive method that does not cause any harm to the animal. A set of 14 types of 4 different strains was evaluated, considering 742 images of footprints of the left hind limb, where the processing, alignment and normalization of these images is done. Our results show that our proposal is obtained of 92.86 % recognition.

**Keywords:** OLOF, Procrustes, Gabor Filter, Warping.

## 1. Introducción

Existe una amplia variedad de sistemas de marcaje para la identificación de los animales de laboratorio, por ejemplo: el pelaje, rasurado en alguna parte específica del cuerpo, así como tintes no tóxicos a prueba de agua. Estos presentan la desventaja de que se desvanecen muy fácilmente y no son eficientes en animales de pelajes oscuros. Existen otros métodos de identificación a corto plazo como: marcas naturales, sexo, textura del pelaje, longitud y cepa. En este trabajo se plantea la metodología para identificación del ratón basado en algoritmos para el procesamiento de imágenes de huellas dactilares que se han empleado ampliamente y de manera efectiva, como: filtros Gabor, filtros ordinales de líneas ortogonales y clasificadores como los mencionados en los artículos<sup>9</sup> y.<sup>1</sup> La metodología presentada a continuación es un método no invasivo que consiste en el mejoramiento de la imagen por medio del contraste y su borde, el filtrado espacial y codificación por medio de un conjunto de filtros Gabor sintonizados para la detección de la huella dactilar del ratón y como clasificador simple la comparación de imágenes codificadas por filtros ordinales. Con este método se obtuvo un 92.86 % de efectividad en el reconocimiento de estas huellas. El trabajo inicia con un resumen de las investigaciones relacionadas con la identificación en animales presentando los algoritmos empleados y su porcentaje de reconocimiento o de error, después se presenta el método de este trabajo con los resultados correspondientes y finalmente las conclusiones.

## 2. Antecedentes y trabajo relacionado

Alli et al.<sup>1</sup> menciona que los animales pueden ser identificados utilizando sus huellas. Varias características contenidas dentro de una huella de estos animales, se pueden utilizar para ayudar en la identificación de un animal. Entre estas características, el más común y más utilizado por los seres humanos para identificar manualmente el animal es el número y tamaño de las gotas en la huella. El uso de las técnicas de procesamiento de imágenes fue para crear un algoritmo para segmentar y extraer la mejor representación posible de la huella que variaba a través del color. La técnica de los componentes utilizados se empleó para contar el número de gotas contenidas dentro de la huella y medir el tamaño de cada burbuja. Usando esta información solamente,

se encontró que una huella podría ser clasificada como de impresión con pezuñas. Finalmente se investigaron técnicas de extracción de características morfológicas para clasificar completamente la huella, por ejemplo: Clasificador Naive Bayes (NBC), Análisis de componentes principales (PCA), Transformación logarítmica, Two Pass Erosión, Umbral basado en el valor medio de píxel, siete momentos invariantes de Hough, distancia euclidiana, cityblock, Canberra y la combinación de estos últimos tres. El sistema reportado cuenta con una tasa de precisión de 97 %.

Shin et al.,<sup>10</sup> discuten el reconocimiento de huellas de insectos. Por medio de escaneos de la huella se extraen segmentos de la misma y las características adecuadas se calculan para los segmentos (o grupo de segmentos) con el fin de discriminar especies de insectos. La selección o identificación de tales características es fundamental para este proceso de clasificación. Propone métodos para la segmentación de la huella automática y extracción de características. En primer lugar, utilizan un método morfológico con el fin de extraer las regiones de la huella de la agrupación de patrones de la misma. En segundo lugar, un algoritmo SOM mejorado y un algoritmo ART2 para la selección automática del umbral, se aplican para extraer segmentos de la huella de la agrupación de las regiones de la huella independientemente del tamaño de la misma o la zancada. En tercer lugar, utilizan una técnica para la transformación de la huella con el fin de averiguar las características apropiadas para los segmentos extraídos por los métodos anteriores. La transformación de la huella construye un nuevo tipo de estructura de datos de las imágenes segmentadas, mediante la definición de las funciones basadas en las líneas de trazo paralelos. Este nuevo tipo de estructura de datos tiene características invariantes a la traslación, rotación y reflexión de las imágenes. Estos se convierten en características triples utilizando funciones diametrales y circular; las características triples son finalmente utilizados para discriminar patrones de huellas de insectos con tasas de precisión de 67 %, 86 % y 100 %.

Nilsson et al.<sup>9</sup> presentan una nueva área de aplicación para el reconocimiento biométrico: la identificación de los animales de laboratorio para reemplazar los métodos invasivos de hoy en día. A través de la identificación biométrica como una técnica no invasiva, se aplica con un espacio de código que está restringido solamente por la singularidad del identificador biométrico en uso, y con una tasa de error que es predecible. Exhiben el patrón de los vasos sanguíneos en una oreja de ratón como un identificador biométrico adecuado y usado para la identificación del mismo, utilizando un total de 50 ratones para las pruebas. Los algoritmos empleados fueron Filtro Gabor, algoritmo basado en la transformada de Hough y correlación de fase obteniendo una tasa de error equitativo (EER) de 2,5 % con una tasa de precisión del 97.5 %.

Hasler et al.<sup>7</sup> presenta dos métodos que dan un paso hacia la automatización del proceso de reconocimiento de huellas de roedores e insectos. Además, estudiaron dos métodos (Análisis de Componentes Principales, un simple clasificador de Naive Bayes) para distinguir las cuatro especies de insectos examinados. Aquí, una combinación de ambos clasificadores resultó superior a usar un solo método.

### 3. Método y técnica

El método se basa de acuerdo al organigrama de la figura 1, donde se realiza un pre-procesamiento a la huella dactilar del ratón, después se extraen las características de este y finalmente para su clasificación se realiza un Matching. En los siguientes temas se describen los conceptos y algoritmos para este trabajo.

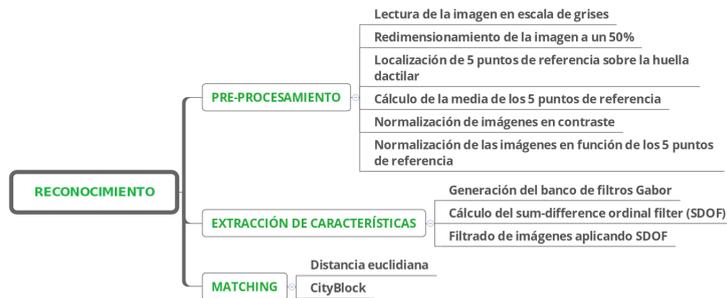


Fig. 1. Organigrama del proceso para la identificación de la huella dactilar del ratón.

#### 3.1. Filtro de mejoramiento del contraste

El objetivo del filtro es realzar el contraste de las líneas de la huella ya que estas no son constantes a lo largo de la imagen. La entrada del filtro es una imagen en niveles de grises. La salida es una imagen en niveles de grises con la particularidad de que la media  $\mu_0$  y la varianza  $\sigma_0$  tienden a valores predeterminados. Donde  $\mu$  es la media y  $\sigma$  es la varianza de la imagen, las ecuaciones empleadas por este filtro son:<sup>8</sup>

$$I'(x, y) = \begin{cases} \mu_0 + \sqrt{\frac{(I(x,y)-\mu)\sigma_0}{\sigma}} & \text{Si } I(x, y) > \mu, \\ \mu_0 - \sqrt{\frac{(I(x,y)-\mu)\sigma_0}{\sigma}} & \text{En otro caso.} \end{cases} \quad (1)$$

Después se aplicó un filtrado espacial de bordes empleando las máscaras de Prewitt ( $P_x$  y  $P_y$ ), después la resta del borde con la imagen contrastada  $I'(x, y)$  y finalmente una normalización de la escala de grises, el resultado de este proceso se muestra en la figura 2.

$$P_x = \begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix} \quad P_y = \begin{bmatrix} -1 & -1 & -1 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{bmatrix} \quad (2)$$

$$P'_x(i, j) = \sum_{x=1}^3 \sum_{y=1}^3 P_x(x, y) I'(i+x, j+y), \quad (3)$$

$$P'_y(i, j) = \sum_{x=1}^3 \sum_{y=1}^3 P_y(x, y) I'(i + x, j + y), \quad (4)$$

$$V = \sqrt{P_x'^2 + P_y'^2}, \quad (5)$$

$$ROI' = I' - V, \quad (6)$$

$$ROI'' = \sum_{x=1}^{N_1} \sum_{y=1}^{M_1} \frac{ROI'(x, y) - \min(ROI') * 255}{\max(ROI') - \min(ROI')}. \quad (7)$$

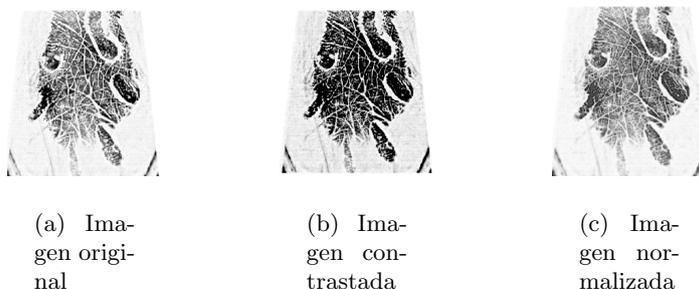
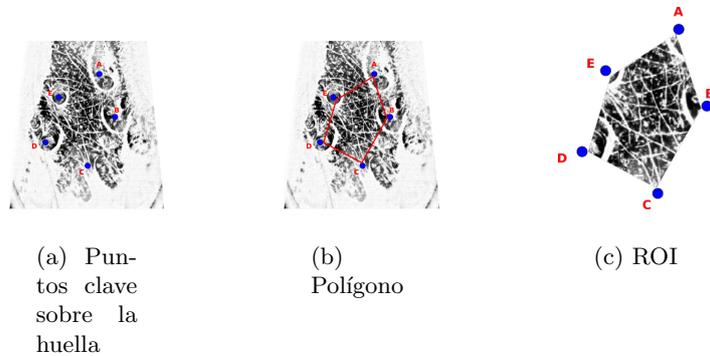


Fig. 2. Proceso del resultado del filtro de mejoramiento de contraste.

### 3.2. Extracción de la ROI

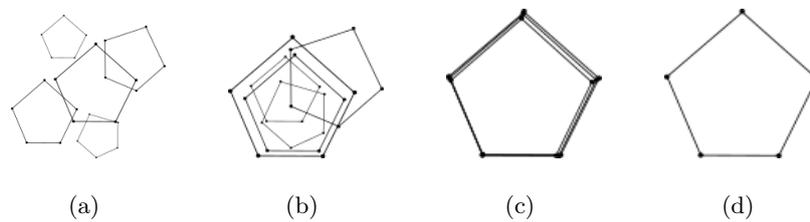
Para realizar la extracción de la región de interés de la imagen de la huella dactilar del ratón, figura 3(c), se procedió de la siguiente manera: se colocaron manualmente 5 puntos referencia (landmarks) A, B, C, D y E, sobre los cojinetes en 742 imágenes, figura 3(a), de tal forma que la ROI está limitada por estos 5 landmarks formando un polígono como se muestra en la figura 3(b).

La alineación de las imágenes se realiza mediante una superposición de Procrustes y Warping. Una superposición de Procrustes minimiza las diferencias entre las configuraciones de landmarks basado en mínimos cuadrados, por lo que también se le conoce como Generalized least square superimposition o GLS, lo que se traduce como superposición generalizada de mínimos cuadrados. Se basa en tres pasos para remover toda la información no relacionada a la forma. La variación en el tamaño se elimina al ajustar cada configuración de landmarks de modo que tenga un tamaño de centroide igual a 1 (El tamaño del centroide, es una medida de tamaño que cuantifica la extensión de los landmarks entorno a su centro de gravedad). La variación en la posición es removida mediante



**Fig. 3.** Región de interés de la huella dactilar del ratón (ROI).

el desplazamiento de las configuraciones de manera que compartan la misma posición de su centro de gravedad. Por último las configuraciones se rotan hasta encontrar una orientación óptima para cada una de las configuraciones, figura 4,<sup>2, 4, 5</sup>



**Fig. 4.** Representación generalizada de la superposición de procrustes: en el inciso a) se muestran las figuras originales, en b) las diferencias por posición a la traslación, en c) la configuración de los landmarks escalados y finalmente la rotación de configuraciones en d).

Después de obtener la forma media (forma base) conformado por los 5 landmarks, se aplica un Warping, el algoritmo fue tomado y adaptado de.<sup>12</sup> Siguiendo el diagrama de la figura 5, se obtienen las imágenes alineadas en función de los 5 landmarks.



Fig. 5. Diagrama de flujo para la extracción de la ROI.

### 3.3. Características ordinales de líneas ortogonales (OLOF) y Matching

El trabajo propuesto por<sup>11</sup> para calcular el filtro ordinal de líneas ortogonales es diseñado como sigue:

$$F(x, y, \theta) = G(x, y, \theta) - G(x, y, \theta + \pi/2). \quad (8)$$

En el dominio espacial, un filtro Gabor 2d es el resultado de una función Gaussiana modulada por una onda sinusoidal. Estas funciones operan en el conjunto de los números complejos, cuya parte real es la función de Gabor simétrica y la parte imaginaria es la función de Gabor asimétrica. En este trabajo se empleó la parte real que está dada por la expresión 9.

$$G(x_p, y_p) = g e^{-\left(\frac{(x_p+y_p)^2}{\sigma}\right)} \cos(wx_p), \quad (9)$$

$$x_p(x, y, \theta) = x \cos(\theta) + y \sin(\theta), \quad (10)$$

$$y_p(x, y, \theta) = -x \sin(\theta) + y \cos(\theta), \quad (11)$$

donde  $\sigma$  es la varianza de la Gaussiana para este caso es de 32,  $w = 2\pi/k$  es la frecuencia espacial,  $k$  es el período espacial cuyo valor es 25 ciclos/imagen,  $g$  es un factor de escala del filtro con valor de 0.8. El tamaño del conjunto de filtros gabor es de  $33 \times 33$  pixeles, estos valores son tomados conforme a los artículos,<sup>13</sup> y experimentales.

El OLOF describe la información de orientación de segmentos de línea negativa en las impresiones de palma de mano. El esquema que propone<sup>6</sup> para codificación de características de la impresión de la palma, es rápido y utiliza el OLOF simplificado porque los coeficientes de la función gaussiana están en punto flotante y el resultado es el coste computacional. Así el filtro ordinal suma diferencia (SDOF), sólo contiene coeficientes enteros, el cual es definido como la siguiente ecuación 12:

$$SDOF(x, y, \theta) = \begin{cases} 1 & F(x, y, \theta) \geq k, \\ -1 & F(x, y, \theta) < -k, \\ 0 & \text{En otro caso,} \end{cases} \quad (12)$$

donde  $k > 0$  y  $\theta$  es la orientación del filtro, de acuerdo a nuestros experimentos elegimos  $k$  como:

$$k = 0,1 \times E, \quad (13)$$

donde  $E$  es la máxima intensidad de  $G(x, y, \theta)$ . Después se realiza un filtrado espacial para cada ROI y los tres SDFOs de acuerdo a la ecuación 14:

$$I_{SDOF}(i, j, \theta) = \sum_{x=1}^{33} \sum_{y=1}^{33} SDOF(x, y, \theta) ROI(i + x, j + y), \quad (14)$$

donde  $\theta = 0, \pi/6, \pi/3, i = 1, 2, 3, \dots, N$  y  $j = 1, 2, 3, \dots, M$  son los índices de la ROI de tamaño  $N \times M$ .

La clasificación se realiza mediante la comparación del resultado de la ecuación 14 de la imagen de entrada y la base de datos mediante las fórmulas de distancia: Euclidiana y City Block.

$$Euclidean = \sqrt{\sum_{i=1}^k |I_{SDOF_E}(i) - I_{SDOF_i}(i)|^2}, \quad (15)$$

$$CityBlock = \sqrt{\sum_{i=1}^k |I_{SDOF_E}(i) - I_{SDOF_i}(i)|}. \quad (16)$$

En la figura 6, se muestra la gráfica en 2D del filtro Gabor par y el conjunto de 6 filtros empleados para la codificación en las orientaciones  $\theta = 0, \pi/6, \pi/3, \pi/2, 2\pi/3, 5\pi/6$ .

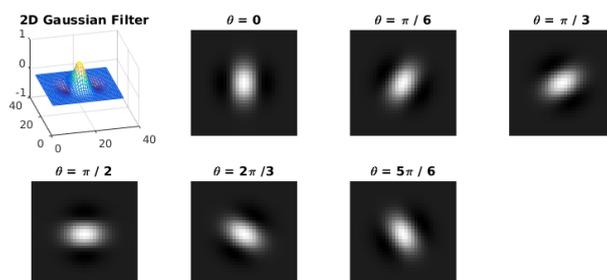


Fig. 6. Gráfica en 2D del Filtro Gabor y el conjunto de 6 filtros.

#### 4. Resultados

Se obtuvieron 742 imágenes de huellas planares de la extremidad posterior izquierda de 14 ratas de las cepas: Long evans, SD, Wistar y Zucker, donde

4 ratas fueron etiquetadas como: Long\_ evans0, Long\_ evans1, Long\_ evans2 y Long\_ evans3; 7 como: Sd1, Sd6, Sdsin\_ num, Sd\_ taiep0, Sd\_ taiep1, Sd\_ taiep2 y Sd\_ ataiep3; 2 como: Wistar\_ sinnum y Wistar\_ sinnum; finalmente uno como: Zucker80, usando un lector biométrico modelo U.ARE.U 4500 de la marca digitalPersona. En la figura 7(a), muestra la ROI y sus imágenes codificadas figuras 7(b), 7(c), 7(d), de acuerdo a la ecuación 14.

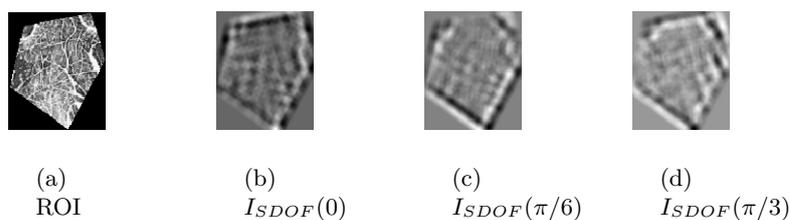


Fig. 7. ROI y sus correspondientes imágenes codificadas.

Se emplearon 280 imágenes de entrenamiento de las 14 clases (ratas) y 462 imágenes para examinación, utilizando la metodología planteada se obtuvo una efectividad con Euclidean de 89.18% y con CityBlock de 92.86%, estos datos fueron determinados mediante una matriz de confusión. Las figuras 8 y 9 muestran los porcentajes de las huellas dactilares correctamente reconocidas.

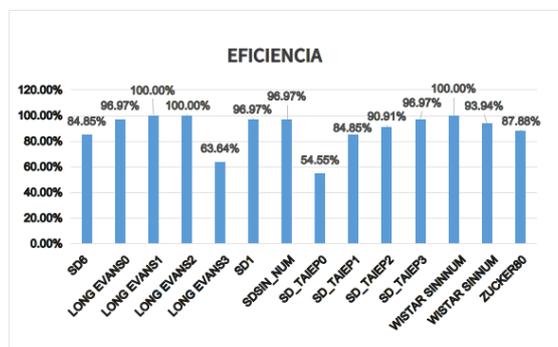


Fig. 8. Gráfica del resultado de la efectividad utilizando Euclidean.

## 5. Conclusiones

Los resultados obtenidos son buenos ya que se obtuvo un porcentaje de 92.86% de imágenes correctamente reconocidas de las huellas dactilares de ratón

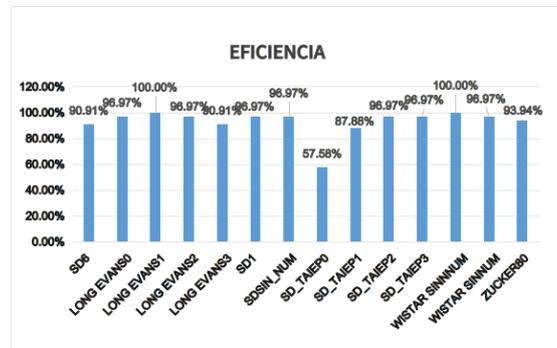


Fig. 9. Gráfica del resultado de la efectividad utilizando CityBlock.

de acuerdo a la metodología planteada, aunque para la clase Sd\_taiep0 solo se reconoce un 57.58 % de las huellas puede ser debido a la adquisición de la huella dactilar e incluso del sensor debido a que se empleó uno para el humano entonces podría utilizarse algún otro con mejores características.

El mejoramiento de contraste planteado es importante en este trabajo debido a que realza la región de interés y consecuentemente normaliza todas las imágenes en escala de grises en función de la media y varianza deseada aun así es necesario mejorar esta etapa ya que en algunos casos el reconocimiento es del 57%. Cabe destacar que el conjunto de filtros Gabor seleccionado al periodo de 25 ciclos/imagen es obtenido de manera experimental y se encontró este valor ya que da mejores resultados al momento de realizar el reconocimiento de la huella dactilar del ratón consecuentemente hay una relación entre las distancias de las grietas de la huella como es el caso de las huellas dactilares del humano, este es un nuevo campo de estudio que no se ha trabajado directamente pero existen investigaciones que se relacionan de manera indirecta como se ha presentado en este trabajo.

El OLOF empleado para las huellas dactilares del ratón es funcional por los resultados obtenidos y es también rápido debido a que se reduce el número de operaciones en punto flotante, ligeramente fue modificado el factor de  $k$ , se llegó a este valor de manera experimental y observando el resultado dado por la ecuación 14. Los parámetros mencionados fueron encontrados de manera experimental para este tipo de huellas, donde se adecuaron a las normalizaciones que se plantean desde la escala de grises y la alineación de las imágenes.

Otro aspecto importante es que se consideraron huellas dactilares de ratones de diferentes tamaños y edades, desde los 100 gramos hasta los 600 gramos con edades que van desde los 3 meses a los 9 meses de edad, estos parámetros no afectan drásticamente, pero son factores que se tienen que considerar. Un factor que afectaría más la metodología planteada es si se añade una clase más, ya que los rasgos procesados por el filtrado espacial son afectados, entonces la compensación para esto es incrementar la base de datos de las huellas dactilares y un incremento en el tamaño de la matriz del filtro Gabor, mientras sea más

grande el tamaño de esta máscara se resaltan más los detalles en las imágenes permitiendo diferenciar entre los ratones, en este caso se lograron buenos resultados con matrices de  $33 \times 33$  píxeles y considerando el coste computacional. El trabajo a futuro es emplear algoritmos basados en el análisis de minucias y otros algoritmos para el mejoramiento de la imagen, además del empleo de redes neuronales para su clasificación.

**Agradecimiento.** Agradecimiento al Bioterio “Claude Bernard” de la Benemérita Universidad Autónoma de Puebla por utilizar sus instalaciones y la colaboración de M.V.Z. César Olvera Pérez para la adquisición de las huellas dactilares de los ratones.

## Referencias

1. Alli, M.N., Viriri, S.: Animal identification based on footprint recognition. In: Adaptive Science and Technology (ICAST), 2013 International Conference on. pp. 1–4. IEEE (2013)
2. Bookstein, F.L.: Biometrics, biomathematics and the morphometric synthesis. *Bulletin of mathematical biology* 58(2), 313–365 (1996)
3. Chen, L., Lu, G., Zhang, D.: Effects of different gabor filters parameters on image retrieval by texture. In: Multimedia Modelling Conference, 2004. Proceedings. 10th International. pp. 273–278. IEEE (2004)
4. Cintas, C., Delrieux, C., Bianchi, G.S., Defossé, N.: Procesamiento inteligente de imágenes: Posicionamiento de vértebras mediante landmarks y redes neuronales. In: XIV Workshop de Investigadores en Ciencias de la Computación (2012)
5. Gower, J.C., Dijksterhuis, G.B.: Procrustes problems, vol. 30. Oxford University Press on Demand (2004)
6. Han, Y., Tan, T., Sun, Z., Hao, Y.: Embedded palmprint recognition system on mobile devices. *Advances in Biometrics* pp. 1184–1193 (2007)
7. Hasler, N., Klette, R., Agnew, W.: Footprint recognition of rodents and insects. Tech. rep., CITR, The University of Auckland, New Zealand (2004)
8. Labati, R.D., Piuri, V., Scotti, F.: Touchless Fingerprint Biometrics. CRC Press (2015)
9. Nilsson, K., Rognvaldsson, T., Cameron, J., Jacobson, C.: Biometric identification of mice. In: Pattern Recognition, 2006. ICPR 2006. 18th International Conference on. vol. 4, pp. 465–468. IEEE (2006)
10. Shin, B.S., Cha, E.Y., Kim, K.B., Cho, K.W., Klette, R., Woo, Y.W.: Effective feature extraction by trace transform for insect footprint recognition. *Journal of Computational and Theoretical Nanoscience* 7(5), 868–875 (2010)
11. Sun, Z., Tan, T., Wang, Y., Li, S.Z.: Ordinal palmprint representation for personal identification [representation read representation]. In: Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on. vol. 1, pp. 279–284. IEEE (2005)
12. Tzimiropoulos, G., Pantic, M.: Optimization problems for fast aam fitting in-the-wild. In: Proceedings of the IEEE international conference on computer vision. pp. 593–600 (2013)



# Clasificación de hiperintensidades presentes en la sustancia blanca en imágenes de resonancia magnética usando extracción de características y redes neuronales

Miguel de Jesús Martínez Felipe, Oleksiy Pogrebnyak,  
Jesús Alberto Martínez Castro, Edgardo Manuel Felipe Riverón

Instituto Politécnico Nacional, Centro de Investigación en Computación,  
Ciudad de México, México

b160670@sagitario.cic.ipn.mx, {olek, macj, edgardo}@cic.ipn.mx

**Resumen** En este trabajo, se muestra un modelo para clasificar lesiones presentes en la sustancia blanca, usando como clasificador una Red Neuronal Multicapa con Retropropagación (RNMR), la cual es alimentada por un vector característico, cuya propiedad deriva de la extracción de características del histograma y el análisis de textura de las Imagen de Resonancia Magnética (IRM). Dado que se ocupa un aprendizaje supervisado con conocimiento previo, es importante etiquetar las imágenes segmentadas como lesión y no lesión, para lo cual un experto determina las áreas con lesión dentro de las imágenes. Para la serie de entrenamiento se ocuparon 29 imágenes y como resultado, la precisión de clasificación esta por encima del 95 %, para ello se puede determinar precisamente si una zona presenta lesión dentro de la sustancia blanca.

**Palabras clave:** imágenes de resonancia magnética, redes neuronales, extracción de características, hiperintensidades de la materia blanca.

## White Matter Hyperintensities Classification in Magnetic Resonance Imaging Using Feature Extraction and Neural Networks

**Abstract.** In this work, a model to classify lesions present in white matter is shown, using as classifier a Back-Propagation Multilayer Neural Network (BPMNN), which is fed by a featured vector, whose properties derived from the feature extraction of histogram and texture analysis of magnetic resonance images; given that a supervised learning with previous knowledge is used, it is important to label the segmented images as lesion and non-lesion, for which an expert determines areas with lesion from the images. To train the neural network, 29 images was occupied and as a result of classifying we obtain over 95% of accuracy, so we can determine accurately which areas are lesion in white matter.

**Keywords:** Magnetic Resonance Imaging, Neuronal Network, Feature Extraction, White Matter Hiperintensities.

## 1. Introducción

Una lesión de sustancia blanca se manifiesta mediante la presencia de un daño isquémico presente en el cerebro, es decir un daño celular causado por la disminución transitoria o permanente del riego sanguíneo. Las técnicas empleadas en las áreas médico biológicas, específicamente en neurología, detectan la presencia de este tipo de lesiones empleando entre otras, tomografía por emisión de positrones (PET), Tomografía Axial Computarizada (TAC) e IRM. Las lesiones de sustancia blanca encontradas en IRM se manifiestan como zonas hiperintensas (áreas de alta intensidad en un examen de resonancia magnética del cerebro) y pueden encontrarse en la región subcortical, es decir la parte interna o basal del cerebro, así como en la región periventricular, es decir el área que rodea los ventrículos del cerebro [10].

Uno de los casos de lesión por sustancia blanca es la desmielinización, presente en pacientes con Esclerosis Múltiple (EM). Sin embargo, es importante resaltar que la existencia de una lesión por sustancia blanca, no necesariamente implica un diagnóstico de EM. La esclerosis múltiple es una enfermedad que consiste en la aparición de lesiones desmielinizantes, neurodegenerativas y crónicas del sistema nervioso [1].

La precisa identificación y detección de lesiones presente en la sustancia blanca en el cerebro ayuda a detectar enfermedades, no sólo de esclerosis múltiple, sino también hipertensión, diabetes u otras patologías; es por ello que los investigadores se han enfocado en múltiples trabajos de inmersión en el área de la computación anudada a la neurociencia con la finalidad de detectar hiperintensidades de sustancia blanca en el cerebro mediante metodologías de identificación de hiperintensidades automáticas y manuales. Un gran número de métodos [12],[2] y [15] , logran identificar y detectar dichas lesiones automáticamente con una tasa de Verdaderos Positivos (VP) mayor al 90%, es decir que la precisión de encontrar hiperintensidades de sustancia blanca en comparación con la detección visual y segmentación manual por radiólogos expertos es mayor, dichas metodologías pueden ser herramientas muy útiles para identificar hiperintensidades.

El artículo se encuentra organizado así: Sección 2 trabajos relacionados; Sección 3 presenta la metodología usada en este trabajo. Es dividida en 4 subsecciones; Sección 3.1 obtención de la información a priori previamente segmentada; Sección 3.2 muestra la fase de procesamiento de la imagen, Sección 3.3 explica la fase de extracción de características, Sección 3.4 Implementación del entrenamiento y clasificación de la red neuronal; Sección 4 se describen los resultados obtenidos por el clasificador de la red neural. Y finalmente, se describen las conclusiones en la Sección 5. La estructuración del modelo se puede apreciar en la Figura 1.

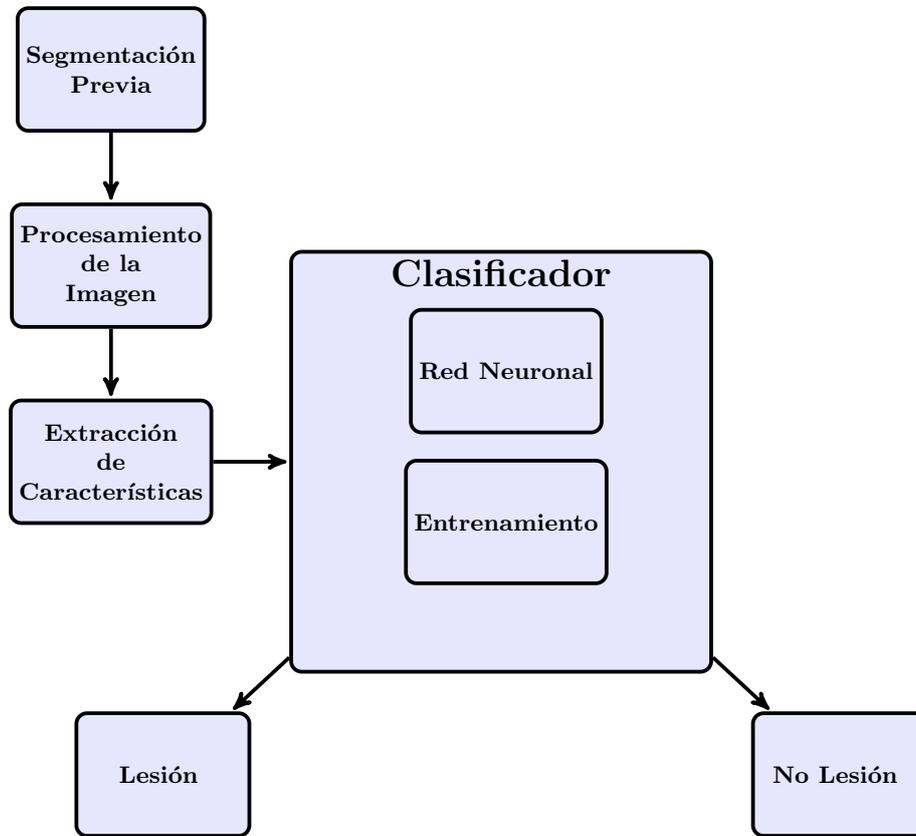


Fig. 1: Diagrama de flujo de la metodología.

## 2. Trabajos relacionados

Existen diversos métodos de trabajo empleados para la identificación y segmentación automática de hiperintensidades presentes en la sustancia blanca, tanto supervisados como no supervisados[8]. Por parte de los aprendizajes supervisados se utilizan IRM previamente segmentadas y dado que se desean resultados con mayor robustez [5], se determinan entradas con información a priori con la finalidad de dar paso a la extracción de rasgos. Una vez obtenida esta extracción se procede a entrenar el clasificador y así generar una función que identifique las salidas deseadas; algunos casos son: la implementación de redes neuronales [9],[16] y [17]; el método del K-vecino más cercano [19], [3] árboles de probabilidad [18],[2]. Las fortalezas de usar aprendizajes supervisados son incluir información local, determinar marcos probabilísticos y utilizar una gran lista de clasificadores, mientras que las debilidades son la necesidad de crear bases de datos con previas segmentaciones y que las previas segmentaciones quizá

contengan errores. Por parte de los métodos no supervisados no se requiere un conjunto de entrenamiento que permita conocer las imágenes segmentadas más bien consiste en aplicar un algoritmo no supervisado para definir específicamente las propiedades usadas en cada imagen de tal forma que las lesiones se encuentran utilizando reglas empíricas, algunos trabajos pueden ser el agrupamiento (Clustering) [7]; lógica difusa[13] y algoritmo Graph Cut [6]. Las fortalezas de ocupar aprendizajes no supervisados son determinar una guía de la segmentación de las lesiones y además sublesiones especiales pueden ser segmentadas, por otro lado las debilidades dependen bastante de la calidad con que se segmente el tejido y la especificación de reglas para cada tipo de lesión. Por ende el presente trabajo se inclinó ante la ventaja inherente de los algoritmos supervisados.

### 3. Metodología propuesta

La Metodología para clasificar las hiperintensidades presentes en la materia blanca, que se catalogan en este trabajo como lesión y no lesión se lleva a cabo mediante una base de datos de IRM axiales del cerebro ponderadas en T2-FLAIR previamente segmentadas [12], estas imágenes fueron proporcionadas por el Instituto Nacional de Rehabilitación (INR), son 29 imágenes con formato BMP con dimensiones de 640 x 600 píxeles.

#### 3.1. Segmentación previa

Primeramente se toman las imágenes previamente segmentadas seleccionadas por expertos con el algoritmo, posteriormente se realiza la diferencia entre la imagen de materia blanca (eliminación del cráneo y extracción de materia gris es especificado en [12]) con respecto a la imagen segmentada, (dado que la imagen segmentada se encuentra en colores es indispensable convertir la imagen a escala de grises para posteriormente identificar el vector con mayor índice de probabilidad en las hiperintensidades) y así obtener únicamente las hiperintensidades que el algoritmo segmentó, se puede apreciar en la Figura 2 b):

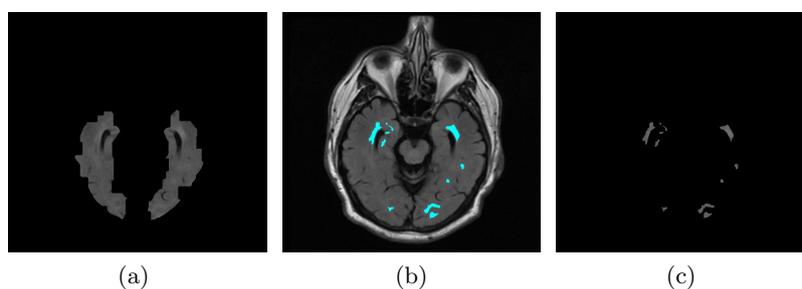


Fig. 2: a) Imagen de materia blanca, b) imagen segmentada, c) obtención de hiperintensidades.

### 3.2. Procesamiento de la imagen

Obtenidas las hiperintensidades es necesario recortar la imagen para fijar un área determinada de trabajo, se obtiene una caja delimitadora (Bounding Box), hallando los límites en el conjunto de hiperintensidades que se pueden observar en la Figura 3. Una vez delimitada el área de trabajo es indispensable adquirir únicamente las Regiones de Interés (ROI's) para cada hiperintensidad individualmente, para esto se binariza la imagen recortada con el umbralado por el método Otsu [11], para posteriormente etiquetar las hiperintensidades de componentes conexas y así obtener cada uno de los objetos, recorriendo los objetos de izquierda a derecha de arriba hacia abajo.

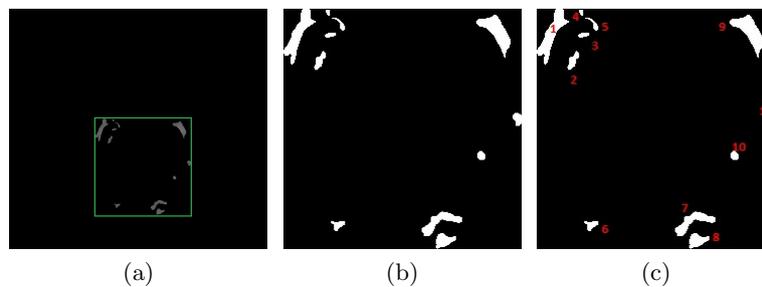


Fig. 3: a) Imagen de la caja delimitadora, b) imagen umbralado por método Otsu, c) imagen etiquetada.

### 3.3. Extracción de características

En esta etapa se extraen las características de cada hiperintensidad obtenida, de forma individual, el principal propósito de la extracción de características es reducir el conjunto de datos de la imagen, calculando algunas propiedades que se pueden ocupar en patrones contenidos en las imágenes, al obtener el vector característico con igual dimensionalidad al número de rasgos obtenidos, es de vital importancia para ejecutar una buena clasificación. Con la finalidad de extraer las características de la ROI's de primer orden el resultado escalar obtenido es almacenado en el vector  $\vec{F} = \{f_1, f_2, \dots, f_6\}$ , donde  $f_n \in \mathbb{R}$  para el vector de primer orden, asumiendo que se tiene una ventana de  $W = w_x \times w_y$  para cada hiperintensidad donde se obtiene la probabilidad de los píxeles con la función  $i(x,y)$  siendo  $x,y$  variables del espacio de la imagen siendo que  $i(x,y)$  puede tomar valores discretos  $[1,G]$  donde  $G = \max(i(x,y))$  es el máximo valor de nivel de gris. Inicialmente se utilizan características de la matemática estadística de primer orden, derivado de la obtención de la intensidad de niveles grises, para lo cual se definen las siguientes características de primer orden:

### Extracción de características del histograma (estadística de primer orden)

$$\text{Media: } \mu = \frac{1}{N} \sum_{i=1}^N i * p(i), \quad (1)$$

$$\text{Varianza: } \sigma^2 = \sum_{i=1}^N (i - \mu)^2 * p(i), \quad (2)$$

$$\text{Skewness: } \mu_3 = \sum_{i=1}^N (i - \mu)^3 * p(i), \quad (3)$$

$$\text{Kurtosis: } \mu_4 = \sum_{i=1}^N (i - \mu)^4 * p(i), \quad (4)$$

$$\text{Max: } Max = \operatorname{argmax} p(i), \quad (5)$$

$$\text{Min: } Min = \operatorname{argmin} p(i). \quad (6)$$

Por parte de las características de la ROI's de segundo orden se considera el centro de masa de cada hiperintensidad para determina una ventana máxima de  $\max = w_x \times w_y$  donde  $i(x,y)$ ,  $i \neq 0$ , a partir de esta ventana se puede obtener la matriz de co-ocurrencia de niveles de gris para posteriormente almacenar los resultados escalares en el vector  $\vec{F} = \{f_7, \dots, f_{11}\}$  y finalmente determinar las 11 características. Tomando así las características de segundo orden derivado de la relación espacial a través de las diferentes ROI's, dichas características pueden ser obtenidas de la matriz de co-ocurrencia de niveles de gris, también nombradas características de textura [14] para lo cual se considera lo siguiente:

### Extracción de características del análisis de textura

$$\text{Entropía: } - \sum_{i=1}^N \sum_{j=1}^M p(i, j) * \log(p(i, j)), \quad (7)$$

$$\text{Contraste: } \sum_{i=1}^N \sum_{j=1}^M p(i, j)^2 * (p(i, j)), \quad (8)$$

$$\text{Energía: } \sqrt{\sum_{i=1}^N \sum_{j=1}^M p^2(i, j)}, \quad (9)$$

$$\text{Correlación: } \frac{\sum_{i=1}^N \sum_{j=1}^M p(i, j) - \mu_i \mu_j}{\sigma_i \sigma_j}, \quad (10)$$

$$\text{Homogeneidad: } \sum_{i=1}^N \sum_{j=1}^M \frac{p(i, j)}{1 + (i - j)^2}. \quad (11)$$

### 3.4. Clasificador red neuronal

El modelo de una red neuronal multicapa con retropropagación se encuentra estructurada primeramente por las entradas o vector característico para alimentar la red neuronal, posteriormente se introducen las capas de neuronas (donde inicialmente se encuentra la capa oculta), es importante mencionar que las capas de neuronas contienen una función de activación que para nuestro caso en particular se implementa una función de activación sigmoidea; en la siguiente etapa se implementa un bloque de retraso, el cual se actualiza en pasos discretos y toma solamente valores enteros, finalmente las salidas son conectadas a las entradas, con la finalidad de ajustar los pesos en cada una de las capas, buscando un mínimo a una función de error, este error decrece con el número de iteraciones y la red neuronal converge a la solución deseada, debido a que en futuro las salidas de la red son ajustadas, estas se calculan a partir de las salidas anteriores se puede apreciar el clasificador en la Figura 4. Las redes recurrentes son potencialmente más potentes que las redes multicapa.

Para este clasificador en específico se utilizó una base de conocimiento previa, de la cual se obtuvieron los rasgos característicos de las lesiones y no lesiones dentro de la sustancia blanca, dicho reconocimiento de patrones se utilizan como entradas para alimentar la red neuronal con retropropagación, siendo ésta, capaz de detectar una separación no lineal entre las características dadas y así poder clasificar los patrones en las salidas correspondientes.

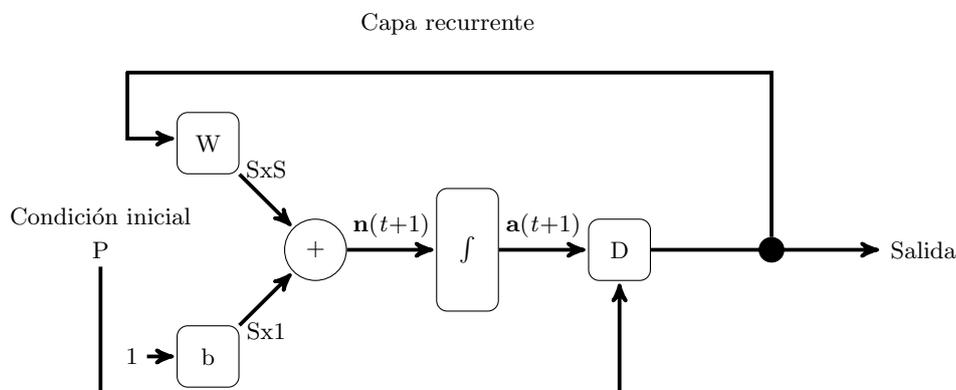


Fig. 4: Red recurrente [4].

**Entrenamiento** Para determinar el vector característico descrito en la sección 3.3, se alimentó la red tomando 24 imágenes de lesión y 5 imágenes de no lesión, siendo un total de 173 subimágenes y 90 subimágenes respectivamente. Al obtener el conjunto de datos característicos, se utiliza la información por secciones para minimizar directamente el error total, por ende se puede actualizar muchas veces las ponderaciones hasta que se cumpla la regla de parada, para ello

es necesario la lectura de varios datos. En este trabajo se utiliza una técnica de optimización, el algoritmo del gradiente conjugado, el cual converge a un mínimo de una función cuadrática en un número finito de iteraciones; Por parte de la clasificación se utiliza la retropropagación previamente explicada.

## 4. Resultados

El grupo de casos del conjunto de datos fueron 70% de las subimágenes como datos de entrenamiento, dentro de las cuales se incluyen los casos de lesión y no lesión, para los datos de validación se ocupó el 15%, de igual forma se utilizó el mismo porcentaje para los datos de prueba, dado que no hay un criterio general para la elección del número de neuronas, inicialmente se ocuparon  $\frac{2}{3}$  de los descriptores de entrada y posteriormente se iteraron pruebas para seleccionar el mejor rendimiento de la red, finalmente se seleccionaron 11 neuronas en la capa oculta.

Para la resolución de la matriz de confusión que se puede observar en la Figura 5 que se obtienen 2 salidas binarias indicando 1 para las lesiones y 0 para no lesiones, si el valor esperado de salida es lesión y la red neuronal clasifica correctamente como lesión, entonces se conoce como un VP, si por otra parte el valor de salida indica que es lesión pero es no lesión entonces se considera como Falso Positivo (FP). Verdaderos Negativos (VN) cuando la salida deseada es no lesión y el resultado esperado también es no lesión, cuando la salida es una lesión pero la red neuronal lo clasifica como no lesión entonces es Falso Negativo (FN). Dichos datos se pueden reflejar en la matriz de confusión indicando que 100% de no lesiones son correctamente clasificadas, 96 % de las lesiones son correctamente clasificadas y 4% de las lesiones son mal clasificadas, la matriz de confusión en general indica que la red neuronal multicapa con retropropagación tiene un alto valor predictivo de clasificación.

## 5. Conclusiones

Este trabajo se aplica en la metodología [12] con la finalidad de derivar un clasificador de hiperintensidades presentes en la sustancia blanca del cerebro; de las cuales se extraen características de estadística de primer orden mediante el histograma, también se ocupan las características de segundo orden análisis de textura correspondiente a las IRM para alimentar la RNMR. Los resultados obtenidos fueron completamente satisfactorios con una precisión del 98.9 %, cuya finalidad es identificar las lesiones y no lesiones presentes en la sustancia blanca. El método propuesto es aplicado únicamente con imágenes T2-Flair y el trabajo futuro a realizar por el presente proyecto es implementar un segmentador automático para precisar la correcta identificación de lesiones en el cerebro.



Fig. 5: Matriz de confusión de la red neuronal.

**Agradecimientos.** Este trabajo fue realizado gracias al apoyo del Consejo Nacional de Ciencia y Tecnología CONACYT y el Instituto Politécnico Nacional SIP-IPN mediante el proyecto No.20171559, COFAA-IPN y BEIFI-IPN.

## References

1. Esclerosis múltiple: Esperanza en la investigación (2012)
2. Akselrod-Ballin, A., Galun, M., Gomori, J.M., Filippi, M., Valsasina, P., Basri, R., Brandt, A.: Automatic segmentation and classification of multiple sclerosis in multichannel mri. *IEEE Transactions on Biomedical Engineering* 56(10), 2461–2469 (Oct 2009)
3. Anbeek, P., Vincken, K., A Viergever, M.: Automated ms-lesion segmentation by k-nearest neighbor classification (01 2008)
4. Demuth, H.B., Beale, M.H., De Jess, O., Hagan, M.T.: *Neural network design*. Martin Hagan (2014)
5. García-Lorenzo, D., Francis, S., Narayanan, S., Arnold, D.L., Collins, D.L.: Review of automatic segmentation methods of multiple sclerosis white matter lesions on conventional magnetic resonance imaging. *Medical image analysis* 17(1), 1–18 (2013)

6. García-Lorenzo, D., Lecoeur, J., Arnold, D.L.: Multiple Sclerosis Lesion Segmentation Using an Automatic Multimodal Graph Cuts, pp. 584–591 (2009)
7. Hill, J.E., Matlock, K., Pal, R., Nutter, B., Mitra, S.: Automated segmentation of ms lesions in flair, dir and t2-w mr images via an information theoretic approach (2016)
8. Lladó, X., Oliver, A., Cabezas, M., Freixenet, J., Vilanova, J.C., Quiles, A., Valls, L., Ramió-Torrent, L., Rovira, L.: Segmentation of multiple sclerosis lesions in brain mri: A review of automated approaches. *Information Sciences* 186(1), 164–185 (2012)
9. Moeskops, P., Viergever, M.A., Mendrik, A.M., de Vries, L.S., Benders, M.J.N.L., Igum, I.: Automatic segmentation of mr brain images with a convolutional neural network. *IEEE Transactions on Medical Imaging* 35(5), 1252–1261 (May 2016)
10. Muñoz-Pérez, M.J., Espinosa-Villaseñor, D.: Deterioro cognitivo y demencia de origen vascular. *Revista Mexicana de Neurociencia* 17(6), 85–96 (2016)
11. Otsu, N.: A threshold selection method from gray-level histograms. *IEEE Transactions on Systems, Man, and Cybernetics* 9(1), 62–66 (Jan 1979)
12. Patino-Correa, L.J., Pogrebnyak, O., Martinez-Castro, J.A., Felipe-Riveron, E.M.: White matter hyper-intensities automatic identification and segmentation in magnetic resonance images. *Expert Systems with Applications* 41(16), 7114–7123 (2014)
13. Pham, D.L.: Robust fuzzy segmentation of magnetic resonance images. In: *Proceedings 14th IEEE Symposium on Computer-Based Medical Systems. CBMS 2001*. pp. 127–131 (2001)
14. Regniers, O., Da Costa, J., Grenier, G., Germain, C., Bombrun, L.: Texture based image retrieval and classification of very high resolution maritime pine forest images. In: *International Geoscience and Remote Sensing Symposium (IGARSS)*. pp. 4038–4041 (2013)
15. Schmidt, P., Gaser, C., Arsic, M., Buck, D., Frschler, A., Berthele, A., Hoshi, M., Ilg, R., Schmid, V.J., Zimmer, C., Hemmer, B., Mhlau, M.: An automated tool for detection of flair-hyperintense white-matter lesions in multiple sclerosis. *NeuroImage* 59(4), 3774–3783 (2012)
16. Tian, D., Fan, L.: A brain mr images segmentation method based on som neural network. In: *2007 1st International Conference on Bioinformatics and Biomedical Engineering, ICBBE*. pp. 686–689 (2007)
17. Valverde, S., Cabezas, M., Roura, E.: Improving automated multiple sclerosis lesion segmentation with a cascaded 3d convolutional neural network approach. *NeuroImage* 155(Supplement C), 159 – 168 (2017)
18. Wels, M., Huber, M., Hornegger, J.: Fully automated segmentation of multiple sclerosis lesions in multispectral mri. *Pattern Recognition and Image Analysis* 18(2), 347–350 (2008)
19. Wu, Y., Warfield, S.K., Tan, I.L., Wells III, W.M., Meier, D.S., van Schijndel, R.A., Barkhof, F., Guttman, C.R.G.: Automated segmentation of multiple sclerosis lesion subtypes with multichannel mri. *NeuroImage* 32(3), 1205–1215 (2006)

# Aplicación del método de bisección para la mejora automática del comportamiento a-todo-momento del algoritmo ASO

Juan C. Moreno-Torres, Marco A. Moreno-Armendáriz

Instituto Politécnico Nacional, Centro de Investigación en Computación,  
CDMX, México

**Resumen.** El comportamiento a-todo-momento dota a un algoritmo de robustez al permitirle ser interrumpido en cualquier punto del proceso, mientras que se garantiza que se entregara la mejor respuesta posible cuando suceda la solicitud de paro. Este trabajo propone una metodología para identificar la relación de los parámetros de un algoritmo de optimización por colonia de hormigas con su comportamiento a-todo-momento, permitiendo implementar una técnica de mejora automática del comportamiento a-todo-momento utilizando una reducción en factores logarítmicos del espacio de búsqueda, disminuyendo la demanda computacional requerida para esta tarea.

**Palabras clave:** bisección, mejora automática, comportamiento a-todo-momento, algoritmo ASO.

## Application of the Bisection Method for the Automatic Improvement of the All-time Behavior of the ASO Algorithm

**Abstract.** The all-time behavior provides a robustness algorithm by allowing it to be interrupted at any point in the process, while ensuring that the best possible response is delivered when the stop request occurs. This work proposes a methodology to identify the relationship of the parameters of an optimization algorithm by colony of ants with its behavior at-all-moment, allowing to implement a technique of automatic improvement of the behavior at-all-moment using a reduction in logarithmic factors of the search space, decreasing the computational demand required for this task.

**Keywords:** Bisection Method, Automatic Improvement, All-time Behavior, ASO Algorithm.

## 1. Introducción

El algoritmo Ant System Optimization (ASO) requiere de seis parámetros que definen su comportamiento al explorar un espacio de soluciones:  $m$ ,  $Q$ ,  $\alpha$ ,  $\beta$ ,  $\rho$  y  $N$  [1]. La combinación de estos parámetros es lo que define la calidad de la optimización y su velocidad de convergencia [3].

Para su selección se han presentado propuestas iterativas, utilización de herramientas evolutivas y variación calendarizada de los parámetros durante la ejecución del algoritmo; sin embargo estas propuestas causan en incremento de la complejidad del algoritmo, dependencia al factor que fue considerado prioritario (calidad de la optimización o tiempo de convergencia en una búsqueda informada), o en el caso de las propuestas evolutivas la adición de sus propios parámetros a definir como óptimos. En 2010 se realizó una compilación que abarca la última década de investigación en este ámbito, en el cual se concluye que la correcta selección de parámetros y su completa comprensión en relación con su impacto temporal ofrece una reducción en complejidad y recursos computacionales, permitiendo eliminar el uso de herramientas adaptativas adicionales [5].

En todas las propuestas comentadas, para definir un conjunto de parámetros como “el mejor” se requiere de una suposición: que se cuenta con un tiempo indefinido para proponer la solución o que se conoce una aproximación al óptimo de antemano. López-Ibáñez en [4] propone un método de selección de parámetros basado en la optimalidad de Pareto, que define “mejor” como aquellos parámetros que deriven en conjuntos no-dominados que generen el mayor hipervolumen. El objetivo de este método de discriminación de combinaciones es generar la respuesta que converja en el menor tiempo posible y con la mayor calidad de optimización. Dicho método aplica exploración iterativa por todo el espacio de soluciones propuesto para cada parámetro; sin embargo este proceso puede resultar costoso en recursos computacionales, aún si solo requiere realizarse una vez para generar una solución para todo un conjunto de problemas de la misma clase, de manera que implementar una reducción del espacio de búsqueda resulta relevante.

El campo de soluciones que se requiere explorar es descrito en la ecuación (1), considerando *iter* como *iteración* y *param* como *parámetro*.

$$\begin{aligned} & \text{No. de combinaciones a explorar con el algoritmo López-Ibáñez}(X, N, Z, W, U, Y) = \\ & X \cdot N \cdot \sum_{h=1}^Z \prod_{j=1}^W \prod_{i=1}^U \left( \binom{Y(i, j, h)}{2} \right) \\ Y(i, j, h) = & \begin{cases} \text{ceil} \left( \left| \frac{(\text{param}(\text{max})_{ijh}) - \text{param}(\text{min})_{ijh}}{\Delta \text{param}_{ijh}} \right| \right) & \text{Si } Z=1, \text{ delta o none} \\ \text{ceil} \left( \left| \frac{(\text{param}(\text{max})_{ijh}) - \text{param}(\text{min})_{ijh}}{\Delta \text{param}_{ijh}} \right| \right) \cdot (\text{iter}_{ijh}) & \text{Si } Z=2, \text{ switch} \end{cases} \quad (1) \end{aligned}$$

donde:

$$\begin{aligned} &\Delta\text{parámetro, parámetro}(\max), \text{parámetro}(\min) \in \mathbb{R} \\ &\quad \text{parámetro}(\max) > \text{parámetro}(\min) \\ &\quad \text{iteración} \in \mathbb{N}, \text{iteración} \leq N \\ Z = \{1, 2\}, \text{comportamiento delta o none} = 1, \text{comportamiento switch} = 2 \\ &\quad X, Y, W, U, Y \in \mathbb{N} \end{aligned}$$

- X: número de problemas de prueba a utilizar.
- N: número de iteraciones máximas a aplicar para cada problema.
- Z: número de comportamientos a analizar, López-Ibañez definió delta, switch y none.
- W: número de parámetros en el algoritmo, para este caso de MMAS es 5, el mismo que para ASO.
- U: número de avances diferentes para cada parámetro, que dependerá de lo definido por el usuario según el algoritmo.
- Y: número de elementos en el dominio de cada parámetro, que dependerá de lo definido por el usuario según el algoritmo.
- $\Delta$ parámetro: avance considerado para el parámetro.
- parámetro(max): valor máximo considerado para el parámetro.
- parámetro(min): valor mínimo considerado para el parámetro.
- iteración: iteración en la que el parámetro realiza el cambio de valor en el comportamiento switch.

El método propuesto en este trabajo se enfoca en generar una reducción del espacio de búsqueda propuesto en [4] que se realiza de manera iterativa por todo su espacio de posibles soluciones, a través de la adaptación del algoritmo de bisección para búsqueda de raíces [2], con el fin de converger a una respuesta con una menor demanda computacional; para alcanzar este objetivo se requiere conocer la relación que mantiene cada parámetro con el comportamiento a-todo-momento del algoritmo estudiado, para lo cual se propone una metodología basada en el análisis del desempeño combinando calidad de optimización y velocidad de convergencia.

## 2. Preliminares

### 2.1. Perfil de desempeño

El perfil de desempeño  $P$  de un algoritmo a-todo-momento, denota la calidad  $q_i(t)$  de la respuesta esperada en función del tiempo  $t_i$  [6].

$$P = \{(t_1, q_1), (t_2, q_2), \dots, (t_n, q_n)\}.$$

### 2.2. Optimalidad de Pareto en perfiles de desempeño

Dados dos perfiles de desempeño  $P_i$  y  $P_j$ , que son resultado de dos algoritmos  $A_i$  y  $A_j$ , en la misma instancia  $n$ , se dice que  $P_i$  es mejor que  $P_j$ , en términos de optimalidad de Pareto, si y solo si  $P_i \neq P_j$  y  $\forall (t_j, q_j) \in P_j, \exists (t_i, q_i) \in P_i$  tal que  $(t_i, q_i) \preceq (t_j, q_j)$  [4].

### 2.3. Hipervolumen de un perfil de desempeño

El hipervolumen  $H(P_i)$  de un conjunto de vectores objetivo con punto de referencia  $(0, \dots, 0)$  es enunciado como:  $H(P_i) = \sum_{k=(0, \dots, 0)}^{(1, \dots, 1)} \overline{P}_i \cdot \gamma(k)$ , de manera que  $\overline{P}_i = \{(\overline{t}_i, \overline{q}_i) | \forall (\overline{t}_i, \overline{q}_i), \exists (t_i, q_i) \in P_i\}$  [4,7], donde

$$\overline{P}_i = \begin{cases} \overline{t}_i = 1 - (0,9 \cdot \frac{t_i - t_{min}}{t_{max} - t_{min}}), \\ \overline{q}_i = 1 - (0,9 \cdot \frac{q_i - q_{min}}{q_{max} - q_{min}}), \end{cases} \quad (2)$$

$$\gamma(k) = \begin{cases} 1 & \text{si } A \succcurlyeq \{k\}, \\ 0 & \text{caso contrario.} \end{cases} \quad (3)$$

$k$  : Vector objetivo en el perfil de desempeño.

## 2.4. Método de bisección

Es un método numérico de búsqueda de una raíz en una función que se sabe se encuentra dentro de un intervalo conocido.

Si una raíz de una función  $f(x)$  ha sido identificada dentro de los intervalos  $(x_1, x_2)$ , entonces  $f(x_1) \cdot f(x_2) < 0$ . Se calcula un  $f(x_3)$ , tal que  $x_3 = (x_1 + x_2)/2$  es el punto medio del intervalo. Si  $f(x_2)f(x_3) < 0$ , entonces la raíz debe encontrarse entre  $(x_3, x_2)$ ; para continuar la búsqueda se sustituye  $x_1$  por  $x_3$ . Si  $f(x_2) \cdot f(x_3) \not< 0$  entonces la raíz debe encontrarse entre  $(x_1, x_3)$  y se realiza la sustitución de  $x_2$  por  $x_3$ . En cualquiera de ambos casos el nuevo intervalo solo contiene la mitad de los elementos del intervalo original. Este procedimiento se continúa hasta que el intervalo es reducido a un valor  $\varepsilon$  definido como límite de búsqueda, tal que  $|x_2 - x_1| \leq \varepsilon$ .

El intervalo original  $\Delta x$  es reducido a  $\Delta x/2$  tras la primer bisección,  $\Delta x/2^2$  tras dos bisecciones, continuando hasta  $n$  bisecciones, donde se tiene  $\Delta x/2^n = \varepsilon$ . Despejando para el número de operaciones necesarias:  $n = \frac{\ln(|\Delta x|/\varepsilon)}{\ln 2}$  [2].

## 3. Propuesta

### 3.1. Adecuación del método de bisección para la mejora automática del comportamiento a-todo-momento

La exploración del algoritmo heurístico en todo el dominio de cada parámetro expresado en la ecuación 1, representa el mayor impacto en cuanto al número de operaciones requeridas en la búsqueda de la mejora del comportamiento a-todo-momento; particularmente se enfocaran la atención en el comportamiento de parámetros constantes que comparte dominio con delta, pues es un caso particular cuando solo hay un  $\Delta$ parámetro $_{ijh}$ , mientras que el comportamiento switch solo es una multiplicidad del segundo, siendo la reducción en switch una derivación de la reducción del comportamiento delta.

El dominio de  $Y(i, j, h)$  con el mayor número de elementos sucede con:  $\min(\Delta$ parámetro $_{ijh})$ , por lo que será considerado como caso base de estudio.

La implementación de un algoritmo de búsqueda por bisección permite reducir la búsqueda en la siguiente expresión (considerando  $iter$  como  $iteración$  y  $par$  como  $parámetro$ ):

$$Yb(i, j, h) = \begin{cases} \left( \frac{\text{ceil} \left( \frac{\ln \left( \left| \frac{\text{par}(\max)_{ijh} - \text{par}(\min)_{ijh}}{\Delta \text{par}_{ijh}} \right| \right)}{\ln 2} \right)}{\text{ceil} \left( \frac{\ln \left( \left| \frac{\text{par}(\max)_{ijh} - \text{par}(\min)_{ijh}}{\Delta \text{par}_{ijh}} \right| \right)}{\ln 2} \right)} \right) & \text{Si } Z=1, \text{ delta o none} \\ \left( \frac{\text{ceil} \left( \frac{\ln \left( \left| \frac{\text{par}(\max)_{ijh} - \text{par}(\min)_{ijh}}{\Delta \text{par}_{ijh}} \right| \right)}{\ln 2} \right)}{\text{ceil} \left( \frac{\ln \left( \left| \frac{\text{par}(\max)_{ijh} - \text{par}(\min)_{ijh}}{\Delta \text{par}_{ijh}} \right| \right)}{\ln 2} \right)} \right) \cdot (\text{iter}_{ijh}) & \text{Si } Z=2, \text{ switch} \end{cases} \quad (4)$$

Las reglas de decisión del algoritmo de bisección propuestas para el recorte del dominio de cada parámetro son las siguientes:

Sea un intervalo de búsqueda  $[a, b]$  con un  $\Delta u$ ,  $c = a + (\text{round}(\frac{1 + \frac{b-a}{\Delta u}}{2}) \cdot \Delta u)$  es el punto medio del intervalo, donde  $a < c < b$ .  $P_i$  es el perfil de desempeño dada una evaluación en el valor  $i$ , y  $H(P_i)$  es el hipervolumen resultante del perfil de desempeño  $P_i$ .

Si  $(H(P_a) > H(P_b) > H(P_c)) \vee (H(P_a) > H(P_c) > H(P_b)) \vee (H(P_c) > H(P_a) > H(P_b))$

entonces el espacio de búsqueda de la siguiente iteración sera  $[a, c]$ .

Si  $(H(P_b) > H(P_a) > H(P_c)) \vee (H(P_b) > H(P_c) > H(P_a)) \vee (H(P_c) > H(P_b) > H(P_a))$

entonces el espacio de búsqueda de la siguiente iteración sera  $[c, b]$ .

Siendo la condición de paro  $\frac{b-a}{\Delta u} = 0$ , ó  $\max(Yb(i, j, h))$  iteraciones.

A continuación se muestra en pseudocódigo el algoritmo de Bisección a-todo-momento para el comportamiento *none* con  $\min(\Delta \text{parámetro})$ .

```

n = No. de parametros del algoritmo
Definición del espacio de búsqueda para cada parámetro  $a_n \leq Param_n \leq b_n$ 
Definición del avance para cada parámetro  $\Delta Param_n$ 
Cálculo de  $P_n = Yb_n$  para cada parámetro
nit = max( $P_n$ ) condición de paro
Cálculo de  $c_n$  de cada parametro
while x ≤ nit do
  while y ≤ n do
    Evaluacion del algoritmo heurístico (ASO) con  $Param_y = a_y$ 
    Evaluacion del algoritmo heurístico (ASO) con  $Param_y = b_y$ 
    Evaluacion del algoritmo heurístico (ASO) con  $Param_y = c_y$ 
    Almacenamiento de perfiles de desempeño y conjunto de parámetros asociados
    Comparación del hipervolumen de entre los tres puntos evaluados
    Aplicación de las reglas de decisión para la actualización de  $c_y$ 
    y++
  end while
  x++
end while
Cálculo del hipervolumen normalizado para todo perfil de desempeño obtenido
Selección del hipervolumen normalizado más alto y extracción de sus parámetros asociados
Fin del algoritmo
    
```

### 3.2. Metodología de selección de rangos

Una característica primordial para el funcionamiento del algoritmo de bisección a-todo-momento es que se debe garantizar un par de elementos  $[a_n, b_n]$

para todo parámetro, entre los cuales se sepa que se encuentra el máximo global con un comportamiento monótono creciente o decreciente, pero de los cuales se desconoce su valor exacto e interacción con otros parámetros.

La metodología propuesta a continuación permitirá identificar los umbrales adecuados para su selección:

1. Identificación del dominio de cada parámetro.
2. Definición de un conjunto de parámetros predeterminados.
3. Cálculo del hipervolumen para variaciones independientes de cada parámetro, manteniendo el resto fijos.
4. Cálculo del hipervolumen con variación en dos parámetros, manteniendo el resto fijos, explorando regiones de interés extraídas del paso anterior.
5. Selección de las regiones con hipervolumenes más altos con comportamientos monótonos donde  $a_n < b_n$ .

Debido a los componentes estocásticos del algoritmo ASO, los pasos 3 y 4 deben ser analizados estadísticamente.

#### 4. Caso de estudio: Mejora automática del comportamiento a-todo-momento por bisección del algoritmo ASO para el TSP

##### 4.1. Selección de rangos de entrada para el algoritmo de bisección a-todo-momento

El dominio de cada parámetro se encuentra definido en [1]. Para todas las pruebas realizadas en esta sección se utilizó las instancia de la librería del agente viajero TSPLIB aplicando Eil51 y KroD100 a menos que se mencione lo contrario, mientras que se realizaron 30 evaluaciones con 300 iteraciones cada una. Los parámetros predeterminados fueron seleccionados en concordancia con las recomendaciones de [1] y [4], y son mostradas en la tabla 1.

El primer análisis corresponde a recorrer por un dominio útil cada parámetro de manera independiente, manteniendo el resto en un valor fijo por todo el proceso, con la finalidad de conocer el impacto individual en el comportamiento a-todo-momento del algoritmo. Dos gráficas son presentadas para cada variable para su posterior análisis. La primera es una representación de caja de todos los resultados, la segunda el comportamiento de la media aritmética, incluyendo sus desviaciones estándar para cada punto. En las figuras 1 a 4 se puede observar el comportamiento de dichas variables.

**Tabla 1:** Parámetros predeterminados y dominios útiles considerados.

$m$	$\alpha$	$\beta$	$\rho$	$Q$
0.8*No. de ciudades	1.0	2.0	0.2	1.0
$1 < m < 120$	$0,1 < \alpha < 5,0$	$0,1 < \beta < 5,0$	$0,1 < \rho < 0,9$	$1 < Q < 10$

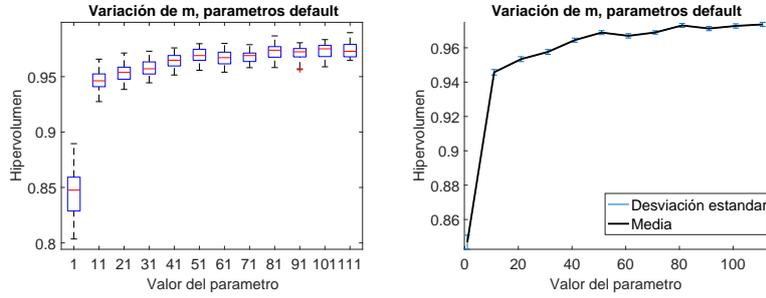


Fig. 1: Comportamiento a-todo-momento con variación de m.

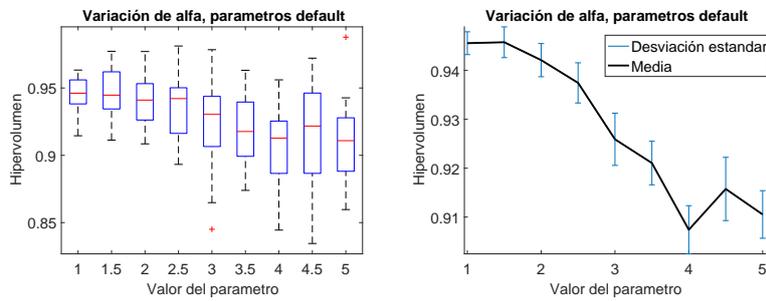


Fig. 2: Comportamiento a-todo-momento con variación de alfa.

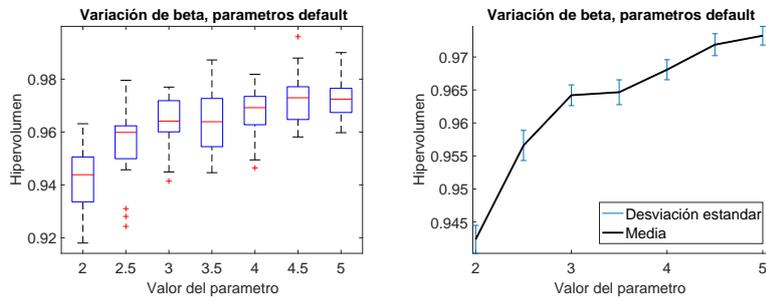


Fig. 3: Comportamiento a-todo-momento con variación de beta.

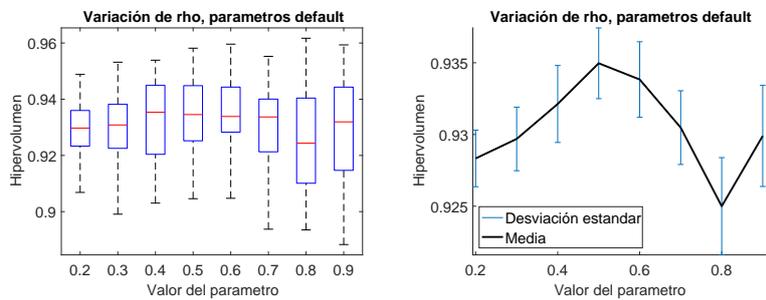


Fig. 4: Comportamiento a-todo-momento con variación de rho.

Con la finalidad de identificar relaciones entre pares de parámetros un enfoque similar se aplicó, excepto que esta vez realizando variaciones sobre dos parámetros de manera secuencial. Los mismos valores predeterminados se aplicaron para los parámetros fijos, así como el mismo número de iteraciones y evaluaciones. El parámetro de comparación elegido para esta tarea fue  $\alpha$ , ya que de las gráficas de comportamiento individual mostró la mayor influencia sobre el comportamiento a-todo-momento, además de encontrarse directamente influenciada por 3 de los 4 otros parámetros en las ecuaciones del algoritmo. En las figuras 5 y 6 se observa la respuesta obtenida.

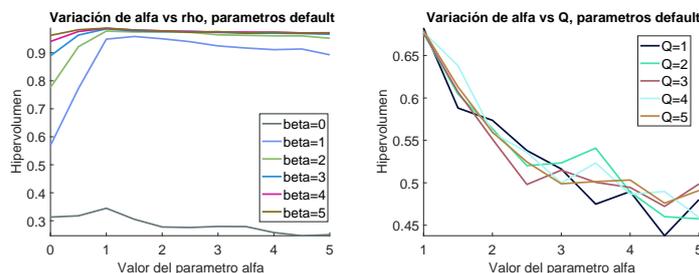


Fig. 5: Comportamiento a-todo-momento con variación de alfa y beta (izquierda), comportamiento a-todo-momento con variación de alfa y Q (derecha).

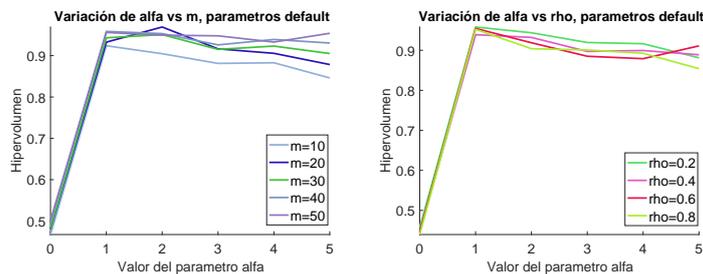


Fig. 6: Comportamiento a-todo-momento con variación de alfa y m (izquierda), comportamiento a-todo-momento con variación de alfa y rho (derecha).

Finalmente, se seleccionaron las regiones que cumplen con las restricciones del algoritmo de bisección a-todo-momento y se procedió a implementar una comparación del algoritmo de mejora del comportamiento a-todo-momento de López-Ibáñez según el pseudocódigo descrito en [4] y el propuesto en este trabajo.

Los algoritmos fueron evaluados 5 veces consecutivas independientes por 200 iteraciones bajo los dominios de la tabla 2:

Tabla 2: Rangos a explorar con los algoritmos de mejora automática del comportamiento a-todo-momento.

$20 \leq m \leq 80$	$0,5 \leq \alpha \leq 3,5$	$2 \leq \beta \leq 5$	$0,4 \leq \rho \leq 0,7$	$1 \leq Q \leq 3$
$\Delta m = 10$	$\Delta \alpha = 0,5$	$\Delta \beta = 0,5$	$\Delta \rho = 0,1$	$\Delta Q = 1$

Para el algoritmo López-Ibáñez el número de elementos a estudiar en ASO con los rangos mostrado arriba corresponde a  $Y(i, j, h) = 1944$  por evaluación;

por otro lado para el algoritmo de bisección a-todo-momento los elementos por evaluación corresponden a  $Yb(i, j, h) = 108$ . La media resultante para cada método se muestra en la tabla 3:

**Tabla 3:** Media de los parámetros resultantes de la mejora automática del comportamiento a-todo-momento.

	m	$\alpha$	$\beta$	$\rho$	Q
Bisección Completo	54	1.3	4.6	0.68	1.8
Algoritmo López-Ibáñez	78	1.6	4.6	0.7	1.6

El comportamiento a-todo-momento con los dos conjuntos de parámetros obtenidos es evaluado a través del hipervolumen para 30 evaluaciones con 300 iteraciones cada una con la finalidad de apreciar las respuestas generadas de manera estadística. La media aritmética y desviación estándar servirán como comparativas, y pueden se pueden ver en la tabla 4.

**Tabla 4:** Comparación de desempeño del hipervolumen entre algoritmos de mejora automática del comportamiento a-todo-momento.

	Media aritmética del Hipervolumen generado	Desviación Estándar del Hipervolumen generado	Mejor respuesta del Hipervolumen generado
Bisección Completo	0.8521	0.0581	0.9588
Algoritmo López-Ibáñez	0.8178	0.0693	0.9621

Se registró el tiempo de procesamiento para las 5 pruebas de cada algoritmo, a continuación se presenta el resultado promedio. Todas las iteraciones se realizaron sobre un procesador Inter i5 de 2.5MHz programadas en MATLAB.

**Tabla 5:** Comparación de desempeño temporal entre algoritmos de mejora automática del comportamiento a-todo-momento.

	Tiempo de ejecución promedio por prueba (segundos)
Bisección Completo	1857
Algoritmo López-Ibáñez	34,991

## 4.2. Análisis de los datos

Las gráficas del comportamiento de  $\alpha$  y  $\rho$  nos permiten identificar un rango claro donde el comportamiento genera el hipervolumen mas alto y los cuales cumplen las características exigidas por el algoritmo propuesto.

Los parámetros  $m$ ,  $Q$  y  $\beta$  permiten como domino todos los números naturales, por lo que el estudio de su comportamiento se ve limitado a los rangos seleccionados que son considerados como factibles a utilizarse. Este tipo de análisis refleja que  $Q$  no mantiene relación con el comportamiento a-todo-momento del algoritmo, ya que su variación no afecta significativamente la respuesta del hipervolumen, debido a esta razón y la limitante de espacio sus gráficas son omitidas del documento. El parámetro  $m$  por otro lado mantiene una relación directamente proporcional con el comportamiento a-todo-momento del algoritmo. Se puede identificar que un valor demasiado alto de  $\beta$  reduciría el elemento heurístico del algoritmo, por lo que debe ser delimitado de manera que no evite la integración de la búsqueda al azar.

El tiempo de ejecución para el caso de estudio utilizando Bisección a-todo-momento en el peor de los casos resultó ser 18.8 veces menor que el algoritmo López-Ibáñez, esto debido al método de búsqueda que redujo las iteraciones necesarias de igual manera 18 veces. Analizando los perfiles de desempeño de los tres algoritmos bajo el criterio de optimalidad de Pareto y teniendo en cuenta la reducción en tiempo de ejecución, el algoritmo de Bisección Completo se perfila como el óptimo.

## 5. Conclusiones

Este trabajo presenta dos principales aportes al análisis y mejora del comportamiento a-todo-momento del algoritmo ASO. El primero propone una metodología para analizar el impacto de cada parámetro con respecto a la respuesta a-todo-momento, el cual no se limita a este caso de estudio específico, sino que es aplicable sobre cualquier algoritmo heurístico que genere un perfil de desempeño con elementos no-dominados y su comportamiento dependa de la selección de un conjunto de parámetros. El segundo aporte radica en la propuesta de la implementación del algoritmo de bisección en la mejora automática del comportamiento a-todo-momento desarrollada en [4], los resultados para la instancia analizada mostraron una reducción de 18.8 veces el tiempo de ejecución del algoritmo original, mientras que convergen a parámetros que ofrecen resultados similares.

**Agradecimientos.** Los autores extendemos un agradecimiento al gobierno Mexicano (Instituto Politécnico Nacional, EDI-IPN, COFAA-IPN, SIP-IPN, BEIFI-IPN, CONACYT, SNI), por proporcionar el apoyo necesario para el desarrollo de este trabajo; los resultados aquí presentados son posibles gracias a estas instituciones.

## Referencias

1. Dorigo, M., Maniezzo, V., Coloni, A.: Ant system: optimization by a colony of cooperating agents. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)* 26(1), 29–41 (1996)
2. Kiusalaas, J.: *Numerical methods in engineering with MATLAB*, vol. 1. Cambridge University Press (2005)
3. Lobo, F., Lima, C.F., Michalewicz, Z.: *Parameter setting in evolutionary algorithms*, vol. 54. Springer Science and Business Media (2007)
4. López-Ibáñez, M., Stützle, T.: Automatically improving the anytime behaviour of optimisation algorithms. *European Journal of Operational Research* 235(3), 569–582 (2014)
5. Stützle, T., López-Ibáñez, M., Pellegrini, P., Maur, M., De Oca, M.M., Birattari, M., Dorigo, M.: Parameter adaptation in ant colony optimization. *Autonomous search* pp. 191–215 (2011)

6. Zilberstein, S.: Using anytime algorithms in intelligent systems. *AI magazine* 17(3), 73 (1996)
7. Zitzler, E., Brockhoff, D., Thiele, L.: The hypervolume indicator revisited: On the design of pareto-compliant indicators via weighted integration. In: *Evolutionary multi-criterion optimization*. pp. 862–876. Springer (2007)



## Uso de tecnologías web para la entrega de datos de sensado de una red de distribución de agua usando la placa de desarrollo Intel Galileo

Juan Antonio Alvarado Elías, Roberto Solís, Oscar Osvaldo Ordaz García,  
José Guadalupe Arceo Olague

Universidad Autónoma de Zacatecas,  
Zacatecas, México

tonnyalvarado94@gmail.com, rsolis@uaz.edu.mx, oscarord27@hotmail.com,  
arceojg@uaz.edu.mx

**Resumen.** Este trabajo expone el análisis, diseño e implementación de un sistema de procesamiento y presentación de datos de presión, flujo y nivel de agua de una Red de Distribución de Agua Potable (RDAP) para su uso en el Internet de las Cosas a través de una placa de desarrollo Intel Galileo. Se establecen los componentes principales del procesamiento del sistema inteligente y los elementos utilizados por la placa de desarrollo Intel Galileo. El diseño y desarrollo del procesamiento de datos se evaluó realizando pruebas de adquisición de voltaje con una fuente variable y se verificaron los resultados experimentales obtenidos. La presentación de datos se compara con los valores esperados en respuesta a la entrada de voltaje en contraste con los datos que se visualizan en un dispositivo LCD (Liquid Crystal Display) y en páginas web apoyándose en tecnologías web como Dweet.io, Freeboard.io y Telegram.

**Palabras clave:** internet de las cosas, placa de desarrollo Intel Galileo, tecnologías web, dweet.io, freeboard.io, telegram.

### Use of Web Technologies for the Delivery of Sensing Data in a Drinking Water Distribution Network Using the Intel Galileo Development Board

**Abstract.** This work exposes the analysis, design, and implementation of a system that processes and presents the pressure, flow and water level data of a Drinking Water Distribution Network (DWDN) for its use in the Internet of Things through an Intel Galileo development board. The main components for the intelligent system processing and the elements used by the Intel Galileo development board are established. The design and development of data processing was evaluated by performing voltage acquisition tests with a variable voltage source and the experimental results obtained were verified. The presentation of data is compared with the expected values in response to the voltage input in contrast to the

data displayed on an LCD (Liquid Crystal Display) device and on web pages based on web technologies such as Dweet.io, Freeboard.io and Telegram.

**Keywords:** Internet of Things, Intel Galileo Development Board, Web Technologies, Dweet.io, Freeboard.io, Telegram.

## 1. Introducción

El sistema de distribución de agua potable es deficiente en las ciudades como Zacatecas, colonias se quedan sin agua diariamente y existen fugas en diferentes espacios en donde se pierde aproximadamente el 40 por ciento del agua potable [1]. Estos problemas podrían minimizarse si existe una forma de obtener mediciones en tiempo real de parámetros como presión, flujo y nivel de agua para la toma de decisiones ya que permiten localizar posibles fugas mediante algoritmos de detección [2].

Actualmente se han realizado varios trabajos para obtener mediciones de los parámetros citados mediante dispositivos lógicos programables [3], sin embargo, es requerido mejorar la sensibilidad, estabilidad y la presentación de los datos para poder implementarse en una RDAP.

Este trabajo está enfocado en procesar y presentar los resultados obtenidos de las lecturas adquiridas en cada ADC (Analog to Digital Converter) de una placa de desarrollo Intel Galileo [4]. Los voltajes hacen referencia a los parámetros que se miden con sensores de presión, flujo y nivel de agua. Para la interpretación de los datos se realiza su procesamiento en Python y después se despliegan los resultados mediante el módulo LCD y mediante las tecnologías web Dweet.io [5], Freeboard.io [6] y Telegram [7]. Dweet.io es un sistema de mensajes y alertas para el Internet de las Cosas, Freeboard.6 es una herramienta de visualización simple y fácil de usar, Telegram es una app de mensajes para desktops y móviles basada en la nube.

El documento está organizado de la siguiente manera: primero se comentan los antecedentes y los trabajos previos; posteriormente se explican los materiales y los métodos para la descripción de la arquitectura, procesamiento y representación de datos; enseguida se comentan los resultados y finalmente las conclusiones.

## 2. Antecedentes y trabajos previos

Actualmente existen instrumentos que permiten realizar lecturas de diferentes parámetros en una RDAP y su comportamiento. El manómetro mide la presión, el rotámetro mide el flujo y el visor de nivel hidrostático mide el nivel de agua [8, 9, 10]. Así como existen instrumentos para las diferentes mediciones que se realizan en una RDAP, también existen sensores que realizan la conversión de los parámetros físicos en corriente eléctrica o voltaje, que deben ser caracterizados. La recolección manual de estos parámetros es ineficiente para la toma de decisiones en la distribución del agua, dado que la precisión y el tiempo requerido para tomar las muestras dependen de la pericia del capturista [11], por lo que monitorear las magnitudes de los sensores a través de una

placa de desarrollo Intel Galileo supone una ventaja para la toma de decisiones en la distribución de agua.

Se han realizado trabajos en Arduino [12, 13] y FPGA [14] para resolver algunos problemas de adquisición y presentación de datos referentes a una RDAP; sin embargo, no los han resuelto del todo, ya que se han encontrado diferentes problemas de conversión, procesamiento y presentación. Este trabajo resuelve los problemas de procesamiento y presentación, mediante el uso de la placa de desarrollo Intel Galileo y el uso de tecnologías Web. Los aspectos a resolver son el procesamiento del cálculo matemático para cada lectura asegurando la asertividad de los valores y la generación de la presentación de resultados en tiempo real mediante tecnologías Web que permita el fácil acceso a la información.

El sistema para la adquisición de datos de presión, flujo y nivel de agua, se compone de cuatro partes fundamentales: (1) la lectura de las magnitudes obtenidas por los sensores, (2) el acoplamiento de los datos obtenidos de los sensores a la placa de desarrollo Intel Galileo mediante un circuito conversor Corriente (I) a Voltaje (V), (3) el procesamiento de los datos obtenidos de los sensores ahora convertidos en magnitud de voltaje, y (4) la presentación de datos mediante el dispositivo LCD, la página de visualización de datos Freeboard.io con ayuda del servicio de mensajes de Dweet.io y la aplicación Telegram. Las cuatro etapas se pueden ver en la Fig. 1.

En este trabajo se presentan las secciones de procesamiento y presentación de datos obtenidos mediante la placa de desarrollo Intel Galileo, para las cuales se necesitan los materiales descritos en la siguiente sección.

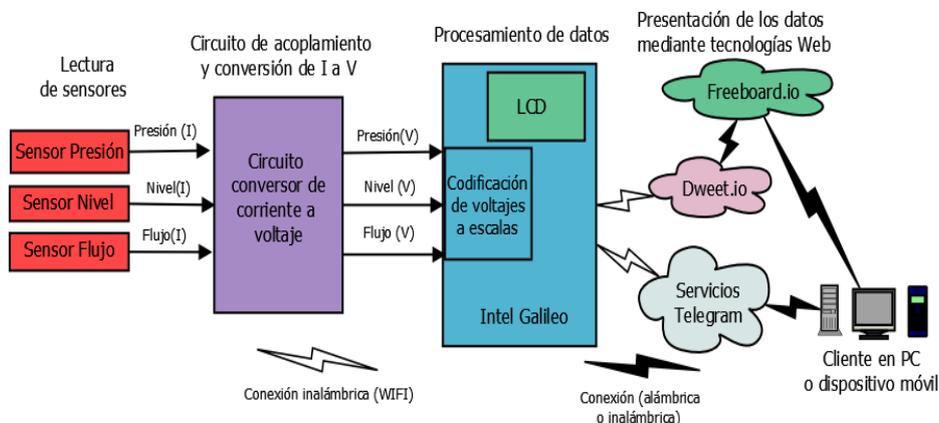


Fig. 1. Diagrama de bloques del sistema de obtención de datos.

### 3. Materiales y métodos

En el proyecto se utilizó la placa de desarrollo Intel Galileo, un Grove Starter Kit compatible con la placa de desarrollo, una unidad extraíble la cual es una memoria microSD desde la cual arranca el sistema operativo instalado, el cual es Linux en la distribución llamada Yocto, utilizada para Internet de las Cosas, un cable On-the-go

(OTG) y un dispositivo WIFI USB para conexión inalámbrica. La placa de desarrollo Intel Galileo contiene un procesador Intel Quark SoC X1000 de 32 bits que funciona a una frecuencia de 400 MHz, la placa dispone de un puerto USB host, un puerto USB cliente, un puerto RS-232 y un zócalo para tarjetas microSD con capacidad de soportar memorias de hasta 32 GB.

La placa dispone de un voltaje de operación configurable entre 3.3 V y 5 V, además de 6 puertos de entrada analógica con un circuito conversor analógico/digital (ADC) modelo AD7298 de Analog Devices con una resolución configurable de 10 bits o 12 bits. La placa tiene la capacidad de soportar un sistema operativo Linux instalado desde una memoria microSD.

El Grove Starter Kit contiene un Shield o placa de acoplamiento con voltaje de operación de 3.3 V y 5 V; conjuntamente contiene un dispositivo LCD [15]. Las especificaciones de la placa determinan que la memoria microSD que soporta es de tipo SD/HC, de clase 4 con capacidad máxima de 32 GB y mínima de 8 GB, la cual debe ser configurada para iniciar desde un dispositivo USB con el sistema operativo Linux especial para Galileo [16].

El cable OTG se recomienda con especificación Hub 2.0, pero puede soportar la especificación Hub 1.0, esto cuando sea compatible con el dispositivo WIFI USB. Para la conexión inalámbrica, es necesario un dispositivo WIFI USB con velocidad de 150Mbps o 300Mbps, con protocolo de conexión 802.11n, siendo necesario para casos especiales instalar los controladores adicionales dependiendo del modelo de dispositivo WIFI USB.

Para la programación del sistema se utilizó una computadora con el controlador FT231 USB UART instalado, para el reconocimiento de la placa de desarrollo Intel Galileo, y el software de conexión serial configurable a una velocidad de 115200 bits por segundo. Es necesario tener una conexión a Internet para la interacción con el sistema de mensajes Dweet.io así como con los servicios de Telegram.

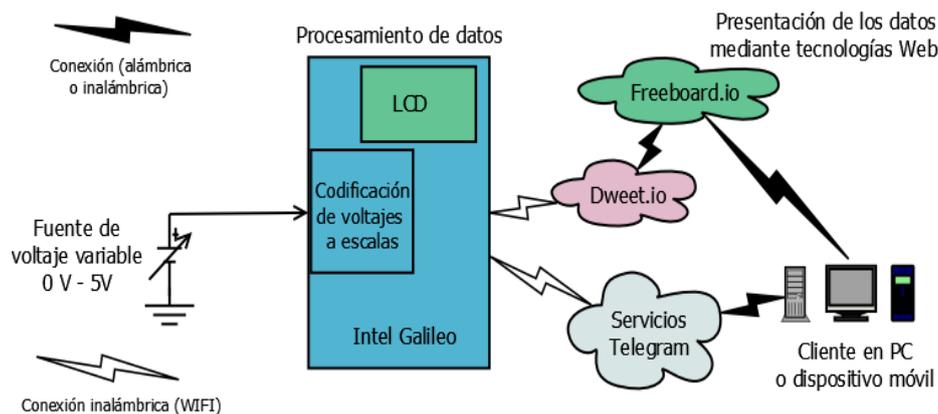


Fig. 2. Diagrama general del proyecto.

El proyecto se realizó utilizando el lenguaje de programación Python [17], el cual es un lenguaje de código abierto, portable, orientado a objetos y dinámico, ya que tiene la

posibilidad de soportar módulos de diferentes lenguajes. Además, posee librerías y características poderosas adaptables a las diferentes necesidades de conexión a través de Internet.

Para conectar todos los materiales anteriores primero debe realizarse la correcta instalación del sistema operativo Linux en una tarjeta micro SD. Para instalarse debe insertarse en la ranura de tarjetas de la placa de desarrollo Intel Galileo y conectarse a la alimentación de voltaje. Acto seguido se debe conectar a la computadora e instalar las librerías de Python requeridas.

Los directorios se manejan igual que en un sistema operativo Linux, solamente debe crearse un directorio de trabajo y de manera opcional se puede trabajar con la herramienta *git* en cualquier plataforma que preste este servicio.

El diagrama general del proyecto se muestra en la Fig. 2, que se enfoca en las etapas de procesamiento y presentación de datos.

Para el análisis de la etapa de procesamiento, se consideraron tres sensores con rango de salida de corriente continua (CC) de 4 a 20 mA:

- Sensor Serie 626 Pressure Transmitter de Dwyer para medir presión con un rango de 0 a 40 Bar de presión, con una precisión de 0.25%. Requiere alimentación de 10 a 30 V<sub>cc</sub> [18].
- Sensor 2551 de Signet para medir el flujo con un rango de 0.5 a 10 m/s con una precisión de  $\pm 1.0\%$ . Requiere alimentación de 24 V<sub>cc</sub> ( $\pm 10\%$ ) [19].
- Sensor ultrasónico serie ToughSonic® modelo TSPC-30S1-232 de SENIX para medir la altura libre de agua en tanques con un rango de 10.16 a 426.72 cm, con precisión del 0.5%. Requiere alimentación de 15 a 30 V<sub>cc</sub> [20].

En la Fig. 3 se muestra gráficamente cómo se desarrolla el flujo de información considerado para el funcionamiento de las etapas de procesamiento y presentación. Inicialmente se realiza la lectura de cada uno de los ADC, para que tales lecturas sean procesadas por el primer módulo de programa y sean presentadas en el LCD.

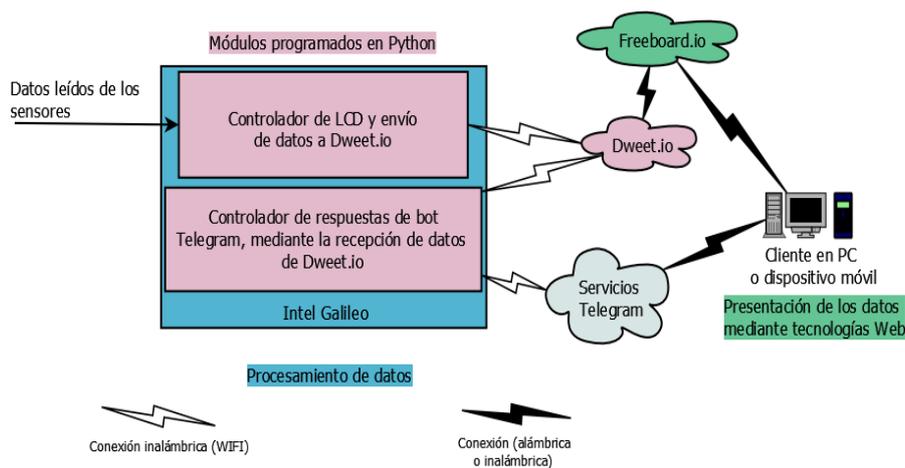


Fig. 3. Flujo de información dentro del micro controlador Galileo.

Una vez realizado el procesamiento de los datos, éstos son enviados a Dweet.io. El nombre Dweet nace de una analogía a Twitter en donde el principio es publicar datos desde “cosas” o aparatos electrónicos conectados a Internet [5]. El segundo módulo del programa, se encarga de la atención exclusiva a un bot de Telegram [21]. El bot es un módulo programado que permite realizar respuestas predefinidas para una aplicación específica. Telegram cuenta con bots en su servicio. Estos son pequeños programas que se encuentran en los servidores y pueden ser “llamados” en cualquier momento para conseguir cientos de funcionalidades diferentes para ser usadas en los chats en la aplicación. Además de estos módulos, se genera otro programa para obtener la lectura de los puertos analógicos A0, A1 y A2, que es donde se conectan los circuitos que convierten la I a V, proveniente de las señales de los sensores de presión, flujo y nivel respectivamente. Para obtener una mejor precisión, cada ADC se configura con una resolución de 12 bits, por lo que se pueden obtener 4,096 valores diferentes de voltaje. Después de tomar la lectura, se realiza el cálculo correspondiente a cada ADC y se almacena en una variable que identifica cada parámetro, para ser presentada en el dispositivo LCD y se agrupan los valores para enviarse a Dweet.io.

**Tabla 1.** Valores mínimos y máximos de medición de los sensores.

Sensor	Rango del Sensor
Presión	0 – 40 bares
Flujo	0.5 – 10 m/s
Nivel	10.16 – 426.72 cm

En la Tabla 1, se muestran los valores de los parámetros mínimo y máximo que son capaces de detectar los diferentes sensores, además de las unidades de medición, los cuales se especifican en las fichas técnicas correspondientes. El rango de voltaje para las entradas de cada ADC fue configurado de 0 a 5 V. Con estas consideraciones se realiza el cálculo matemático con la relación proporcional directa en ecuación **¡Error! No se encuentra el origen de la referencia.:**

$$LS = \left( \frac{V_e}{V_{max}} * (S_f - S_i) \right) + S_i, \quad (1)$$

dónde:

- LS: Lectura del sensor
- Ve: Voltaje de entrada a convertir,
- Vmax: Voltaje de salida máximo,
- Sf: Valor máximo de escala del sensor,
- Si: Valor mínimo de escala del sensor.

Considerando la relación matemática obtenida, se realiza el diseño de un algoritmo para encargarse de la lectura de los valores enviados por los sensores a cada ADC y entonces realizar el procesamiento de datos. Para el código del bot de Telegram se utiliza una plantilla Open Source, realizando las adecuaciones necesarias para cumplir con las expectativas del proyecto.

Para la presentación de datos usando la herramienta web Freeboard.io es necesario realizar el registro de una cuenta en su sitio web. Con la cuenta asignada se puede ingresar con el usuario y contraseña respectivos. Freeboard.io despliega de manera visual

la información enviada a Dweet.io, por lo cual es de vital importancia para el diseño de sistema, entender cómo funciona tal servicio y como debe estructurarse e identificarse la información que se le envía. La Fig. 4 muestra un ejemplo de cómo se despliegan los datos en Freeboard.io.

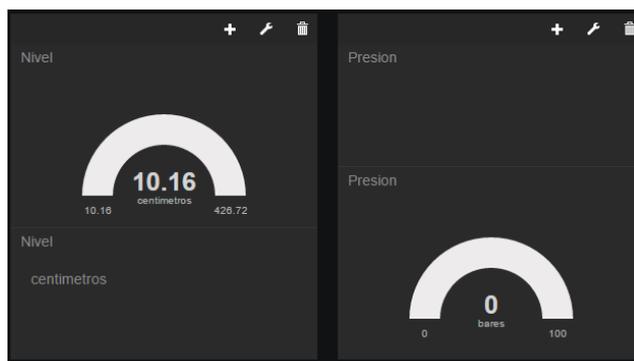


Fig. 4. Ejemplos de paneles en la herramienta web Freeboard.io.

Después de realizar la codificación de los módulos programables, el módulo dedicado al bot de Telegram se ejecutó en modo “background”, de esta manera el módulo encargado de la lectura de cada ADC se puede ejecutar independientemente del módulo dedicado al bot de Telegram. Así se asegura que el sistema se ejecute de manera completa y correcta.

#### 4. Resultados

Las herramientas utilizadas para Internet de las Cosas ofrecen una variedad de formas de trabajar, por lo que es aconsejable utilizar el ambiente que se sugiera para cada una de ellas [22]. Gracias a estas herramientas, los medios de procesamiento y presentación tienen una confiabilidad evidente, los cuales pueden ser consultados en la Tabla 2, donde se demuestra la precisión obtenida en los diferentes medios de presentación.

La ejecución del sistema después de ser analizada se estableció en 3 segundos por iteración del ciclo del algoritmo, primero se realiza el envío de datos y después se realiza una pausa al sistema, la razón de esto es, que el sistema tenga un tiempo para enviar la respuesta. Con 2 segundos se generan retrasos en la ejecución y un entorpecimiento en la entrega de datos desde las aplicaciones web. Por este inconveniente se estableció un tiempo mayor. Sin embargo, con 4 segundos el sistema se torna lento considerando que se espera que la información sea la más reciente. Así fue que probando con iteraciones de 3 segundos se comprobó que es un tiempo ideal con retrasos mínimos y que permite continuar con fluidez el sistema.

La conexión al bot de Telegram tiene como gran ventaja la disponibilidad de información, ya que es una aplicación muy versátil. Pensar en Telegram es considerar que se puede consultar información importante para los interesados del sistema en cualquier

momento y en cualquier lugar. En la Fig. 5 se muestran algunos de los valores solicitados con un teléfono inteligente que tiene instalada la aplicación Telegram; en ella también se puede observar la forma de solicitar cada dato según sea necesario.

**Tabla 2.** Resultados para cada medio de presentación.

V <sub>e</sub>	Valor ideal del sensor			Valor en LCD			Valor en Freeboard			Valor en Telegram		
	P	F	N	P	F	N	P	F	N	P	F	N
0	0	0.5	10.16	0	0.5	10.16	0	0.51	10.16	0	0.51	10.16
0.5	4	1.45	51.81	4.01	1.35	51.8	4.02	1.35	51.81	4.02	1.35	51.81
1	8	2.4	93.47	8.02	2.5	93.57	8.01	2.5	93.57	8.01	2.5	93.57
1.5	12	3.35	135.12	12.01	3.3	135.2	12.02	3.29	135.19	12.02	3.29	135.19
2	16	4.3	176.78	16.03	4.04	176.69	16.02	4.05	176.68	16.02	4.05	176.68
2.5	20	5.25	218.44	40.05	5.24	218.55	40.04	5.25	218.55	40.04	5.25	218.55
3	24	6.2	260.09	24.05	6.04	260.24	24.04	6.05	260.25	24.04	6.05	260.25
3.5	28	7.15	301.75	28.05	7.04	301.54	28.05	7.05	301.55	28.05	7.05	301.55
4	32	8.1	343.4	32.04	8.03	343.5	32.05	8.02	343.51	32.05	8.02	343.51
4.5	36	9.05	385.06	36.07	9.03	385.2	36.06	9.02	385.21	36.06	9.02	385.21
5	40	10	426.72	39.99	9.99	426.61	39.99	9.99	426.62	39.99	9.99	426.62

V: Voltaje de entrada P: Presión medida en bares F: Flujo medido en m/s N: Nivel medido en cm



**Fig. 5.** Solicitud de datos mediante Telegram.

El área de trabajo de los sensores se simula con una fuente de poder variable con la que se realizaron pruebas exhaustivas para el procesamiento y presentación de los datos. La conectividad física se muestra en la Fig. 6.



Fig. 6. Pruebas de procesamiento y presentación de datos respecto a cada elemento.

## 5. Conclusiones

En este trabajo se realiza el procesamiento y la representación de datos referidos a los datos de presión, flujo y nivel de agua. El procesar la información y la presentación de datos en la plataforma de desarrollo Intel Galileo y la codificación con el lenguaje de programación Python, permiten la movilidad y evitan la necesidad de estar frente al dispositivo físico para tomar lecturas del comportamiento de los sensores. La realización de este sistema, permite procesar y representar datos lo cual es el objetivo principal de este trabajo.

Los resultados obtenidos demuestran mejoras de precisión y exactitud al igual que en disponibilidad de los datos comparados con versiones anteriores de este mismo sistema. Uno de los problemas principales a resolver en versiones anteriores del sistema era la latencia con respecto al envío y recepción de datos en las plataformas web. Estos problemas se han resuelto reestructurando el código del programa.

La variación de los valores obtenidos con relación a los valores ideales es mínima y en algunos casos se puede considerar despreciable, esto debido al tipo de dato que se utiliza en la codificación, el cual es de tipo *float*. Cabe recordar que estos tipos de datos son aproximaciones de los valores reales por lo que la variación es insignificante. También se solucionaron ciertos problemas de conexión ya que anteriormente había momentos en los que la conexión fallaba y retrasaba el sistema, sin ser culpa de la conexión a internet precisamente. Esto se debía a ciertas excepciones que no eran controladas

adecuadamente. Al tratar con estas excepciones, las únicas fallas de conexión ahora son provocadas exclusivamente por la falla de la conexión a internet.

Como trabajo futuro se pretende realizar las etapas del sistema de adquisición de datos de presión, flujo y nivel de agua en entornos relevante para validar el prototipo, para lo cual es necesario considerar la tolerancia de error de cada dispositivo y realizar las pruebas para establecer los porcentajes de error del sistema.

## 6. Recomendaciones

Al continuar con el desarrollo del sistema total, es necesario abordar temas de seguridad, control de accesos y cuentas de usuarios. Para la implementación y puesta en marcha del sistema en una RDAP, es necesario considerar personal con conocimientos básicos sobre sistemas operativos Linux para la administración y acceso al mismo.

En caso de un cambio de modelo de sensores, debe planearse el reemplazo que cumpla con la característica de envío a la salida de 4 a 20 mA, ya que de ser así facilita la reconfiguración del código a la nueva escala de los nuevos sensores.

## Referencias

1. Tetreault, V.: El abastecimiento del agua en Zacatecas y Guadalupe ¿Cuál es el problema? en La Jornada Zacatecas. Consultada en <http://ljz.mx/2014/02/26/el-abastecimiento-del-agua-en-zacatecas-y-guadalupe-cual-es-el-problema/> (2014)
2. Informes denominados Actualización de la disponibilidad media anual de agua subterránea, disponibles en la página de Internet de la CONAGUA (<http://www.cna.gob.mx/>)
3. Solís, R. et al.: Beneficios y Desafíos en el uso de una Red Inalámbrica de Sensores para el Monitoreo de una Red de Distribución de Agua Potable, Memorias del CONCYE 2011, 21–23 (2011)
4. Intel. Placa de desarrollo Intel® Galileo Gen 2. [online]. Disponible en: <http://www.intel.es/content/www/es/es/do-it-yourself/galileo-maker-quark-board.html> (2016)
5. Dweet.io: Simple messaging (and alerts) for the Internet of Things. <https://dweet.io/>
6. Freeboard.io: Visualiza el Internet de las Cosas. <https://ricveal.com/blog/freeboard-io/>
7. Telegram: A New Era of Messaging. <https://telegram.org/>
8. Enciclopedia Universal Ilustrada, Europeo-Americana. Tomos XXXII, VII, Hijos de J Espasa editores, Barcelona, España
9. Enciclopedia de la Ciencia y de la Tecnología. Tomo VIII, Editorial Salvat. 1° Edición, Barcelona, España (1964)
10. P. Gerhart, R. Gross, Hochtein J.: Fundamentos de Mecánica de Fluidos. 2° Edición, Addison-Wesley Iberoamericana, USA (1995)
11. Toloza, J., Leiva, L., Riba, A., Carmona, F., Acosta, N.: Desarrollo de un prototipo de sistema portátil para la detección en tiempo-real de la necesidad de riego en cultivos de producción intensiva. En: XVI Congreso Argentino de Ciencias en Computación, 962–971 (2011)
12. Rodríguez, J., Román, J.: Sistema de adquisición de variables hidráulicas en una red de distribución de agua potable utilizando tecnología Zigbee. Universidad Autónoma de Zacatecas (2014)

13. Flow Rate Sensor Interfacing: How to Measure Liquid with an Arduino. <https://diyhack-ing.com/arduino-flow-rate-sensor/>
14. García-Duran, A., Ordaz-García, O., Arce-Olague, J., Hernández-Calviño, M.: Adquisición de datos de sensores de una red de distribución de agua potable. En: Congreso Internacional de Investigación, Academia Journals Córdoba, Veracruz, México, 7(3), 255–259 (2015)
15. Desarrollo de aplicaciones en placas Intel Galileo, Notas del curso impartido por Solís, R., y Ordaz-García, O. Escuela de Verano, Universidad Autónoma de Zacatecas (2016)
16. Galileo Datasheet, Intel 329681-003US (2013)
17. Introducción al Lenguaje Python, Notas del curso impartido por Solís, R. y Ordaz-García, O. Escuela de Verano, Universidad Autónoma de Zacatecas (2016)
18. Series 626 & 628 Pressure Transmitters. Specifications - Installation and Operating Instructions. Bulletin E-111, Dwyer Instruments, Inc. Michigan City, Indiana 46361 USA (2010)
19. Magmeter 2551 Manual. Signet- Senninger LLC, Medidor ciego de flujo electromagnético 2551. 3-2551.090 Rev. L 10/12 (2012)
20. ToughSonic® TSPC Series. Ultrasonic Distance Sensors. Push-button Teachable and/or PC Configurable, Senix Corporation., Rev. 6 (2011)
21. Llegan los bots a Telegram: cómo crear el tuyo propio. <https://www.xatakamovil.com/aplicaciones/llegan-los-bots-a-telegram-como-crear-el-tuyo-propio>
22. Arce, A.: Internet of things. Curso de capacitación de LABSOL (2016)



## **Aplicación del análisis sintáctico automático en la atribución de autoría de mensajes en redes sociales**

Francisco Antonio Castillo Velásquez, Jonny Paul Zavala De Paz,  
Mayra Sánchez Castillo, Adrián Márquez Escandón, Ismael Morales Hernández,  
Jonathan Flores Guzmán

Universidad Politécnica de Querétaro,  
Querétaro, México

francisco.castillo@upq.mx, jonny.zavala@upq.edu.mx,  
{014015299,014014662,014014663,014013727}@upq.edu.mx

**Resumen.** Dentro del Procesamiento del Lenguaje Natural (PLN) se ha considerado que la tarea de análisis sintáctico representa un alto costo computacional, por lo que su aplicación a varias tareas se ha visto limitada. Una de estas es la atribución de autoría de textos (AAT), que se encarga de responder a la cuestión de quién es el autor de un texto, dando algunos ejemplos previos de ese autor (conjunto de entrenamiento). A pesar del costo computacional, los trabajos de clasificación en este campo han dado buenos resultados para textos largos (por ejemplo, libros), pero el estudio de textos cortos ha quedado rezagado. En este trabajo de investigación se propone un modelo computacional basado en n-gramas sintácticos de dependencias para la atribución de autoría de textos (AAT) en Twitter. Como estos mensajes tienen una longitud corta (máximo 140 caracteres) el desempeño del parser sintáctico no es materia de preocupación. La metodología utilizada consistió en la compilación de tweets en español (corpus), su procesamiento en tareas de etiquetación de partes de oración (PosTags) para formar un baseline, la aplicación del análisis sintáctico de dependencias al corpus, la generación de gramas de PosTags y de dependencias sintácticas, la generación de archivos con datos de entrenamiento y clasificación y la aplicación de métodos supervisados de aprendizaje automático a estos archivos. Aunque la mayoría de los resultados no son alentadores, hay un conjunto que permiten ver la factibilidad de la AAT en mensajes de textos cortos.

**Palabras clave:** atribución, autoría, n-gramas, dependencias sintácticas.

### **Application of Automatic Syntactic Analysis in the Authorship Attribution of Social Media Messages**

**Abstract.** It has been considered that the task of parsing represents a high computational cost in Natural Language Processing (PLN), for this reason its application to several tasks has been limited. One of these is the authorship attribution of texts (AAT), which is responsible for answering the question of who is the author of a text, giving some previous examples of that author (training set). In spite of the computational cost, the classification works in this field have given good results for long texts (for example, books), but the study

of short texts has lagged behind. In this research work we propose a computational model based on Syntactic dependency-based N-grams for the attribution of authorship of texts (AAT) on Twitter. As these messages have a short length (maximum 140 characters) the performance of the syntactic parser is not a matter of concern. The methodology used consisted of the compilation of tweets in Spanish (corpus), its processing in tasks of labeling of parts of sentence (PosTags) to form a baseline, the application of the syntactic analysis of dependencies to the corpus, the generation of PosTags N-grams and Syntactic dependency-based N-grams, the generation of files with training and classification data and the application of supervised automatic learning methods to these files. Although most of the results are not encouraging, there is a set that allows to see the feasibility of the AAT in short text messages.

**Keywords:** Attribution, Authorship, N-Grams, Syntactic Dependencies.

## **1. Introducción**

La atribución de autoría hace frente a una pregunta antigua y difícil: cómo asignar un texto de una autoría desconocida o disputada a un miembro de un conjunto de autores candidatos de quienes se tienen ejemplos de textos sin disputa [9]. A pesar de su aplicación a trabajos literarios, la rápida expansión de texto en línea en Internet (blogs, mensajes de correo electrónico, posts en redes sociales, etc.), revelan que las aplicaciones prácticas de la atribución de autoría son asociadas, por lo general, con trabajos forenses [1].

Los enfoques automatizados a este problema involucran el uso de métodos de aprendizaje o estadísticos [13]. Desde el punto de vista del aprendizaje automático, la atribución de autoría puede verse como una tarea de clasificación multi-clase y de etiquetación [11]. Existen dos etapas básicas: primero, los textos pueden representarse apropiadamente como vectores de valores numéricos y, luego, un algoritmo de clasificación puede usar estos vectores para estimar la probabilidad de asociación de un texto a una clase.

Desde entonces se han propuesto cientos de características estilométricas. Estas pueden distinguirse en las siguientes cinco categorías de acuerdo al análisis textual que requieren [13]: características léxicas (frecuencia de palabras funcionales, frecuencia de n-gramas de palabras, medidas de la riqueza del vocabulario, etc.), características de carácter (frecuencia de letras y n-gramas de carácter), características sintácticas (frecuencia de etiquetas POS, mediciones en la estructura de oraciones y frases, frecuencia de reglas de reescritura, etc.), características semánticas (mediciones de sinónimos, mediciones de dependencias semánticas, etc.) y características específicas de la aplicación (tamaño de fuente, color de fuente, frecuencias de palabras específicas, etc.). Hasta ahora, algunos estudios han demostrado que las mediciones más efectivas son las de características léxicas y de carácter.

## **2. Trabajo relacionado**

Existen pocas referencias para trabajos sobre AAT en Twitter. En [19], los autores trabajaron con Ruby para obtener 393 características, como longitud de palabras,

frecuencia de caracteres, palabras funcionales y longitud del texto, para procesarlas en clasificadores implementados en Matlab, con resultados de clasificación correcta no más allá del 40%.

Boutwell [22] realizó pruebas con un clasificador Naïve Bayes para n-gramas de carácter. La autora experimentó con 50 autores y dos conjuntos de entrenamiento (120 y 230). Complementó su trabajo con una serie de pruebas para estudiar el efecto de unir varios tweets en un solo documento.

Otro trabajo relacionado se encuentra en [21], donde los autores trabajaron con un conjunto de características que incluyen n-gramas de caracteres y de palabras, aunque en los experimentos solo se usó un tweet como documento para las pruebas (test). También trabajaron con los conceptos de *k-signature* y patrones flexibles que, al integrarlo con los n-gramas, lograron una mejora en sus resultados (hasta un 70% de clasificación correcta).

En [23] los autores propusieron un conjunto de marcadores estilísticos para la AAT para mensajes de Twitter. Estos marcadores incluyeron emoticones, interjecciones, caracteres de puntuación, abreviaciones y otras características de bajo nivel. Solo trabajaron para tres autores con SVM, alcanzando una clasificación correcta del 63%.

### **3. Metodología usada**

En esta sección describiremos el modelo propuesto de trabajo, desde la compilación del corpus de trabajo, pasando por el proceso de obtención de los n-gramas y generación de estadísticas, hasta la tarea de clasificación.

Los experimentos consistieron en la generación de n-gramas (mediante una herramienta libre de extracción); la compilación de los n-gramas hace referencia al almacenamiento dinámico de los n-gramas únicos (haciendo uso de estructuras matriciales y la implementación de un algoritmo para su manipulación); la refinación, que es la puesta a punto de los n-gramas únicos para que los caracteres no reconocidos por Weka sean detectados y cambiados; la generación de archivos Weka; la aplicación de estadísticas de frecuencias y la aplicación de procesos de clasificación con los modelos de SVM (SMO), Naïve Bayes y J48 para distintos tamaños de perfiles (perfiles). Usamos un baseline tradicional con clasificación *NaïveBayes Multinomial Text* para el texto original como para etiquetas de *PoS*.

#### **3.1 Compilación del corpus**

La parte inicial del trabajo fue la compilación de un corpus de 600 tweets en español, intentando verificar la autenticidad de los mismos y seleccionando aquellos que no contuvieran caracteres de tipo emoticón. Este corpus está disponible para la comunidad investigadora.

A continuación, se muestran algunos ejemplos de tweets que forman parte del corpus. Se ha respetado la redacción original.

*Algunas mentiras son de carne y hueso.*

*Mi instinto de supervivencia me hizo que me alejara de ti.*

*No llores porque se terminó, sonríe porque te vas a ahorrar los regalos de Navidad.*

*Cómo cambian las cosas!!  
 Cuando algo se me mete en la cabeza...imposible parar hasta que lo consigo  
 Un intercambio de mente quiero.*

### 3.2 Generación de n-gramas

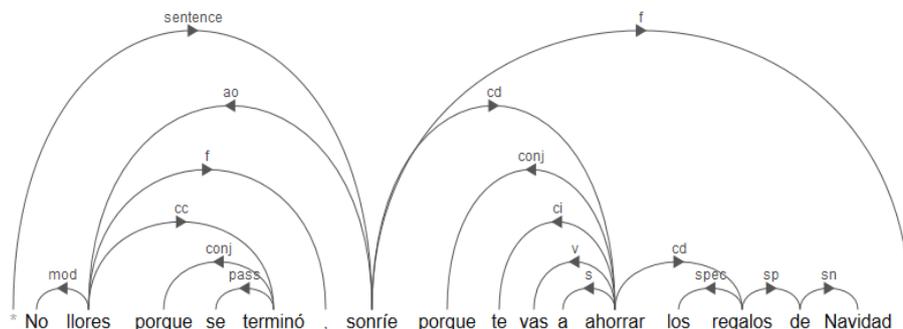
Tomando como punto de referencia la segunda oración del ejemplo 1 (“*Cómo cambian las cosas!!*”) podemos generar los bigramas de caracteres *Có, óm, mo, o\_, \_c, ca, am, mb, bi, ia, an, n\_, \_l, la, as, s\_, \_c, co, os, sa, as, s!* y *!!*. También podemos generar los trigramas *Cóm, ómo, mo\_, o\_c, \_ca, cam, amb, mbi, bia, ian, an\_, n\_l, \_la*, etc. La idea de trabajar con gramas es muy simple y tiene la ventaja adicional que puede aplicarse prácticamente para cualquier lenguaje.

Nuestro modelo hace un análisis estadístico de las apariciones de los gramas en cada una de las oraciones. Se pretende obtener un conjunto de características definitorias del estilo de un autor basado en este fundamento léxico de caracteres. Un análisis de este nivel (superficial) no necesita de un procesamiento profundo de las oraciones, como lo hace un análisis sintáctico (tanto de dependencias como de constituyentes).

El modelo del trabajo se resume en la figura 3, donde el paso inicial (generación del corpus) se dividió en dos conjuntos: los tweets originales y las dependencias generadas por el parser sintáctico FreeLing. Para la misma frase de ejemplo, el parser de dependencias del FreeLing regresa el árbol de dependencias de la Figura 1 y el análisis en formato CONLL (Figura 2).

**Tabla 1.** Rutas de dependencias para el grafo de la Figura 2.

ao mod	cd v
ao cc conj	cd s
ao cc pass	cd cd spec
ao f	cd cd sp sn
cd conj	f
cd ci	



**Fig. 1.** Árbol de dependencias para una oración en español, generado por FreeLing.

Aplicación del análisis sintáctico automático en la atribución de autoría de mensajes en redes sociales

```

1 No no RN RN pos=adverb|type=negative --- 2 mod --
2 llores llorar VMSP250 VMS pos=verb|type=main|mood=subjunctive|tense=present|person=2|num=singular --- 7 ac - llorar.00
3 porque porque CS CS pos=conjunction|type=subordinating --- 5 conj --
4 se se P00GN00 PO pos=pronoun|gen=common|num=invariable --- 5 pass --
5 terminó terminar VMIS350 VMI pos=verb|type=main|mood=indicative|tense=past|person=3|num=singular --- 2 cc - terminar.00
6 , , Fc Fc pos=punctuation|type=comma --- 2 f --
7 sonríe sonreír VMIP350 VMI pos=verb|type=main|mood=indicative|tense=present|person=3|num=singular --- 0 sentence - sonreír.00
8 porque porque CS CS pos=conjunction|type=subordinating --- 12 conj --
9 te te FP2CS00 FP pos=pronoun|type=personal|person=2|gen=common|num=singular --- 12 ci --
10 vas ir VMIF250 VMI pos=verb|type=main|mood=indicative|tense=present|person=2|num=singular --- 12 v - ir.00
11 a a SP SP pos=adposition|type=preposition --- 12 s --
12 ahorrar ahorrar VMN0000 VMN pos=verb|type=main|mood=infinitive --- 7 cd - ahorrar.00
13 los el DA0MPO DA pos=determiner|type=article|gen=male|num=plural --- 14 spec --
14 regalos regalo NCMPO00 NC pos=noun|type=common|gen=male|num=plural --- 12 cd --
15 de de SP SP pos=adposition|type=preposition --- 14 sp --
16 Navidad navidad NP000000 NP pos=noun|type=proper --- 15 sn --
17 . . Fp Fp pos=punctuation|type=period --- 7 f --
    
```

Fig. 2. Análisis CONLL para la misma oración de la figura 2, generado por FreeLing.

La información CONLL es transferida a un módulo programado en Java para obtener todas las rutas de dependencias posibles. Para el ejemplo se obtendrían las rutas mostradas en la Tabla 1.

Estas rutas son procesadas con text2ngram [17] para obtener los gramas de dependencias. Por ejemplo, los bigramas posibles para el conjunto de la Tabla 1 se muestran en la Tabla 2 y para los trigramas en la Tabla 3.

Tabla 2. Bigramas posibles para las rutas de dependencias de la Tabla 1.

ao mod	ao f	cd cd
ao cc	cd conj	cd spec
cc conj	cd ci	cd cd
ao cc	cd v	cd sp
cc pass	cd s	sp sn

Tabla 3. Trigramas posibles para las rutas de dependencias de la Tabla 1.

ao cc conj	cd cd sp
ao cc pass	cd sp sn
cd cd spec	

El único cuatrigrama posible es cd cd sp sn.

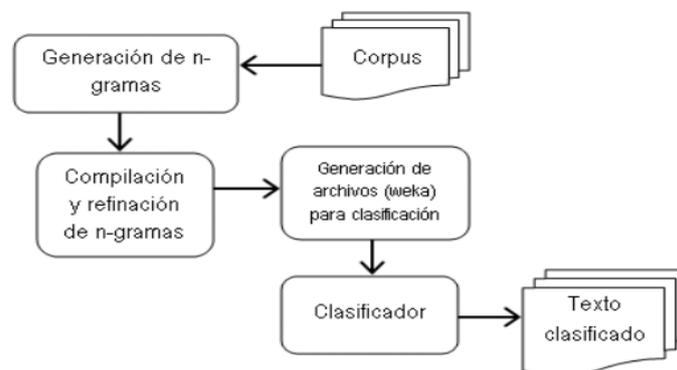


Fig. 3. Modelo de trabajo propuesto.



ejemplo, la primera terna de valores del último registro (5,1,0) significa que los gramas *sn\_sp\_sn*, *sp\_sn\_spec* y *cc\_sn\_spec* aparecen cinco, una y cero veces en el tweet, respectivamente. El último valor, que en nuestro ejemplo no es numérico, representa al autor del tweet.

El proceso de entrenamiento y clasificación también se llevó acabo con el software Weka, que proporciona una diversidad de métodos. En particular, fueron usados los clasificadores NaiveBayes, Optimización Mínima Secuencial (SMO - Support Vector Machines) y árboles de decisión (J48), ya que estos han mostrado buenos resultados en otros trabajos de investigación, como en [12]. La configuración de la clasificación fue con una validación cruzada de 10 iteraciones y un porcentaje de división de 2/3.

#### 4. Resultados experimentales

Los experimentos fueron desarrollados sobre los datos de un corpus para el problema de atribución de autoría. Con el corpus de 600 tweets se hicieron dos pruebas: una con 300 y otra con 600, o su equivalente, para 3 y 6 autores, ya que se recopilaban 100 tweets para cada autor. En la tabla 4 se muestra la información del baseline, con clasificación del algoritmo NaïveBayes Multinomial Text para el texto original de los tweets como para las partes de oración.

**Tabla 4.** Baseline para 100 instancias por autor en texto original y partes de oración, con clasificación por NaïveBayes Multinomial Text.

# Aut	Autores	Texto Original	PoSTags
3	CoffeeJaay egolepsia LuSagania	0.53	0.52
3	Soymarioruiz Mzavalagc Raul_Jimenez9	<b>0.67</b>	0.63
6	CoffeeJaay egolepsia LuSagania Soymarioruiz Mzavalagc Raul_Jimenez9	0.47	0.44

En los resultados que se mostrarán a continuación, usamos el término "*profile size*" para representar los primeros n-gramas/sn-gramas más frecuentes; por ejemplo, un tamaño del profile de 30 significa que se usaron solo los primeros 30 n-gramas más frecuentes. Probamos varios umbrales para el profile y seleccionamos 4 de ellos, como se muestra en todas las tablas de resultados.

Cuando alguna celda de la tabla contiene NA (*not available, no aplica*) significa que nuestros datos fueron insuficientes para obtener el número correspondiente de n-gramas. Sucede solo con los bigramas, ya que en general hay menos bigramas que

trigramas, etc. En estos casos el número total de todos los bigramas es menor que el tamaño del profile.

**Tabla 5.** Resultados de la clasificación para el corpus de 3 autores, 60 instancias.

tamaño del profile	clasificador	tamaño del n-grama		
		2	3	4
15	NB	0.48	0.38	0.30
	SVM	0.53	0.52	0.35
	J48	0.40	0.45	0.28
30	NB	0.64	0.47	0.42
	SVM	<b>0.70</b>	0.50	0.40
	J48	0.53	0.55	0.30
60	NB	<b>0.69</b>	0.48	0.47
	SVM	0.64	0.38	0.42
	J48	0.50	0.38	0.32
90	NB	0.62	0.48	0.42
	SVM	0.60	0.38	0.40
	J48	0.52	0.38	0.33

La tarea de clasificación consiste en seleccionar características para construir el modelo de espacio de vectores, algoritmos supervisados de entrenamiento y clasificación –decidir a qué clase pertenece el texto –en nuestro modelo de espacio de vectores. En este trabajo presentamos resultados para tres clasificadores: SVM (SMO), Naïve Bayes y J48.

**Tabla 6.** Resultados de la clasificación para el corpus de 3 autores, 150 instancias.

tamaño del profile	clasificador	tamaño del n-grama		
		2	3	4
15	NB	0.50	0.45	0.45
	SVM	0.54	0.50	0.41
	J48	0.43	0.53	0.43
30	NB	0.51	0.47	0.45
	SVM	<b>0.60</b>	0.48	0.42
	J48	0.44	0.43	0.45
60	NB	0.53	0.51	0.45
	SVM	<b>0.59</b>	0.47	0.48
	J48	0.45	0.45	0.45
90	NB	0.51	0.49	0.49
	SVM	0.55	0.47	0.47
	J48	0.46	0.49	0.45

En los resultados mostrados en la tabla 5, es interesante notar que los que representan una mayor exactitud son dados por los clasificadores SVM y Naïve Bayes para bigramas, alcanzando un 70% el primero de ellos; de hecho, los mejores resultados están para estos dos clasificadores en bigramas, salvo el caso de NB con un profile de 15.

En la tabla 6 se muestran los resultados de clasificación con 150 instancias, es decir, 50 tweets para cada uno de los 3 autores. Los mejores resultados los arroja el clasificador SVM para un tamaño de profile de 30 y 60 bigramas.

En la tabla 7 se muestran los resultados de clasificación para 3 autores, 100 tweets de cada uno de ellos, dando un total de 600 instancias. De nueva cuenta el clasificador SVM obtiene los mejores resultados para cualquier tamaño de profile.

Con las tablas 5-7, que representan los corpus de 3 autores, en general hay un mejor comportamiento con SVM, aunque las cifras no son tan alentadoras como se esperaban. La clasificación más alta está entre 57%-60%, para bigramas.

**Tabla 7.** Resultados de la clasificación para el corpus de 3 autores, 300 instancias.

tamaño del profile	clasificador	tamaño del n-grama		
		2	3	4
15	NB	0.49	0.45	0.43
	SVM	0.52	0.43	0.38
	J48	0.49	0.40	0.39
30	NB	0.50	0.46	0.42
	SVM	<b>0.57</b>	0.45	0.41
	J48	0.54	0.43	0.38
60	NB	0.54	0.47	0.40
	SVM	<b>0.57</b>	0.45	0.43
	J48	0.51	0.40	0.38
90	NB	0.56	0.49	0.40
	SVM	<b>0.59</b>	0.45	0.44
	J48	0.51	0.41	0.38

En la tabla 8 se muestran los resultados de clasificación para el corpus más grande compilado: 6 autores, 100 tweets para cada uno de ellos, dando un total de 600 instancias. Los resultados bajan drásticamente, no dando posibilidad de augurar mejores resultados con más autores y/o registros.

En este caso, las cifras del baseline (Tabla 4) resultan mejores, ya que alcanzaron un 47%, contra un 37%, que fue el mejor resultado para los gramas de dependencias sintácticas.

Se desarrolló una interfaz para la demostración del método, donde es posible configurar la elección de archivos a compilar, la frecuencia y tamaño de los n-gramas, entre otros parámetros.

Aunadas a estas aportaciones científicas, las de carácter técnico son la creación de una herramienta estadística para visualizar diversos resultados de los procesos

involucrados y el corpus procesado para los experimentos (disponible para la comunidad científica [18]).

**Tabla 8.** Resultados de la clasificación para el corpus de 6 autores, 600 instancias.

tamaño del profile	clasificador	tamaño del n-grama		
		2	3	4
15	NB	0.32	0.28	0.25
	SVM	0.34	0.29	0.23
	J48	0.31	0.29	0.24
30	NB	0.36	0.31	0.27
	SVM	0.36	0.30	0.27
	J48	0.30	0.26	0.24
60	NB	0.36	0.31	0.26
	SVM	<b>0.37</b>	0.31	0.26
	J48	0.31	0.28	0.24
90	NB	<b>0.37</b>	0.31	0.26
	SVM	<b>0.37</b>	0.32	0.28
	J48	0.36	0.28	0.22

## 5. Conclusiones y trabajo futuro

Los resultados de los experimentos demuestran la factibilidad de usar modelos computacionales simples para la tarea de atribución de autoría de textos cortos, como correos electrónicos, tweets, mensajes de chats, etc., aunque solo para un conjunto reducido del corpus y n-gramas pequeños (la clasificación correcta más alta fue de 70% para solo 3 autores). La contribución más importante es la simplicidad y la generalidad del modelo (puede ser aplicados a cualquier lenguaje). Como trabajo futuro se pretende trabajar con la definición de nuevas características que, aunadas a los gramas de rutas de dependencias, mejoren substancialmente los resultados de clasificación; usar datasets estándares, como los que proporciona PAN-CLEF, para poder comparar nuestro método con otros que ya los han usado; y por último, probar otros métodos de clasificación, como Random Forest.

## Referencias

1. Abbasi A., Chen, H.: Sentiment Analysis in Multiple Languages: Feature Selection for Opinion Classification in Web Forums. *ACM Trans. Information Systems*, 26(3), 9–12 (2008)
2. Argamon, S., Levitan, S.: Measuring the usefulness of function words for authorship attribution. In: *Proc. of the Joint Conference of the Association for Computers and the Humanities and the Association for Literary and Linguistic Computing* (2005)

3. Agarwal, A.: Contextual Phrase-Level Polarity Analysis using Lexical Affect Scoring and Syntactic N-Gram. In: Proceedings of the 12th Conference of the European Chapter of the ACL (EACL), 24–32 (2009)
4. Baayen, H.: Outside The Cave of Shadows: Using Syntactic Annotation to Enhance Authorship Attribution. *Literary and Linguistic Computing*, 121–131 (1996)
5. Chaski, C.: Who wrote it? Steps Towards a Science of Authorship Identification. *National Institute of Justice Journal*, 15–21 (1997)
6. De-Marneffe, M.: Generating Typed Dependency Parses from Phrase Structure Parses. In: Proc. of LREC (2006)
7. Gamon, M.: Linguistic correlates of style: Authorship Classification with Deep Linguistic Analysis Features. In: Proceedings of COLING, 611–617 (2004)
8. Hollingsworth, C.: Syntactic Stylometric: Using Sentence Structure for Authorship Attribution. Ms. Thesis, University of Georgia (2012)
9. Juola, P.: Questioned Electronic Documents: Empirical Studies in Authorship Attribution. In: Olivier and Sheno, Eds. *Research Advances in Digital Forensics II Heidelberg*, Springer, 5–8 (2006)
10. López, A.: Atribución de Autoría utilizando distintos tipos de Características a través de una nueva Representación. Tesis de maestría, INAOE, Puebla, México, 11 (2012).
11. Sebastiani, F.: Machine Learning in Automated Text Categorization. *ACM Computing Surveys (CSUR)*, 34 (1), 1–47
12. Sidorov, G., Velasquez F., Stamatatos, E., Gelbukh, A., Chanona-Hernández, L.: Syntactic Dependency-based N-grams as Classification Features. *LNAI 7630*, 1–11 (2012)
13. Stamatatos, E.: A Survey of Modern Authorship Attribution Methods. *Journal of the American Society for Information Science and Technology*, 538–566 (2009)
14. Stefanova, M.: El Potencial Discriminatorio de las Secuencias de Categorías Gramaticales en la Atribución Forense de Autoría de Textos en Español. Tesis doctoral, 41–63 (2009)
15. Sidorov, G., Velasquez F., Stamatatos, E., Gelbukh, A., Chanona-Hernández, L.: Syntactic N-grams as Machine Learning Features for Natural Language Processing. *Expert Systems with Applications*, 41(3), 853–860 (2014)
16. Sidorov, G.: Construcción no lineal de n-gramas en la lingüística computacional: n-gramas sintácticos, filtrados y generalizados. 166 p. (2013)
17. N-Gram Extraction Tool. <https://homepages.inf.ed.ac.uk/lzhang10/ngram.html>
18. Carpeta compartida de recursos del presente artículo. [https://drive.google.com/open?id=0BwJ\\_YuKc8LgKWFgzc1JHYlhIVGs](https://drive.google.com/open?id=0BwJ_YuKc8LgKWFgzc1JHYlhIVGs)
19. Castro, A.: Author Identification on Twitter. In: Hardesty, *Third IEEE International Conference on Data Mining*, 705–708 (2003)
20. Green, R. M., Sheppard, J. W.: Comparing Frequency- and Style-Based Features for Twitter Author Identification. In: C. Boonthum-Denecke & G. M. Youngblood (eds.), *FLAIRS Conference: AAAI Press* (2013)
21. Schwartz, R., Tsur, O., Rappoport, A, Koppel, M.: Authorship attribution of micro-messages. In: *EMNLP 2013 Conference on Empirical Methods in Natural Language Processing, Proceedings of the Conference*, 1880–1891 (2013)
22. Boutwell, S.: Authorship Attribution of Short Messages Using Multimodal Features. Master's thesis, Naval Postgraduate School (2011)
23. Silva, R., Laboreiro, G., TimGrant, L., Oliveira, E., Maia, B.: 'twazn me!!! ;(' automatic authorship analysis of micro-blogging messages. In: Proc. of the 16th inter-national conference on Natural language processing and information systems, *NLDB'11*, 161–168, Berlin, Heidelberg (2011)



## Implementación de algoritmos genéticos para el diseño, optimización y selección de vigas

Valentín Vázquez Castillo<sup>1</sup>, Derlis Hernández Lara<sup>1</sup>,  
Emmanuel Alejandro Merchán Cruz<sup>1</sup>, Ricardo Gustavo Rodríguez Cañizo<sup>1</sup>,  
Edgar Alfredo Portilla Flores<sup>2</sup>

<sup>1</sup>Instituto Politécnico Nacional,  
Escuela Superior de Ingeniería Mecánica y Eléctrica Unidad Azcapotzalco,  
México

<sup>2</sup>Instituto Politécnico Nacional, Centro de Innovación y Desarrollo Tecnológico en Cómputo,  
México

{valentinvc29, derlis392}@hotmail.com,  
{eamerchan, rgridriguez, aportilla}@ipn.mx

**Resumen.** En este trabajo se presenta la utilización de heurísticas para apoyar al diseño de sistemas mecánicos, como una propuesta de metodología de diseño robusta. Se emplearon los algoritmos genéticos en el diseño, optimización y selección de vigas utilizando la metodología de diseño paramétrico, que ayuda a variar las dimensiones (parámetros) de las secciones transversales de las vigas de estudio, con el fin de optimizarlas para soportar cargas específicas sin fracturarse. Los resultados experimentales conllevan a que este tipo de implementaciones son de gran utilidad para resolver problemas de optimización numérica en el diseño y selección de vigas para diversas aplicaciones.

**Palabras clave:** algoritmos genéticos, optimización, diseño paramétrico, selección de vigas.

## Implementation of Genetic Algorithms for the Design, Optimization, and Selection of Beams

**Abstract.** In this paper, the use of heuristic techniques to support the design of mechanical systems is outlined as a proposal for a robust-design methodology. Genetic algorithms were used in the design, optimization, and beams selection for a series of applications using parametric-design methodology, which help vary the dimension or parameters of the studied beams cross sections, so that the area could be optimized applying a specific force without fracture the beam. The experimental results suggest that these types of implementations are of great help when solving numeric-optimization problems in the design of mechanical systems, as well as in the areas of beams enhancement and selection for specific applications, such as the presented case study.

**Keywords:** genetic algorithms, optimization, parametric design, beams selection.

## 1. Introducción

Uno de los principales objetivos al implementar heurísticas en problemas de optimización, es el de resolver situaciones complejas y buscar soluciones factibles dentro de un intervalo definido por las cotas de diseño establecidas. Para este estudio se realiza la optimización y selección de diferentes perfiles de vigas mediante algoritmos genéticos, en donde se obtendrá la viga y el perfil adecuados para las cargas y esfuerzos requeridos en situaciones específicas.

En Biomecánica se emplean técnicas de cómputo inteligente como lo son: algoritmos genéticos, redes neuronales y lógica difusa para la optimización en la síntesis de mecanismos para prótesis robóticas. Una tesis con este enfoque es “Diseño de mecanismos utilizando algoritmos genéticos con aplicación en prótesis para miembro inferior” donde la autora encontró las longitudes de los eslabones y la medida de los ángulos para un mecanismo óptimo con el que se reproduzca el proceso de marcha humana como caso de aplicación [1].

En el uso de aplicaciones tecnológicas como el Sistema de Posicionamiento Global (GPS por sus siglas en inglés “Global Positioning System”), la implementación de algoritmos genéticos permite optimizar el tiempo y la selección de la mejor ruta para llegar a un destino específico [2].

En trabajos como el presentado en [3] se aborda el diseño óptimo de columnas de concreto reforzado haciendo uso de un algoritmo genético, con el cual se minimiza el costo de una viga considerando no sólo los esfuerzos permisibles por el elemento, sino también los costos del concreto, el acero y el recubrimiento utilizados. El modelo de diseño óptimo se basa en el propuesto por Chakrabarty [4], aunque incluye ciertas modificaciones que lo hacen más apropiado para aplicaciones prácticas.

En [5] se han aplicado métodos de optimización con criterios múltiples a problemas de diseño de vigas. Asimismo, en [6] propusieron un modelo para optimizar el diseño de secciones de concreto reforzado en el cual consideraron los costos del acero, el concreto y el recubrimiento.

### 1.1 Algoritmos genéticos

Los algoritmos genéticos (AG) son simulaciones de la selección natural que pueden resolver problemas de optimización, es decir, son considerados como una estrategia de búsqueda que imita la teoría de la evolución de la vida en la naturaleza. Ahora bien, si un algoritmo genético es una técnica de búsqueda iterativa inspirada en los principios de selección natural, es importante aclarar que los AG no buscan modelar la evolución biológica, sino derivar estrategias de optimización. La idea de los AG es optimizar (hallar el máximo o mínimo) una función objetivo utilizando los principios de selección natural sobre los parámetros de la función [7].

El proceso de ejecución del AG se inicia creando una población de  $n$  individuos, donde cada uno contiene dentro de su cromosoma la información del problema a

optimizar, en este caso específico, los parámetros de diseño para cada una de las secciones transversales de una viga, es decir, sus dimensiones.

Luego se establece una función objetivo que evaluará a cada uno de los individuos para determinar su aptitud. Enseguida se realiza la operación de cruce de genes del cromosoma entre dos individuos, operación realizada mediante la ruleta de selección, la cual consiste en seleccionar aleatoriamente dos individuos con base en su aptitud, lo que implica que la probabilidad de selección será mayor conforme tenga mejor aptitud, así como aquellos individuos que tengan aptitud igual a cero serán despreciados en la selección del cromosoma del individuo para compartir genes con otro individuo y crear un descendiente.

Finalmente dependiendo del problema, el diseñador determinará la probabilidad de mutación para el AG, es decir, la modificación de los genes en el cromosoma de un individuo lo cual puede o no mejorar la aptitud del mismo, finalizando de esta forma la primera generación del AG representada en la figura 1.

Sin embargo, es importante que el número de descendientes sea igual al número de individuos de la población creada al inicio del algoritmo, ya que estos descendientes pasarán a ser la población inicial para la generación 2, repitiendo el proceso de la Figura 1 hasta encontrar los valores óptimos que satisfagan el problema a optimizar.

El pseudocódigo del AG implementado se muestra en el Algoritmo 1.

---

#### **Algoritmo 1. Algoritmo Genético**

---

```
1 BEGIN /*Inicio del algoritmo*/
2 Generar una población inicial;
3 Obtener dimensiones de la cadena de cada individuo para cada parámetro;
4 Computar el momento de inercia y deflexión para cada individuo;
5 Computar la función de evaluación de cada individuo;
6 WHILE NOT Terminando DO
7   BEGIN /*Generar nueva población*/
8   FOR Número de individuos/2 DO
9     BEGIN /*Proceso de reproducción*/
10    Seleccionar por ruleta dos individuos con base a su aptitud de la generación anterior;
11    Cruzar los genes de los cromosomas de los dos individuos de forma aleatoria;
12    Mutar los descendientes con cierta probabilidad;
13    Computar el momento de inercia y deflexión con los parámetros de cada individuo;
14    Computar la función de evaluación de los dos descendientes mutados;
15  END
16 IF la población ha convergido a la solución THEN
17   Break;
18 END
19 END
```

---

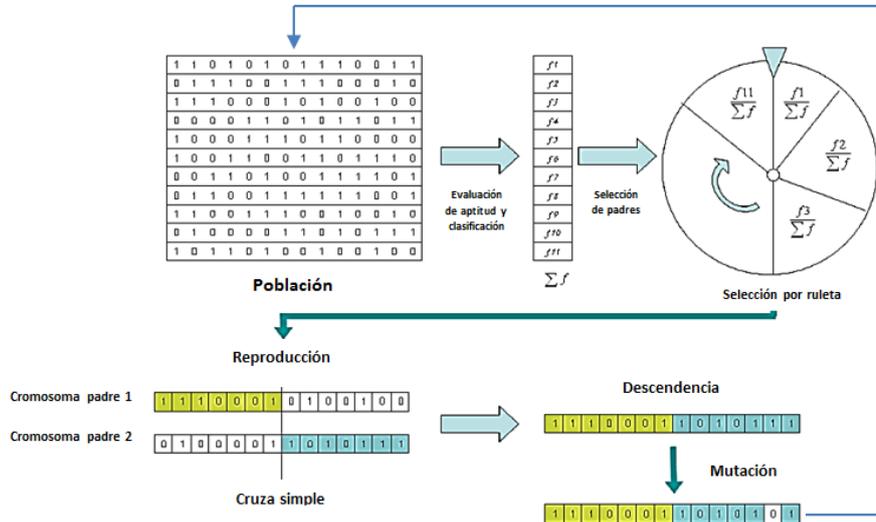


Fig. 1. Descripción general de un AG [8].

## 2. Implementación computacional para la optimización de vigas

La implementación del algoritmo se programó en el entorno de MATLAB® R2015a, y las corridas se llevaron a cabo en una plataforma computacional con las siguientes características: procesador Inter Core i5 @ 1.80 GHz, con 8Gb de memoria RAM y Sistema operativo Windows 8.

Las secciones transversales que se optimizaron son tipo: rectangular (a), T (b), I (c) tubular cilíndrica (d) y tubular cuadrada (e), tal como se muestran en la figura 2.

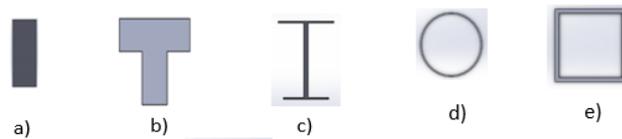


Fig. 2. Secciones transversales a optimizar con el AG.

La figura 3 muestra los parámetros que definen la sección trasversal rectangular, los cuales son implementados como variables dentro del AG.

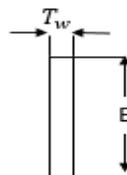


Fig. 3. Parámetros a evaluar en el AG para la sección rectangular.

Se sabe que la deflexión máxima de vigas en voladizo se determina por la ecuación:

$$y_{\text{máx}} = -\frac{PL^3}{3EI'} \quad (1)$$

donde:

- P**: es la carga que se ejerce sobre la viga.
- L**: es la longitud de la viga.
- E**: es el módulo de elasticidad del material.
- I**: es el momento de inercia.

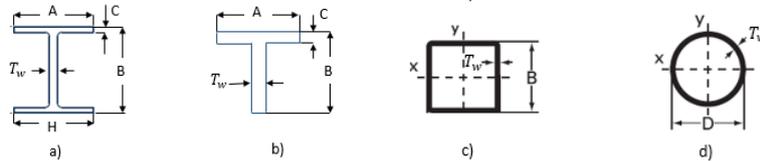
En el caso de la sección transversal rectangular el momento de inercia **I** se calcula por la ecuación:

$$I_{xxSR} = \frac{T_w B^3}{12}. \quad (2)$$

Se implementó una restricción para esta sección transversal dentro del proceso del AG para obtener valores lógicos, la cual está dada por:

$$0 < T_w < B. \quad (3)$$

De la misma forma se parametrizó cada sección trasversal con la finalidad de obtener el momento de inercia, así como el área y poder determinar la función objetivo implementada en el AG. La figura 4 muestra los parámetros que se requieren en la obtención de dichos datos del problema para cada una de las secciones.



**Fig. 4.** Parametrización de las secciones transversales de vigas a optimizar.

Para las secciones transversales tubulares circular (ecuación 4) y cuadrada (ecuación 5) se obtuvieron por fórmula y están expresadas como:

$$I_{xxSTci} = \frac{\pi(D^2 - d^2)}{4}, \quad (4)$$

$$I_{xxSTcu} = \frac{(s^4)}{12} - \frac{(s^4)}{12}, \quad (5)$$

Dado que las secciones transversales tipo **I** y **T** son perfiles compuestos, se utilizó el método del teorema del eje paralelo para el cálculo de los momentos de inercia en cada caso.

Retomando la ecuación 1 para el cálculo de deflexión de una viga en cantiléver, y considerando que se necesita determinar una deflexión máxima conocida, es decir, manejando el dato como una constante, así como también la longitud total de la viga,

el material y la carga ejercida en el extremo de la viga tal como se muestra en la figura 5, se despeja el momento de inercia de la ecuación y se obtiene que:

$$I = -\frac{PL^3}{3E y_{\text{máx}}}. \quad (6)$$

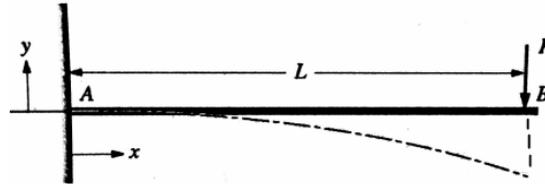


Fig. 5. Aplicación de una carga **P** en el extremo de una viga en cantiléver.

De esta forma las ecuaciones que describen los momentos de inercia, se obtienen sustituyendo los parámetros del perfil en la ecuación 4, respecto a la figura 4d para el perfil tubular cilíndrico, quedando como sigue:

$$I_{xx_{STci}} = \frac{\pi}{64} (D^4 - (D - 2T_w)^4), \quad (7)$$

$$A_{total_{STci}} = \pi \left(\frac{D}{2}\right)^2 - \left(\frac{D - 2T_w}{2}\right)^2 \quad (8)$$

Para el perfil tubular cuadrado, quedan las siguientes ecuaciones:

$$I_{xx_{STcu}} = \frac{(B^4)}{12} - \frac{(B - 2T_w)^4}{12} \quad (9)$$

$$A_{total_{STcu}} = B^2 - (B - 2T_w)^2 \quad (10)$$

Finalmente, se plantea la generación de la población del AG por medio de los parámetros de diseño de cada sección transversal. Para esto, se crea una población inicial de 1000 individuos. En el caso del perfil rectangular, cada individuo contiene en su cadena un código binario creado aleatoriamente por 38 genes o bits, los cuales proporcionan la información de los parámetros  $T_w$  y **B**, donde 19 bits representan el valor para cada parámetro o cromosoma tal como se muestran en la figura 6.

B										Tw																																
0	1	1	1	0	0	1	1	0	0	0	0	0	1	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	1	1	1	1	0	1	0	0	0	1	0	1
Entera					Decimal															Entera					Decimal																	

Fig. 6. Cromosomas de un individuo de la población para el cálculo de la sección transversal rectangular.

Como se observa en la figura 6, los primeros 5 genes del cromosoma representan el valor entero y los siguientes 14 el valor decimal, proporcionando el valor total del parámetro **B**. De tal forma que para tener el valor decimal de **B** se debe hacer una conversión de código binario a decimal para la parte entera y decimal, sumarlos y

finalmente obtener su valor total para el parámetro **B** en forma de dato flotante. Estas operaciones ocurren dentro del AG para cada individuo.

Esto sucede para el resto de los perfiles, sin embargo, existen variaciones en cuanto al número de genes en los cromosomas de los individuos en cada perfil, debido a la complejidad de los parámetros de diseño que forman cada una de las distintas secciones transversales de estudio, establecidas en el Manual de Construcción en Acero, propuestas por el Instituto Mexicano de la Construcción del Acero, A.C. (IMCA) [9].

La Tabla 1 muestra el tamaño de población, así como la longitud del cromosoma que forman el resto de los perfiles estructurales para optimizar el problema de estudio.

**Tabla 1.** Características de la población de los diferentes perfiles de estudio.

Tipo de perfil	Número de individuos	Longitud del Cromosoma
I	15,000	91
T	2,500	72
Tubular circular	1,000	38
Tubular cuadrado	2,500	33

Es importante tener presente que la longitud del cromosoma aumentará conforme los parámetros de diseño incrementen, y en ocasiones se necesitará de una población mayor de individuos para converger a la solución tal como se observa en la tabla 1.

**Tabla 2.** Restricciones para la generación de forma de los distintos perfiles.

Perfil	Restricciones de construcción
I	$B < 2A$ y $B < 2H$ $0.10B < A < 0.80B$ $0.10B < H < 0.80B$ $0.10 < C < 0.35B$ $0.10 < T_w < 0.25B$ $A < 1.75H$ y $H < 1.75A$
T	$C \neq 0$ $T_w \neq 0$ $A < B < 3A$ $B \geq 1$ y $A \geq 1$
Tubular circular	$D > (D - 2T_w)$ $(D - 2T_w) < 0.90D$ $(D - 2T_w) \neq 0$ $D \geq 1$
Tubular cuadrado	$B - 2T_w > 1$ $B > (B - 2T_w)$ $B < 17.5T_w$

Tomando en consideración que la generación 1 creada aleatoriamente puede formar áreas sin lógica alguna, y con fundamento al Manual de Construcción en Acero [9]

mencionado anteriormente, se establecieron ciertas restricciones para cada uno de los perfiles en el AG, las cuales están descritas en la tabla 2.

Finalmente, una vez obtenida la primera generación es posible que un porcentaje pequeño caiga dentro de las restricciones determinadas en la Tabla 2, sin embargo, para que las siguientes generaciones mejoren, se establece una función objetivo, la cual se encargará de evaluar la aptitud de cada individuo y en dado caso que viole alguna de las restricciones, no se tomará en consideración en la operación de cruce del AG.

La Tabla 3 muestra cada una de las funciones objetivo que se determinaron para evaluar la aptitud de los individuos de la población generada, a través del análisis de los parámetros que permiten obtener la menor área posible y logren la deflexión requerida por el problema para que no se fracture la viga, es decir, minimizando el área total a través de las dimensiones de diseño para cada perfil, así como el tamaño del error determinado por la deflexión máxima y la deflexión deseada en el problema.

De esta forma las ecuaciones de 14-18 ilustran cómo influyen los parámetros de área y el error entre la deflexión deseada, y la deflexión que se genera a través de los parámetros de diseño de viga de cada uno de los individuos generados por la población inicial y consecuente en cada generación.

**Tabla 3.** Función objetivo a minimizar para cada perfil estructural de vigas.

Perfil	Función objetivo
Rectangular	$f(x) = \frac{1}{Y_{m\acute{a}x} - Y_{Y_{m\acute{a}x}}} + \frac{1000}{B * T_w}$ (14)
I	$f(x) = \frac{1e7}{Y_{m\acute{a}x} - Y_{Y_{m\acute{a}x}}} + \frac{1e9}{A + H + B + (A_{total})} + \frac{1e13}{abs(I_{xx} - I_{xx})}$ (15)
T	$f(x) = \frac{1e4}{Y_{m\acute{a}x} - Y_{Y_{m\acute{a}x}}} + \frac{1e8}{A_{total}}$ (16)
Tubular circular	$f(x) = \frac{10}{Y_{m\acute{a}x} - Y_{Y_{m\acute{a}x}}} + \frac{1000}{\pi(\frac{D}{2})^2 - (\frac{D - 2T_w}{2})^2}$ (17)
Tubular cuadrada	$f(x) = \frac{10}{Y_{m\acute{a}x} - Y_{Y_{m\acute{a}x}}} + \frac{1000}{B^2 - (B - 2T_w)^2}$ (18)

### 3. Resultados

Para analizar los resultados del algoritmo implementado, se tomaron datos de problemas ejemplo del libro Resistencia de Materiales del autor Robert L Mott [10]. Específicamente problemas de vigas en voladizo, en donde se propone usar acero estándar y soportar una deflexión máxima establecida.

El AG se implementó para tomar como base el análisis de dichos problemas y optimizar cada una de las 5 secciones transversales de estudio en este trabajo. Los datos constantes son longitud **L**, carga **P**, deflexión máxima **y<sub>max</sub>**, y el módulo de elasticidad

de un acero estándar, se obtuvieron los resultados que se muestran en la tabla 4, los cuales cumplen con las características requeridas por el diseño.

**Tabla 4.** Resultados promedio del AG.

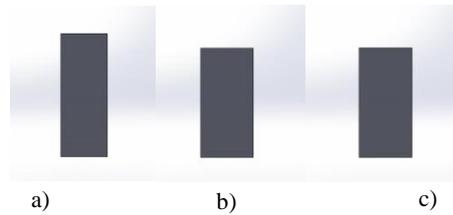
	$B$ (in)	$C$ (in)	$H$ (in)	$T_w$ (in)	$A$ (in)	$A_{total}$ (in <sup>2</sup> )	$I_{xx}$ (in <sup>4</sup> )	$\gamma_{max}$ (in)
	4.097	0.1024	2.8392	0.227	2.3124	1.45754284	4.3214	0.083334
	$T_w$ (in)	$B$ (in)	$A$ (in)	$C$ (in)	$A_{total}$ (in <sup>2</sup> )	$I_{xx}$ (in <sup>4</sup> )	$\gamma_{max}$ (in)	
	1.0404	3.1395	1.0489	1.0906	3.2756059	4.3231	0.083335	
	$B$ (in)		$T_w$ (in)		$A_{total}$ (in <sup>2</sup> )	$I_{xx}$ (in <sup>4</sup> )	$\gamma_{max}$ (in)	
	2.8896		0.417		4.1242968	4.322015	0.0832945	
	$D$ (in)		$T_w$ (in)		$A_{total}$ (in <sup>2</sup> )	$I_{xx}$ (in <sup>4</sup> )	$\gamma_{max}$ (in)	
	4.7436		0.1106		1.609782863	4.321647	0.0833016	
	$B$ (in)		$T_w$ (in)		$A_{total}$ (in <sup>2</sup> )	$I_{xx}$ (in <sup>4</sup> )	$\gamma_{max}$ (in)	
	3.2869		1.4604		4.80018876	4.321655	0.0833014	

**Tabla 5.** Características del AG para converger a la solución.

Tipo de perfil	No. de individuos	No. de generaciones	No. de corridas
I	15,000	17	50
T	2,500	81	50
Tubular Circular	1,000	16	50
Tubular Cuadrada	2,500	16	50
Rectangular	1,000	24	50

Estos resultados son valores promedio de haber realizado múltiples corridas del algoritmo y de una selección de muestras de estudio de 50 corridas, las cuales convergieron a la solución. La tabla 5 muestra un resumen de las características del algoritmo para cada uno de los perfiles.

Las figuras 7, 10, 11, 12 y 13 muestran gráficamente cómo evoluciona la solución de cada generación hasta converger a la solución óptima.



**Fig. 7.** Comportamiento del AG para el perfil rectangular. Generación 1 (a), 6 (b) y 24 (c).

Se puede ver el salto que tienen los individuos en su valor de aptitud en la evolución del algoritmo. Este comportamiento se repite en cada una de las secciones transversales de estudio en este trabajo.

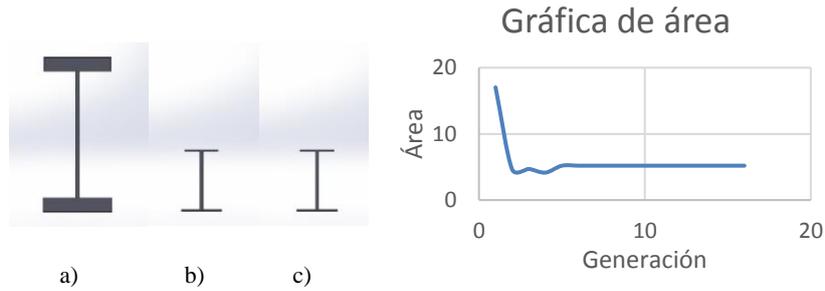


**Fig. 8.** Comportamiento del valor de aptitud durante la evolución del perfil rectangular en específico y en general para todos los perfiles.

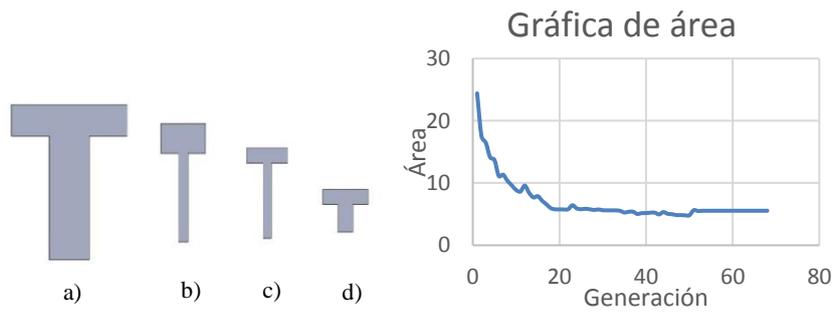


**Fig. 9.** Obtención del área óptima para perfil rectangular.

Uno de los objetivos que cumple el AG implementado es optimizar la sección transversal de las vigas. Esto se obtiene conforme el algoritmo evoluciona, al encontrar el área óptima, lo cual puede observarse en las gráficas de las Figuras 9, 10, 11, 12 y 13 para cada perfil respectivamente.

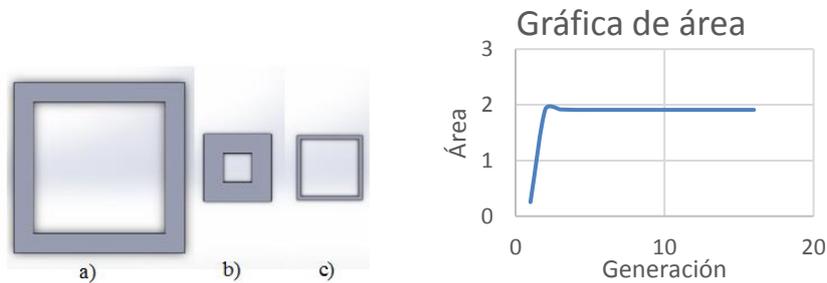


**Fig. 10.** Comportamiento del AG para el perfil I. Generación 1 (a), 6 (b) y 12 (c).



**Fig. 11.** Comportamiento del AG para el perfil T. Generaciones: 1(a), 13 (b), 29 (c) y 56 (d).

En los resultados de la sección transversal tipo I, mostrados geoméricamente en la figura 10, se observa cómo el algoritmo minimiza los parámetros de la sección transversal entre la primera y última generación, cumpliéndose este objetivo en todas las secciones analizadas.



**Fig. 12.** Comportamiento del AG, perfil tubular cuadrado. Generaciones: 1 (a), 7 (b) y 16 (c).

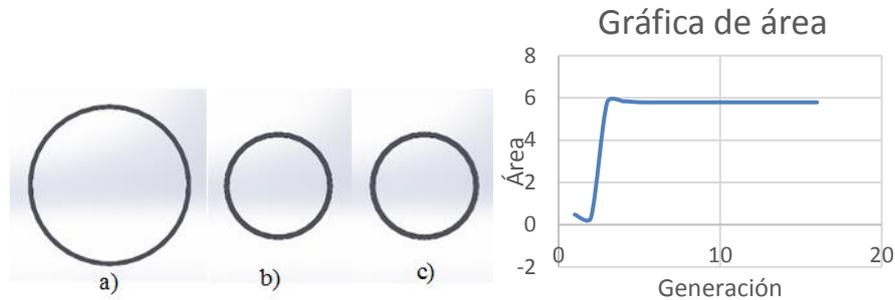


Fig. 13. Comportamiento del algoritmo, perfil tubular cilíndrico. Generaciones: 1, 7 y 16.

### 3.1 Discusión

Los resultados obtenidos por el AG implementado para cada uno de los perfiles, muestran como los individuos de cada generación mejora a la anterior, además, cumple con el objetivo de optimizar la sección transversal para cada perfil, sin embargo, es importante desarrollar cada perfil con una función objetivo diferente dentro del algoritmo genético, dado que cuando se implementó el desarrollo para las secciones transversales tipo **T**, **I** y rectangular en una sola función, los resultados del algoritmo arrojaban las características del perfil rectangular sin tener oportunidad de evaluar como usuario las características de los otros perfiles, esto para fines de diseño es de suma importancia, ya que como diseñador teniendo los valores óptimos de cada perfil, la selección de la sección transversal se elegirá de acuerdo a la aplicación a realizar.

En la tabla 6, se muestran numéricamente los valores de cada parámetro de las soluciones del algoritmo implementado, donde se puede observar las variaciones mínimas que tienen los parámetros de diseño de la sección trasversal para una viga tipo **T** de las analizadas en este trabajo para diferentes corridas del AG.

Tabla 6. Parámetros de la viga **T** dentro de 30 corridas.

N	i	B	Tw	A	C	Área Total	Ixx	y <sub>max</sub>	F.O.
1	68	3.1416	1.0377	3.1351	1.0929	5.5522	4.32168993	0.08330075	1.34E+10
2	63	3.4927	0.6552	2.9842	1.3282	5.3817	4.32173629	0.08329985	6.83E+10
3	54	3.1238	1.0769	3.0813	1.1216	5.6121	4.32175775	0.08329944	1.79E+10
4	86	3.3885	0.7496	2.997	1.2648	5.3825	4.32173934	0.08329979	4.88E+10
5	119	3.1738	1.0068	3.0386	1.1579	5.5480	4.32173774	0.08329983	5.74E+10
6	46	3.1492	1.0235	3.1483	1.0868	5.5324	4.32175008	0.08329959	2.43E+10
7	52	3.5167	0.6283	3.0641	1.2988	5.3731	4.32172575	0.08330006	1.76E+11
8	44	3.2493	0.9047	3.0213	1.191	5.4605	4.3217377	0.08329983	5.76E+10
9	105	3.442	0.6879	3.0979	1.237	5.3489	4.32181104	0.08329841	6.32E+09
10	221	3.753	0.5238	2.5988	0.7155	3.4504	4.32175447	0.0832995	2.02E+10

11	32	3.3553	0.7556	3.2292	1.1185	5.3019	4.32174541	0.08329968	3.10E+10
12	70	3.4561	0.6994	2.8696	1.371	5.3925	4.32169511	0.08330065	1.55E+10
13	32	3.4196	0.6976	3.2002	1.1683	5.3093	4.3217289	0.0833	2.54E+12
14	138	3.5381	0.6283	2.8707	1.4169	5.4002	4.32175733	0.08329945	1.81E+10
15	59	3.5362	0.6197	2.9849	1.3536	5.3929	4.32170043	0.08330054	1.84E+10
16	166	3.3741	0.7851	2.8391	1.3471	5.4159	4.32170348	0.08330049	2.06E+10
17	78	3.4829	0.6685	2.9341	1.3501	5.3871	4.32168552	0.08330083	1.20E+10
18	60	3.136	1.0496	3.1192	1.1012	5.5705	4.32177321	0.08329914	1.17E+10
19	1	3.284	0.8466	3.122	1.1463	5.3885	4.3217148	0.08330027	3.74E+10
20	31	3.3185	0.8015	3.1671	1.1357	5.3463	4.32170985	0.08330036	2.76E+10
21	32	3.2794	0.8401	3.2083	1.0953	5.3489	4.32168293	0.08330088	1.14E+10
22	64	3.1651	1.0092	3.0952	1.1221	5.5349	4.32173765	0.08329983	5.80E+10
23	61	3.1181	1.0791	3.1165	1.0994	5.6046	4.32175945	0.08329941	1.69E+10
24	69	3.1595	1.0206	3.0798	1.1299	5.5512	4.32178357	0.08329894	9.47E+09
25	46	3.1528	1.0181	3.1479	1.0879	5.5268	4.32180765	0.08329848	6.59E+09
26	460	3.3959	0.7684	2.7922	1.384	5.4103	4.32177755	0.08329906	1.06E+10
27	91	3.4161	0.7255	2.9717	1.293	5.3827	4.32173978	0.08329979	4.68E+10
28	78	3.1531	1.0261	3.1023	1.115	5.5503	4.32170436	0.08330047	2.13E+10
29	63	3.387	0.7358	3.1276	1.1911	5.3410	4.32172872	0.0833	1.55E+13
30	39	3.4507	0.7098	2.823	1.3941	5.3953	4.32169265	0.08330069	1.44E+10

#### 4. Conclusiones

Se ha logrado implementar un algoritmo que es capaz de optimizar cinco secciones transversales para el análisis de una viga en cantiléver sometida a una carga en el extremo, de esta manera, el diseñador podrá ahorrar tiempo en el proceso de diseño cuando requiera de un análisis de este tipo de elementos, variando dentro del código del algoritmo la longitud **L** de la viga, así como la carga **P** sometida en el extremo.

#### 5. Trabajo a futuro

Con el fin de continuar con el desarrollo alcanzado por este trabajo, se propone implementar otros perfiles estructurales, con el objetivo de tener un catálogo final de las dimensiones de cada sección transversal y así el diseñador pueda hacer uso de los resultados con base a la aplicación específica.

**Agradecimientos.** Los autores agradecen al Instituto Politécnico Nacional, a la Sección de Estudios de Posgrado e Investigación de la Escuela Superior de Ingeniería Mecánica

y Eléctrica Unidad Azcapotzalco, y al Consejo Nacional de Ciencia y Tecnología por el apoyo brindado.

## **Referencias**

1. González, E. L.: Diseño de Mecanismos utilizando algoritmos genéticos con aplicación en prótesis para miembro inferior. Instituto Politécnico Nacional, México, 1–204 (2010)
2. Abed, M. M., Mohd, G. A., Ibraheem, H. R., Salama, M. A., Sharifuddin, A. M., Ahmed, I. D.: Solving vehicle routing problem by using improved genetic algorithm for optimal solution. *Journal of Computational Science* 21, 255–262 (2017)
3. Coello, C. A., Santos, F. H., Farrera, F. A.: Diseño óptimo de vigas de concreto reforzado mediante algoritmos genéticos. Universidad Autónoma de Chiapas, México, 1–16 (1995)
4. Chakrabarty, B. K.: Model for optimal design of reinforced concrete beams. *Computers and Structures* 42(3), 447–451 (1992)
5. Osyczka, A.: Multicriteria optimization for engineering design. En: John S. Gero, editor, *Design Optimization*, Academic Press, 193–227 (1985)
6. Prakash, A., Agarwala, S. K., Singh, K. K.: Optimum design of reinforced concrete sections. *Computers and Structures* 30(4), 1009–1011 (1988)
7. Dan, S.: *Evolutionary optimization algorithms*. Wiley, EE.UU. (2013)
8. Merchán, E. A.: Soft-computing techniques in the trajectory planning of robot manipulators sharing a common workspace. University of Sheffield, U.K., 1–204 (2005)
9. IMCA.: *Manual en construcción en acero*, Limusa, México (2002)
10. Mott, R. L.: *Resistencia de materiales*. Pearson, México (2009)

## **Sistema para la medición, visualización y registro de la radiación solar**

Ismael Díaz Rangel, María del Consuelo Escoto Ponce de León,  
Enrique José Tinajero Pérez, Luis Francisco Hernández Villalobos,  
Gustavo Angel Pioquinto Pérez

Universidad Autónoma del Estado de México, Ecatepec de Morelos,  
México

ismael1099@hotmail.com, consuelo\_escoto@hotmail.com, ejtinajerop@uaemex.mx,  
luis123456z114@gmail.com, gustavo\_pioquinto@hotmail.com

**Resumen.** La luz solar es esencial para la vida, una exposición moderada a la radiación ultravioleta contribuye a la síntesis de vitamina D, pero exponerse mucho puede ser dañino para la piel, ojos y el sistema inmunitario. Uno de los mayores problemas al exponerse al sol es el cáncer de piel. En México ocurren alrededor de 1,000 casos anuales; los estados con mayor registro de muertes por cáncer son: Ciudad de México, Estado de México, Veracruz y Jalisco [1]. De acuerdo con la doctora Minerva Gómez, Presidenta de la Fundación Mexicana de Dermatología (FMD), este tipo de cáncer ocupa el segundo lugar en frecuencia en nuestro país: el primero en el caso de los hombres, y el segundo respecto a las mujeres [2]. La exposición directa al sol es una amenaza para todas las personas, de manera particular para los menores de edad, por ello es importante contar con medidas preventivas oportunas. La idea de este proyecto es proporcionar a las personas información en tiempo real de la radiación solar a la que se exponen, esto mediante dispositivos que permitan medir el nivel de radiación ultra violeta (UV) y mostrar visualmente la mediación. La propuesta realiza mediciones con múltiples sensores, y mediante un algoritmo ad-hoc mejora el nivel de precisión, el algoritmo también determina el correcto funcionamiento de sensores, haciendo al sistema más confiable; adicionalmente se guarda un registro de las mediciones, lo cual podrá ser útil para análisis a largo plazo. Todo es implementado usando dispositivos de bajo costo.

**Palabras clave:** UVU, cáncer, sensores, medición.

## **System for the Measurement, Visualization and Recording of Solar Radiation**

**Abstract.** Sunlight is essential for life, moderate exposure to ultraviolet radiation contributes to the synthesis of vitamin D, but much exposure can be harmful to skin, eyes and immune system. One of the biggest problems when exposed to the sun is skin cancer. In Mexico, around 1,000 cases occur annually; the states with the highest record of cancer deaths are: Mexico City, Mexico State, Veracruz y

Jalisco [1]. According to Dr. Minerva Gómez, President of the Mexican Foundation of Dermatology (FMD), this type of cancer is the second most common in our country: the first in the case of men, and the second in regard to women [2]. Direct exposure to the sun is a threat to all people, particularly for children, so it is important to have timely preventive actions. The idea of this project is provide to the people real-time information about the solar radiation to which they are exposed, using devices that measure the level of ultraviolet (UV) radiation and visually show the measurement. The proposal makes measurements with multiple sensors, and using an algorithm ad-hoc improves accuracy level, the algorithm also determines the correct functioning of sensors, making the system more reliable. Additionally, a record of the measurements is saved, which may be useful for long-term analysis. Everything is implemented using low-cost devices.

**Keywords:** UV Index, Cancer, Sensors, Measurement.

## **1. Introducción**

Todos estamos expuestos a la radiación UV procedente del sol y de numerosas fuentes artificiales. La región UV abarca el intervalo de longitudes de onda de 100 a 400 nm y se divide en las siguientes tres bandas: UV-A (315–400 nm), UV-B (280–315 nm) y UV-C (100–280 nm). Cuando la luz solar atraviesa la atmósfera, el ozono, el vapor de agua, el oxígeno y el dióxido de carbono absorben toda la radiación UV-C y aproximadamente el 90% de la radiación UV-B. La atmósfera absorbe la radiación UV-A en menor medida. En consecuencia, la radiación UV que alcanza la superficie terrestre se compone en su mayor parte de rayos UV-A, con una pequeña parte de rayos UV-B; no obstante, los rayos UV-B son responsables de las quemaduras de sol y el cáncer de piel [1].

En el ser humano, una exposición prolongada a la radiación UV solar puede producir efectos agudos y crónicos en la salud de la piel, los ojos y el sistema inmunitario. Las quemaduras solares y el bronceado son los efectos agudos más conocidos de la exposición excesiva a la radiación UV; a largo plazo, se produce un envejecimiento prematuro de la piel como consecuencia de la degeneración de las células, del tejido fibroso y de los vasos sanguíneos inducida por la radiación UV. La radiación UV puede producir también reacciones oculares de tipo inflamatorio, como la queratitis actínica [2].

Se considera que la conducta de las personas con respecto al sol es la razón principal del aumento de las tasas de cáncer de piel en las últimas décadas. El incremento de las actividades al aire libre y los nuevos hábitos al tomar el sol ocasionan a menudo una excesiva exposición a la radiación UV. Además de sus ventajas para la salud, los programas educativos eficaces pueden favorecer las economías de los países, reduciendo la carga financiera que suponen para los sistemas de atención de salud los tratamientos del cáncer de piel y de las cataratas. En todo el mundo se gastan miles de millones de dólares en el tratamiento de estas enfermedades, que en muchas ocasiones se podían haber prevenido o retrasado [3].

El IUV es una medida de la intensidad de la radiación UV solar en la superficie terrestre (Figura 1); este índice se expresa como un valor superior a cero, y cuanto más alto, mayor es la probabilidad de lesiones cutáneas y oculares, y menos tardan en producirse esas lesiones.

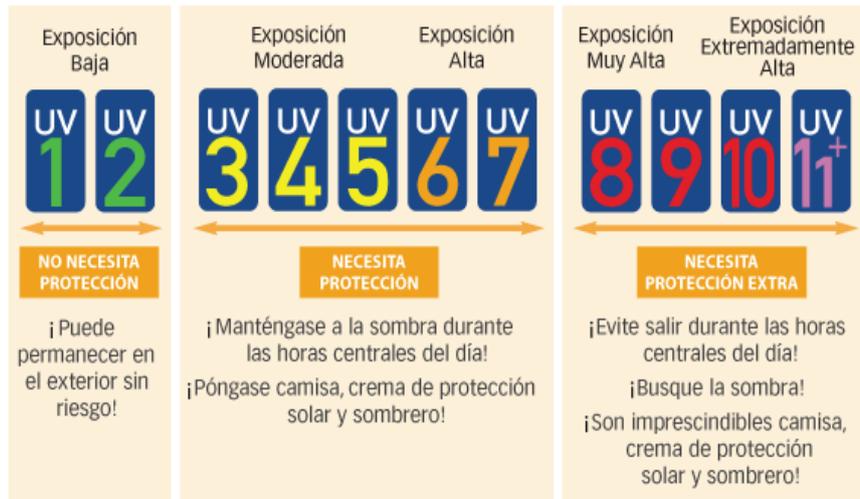


Fig. 1. Escala del índice de radiación solar.

De acuerdo con cifras de la Organización Mundial de la Salud, el cáncer de piel es el más frecuente entre los seres humanos y cada día se diagnostican 2 millones de casos en todo el mundo [4].

En palabras de la doctora Minerva Gómez, Presidenta de la Fundación Mexicana de Dermatología (FMD), este tipo de cáncer ocupa el segundo lugar en frecuencia en nuestro país: el primero en el caso de los hombres, y el segundo respecto a las mujeres [5]. Es importante considerar que es necesario sensibilizar a la población e incitar la toma de medidas de prevención, como es la protección solar particularmente en niños; por lo cual, este proyecto propone crear un sistema simple, fiable y económico de proporcionar información en tiempo real de los niveles de radiación solar a la que nos exponemos, y con eso motivar a tomar medidas de protección, esperando disminuir los casos de cáncer en la piel. Se podría pensar que los niveles de radiación UV nos son tan altos en sitios como en la Ciudad de México, sin embargo hay días en los que se llega a registrar el nivel 11, el cual es el considerado el más alto en la escala UV.

Ya existen trabajos relacionados [6], [7] y [8], esta propuesta se diferencia en que las mediciones realizadas son almacenadas en una memoria MicroSD en intervalos de cinco minutos, los cuales pueden ser recuperados para observarse y analizarse en cualquier software de hojas de cálculo; junto con el nivel de radiación se almacena la hora y fecha exacta de la medición, gracias a que se incorpora un reloj de tiempo real. Además, mediante comunicación inalámbrica por radiofrecuencia, el nivel de radiación es enviado (hasta un máximo de 1km de manera ideal) a un display, dando la posibilidad

de implementar el sistema en escuelas y cualquier espacio público para que la gente conozca, en tiempo real, el nivel de radiación al que se expone al realizar actividades al aire libre, y así tomar medidas preventivas adecuadas.

## 2. Metodología

El diseño de la propuesta está conformado por dos subsistemas: Emisor (fig. 2), realiza lecturas de los cinco sensores de radiación UV que incorpora, lectura de la hora y fecha, almacenamiento de la información en una MicroSD y transmisión inalámbrica; Receptor (figura. 4), recibe la información y la muestra en el display que incorpora. Microcontroladores realizan todas las funciones lógicas necesarias.

### 2.1 Emisor

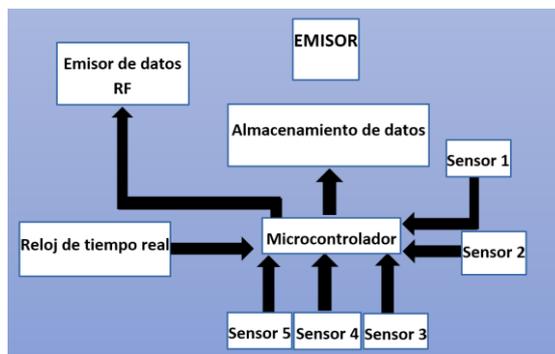


Fig. 2. Diagrama a bloques del emisor.

**Sensores.** - Se utilizaron los sensores ML8511; lo que se indica a continuación es realizado por cada uno de los cinco sensores: el microcontrolador realiza ocho lecturas consecutivas, con un breve intervalo entre cada una, se obtiene el promedio y se guarda en una variable.

**Reloj de tiempo real (RTC).** - Se utilizó el DS13307; este bloque lee del RTC la fecha y hora, la información es almacenada en una variable.

**Almacenamiento de datos.** -La información de fecha, hora y nivel de radiación es almacenada en una memoria MicroSD, utilizando un MicroSD Shield.

**Emisor de datos.** - Mediante el transceptor RF NRF24L01, se transmite la información del índice de radiación UV utilizando el standard de radiofrecuencia a 2.4GHZ.

**Microcontrolador Atmega328.**- Adicionalmente a la generación de señales de control para los diferentes módulos del sistema, se realiza la decodificación las lecturas de los sensores para transfórmalas en el formato de IVU; además, incorpora un algoritmo para determinar y descartar los sensores que están operando de manera fallida; el algoritmo también cuenta con un método que permite obtener una medición

con precisión mejorada, utilizando la información de los sensores que operan correctamente.

El diagrama esquemático del sistema emisor se muestra en la figura 3.

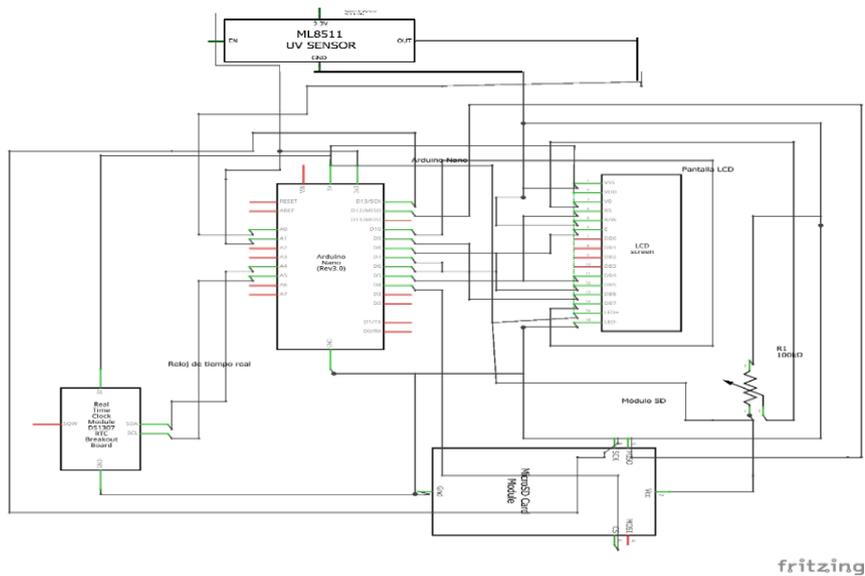


Fig. 3. Diagrama esquemático del emisor.

## 2.2 Receptor

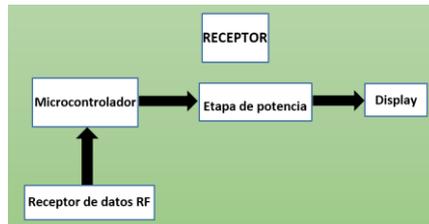


Fig. 4. Diagrama a bloques del receptor.

**Receptor de datos.** - El microcontrolador realiza una lectura al NRF24L01, en caso de tener un dato se guarda en un registro.

**Etapa de potencia.** - El microcontrolador decodifica el dato recibido correspondiente al nivel de radiación, y lo transmite en formato de siete segmentos hacia arreglos de transistores en Darlington, poniéndolos en corte o saturación.

**Display.** - A la etapa de potencia se conectan tiras de LED, las cuales encienden o se apagan de acuerdo el número que representará al nivel de radiación.

En la figura 5 se presenta el diagrama esquemático del Receptor.

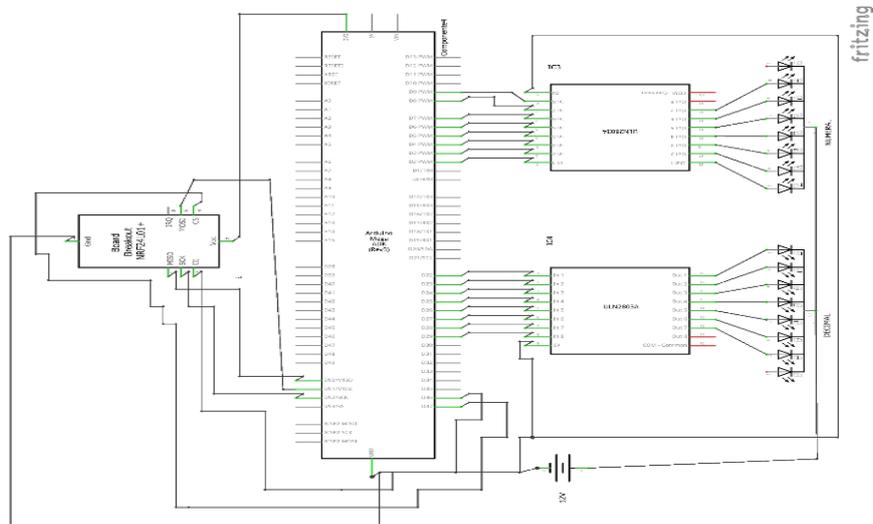


Fig. 5. Diagrama esquemático del receptor.

A continuación, se presenta en la figura 6 el diagrama de flujo del emisor:

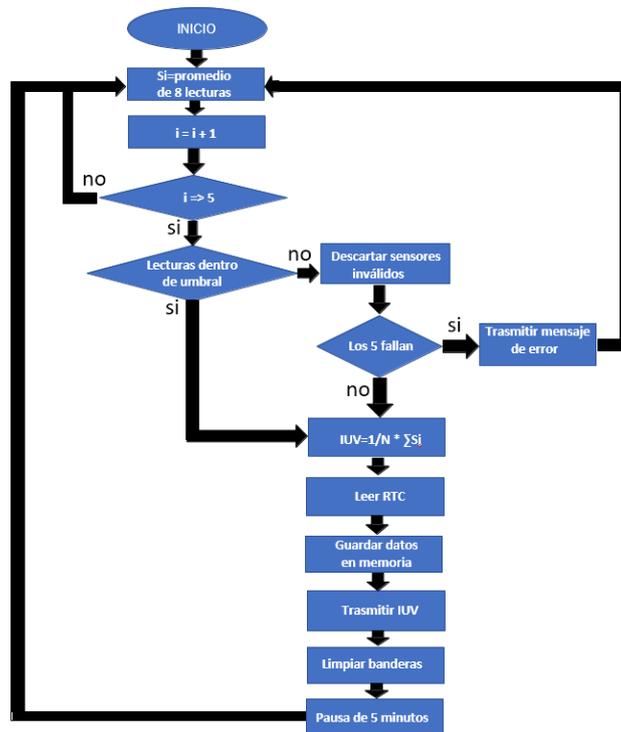


Fig. 6. Diagrama de flujo del emisor.

Se estableció realizar ocho lecturas consecutivas de cada sensor con un intervalo de 100 ms entre cada una; posteriormente se promedian las mediciones, y el resultado se guarda en una variable (una para cada sensor); esto tiene la finalidad de reducir el margen de error propio del sensor, mejorando así la precisión del resultado. Para determinar los valores mínimos y máximos que proporcionan los sensores, se realizó el siguiente procedimiento:

Primero, se determinó el valor mínimo que entrega el sensor (cuando no hay radiación UV en el entorno), el resultado se obtuvo con la ecuación 1:

$$V_{\text{mínimo}} = \frac{\text{LecSen} * 3.3}{\text{LecSenRef}}, \quad (1)$$

donde:

- $V_{\text{mínimo}}$ .- es el valor que entrega el sensor cuando la radiación es cero.
- LecSen.- es el resultado del ADC del microcontrolador al realizar una lectura del sensor.
- LecSenRef.- es el resultado del ADC al realizar una lectura del sensor sobre el voltaje de referencia (3.3 V).

Para conseguir el valor máximo se aplicó una radiación UV de intensidad nivel 15, y se hizo el procedimiento anterior, arrojando los siguientes resultados:

$$V_{\text{mínimo}} = 0.99,$$

$$V_{\text{máximo}} = 2.8.$$

Estos valores indican que cuando la radiación aplicada al sensor es cero, este entrega 0.99 volts, y cuando la radiación es nivel 15, entrega 2.88 volts. En la hoja de datos del sensor, el fabricante indica que trabaja de manera lineal, por lo que la ecuación que modela su funcionamiento ideal se determinó con la ecuación 2:

$$IUV = m * \text{LecSen} + b, \quad (2)$$

donde:

- IUV.- índice de radiación ultravioleta.
- m.- pendiente de relación lineal.
- b.- ordenada al origen.

Para calcular la pendiente (m) se usó la fórmula de la ecuación 3:

$$m = \frac{\text{Índice de radiación máximo}}{V_{\text{máximo}} - V_{\text{mínimo}}}. \quad (3)$$

Sustituyendo los valores obtenidos en los procedimientos anteriormente mencionados, se tiene:

$$m = \frac{15-0}{2.8-0.99} = 8.29.$$

Para la obtención de b, se hace un despeje de la ecuación 2, teniendo:

$$b = IUV - m * LecSen. \quad (4)$$

Al sustituir con valores conocidos se tiene:

$$b = 15 - (8.29 * 2.8) = -8.2.$$

Utilizando la información obtenida con estos experimentos, se pueden descartar sensores que se encuentren en mal estado, y esto será cuando en sus lecturas se observe un voltaje inferior a 0.99 o superior a 2.88, lo cual fue utilizado en el algoritmo para identificar sensores en mal estado.

Una vez identificados los sensores con funcionamiento correcto (N sensores), se procede a obtener el promedio de sus mediciones. Si los 5 sensores están dañados se envía al Display un mensaje de error (letra “e” en código de 7-segmentos); adicionalmente, hay 5 LED, uno para cada sensor, que en caso de mal funcionamiento se envía una señal para que el usuario sepa de dicha situación y proceda a su revisión.

En caso de que cuatro o los cinco sensores operen correctamente, se descartará el de mayor y el de menor resultado, promediando los restantes; esta acción tiene como finalidad mejorar la precisión de las mediciones. El resultado será codificado a formato de 7-segmentos y transmitido al módulo de RF, Para los demás casos se realizarán las mismas acciones pero sin eliminar datos mínimos y máximos.

Adicionalmente, se accede al RTC para obtener los datos de fecha y hora; posteriormente, se almacenan mediante el módulo MicroSD. A continuación, se presenta en la figura 6 el diagrama de flujo del emisor:

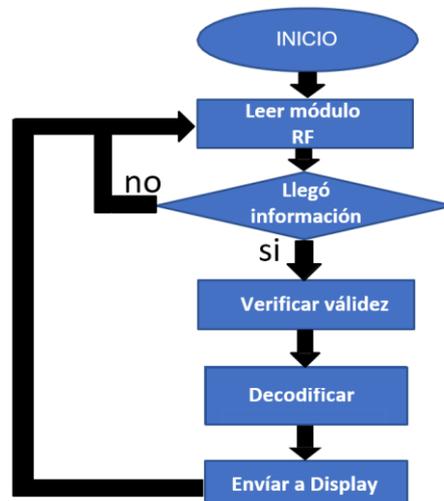


Fig. 7. Diagrama de flujo del receptor

En este subsistema se inicia revisando si llegaron datos provenientes del Emisor, se verifica su validez; de ser positiva, se procede a decodificar la información para su envío al display de 7-segmentos.

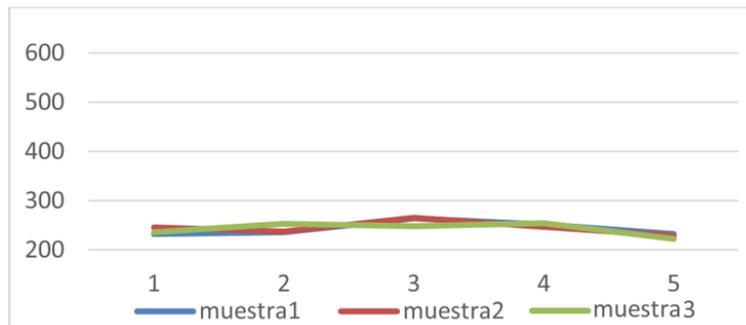
### 3. Resultados

Para saber la precisión de los sensores, se realizaron múltiples mediciones de manera simultánea a los cinco sensores, se utilizaron lámparas UV a una distancia de 5 cm; en la tabla I muestran los resultados de la prueba.

**Tabla 1.** Resultados de lecturas aplicando una radiación UV constante.

Sensor	Muestra 1		Muestra 2		Muestra 3	
	Lectura	IUV	Lectura	IUV	Lectura	IUV
1	233	1.24	236	1.36	237	1.41
2	236	1.36	238	1.45	244	1.69
3	235	1.32	236	1.36	239	1.49
4	242	1.61	247	1.81	245	1.73
5	233	1.24	229	1.08	233	1.24

La figura 8 corresponde a la gráfica de la tabla 1, donde se aprecia claramente el alto nivel de precisión de los sensores.



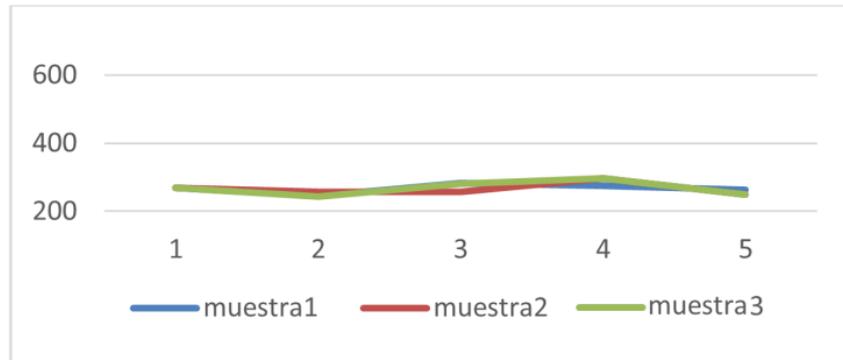
**Fig. 8.** Gráfica con mediciones de los cinco sensores.

Se realizaron más prueba aplicando diferentes niveles de radiación; la tabla 2 corresponde a la aplicación de un menor nivel de radiación.

**Tabla 2.** Resultados de lecturas aplicando una radiación UV constante.

Sensor	Muestra 1		Muestra 2		Muestra 3	
	Lectura	IUV	Lectura	IUV	Lectura	IUV
1	216	0.55	217	0.59	220	0.72
2	212	0.39	216	0.55	215	0.51
3	212	0.39	212	0.39	218	0.64
4	216	0.55	209	0.27	212	0.39
5	210	0.31	218	0.64	216	0.55

La figura 9 es la gráfica de la tabla II, donde se aprecia nuevamente el alto nivel de precisión de los sensores.



**Fig. 9.** Gráfica con mediciones de los cinco sensores.

En las dos pruebas presentadas, se aprecia claramente el alto nivel de precisión con que los sensores ML8511 realizan las estimaciones de radiación UV. Adicionalmente se hizo una prueba comparativa con un sistema de medición de la radiación UV profesional, el cual pertenece al SIMAT (Sistema de Monitoreo Atmosférico de la Ciudad de México); para ello se realizaron las peticiones correspondientes a la Secretaría del Medio Ambiente. Para las pruebas se colocó el sistema de esta propuesta a un costado del equipo UV-BIOMETER (Figura 10).



**Fig. 10.** Radiómetro UV-BIOMETER.

Se tomaron lecturas durante dos días, la figura 11 corresponde a una gráfica comparativa entre ambos sistemas, donde se aprecia claramente el alto nivel de similitud de las mediciones entre ambos sistemas.

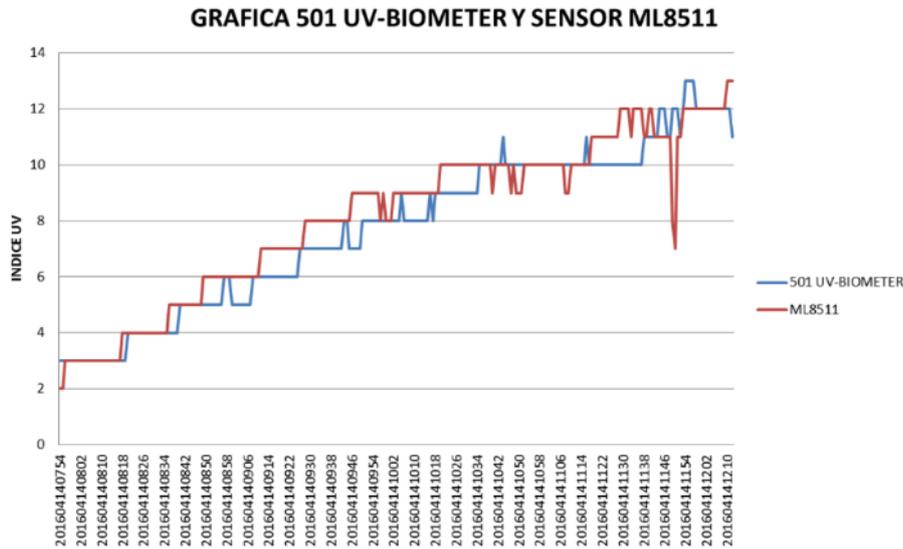


Fig. 11. Gráfica comparativa de mediciones ML8511 vs UV-BIOMETER.

#### 4. Conclusiones

En la investigación realizada para conocer los efectos de las radiación UV sobre las personas se pudo constatar, que si bien es beneficiosa para los organismos, una sobre exposición provoca efectos negativos en la salud; desafortunadamente cada año se reportan más casos de cáncer en la piel, siendo este tipo de cáncer el más común en mujeres y el segundo en hombres, estas cifras México.

En la revisión del estado del arte se encontraron varias propuestas, pero ninguna consiste en un sistema que utilice varios sensores para mejorar la precisión, ni de algoritmos para descartar/indicar problemas de funcionamiento de los mismos. Otra importante diferencia, es que uno de sus subsistemas realiza las mediciones, y apoyándose en un reloj de tiempo real y un módulo MicroSD guarda un histórico de las lecturas cada cinco minutos, y que las mediciones son transmitidas inalámbricamente por radiofrecuencia al segundo subsistema, el cual recibe la información, codifica y la envía a un display; se cuenta con una etapa de potencia que permitió usar tiras LED para tener la capacidad de incorporar un display de gran tamaño, lo cual permite su visualización a distancia, haciéndolo apto para su instalación en espacios “abiertos” como patios de escuelas y parques.

Una de las ventajas más importantes de esta propuesta, es que el nivel de precisión es cercano a radiómetros profesionales, pero costando una pequeña fracción en comparación.

Las pruebas realizadas mostraron de manera satisfactoria el correcto funcionamiento de todos los elementos que conforman a esta propuesta, quedando como trabajo futuro agregar una carcasa que permita su fácil instalación y protección a la intemperie.

## **Referencias**

1. Arulmozhi, B., Mekala, K., Sivasankari, A.: Storage-Less and Converter-Less Energy Harvesting using Internet of Thinking Technology. *International Research Journal of Advanced Engineering*, pp. 36–40 (2016)
2. Bienestar180: Recuperado de <http://www.salud180.com/salud-dia-dia/cancer-de-piel-crece-en-mexico> (2017)
3. Diffey, B.L.: Sources and measurement of ultraviolet radiation. *Methods*, pp. 44–13 (2002)
4. LAPIS Semiconductor: Recuperado de <http://www.lapis-semi.com/en/semicon/sensor/ml8511.html> (2016)
5. OMS Organización Mundial de la Salud: Recuperado de <http://www.who.int/mediacentre/news/notes/2005/np07/es/> (2016)
6. Paredes, C. Ibrahim, P.: Diseño de un sistema de señalización utilizando sensores fotovoltaicos para la prevención de los efectos de la radiación solar para los estudiantes de la EPIE. Universidad Nacional del Altiplano (2017)
7. Pérez, T. M.: Diseño de un radiómetro ultravioleta, para su aplicación en modelos de radiación UV. México D.F., UNAM (2014)
8. Sanz, A. Ozono, L.: Recuperado de <http://www.pionero.cu/sites/default/files/especiales/sitios/ozono/cuidadoconsol-22feb09.htm> (2016)
9. Riveros, T.: Efectos de la radiación ultravioleta en el ser humano, aspectos clínicos. Comisión de fotobiología de Sochiderm (2016)

# WannaCry: Análisis del movimiento de recursos financieros en el blockchain de bitcoin

Víctor Gabriel Reyes Macedo, Moisés Salinas Rosales

Instituto Politécnico Nacional,  
México

vg.reyesmacedo@gmail.com, msalinasr@ipn.mx

**Resumen.** Un ransomware es un tipo de malware mediante el cual un atacante extorsiona al usuario de un equipo de cómputo para que éste le realice un pago que generalmente es operado en el sistema Bitcoin. Este estudio pretende medir el movimiento de los flujos de pago correspondientes a ataques del ransomware *WannaCry*, con el objetivo de proporcionar información sobre los métodos que utilizan los cibercriminales para mover y ocultar el rastro de los recursos financieros resultantes en el blockchain de Bitcoin.

**Keywords:** ransomware, bitcoin, cibercrimen, WannaCry.

## WannaCry: Analysis of Financial Resources Movements for Bitcoin Blockchain

**Abstract.** A ransomware is a type of malware through which an attacker extorts the user of a computer system, so that he can make a payment that is generally operated in the Bitcoin system. This study aims to measure the payment flows corresponding to ransomware *WannaCry*, in order to provide information about methods used by cybercriminals to move and hide the trace of the financial resources on Bitcoin Blockchain.

**Keywords:** Ransomware, Bitcoin, Cybercrime, WannaCry, Networks Analysis.

### 1. Introducción

El 12 de mayo de 2017, miles de computadoras alrededor del mundo fueron infectadas de manera simultánea por un tipo de malware conocido como *ransomware*, cuyo objetivo consiste en cifrar la información resguardada en un equipo de cómputo y exigir un pago a cambio de la herramienta de descifrado

para recuperar los archivos comprometidos. En este caso los atacantes pidieron un pago de aproximadamente \$300 USD por cada equipo infectado.

Durante este ataque se vieron afectadas empresas, instituciones educativas, hospitales y oficinas de gobierno entre otras en alrededor de 100 países, convirtiéndose así en la mayor infección por ransomware de la historia.

No obstante, no es la primera vez que algo así sucede, ya que durante los últimos 5 años este tipo de infecciones se han incrementado en 400% [9].

En general, el pago de estos rescates es operado mediante el sistema Bitcoin[10], que entre otras cosas ofrece a los atacantes la ventaja de ser un sistema *pseudo-anónimo*[4] pues si bien los pagos no están vinculados con personas en particular, si quedan registrados en el *blockchain* del sistema, que funciona como un registro histórico de todas las transacciones que se ejecutan. Esta característica ha permitido llevar a cabo estudios sobre la verdadera capacidad de anonimato en el sistema.

Este documento presenta los avances de una propuesta de análisis de los pagos asociados a este tipo de ciberdelito tomando como caso de estudio el malware conocido como *WannaCry*, del cual se toma como objeto de estudio la forma en que distribuye los recursos captados luego del ataque global y la manera en que, a través de la red de Bitcoin, procede al blanqueo del capital obtenido, utilizando durante el proceso algunas estrategias para maximizar en la medida de lo posible su anonimato.

En la segunda sección de este artículo se abordan de manera general los antecedentes necesarios para comprender la amenaza y el modo de operación del ransomware así como la idea general del funcionamiento del sistema Bitcoin.

En la tercera sección se presenta el contexto del caso de estudio que ocupa a este documento, además discute la metodología utilizada para llevar a cabo el análisis y la recolección de los datos.

Finalmente se presentan los resultados y conclusiones del caso, con el objetivo de aportar información que coadyuve a los esfuerzos internacionales por detener esta modalidad de delito.

## 2. Antecedentes

Si bien las primeras amenazas de ransomware datan de 1989 [11], es a partir del año 2013 que el número de ataques de este tipo se ha incrementado más de un 500% [5]. Dada la tasa de incremento que se ha observado, resulta importante comprender a fondo este fenómeno para lo cual se pone un breve contexto a continuación.

### 2.1. Ransomware: Una amenaza cibernética

El modo frecuente de infección por ransomware es mediante la descarga de archivos de procedencia desconocida. Un escenario común es en empresas en las que los empleados reciben un correo con un archivo adjunto que resulta tener el

malware oculto, o bien instalan complementos que incluyen el malware oculto. Una vez descargado, éste se ejecuta y comienza con la extorsión.

Diversos portales web especializados señalaron al ransomware como la amenaza más importante del año 2016 dado el incremento que este tipo de malware tuvo a nivel mundial generando situaciones que podrían poner en peligro vidas humanas, pues entre sus objetivos existen equipos de cómputo en hospitales que contienen información sensible.

Por su forma de operar se propone una clasificación en tres tipos:

1. **De Bloqueo:** Es un tipo de ransomware que impide el funcionamiento normal de un dispositivo, impidiendo al usuario hacer uso de éste. Es posible encontrar esta variación en el entorno *IoT* y uno de los más casos más comunes ha sido el llamado *virus policía*.
2. **De Cifrado:** Es la variedad más común y tal vez la más conocida. Como ya es sabido, cifra los archivos de una amplia variedad de extensiones, cubriendo prácticamente todos aquellos relacionados con la vida personal de las víctimas o con sus actividades laborales o de negocios. Incluso ha evolucionado para cifrar discos duros.
3. **De Control:** Particularmente peligroso, toma el control de sistemas completos (que pueden ir desde la apertura electrónica de puertas en un edificio hasta plantas de potabilización de agua) hasta recibir el pago solicitado.

Uno de los casos más estudiados es el de *CryptoLocker*, el cual se dió a conocer el 5 de septiembre de 2013 siendo un tipo de malware distribuido a través de correo electrónico spam [7] y que durante su período de actividad logró recaudar cerca de \$1.1 millones de dólares [12], con lo cual se convirtió en el mayor ataque en su tipo hasta ese momento, siendo superado en número de infecciones por *WannaCry* en 2017.

El beneficio financiero que representa el uso de ransomware para los ciberdelincuentes ha propiciado la aparición en la *deep web* de servicios de infección por este medio sobre pedido, de la misma forma que otros servicios ilegales pueden ser adquiridos y los cuales son pagados comúnmente mediante una comisión sobre los pagos generados por la infección.

Una propuesta de identificación de ransomware en la red de Bitcoin se muestra mediante la aplicación de BitIodine [12] en la cual el autor logra agrupar en un clúster todas las direcciones asociadas a *CryptoLocker*.

## 2.2. Sistema de pagos electrónicos *Bitcoin*

Bitcoin es un sistema de pagos electrónico con base en una red *peer to peer* propuesto en 2009 por Satoshi Nakamoto [10].

Su arquitectura consiste en:

- Participantes,
- Transacciones,
- Blockchain.

Los **participantes** son *los usuarios* quienes usan Bitcoin como medio de pago para comerciar y *los mineros* quienes validan las transacciones y emiten monedas.

Las **transacciones** en el sistema se llevan a cabo con la interacción de tres elementos: las *direcciones*, las *llaves* y un *wallet*. Las direcciones son identificadores compuestos por cadenas de entre 27 y 31 caracteres alfanuméricos generados mediante protocolos de criptografía asimétrica, por lo que a cada dirección le corresponde una *llave pública* y una *llave privada*. Su función es similar a la de una cuenta bancaria, en la que se pueden enviar y recibir bitcoins. Cada usuario puede tener tantas direcciones como necesite, y para administrarlas es necesario contar con un *wallet* que, de manera básica, es la colección de llaves que dan acceso a las direcciones del usuario.

Suponga que se desea hacer una transacción entre los usuarios  $A$ , que enviará 1 BTC y  $B$  quien lo recibirá. Entonces el usuario  $B$  debe indicar una dirección para recibirlo, por ejemplo 12BF3DZaoq5sCHLQGDgNqUBrKChM2tXvq9. El usuario  $A$  accederá a su *wallet*, e ingresará el monto a enviar y la dirección de destino, y el *wallet* tomará 1 BTC de alguna de las direcciones a las que tiene acceso para poder completar la transacción.

El **blockchain** es el registro histórico de todas las transacciones que se llevan a cabo. En éste, quedan asentados los montos, las direcciones de origen de los recursos, las direcciones de destino, el momento en el cual se realiza un pago y un ID para cada transacción. Todas las transacciones se organizan por conjuntos llamados *bloques*.

El lector podrá encontrar una exposición detallada del funcionamiento del sistema Bitcoin en [2].

Las características que hacen a este sistema atractivo a los ciber delincuentes son la cotización que ha pasado en un año de un aproximado de \$800 USD a alcanzar los \$4340 USD <sup>1</sup> y la posibilidad de generar transacciones de manera fácil, económica, rápida y global y que no están directamente vinculadas a ninguna persona. Además los pagos son irreversibles, por lo que no hay posibilidad de cancelar una transacción.

No obstante, si una dirección fue asociada de manera pública a una entidad, sí es posible señalarla como tal. Es el caso, por ejemplo, de algunas fundaciones que reciben donaciones en Bitcoin, para lo cual hacen pública la dirección y por ello se sabe a quién pertenece. Algo similar sucede con los ataques de ransomware, los cuales indican a las víctimas a qué dirección deben de realizar sus pagos, con lo que es posible asegurar que tales direcciones pertenecen a la entidad responsable del ataque.

### 3. Presentación del caso: WannaCry

Como se dijo anteriormente, el ataque mediante el ransomware WannaCry ha sido uno de los más agresivos de la historia, por lo que se decidió utilizar

<sup>1</sup> [https://poloniex.com/exchange#btc\\_xrp](https://poloniex.com/exchange#btc_xrp) al 25 de agosto de 2017 18:39 hrs GMT-5

este caso como referencia para plantear un análisis de la forma en que maneja los recursos captados en la red de Bitcoin. Este ataque en particular aprovechó una vulnerabilidad de los sistemas operativos de Microsoft, entre ellos los Microsoft Windows 7, 8.1 y 10, así como Microsoft Windows Vista SP2 y Server 2008/2012/2016 que no contaban con la actualización necesaria para corregir la vulnerabilidad que el malware aprovechó<sup>2</sup>.

Después de captar los recursos, las direcciones permanecieron sin movimientos hasta el día 3 de agosto de 2017.

#### 4. Recolección de datos

Para iniciar el proceso de análisis se consideraron tres direcciones abiertamente asociadas a WannaCry (sobre las cuales se pedía realizar el pago del rescate) obtenidas en [bitcointalk.com](http://bitcointalk.com), a las cuales llamaremos en adelante direcciones semilla<sup>3</sup>, siendo las siguientes:

115p7UMMngo1pMvkhHijcRdfJNXj6LrLn  
12t9YDPgwueZ9NyMgw519p7AA8isjr6SMw  
13AM4VW2dhxYgXeQepoHkHSQuy6NgaEb94

Posteriormente se recogieron datos sobre las transacciones en dichas direcciones, primero mediante el uso de la herramienta **rusty-blockparser** desarrollada por Michel Spagnuolo como una versión mejorada de **BitIodine**[12]. Con dicha herramienta se obtuvieron los datos correspondientes al TxID (ID de la transacción), fecha, hora, monto y destino de la transacción. Mediante un análisis en [blockchain.info](http://blockchain.info) se obtuvieron también las direcciones de origen.

Los datos de las transacciones consideradas corresponden al período comprendido entre el 12 de mayo y el 18 de agosto de 2017, que es el mismo período comprendido entre el inicio del ataque y el momento en que se pierde el rastro de los fondos debido a que éstos entraron en servicios de exchange que entre otras cosas reciben miles de transacciones en una misma dirección y dispersan los fondos en diferentes montos a una gran velocidad.

Cabe mencionar que las transacciones que se usaron para el análisis son aquellas que se ejecutaron después de la dirección semilla. Se considera que las direcciones que alimentaron a las semillas pertenecen a las víctimas del ataque, por lo que no son relevantes para el estudio del flujo de recursos en la red. No obstante fueron útiles para conocer los montos promedio de pagos de rescate así como los horarios y fechas en que dichos pagos fueron hechos.

Para un mejor análisis, los datos obtenidos se concentraron en un archivo .csv en el cual se detallaron aspectos como la fecha y hora del pago, la dirección de origen, la dirección de destino, el ID de la transacción y el monto enviado o recibido para cada dirección.

<sup>2</sup> <https://technet.microsoft.com/en-us/library/security/ms17-010.aspx>

<sup>3</sup> <https://bitcointalk.org/index.php?topic=1916199.0>

## 5. Medición

Como herramientas de medición se utilizó un software libre llamado *Gephi*<sup>4</sup>. Además se hizo un análisis con enfoque estadístico para extraer características como las medidas de centralidad y dispersión de los pagos y posibles correlaciones que puedan existir entre los flujos de bitcoins que realizan los responsables de un ransomware y la cotización de la moneda. A continuación se presentan los resultados de las mediciones con ambos enfoques.

### 5.1. Enfoque en redes

El estudio de las transacciones en la red de Bitcoin implica generar la **matriz de adyacencia** correspondiente, en la cual cada entrada se define con valor 1 si las direcciones se conectan al enviar o recibir recursos, o con un 0 en caso contrario. De manera formal, la matriz de adyacencia  $A(G) = (a_{ij})$  de una red  $G = (X, E)$  se define de la siguiente forma:

$$a_{ij} = \begin{cases} 1 & \text{si } \{i, j\} \in E, \\ 0 & \text{si } \{i, j\} \notin E. \end{cases} \quad (1)$$

En donde  $X$  es el conjunto de direcciones Bitcoin (que tomarán el papel de nodos en la red) y  $E$  es el conjunto de transacciones (que para efecto práctico representan los enlaces dirigidos de la red). A partir de la construcción de esta matriz, es posible medir los siguientes parámetros.

El **grado de un nodo** es una medida de centralidad que describe la estructura de una red en términos de la conectividad individual de los nodos, es decir, el número de enlaces con los que conecta.

Si el resultado es un valor numérico alto, se considera que el nodo está bien conectado con la red. Si el valor numérico es bajo, entonces se dice que el nodo tiene una conexión débil. Este valor indicaría, de manera práctica, las veces que se usa una dirección Bitcoin tanto para recibir pagos como para realizarlos. En este caso el valor promedio obtenido en la muestra de direcciones es de 1.053 lo cual indica que cada dirección se usa en promedio una vez, es decir, recibe un pago y realiza un pago para luego no volver a ser utilizada. Esto coincide con algunas recomendaciones de privacidad [4] que sugieren hacer uso de cada dirección una vez y durante períodos cortos para maximizar el nivel de anonimato.

La **longitud de ruta** es el número de pasos que toma llegar de un punto  $A$  a un punto  $B$  de la red.

Esta cantidad tiene un valor bajo para vértices cuya separación respecto a otros es corta, lo que significa que el flujo a través de ellos es más rápido. En [3] asocian los valores altos de este parámetro en la red de Bitcoin a actividades posiblemente ilegales, toda vez que para ocultar el rastro de información y pasar desapercibidos los ciberdelincuentes hacen pasar los pagos por varias direcciones a través de una ventana de tiempo relativamente pequeña. El valor obtenido en este parámetro durante la medición es de 3.9180, es decir, los fondos obtenidos

<sup>4</sup> <https://gephi.org/>

se trasladaron en promedio a través de aproximadamente 4 direcciones antes de llegar a su destino final. La ruta más larga encontrada fue de 11 pasos.

En ciertos casos, es posible que no existan rutas entre un par de vértices de una red. Incluso, es posible que en una red existan subconjuntos de nodos entre los cuales no existe una ruta. A una red con esta característica se le conoce como *red no conectada*. Los grupos de vértices en una red no conectada son conocidos como **componentes de la red** y se definen como un subconjunto de los vértices tales que existe al menos un enlace entre los miembros del subconjunto y ningún otro nodo pueda ser agregado si se conserva esta propiedad[6]. En términos más sencillos, son grupos de nodos aislados y son una referencia de qué tan fuertemente conectada está la red. En la medición realizada, el resultado obtenido es de 1386 componentes débilmente conectados, lo que indicaría una red sumamente débil en la que básicamente cada nodo es por sí mismo un componente.

**La centralidad del eigenvector** es la extensión de la medida de centralidad de los nodos. Su valor indica el nivel de conectividad de un grupo de nodos respecto al resto de la red. El valor obtenido mediante 100 iteraciones es de 0.1092 y al correr mil iteraciones fue de aproximadamente 0.05, con tendencia a ser menor mientras mayor sea el número de iteraciones. Este resultado indica que en realidad la dispersión que realizan los responsables evita que haya nodos que mantengan mayor influencia en la red, contrario a una suposición inicial en la que se consideró que habría oportunidad de identificar nodos con mayor conectividad para realizar la dispersión.

## 5.2. Enfoque estadístico

De los resultados estadísticos, se obtuvo que el monto promedio de los rescates fue de 0.14751963 BTC con una desviación estándar de 0.143762402 BTC, es decir un aproximado de \$ 266 USD cada pago a un tipo de cambio promedio de \$1,840.86468 USD/BTC con desviación estándar de \$237.15 USD. El rango de los pagos fue desde 0.00000563 BTC hasta 1.999 BTC.

De las direcciones semilla aquí analizadas, los responsables del ataque obtuvieron 51.92690943 BTC que equivale aproximadamente a \$95,590.4135112 USD al tipo de cambio promedio, este dato coincide con lo señalado en diferentes medios informativos en los que se ha hablado de que el monto final no fue tan grande como se habría esperado de un ataque global.

Además, contrario a una suposición inicial, no se observó correlación entre los pagos de rescate a WannaCry y la cotización del bitcoin. Tampoco hay signos de correlación entre la cotización de la moneda y los movimientos de recursos encontrados en la red, por lo que se puede descartar la posibilidad de que los responsables hayan comenzado a mover el dinero de los rescates desde las semillas hasta sus destinos finales orientados por un alza en precios, o visto desde otra perspectiva, es posible descartar que el movimiento de flujos y el mismo ataque tuvieran influencia en el precio de la moneda.

## 6. Conclusiones

Estas direcciones presentaron indicios del proceso conocido como *peeling chain* descrito por Sarah Meiklejohn en 2013 [8], mediante el cual se lleva a cabo la dispersión de fondos a través de pagos por montos pequeños. El propósito de esto es dificultar el rastreo de los recursos. Es frecuente encontrar esta característica en flujos de bitcoins asociados con actividades sospechosas, y una forma de medirlo es mediante la longitud de ruta, que es una medida de la eficiencia en el transporte de la información. Para el caso que nos ocupa, este valor resultó en un valor promedio de 3.98, que debe verificarse contra los valores de muestras adicionales de ransomware.

Se observa además que el grado de los nodos (las direcciones bitcoin) utilizados por WannaCry tiene un valor promedio de 1.084, que indicaría que las direcciones usadas permanecen activas por períodos cortos y son poco reutilizadas. Esta característica es un indicativo de posible actividad sospechosa debido a que la actividad breve es difícil de rastrear[4].

También se concluye que la red formada por las direcciones involucradas en este estudio es una red débilmente conectada, en la que la mayoría de los nodos se mantienen aislados.

El rastreo arrojó además que para el caso de WannaCry los fondos entraron a servicios de *exchange* y *trading* ofrecidos por las operadoras más importantes del mundo, entre las identificadas se encuentran *Polonix*, *HitBTC.com*, *BitStamp*, *Huobi*, *WhaleClub*, *Mercado Bitcoin* y *ShapeShift*. Este resultado es importante debido a que con el apoyo de autoridades y empresas, se pueden dar pasos en la dirección de prevenir este tipo de actividades. La identificación se hizo mediante la plataforma <https://www.blockseer.com/>.

Por último cabe señalar que, según los resultados de esta muestra, el ataque de WannaCry no influyó de manera significativa en la cotización de la moneda de Bitcoin, y que las variaciones de dicha cotización tampoco parecen haber afectado o influido en la dispersión de los fondos obtenidos.

## 7. Trabajo futuro

Para finalizar con esta investigación debe tomarse una muestra aleatoria de direcciones bitcoin que correspondan al mismo período analizado para WannaCry, con la finalidad de comparar comportamientos sospechosos con usuales. También se requiere someter los datos a un análisis estadístico con la finalidad de detectar patrones de comportamientos en cuanto al proceso de *peeling chain* y a los timestamps, con la finalidad de detectar las zonas horarias en la que se realizan los pagos y establecer con ello un marco de referencia.

## Referencias

1. Androulaki, E., Karame, G. O., Roeschlin, M., Scherer, T., Capkun, S.: Evaluating user privacy in bitcoin. International Conference on Financial Cryptography and Data Security, pp. 34–51 (2013)

2. Antonopoulos, A. M.: Mastering Bitcoin: unlocking digital cryptocurrencies. (2014)
3. Baumann, A., Fabian, B., Lischke, M.: Exploring the Bitcoin Network. WEBIST (1), pp. 369–374 (2014)
4. Herrera-Joancomartí, J.: Research and challenges on bitcoin anonymity. In: Data Privacy Management, Autonomous Spontaneous Security, and Security Assurance, pp. 3-16 (2015)
5. Kharraz, A., Robertson, W., Balzarotti, D., Bilge, L., Kirda, E.: Cutting the Gordian knot: A look under the hood of ransomware attacks. In: International Conference on Detection of Intrusions and Malware, and Vulnerability Assessment, pp. 3–24 (2015)
6. Lewis, T. G.: Network science: Theory and applications. (2011)
7. Liao, K., Zhao, Z., Doupe, A., Ahn, G. J.: Behind closed doors: measurement and analysis of CryptoLocker ransoms in Bitcoin. In: Electronic Crime Research (eCrime) APWG Symposium, pp. 1–13 (2016)
8. Meiklejohn, S., Pomarole, M., Jordan, G., Levchenko, K., McCoy, D., Voelker, G. M., Savage, S.: A fistful of bitcoins: characterizing payments among men with no names. In: Proceedings of the 2013 conference on Internet measurement conference, pp. 127–140 (2013)
9. Nieuwenhuizen, D.: A behavioural-based approach to ransomware detection. pp. 1–3 (2017)
10. Nakamoto, S.: Bitcoin: A peer-to-peer electronic cash system. (2008)
11. Smith, J.: Ransomware Incident Response for Law Enforcement. Doctoral dissertation, Utica College (2017).
12. Spagnuolo, M., Maggi, F., Zanero, S.: Bitiodine: Extracting intelligence from the bitcoin network. In: International Conference on Financial Cryptography and Data Security, pp. 457–468 (2014)



## Algoritmo genético para el problema logístico de asignación de la mochila (Knapsack Problem)

Gladys Bonilla Enríquez, Diana Sánchez Partida,  
Santiago Omar Caballer Morales

Universidad Popular Autónoma del Estado de Puebla A.C., Puebla,  
Puebla, México

{gladys.bonilla, diana.sanchez, santiagoomar.caballero}@upaep.mx

**Resumen.** El presente trabajo aborda el caso de estudio de una empresa comercializadora nacional mediante el Problema de la Mochila o Knapsack (KP). Este problema logístico consiste en determinar los productos a almacenar que tengan la mejor relación costo-beneficio para la empresa sin afectar su capacidad de almacenamiento, por lo cual es un problema Binario Multi-criterio del tipo NP-duro. Si bien existen herramientas exactas que pueden resolver algunas instancias de este problema en la práctica pueden tener un costo significativo para la empresa limitando su adquisición. Debido a esto se desarrolló una propuesta de acceso libre de un Algoritmo Genético (AG) para resolver el KP de una manera factible para la empresa. El AG utiliza operadores de cruzamiento aleatorio, cruzamiento uniforme y mutación. El desempeño del AG fue evaluado con un valor óptimo, mostrando un porcentaje de error menor al 4% en comparación de una herramienta comercial exacta.

**Palabras clave:** algoritmo genético, Knapsack binario multicriterio, optimización logística.

### Genetic Algorithm for the Logistic Knapsack Problem

**Abstract.** The present work addresses the case study of a national trading company through the Knapsack Problem (KP). This logistic problem consists on determining the products to be stored that have the best cost-benefit ratio for the company without affecting its storage capacity. This is the reason why the KP a binary multi-criterion problem of NP-hard complexity. While there are exact tools that can solve some instances of this problem, in practice these can have a significant cost to the company which may limit their acquisition. Due to this, a free-access proposal which consists of a Genetic Algorithm (GA) was developed to solve the KP in a feasible way for the company. The GA uses random crossing, uniform crossing and mutation operators. The performance of the GA was evaluated with an optimal value, showing a percentage error smaller than 4% when compared to an exact commercial tool.

**Keywords:** Genetic Algorithms, Binary Multi-criterion Knapsack Problem, Logistic Optimization.

## **1. Introducción**

En el presente trabajo, el Problema de la Mochila o Knapsack (KP) Binario Multi-criterio (Martello & Toth, 1990) es abordado como estrategia de solución para una empresa que presenta un problema de optimización de su inventario para maximizar su ganancia sin afectar sus restricciones de capacidad. El modelo cuenta con una variedad de aplicaciones tales como la selección de proyectos que den la mayor ganancia al inversionista, determinar los productos que puedan caber en un contenedor para optimizar áreas de almacenamiento con espacios reducidos o la transportación de productos con características específicas. En conjunto con otras técnicas, KP puede ser de gran apoyo para resolver distintas instancias en la industria.

El KP está clasificado como un problema combinatorio NP-duro en donde la complejidad computacional que clasifica los problemas como NP-duro o NP-completo indica que la diferencia entre problemas “factibles” e “intratables” se basa en el tiempo que un algoritmo podría tardar en dar una solución a un problema con  $n$  objetos (MacGregor & Ormerod, 1996). Es debido a esto que los problemas de optimización combinatoria sirven como modelos para un gran número de problemas reales y se estudian para construir algoritmos que sean efectivos en términos de complejidad y de la calidad de las soluciones devueltas (Angel, Bampis, & Gourvès, 2014).

Si bien existen herramientas computacionales exactas para resolver problemas combinatorios, estas pierden eficiencia conforme el número de variables o factores incrementa, requiriendo mayor tiempo y recurso computacional. De igual manera, herramientas comerciales pueden tener un costo significativo para su uso por parte de pequeñas y medianas empresas. Debido a esto se han desarrollado diversos métodos meta-heurísticos para tener alternativas más rápidas y alcanzables para este tipo de problemas. Algunos trabajos reportados en la literatura que han abordado el KP con diferentes enfoques son los siguientes:

- En (Mencarelli, D’Ambrosio, Di Zio, & Martello, 2016) se presentó una propuesta de algoritmo meta-heurístico para el KP múltiple no lineal, con funciones de ganancia y peso separables no convexas.
- En (Gao, Lu, Yao, & Li, 2017) se describió el problema KP multidimensional de opción múltiple (MMKP) como un problema importante de optimización combinatoria NP-duro con muchas aplicaciones. Los resultados experimentales mostraron que un algoritmo meta-heurístico desarrollado para la resolución del mismo pudo ser muy competitivo "reduciendo y solucionando" el KP en 18 de 37 casos.
- En (Avci & Topaloglu, 2017) se presentó el KP Múltiple Cuadrático (QMKP, por sus siglas en inglés) como una variante del problema KP clásico en el que se consideran múltiples mochilas y cuyo objetivo es maximizar una función objetivo cuadrática sujeta a restricciones de capacidad.

Entre los meta-heurísticos más utilizados para resolver problemas combinatorios se destacan a los Algoritmos Genéticos (AG) con las siguientes características:

- Son procesos adaptativos para la búsqueda de soluciones en espacios complejos inspirados en la evolución biológica (Herrera, Lozano, & Verdegay, 1995).
- Los AG permiten explorar una cantidad más amplia de posibles soluciones que los programas tradicionales (Holland J. H., 1992).
- Operan de forma simultánea con varias soluciones, en vez de trabajar de forma secuencial como las técnicas tradicionales (Vergara Canizales, 2005).
- Cuando se usan para problemas de optimización (por ejemplo, maximizar una función objetivo) resultan menos afectados por los máximos locales (falsas soluciones) que las técnicas tradicionales (Vergara Canizales, 2005).
- Usan operadores probabilísticos en vez de los típicos operadores determinísticos de otras técnicas (Vergara Canizales, 2005).

Dado que el KP es un problema con la característica NP-duro se decide abordar el problema con un AG. Estos algoritmos pueden dar buenos resultados con una selección de las mejores soluciones (individuos) a las cuales se aplican operadores genéticos para su diversificación. De esta manera se produce una nueva *población* de posibles soluciones, la cual reemplazará a la anterior verificando la propiedad de que la población siguiente posea mejores características en comparación con la población anterior. Así, a lo largo de diversas iteraciones o *generaciones* del AG, se espera que las buenas características se hereden (mantengan) a través de las poblaciones mejorando así la factibilidad de las soluciones (Scenna, Anaut, Passoni, & Meschino, 2013).

De esta manera el presente trabajo describe el desarrollo de un AG para resolver el KP de un caso de estudio. Uno de los objetivos del desarrollo de esta meta-heurística en una plataforma de libre acceso es que pueda proporcionar resultados cercanos a los obtenidos con sistemas restringidos por costos. El desarrollo del AG para la solución de la instancia considerada presenta más de una restricción y en su evaluación tuvo un desempeño significativo favorable al compararse con software especializado. Por lo tanto este desarrollo puede ser utilizado por pequeñas y medianas empresas con problemáticas que puedan modelarse mediante el KP y que tengan limitantes en cuanto a recursos o software especializado para resolverlo.

## **2. Marco teórico**

El KP es un problema de tipo no-determinístico (NP) ya que existe una combinación exponencial de instancias que en su totalidad no pueden ser resueltas (NP-duro). Estos temas los aborda la teoría de la complejidad computacional que clasifica los problemas como problemas NP-duro y NP-completo, en donde la diferencia entre problemas “factibles” e “intratables” se basa en el tiempo que un algoritmo podría tardar en dar una solución al mismo con  $n$  objetos (MacGregor & Ormerod, 1996). El KP forma parte de una lista histórica de problemas NP-duros descrita en (Karp, 1972).

Dentro de los trabajos base que abordan el KP se encuentran aquellos reportados en (Martello & Toth, 1990), (Karp, 1972) y (Kellerer, Pferschy, & Pisinger, 2004). En (Iwama & Taketomi, 2002) se presenta el caso del KP en donde cada elemento a

incorporar en la mochila tiene pesos y valores iguales. Otra variante del KP se presenta en (Disser, Klimm, Megow, & Stille, 2014) en donde la capacidad de la mochila es desconocida. En (Klamroth & Wiecek, 2000) se discuten diferentes modelos de KP como aquellos con múltiples criterios binarios con más de una restricción, con múltiples periodos y con dependencia del tiempo. El diseño de un algoritmo híbrido se presenta en (Carraway, Schmidt, & Weatherford, 1993) para el KP lineal estocástico en donde los costos se conocen con certeza, pero los rendimientos son variables aleatorias independientes normalmente distribuidas. En este caso, el objetivo es maximizar la probabilidad de que el rendimiento total sea igual o superior a un valor objetivo especificado. El trabajo reportado en (Angel, Bampis, & Gourvès, 2014) presenta de manera general los problemas multi-criterio en optimización combinatoria, identificando cuatro enfoques generales de la aproximación polinomial con garantías de rendimiento (enfoque de ponderación de criterios, enfoque simultáneo, enfoque presupuestario, y enfoque de la curva de Pareto). Finalmente, en (Kalai & Vanderpooten, 2011) se presenta una versión robusta del KP utilizando un criterio de rechazo máximo-mínimo o mínimo-máximo. En este trabajo, se discuten los inconvenientes de dichos criterios y se propone un nuevo enfoque de robustez, denominado robustez- $\alpha$  lexicográfica. Se muestra que la complejidad del problema lexicográfico  $\alpha$ -robusto no aumenta en comparación con la versión máximo-mínimo.

Dado que se conoce la complejidad matemática y computacional del KP, para el presente trabajo se considera la herramienta de Algoritmos Genéticos (AG). Esto dado que proporcionan una alternativa a las técnicas tradicionales de optimización mediante el uso de búsquedas dirigidas aleatoriamente para localizar soluciones óptimas en escenarios complejos (Srinivas & Patnaik, 1994). Los AG tienen su origen a principios de los 60's por Holland quien desarrolló los fundamentos de los AG basándose en los conceptos evolutivos y reproductivos de la teoría de Darwin. Esto en un intento por imitar de manera artificial el proceso natural de la evolución de los seres vivos (Holland J. H., 1975). Bajo este contexto, los AG permiten evolucionar a una población de individuos al someterlas a combinaciones aleatorias que se asemejan a las presentes en la evolución biológica, decidiendo cuál de estas combinaciones es más adaptable al medio que se plantea (en este caso para obtener una ganancia máxima) (Srinivas & Patnaik, 1994). La evolución de una población se obtiene mediante operaciones de reproducción conocidos como mutación y cruzamiento (Pedemonte, 2003). En las siguientes secciones se presentan los detalles del modelo KP y el AG desarrollado para su resolución.

### 3. Descripción del modelo KP

El modelo del KP parte de la suposición de que un escalador tiene que llenar su mochila. Para ello debe seleccionar una cantidad de varios posibles objetos, los cuales le pueden brindar el mayor beneficio sin exceder la capacidad de la mochila. El KP puede ser matemáticamente formulado con un vector de variables binarias  $x_j$  en donde:

$$x_j = \begin{cases} 1 & \text{Si el elemento } j \text{ es seleccionado,} \\ 0 & \text{en otro caso.} \end{cases} \quad (1)$$

Entonces, si  $p_j$  es una medida de importancia (en este caso, ganancia) para un elemento  $j$ ,  $w_j$  representa el tamaño de dicho elemento, y  $cv$  es el tamaño de la mochila, el problema refiere a la selección de la cantidad de todos elementos cuyos vectores binarios  $x_j$  satisfagan las siguientes restricciones:

$$\sum_{j=1}^n w_j x_j \leq cv, \quad (2)$$

$$x_j \in \{0,1\}, j = 1, \dots, n, \quad (3)$$

que deben contribuir a maximizar la siguiente función objetivo:

$$\sum_{j=1}^n p_j x_j. \quad (4)$$

Para este caso se extendió el modelo con una restricción adicional de tamaño:  $cv$  se considera la capacidad volumétrica de la mochila y  $cz$  se considera la capacidad en peso de la mochila. De esta manera se tienen dos variables de tamaño para cada elemento:  $w_j$  el tamaño en volumen ( $\text{cm}^3$ ) de cada elemento, y  $z_j$  el tamaño en peso (kg) del elemento, lo cual da la siguiente restricción:

$$\sum_{j=1}^n z_j x_j \leq cz. \quad (5)$$

## 4. Desarrollo de la implementación

El modelo KP se consideró para resolver un problema de almacenamiento para una empresa en México. Esta empresa tiene 250 productos y las siguientes restricciones de volumen y peso: capacidad = 169,750  $\text{m}^3$ , peso = 545 kilogramos. La determinación de estos 250 productos se realizó mediante una clasificación del tipo ABC en base a su costo y frecuencia de pedido. En la Tabla 1 se muestran los datos de cada producto respecto a su ganancia (en pesos mexicanos), volumen (en centímetros cúbicos) y peso (en kilogramos).

Es importante señalar que el modelo KP normalmente es encontrado con una restricción. Para este trabajo son dos las restricciones que se consideraron de acuerdo al requerimiento del problema real.

### 4.1. Programación del AG

La Figura 1 muestra la estructura del AG desarrollado para resolver el KP del caso de estudio. La programación del AG fue realizada mediante el software Octave. Como se muestra, el AG opera sobre una población con  $P$  soluciones que se generaron aleatoriamente. Estas soluciones se llaman individuos y para cada

individuo  $i \in P$  se define una función que evalúa su adaptabilidad al entorno, es decir, que determine qué tan factible u óptimo es para resolver el problema (Olivera, 2004).

**Tabla 1.** Datos del problema. Fuente propia.

$j$	$p_j$	$w_j$	$z_j$																
1	6985	400	3	51	6419	853	4	101	4746	581	3	151	7645	495	5	201	5947	802	5
2	5716	1272	4	52	4956	655	3	102	4245	556	5	152	5802	343	4	202	4051	361	3
3	5907	128	4	53	8256	1095	3	103	5275	1010	5	153	4171	1344	4	203	3662	1451	5
4	5177	751	4	54	5871	646	3	104	5843	482	5	154	5669	850	3	204	8479	955	5
5	8195	794	5	55	4724	547	4	105	4485	829	4	155	5544	878	3	205	8296	459	5
6	3818	1040	5	56	5123	608	3	106	4630	228	3	156	6195	137	5	206	9257	828	3
7	8681	405	3	57	5945	795	3	107	8093	1467	4	157	9357	1293	5	207	7623	603	3
8	5368	1414	4	58	4517	1427	4	108	6078	942	4	158	5747	434	4	208	3693	1283	5
9	7653	655	5	59	9383	433	4	109	3946	1298	3	159	9037	929	3	209	4039	465	5
10	5482	367	3	60	4109	1158	4	110	4592	568	4	160	5882	1110	5	210	6446	999	5
11	8004	381	5	61	6419	905	5	111	3887	433	5	161	6741	1514	5	211	8330	918	3
12	9112	924	3	62	5514	592	5	112	4598	983	5	162	8558	659	4	212	4086	1281	3
13	5139	233	3	63	4703	1380	4	113	7030	927	5	163	4990	301	4	213	4859	309	5
14	5725	1511	5	64	3860	1315	4	114	6486	821	5	164	6466	1295	3	214	3609	994	5
15	7826	1048	5	65	5849	1184	4	115	4781	1352	3	165	7762	394	3	215	8633	380	5
16	7816	1064	4	66	5472	182	4	116	3510	1485	4	166	9278	1419	5	216	9821	1043	4
17	4077	1078	5	67	3591	1426	3	117	3676	1081	5	167	6031	1197	4	217	9500	473	3
18	3757	220	4	68	8085	1240	3	118	5259	867	4	168	7314	1019	4	218	5692	377	5
19	5954	405	5	69	4069	1340	4	119	8452	326	5	169	8186	385	4	219	6582	471	3
20	3196	819	5	70	9395	1070	5	120	6836	479	4	170	7042	1277	5	220	7506	1231	3
21	3325	772	4	71	6675	1158	5	121	6952	1311	4	171	5084	918	3	221	8216	1485	5
22	4586	824	5	72	4398	1515	5	122	3152	866	4	172	9443	870	5	222	9808	328	5
23	4199	1216	4	73	9150	865	3	123	7021	543	3	173	3853	1270	4	223	6115	1497	4
24	5574	512	5	74	6864	606	5	124	7889	434	5	174	6946	1238	4	224	6003	1143	5
25	8184	122	4	75	8550	847	4	125	4897	1358	5	175	9542	860	5	225	9725	352	5
26	4157	465	3	76	9683	483	3	126	7558	1103	5	176	6595	1408	5	226	8048	367	3
27	6638	569	4	77	9768	1500	4	127	3448	384	4	177	7636	187	3	227	7906	1231	3
28	6873	680	4	78	9186	360	3	128	3227	329	4	178	3411	720	5	228	4915	795	3
29	5122	1491	5	79	8843	1287	4	129	6201	1535	3	179	9834	361	5	229	6784	1148	4
30	4681	965	5	80	4065	493	3	130	6030	451	4	180	6201	398	5	230	7021	349	3
31	5013	340	5	81	7292	1263	5	131	7667	603	4	181	3539	1025	4	231	4900	929	3
32	4586	892	5	82	3589	801	5	132	8840	1094	3	182	4753	359	4	232	6116	1420	5
33	3749	1081	5	83	9630	1277	4	133	8968	1109	4	183	5948	380	4	233	6999	386	3
34	6377	451	5	84	8381	486	3	134	7692	620	5	184	6402	631	5	234	4773	949	4
35	5081	1152	3	85	5997	917	4	135	6285	707	5	185	7857	1309	4	235	6674	1049	3
36	7413	286	4	86	4950	255	4	136	7344	856	5	186	8444	1447	4	236	6647	333	3
37	8087	451	5	87	5635	1114	4	137	9099	981	4	187	6921	642	5	237	4578	1159	5
38	4866	599	3	88	8505	516	3	138	3508	220	5	188	8901	1516	5	238	8897	839	5
39	7641	1105	5	89	6593	1409	4	139	3145	396	3	189	6428	712	5	239	5081	1121	4
40	8131	1077	5	90	8478	327	5	140	3330	716	5	190	9870	974	5	240	9644	650	4
41	9928	925	3	91	7032	1185	3	141	8565	1085	3	191	7180	405	4	241	3941	704	3
42	7961	745	3	92	3077	634	4	142	8017	1414	3	192	7132	1368	5	242	5099	607	3
43	9477	751	4	93	3134	452	4	143	5467	699	5	193	5557	880	5	243	8854	193	5
44	3655	214	4	94	8525	345	3	144	9030	1256	3	194	3106	572	3	244	7636	1368	4
45	3780	860	3	95	6369	152	4	145	3435	305	4	195	6303	1284	5	245	3431	683	4
46	4803	1083	4	96	7574	1313	5	146	9267	861	4	196	8695	735	3	246	6859	1092	5
47	3798	310	5	97	3346	1385	5	147	9195	437	5	197	6163	607	5	247	8933	1190	5
48	8216	1244	3	98	7667	253	3	148	7626	214	3	198	9333	380	5	248	7412	1511	3
49	3178	781	3	99	7013	1503	3	149	5160	1174	5	199	7857	813	3	249	8730	839	4
50	9705	388	4	100	7367	580	3	150	7619	1420	3	200	3922	1395	5	250	7354	754	5

En este caso, cada individuo representa un vector de asignación de elementos en la mochila  $x_j$  (vector de 1's y 0's) y la factibilidad del mismo se evalúa en base al valor de la función objetivo (Ecuación 4) con los respectivos valores de las variables  $x_j$ . Si la asignación cumple con las restricciones de capacidad (Ecuación 2 y Ecuación 5) la solución recibe el valor de la función objetivo correspondiente. Si no se cumplen dichas restricciones entonces la factibilidad recibe un valor de 0 dado que es un problema de maximización.

El AG se ejecuta durante 100 generaciones o iteraciones. En cada iteración se aplican los operadores evolutivos sobre un conjunto de  $i$  individuos denominados

*padres* para recombinarlos y modificarlos con el propósito de generar nuevos individuos denominados *hijos* y constituir una nueva población. Los individuos *padres* se seleccionan mediante el operador probabilístico conocido como *Ruleta* el cual da privilegios a los individuos con mejor factibilidad en la población para su reproducción.

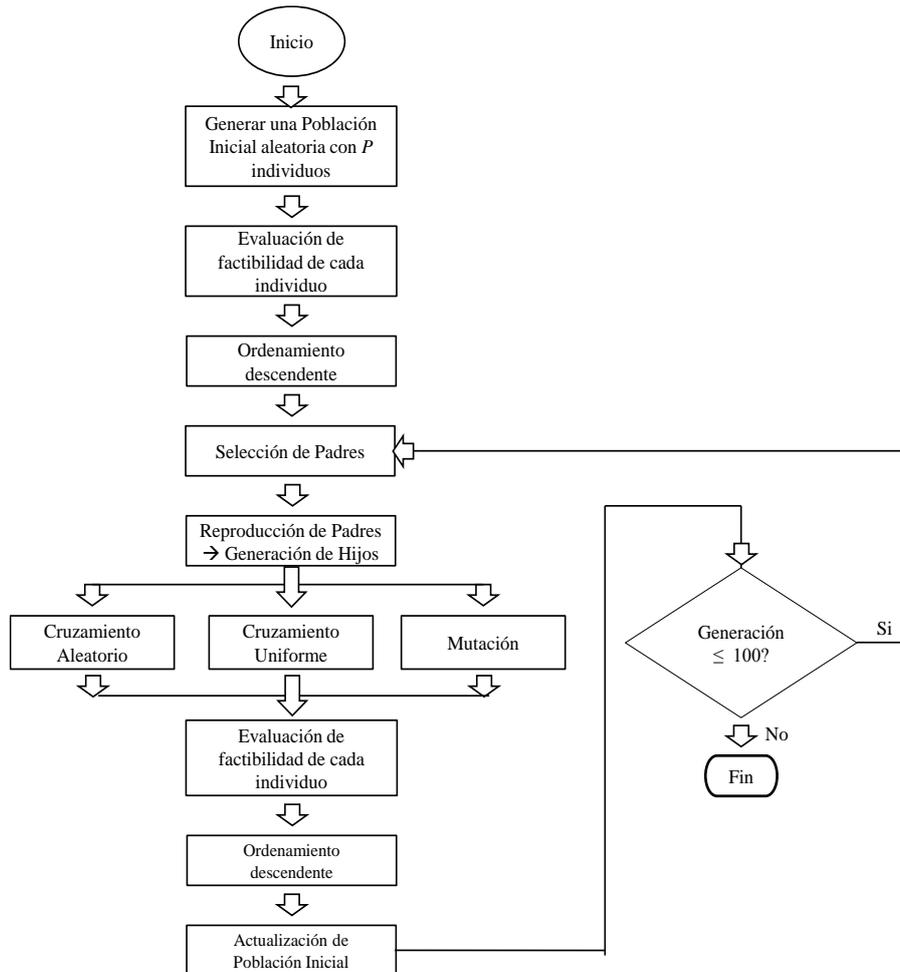


Fig. 1. Diagrama del AG. Fuente propia.

La manera en que los individuos *padres* se reproducen (recombinan) para generar nuevos individuos *hijos* es por medio de los operadores conocidos como *cruzamiento* y *mutación*. Usualmente los operadores de cruzamiento toman dos individuos  $p1$  y  $p2$  *padres* y generan dos individuos  $h1$  y  $h2$  *hijos* mediante la aplicación de la regla

probabilística. Los operadores de mutación solo necesitan un individuo *padre* para generar un individuo *hijo*.

Una vez que se han generado los individuos *hijos* se hace la actualización de la población con los mejores  $P$  individuos existentes considerando a todos los individuos. Esta población actualizada será la nueva población para la siguiente generación o iteración del AG. Este proceso de depuración se presenta en la Figura 2.

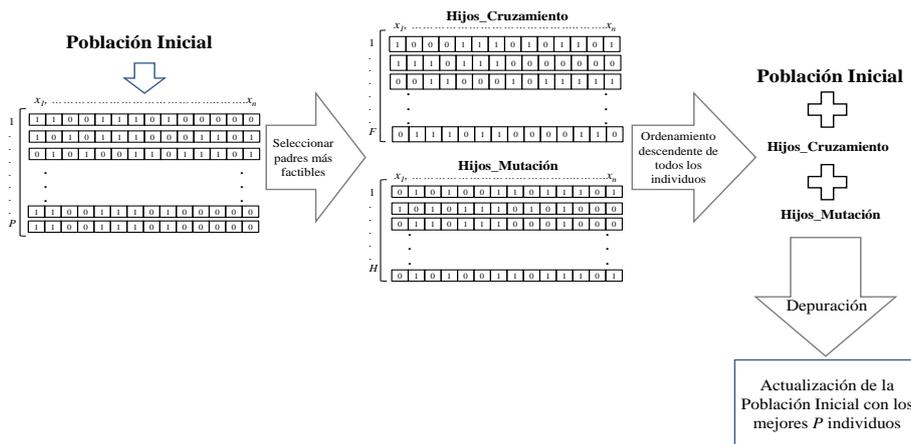


Fig. 2. Proceso depurativo de la población del AG.

A continuación se describen los operadores de cruzamiento y mutación del AG propuesto. Estos operadores fueron seleccionados dada la naturaleza binaria de la variable de decisión  $x_i$ .

- a) *Cruzamiento Aleatorio*: El cruzamiento se lleva a cabo entre Padre1 y Padre2 generando los hijos h1\_cruzamiento y h2\_cruzamiento. Los puntos de cruzamiento para la recombinación de información se seleccionan de manera aleatoria. La Figura 3 muestra el funcionamiento de este operador.

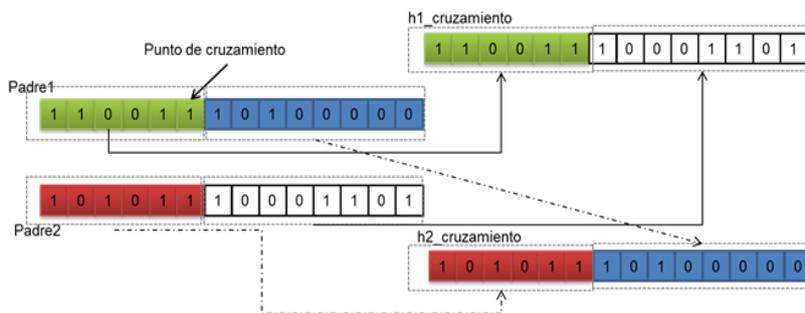


Fig. 3. Cruzamiento aleatorio. Fuente propia.

- b) *Cruzamiento Uniforme*: Se genera aleatoriamente un vector plantilla del mismo tamaño que los padres (en este caso, de tamaño 250) con 1's y 0's. Partiendo de

dos soluciones (Padre1 y Padre2) se generan dos hijos h1\_uniforme y h2\_uniforme de la siguiente manera con la plantilla generada (véase Figura 4):

- Para h1\_uniforme: Si plantilla = 0 tomar el valor de Padre1, de lo contrario tomar el valor de Padre2
- Para h2\_uniforme: Si plantilla = 0 tomar el valor de Padre2, de lo contrario tomar el valor de Padre1

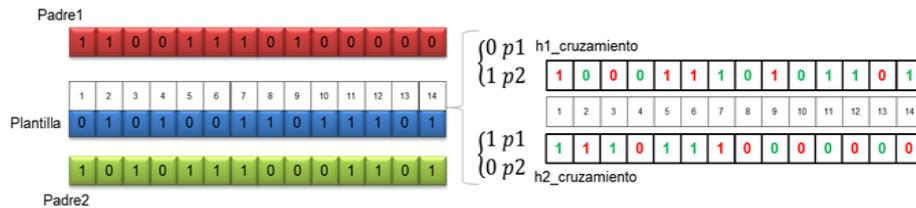


Fig. 4. Cruzamiento uniforme. Fuente propia.

- c) *Mutación*: Para obtener un hijo h\_mutación se toma un individuo Padre y se cambia de valor un elemento de  $x_j$  (de 0 a 1 y viceversa). Este proceso es ejemplificado en la Figura 5.

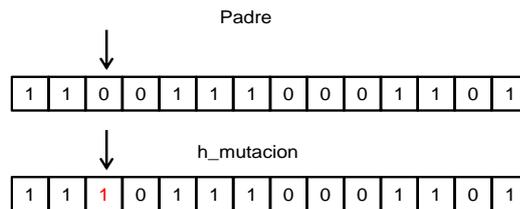


Fig. 5. Mutación. Fuente propia.

## 4.2. Comparación con Lingo

Para comparar el desempeño del AG se utilizó el software comercial Lingo. Esta es una herramienta de programación lineal que puede aplicar métodos como Branch & Bound para la resolución óptima de algunos tipos de problemas combinatorios (Martello & Toth, 1990).

Tabla 2. Comparativa del modelo exacto vs. AG. Fuente propia.

Comparativa		
	Lingo	AG Octave
	Óptimo	Factible
Ganancia	1093549	1058253
Productos	143	142
% Error		3.23%

La programación del modelo consiste en la maximización de la ganancia, considerando las restricciones de volumen y peso. La Figura 6 muestra los resultados obtenidos por Lingo que da una solución óptima del problema abordado.

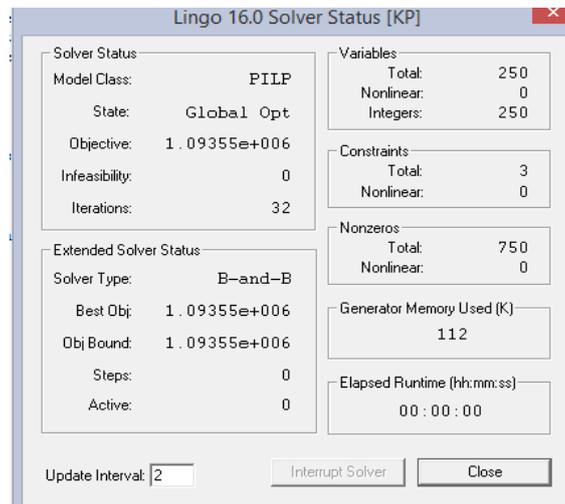


Fig. 6. Resultados con Lingo del KP del caso de estudio. Fuente Lingo.

De acuerdo al modelo de maximización de ganancia el software comercial Lingo da un resultado de 1,093,550.

## 5. Resultados

La Figura 7 muestra la convergencia del AG hacia un valor máximo de ganancia. En el eje horizontal se muestra la iteración o generación del AG en tanto que el eje vertical muestra el rango de la mejores ganancias totales considerando los productos asignados en cada iteración. Aproximadamente en la iteración 50 se alcanza el valor de convergencia que es de 1,058,253.

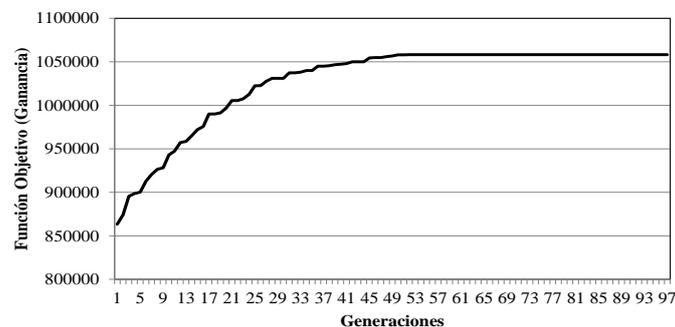


Fig. 7. Convergencia del AG. Fuente propia.

En la Tabla 2 se presenta la comparativa de los resultados de ambas herramientas. De acuerdo a la maximización de la ganancia con Lingo se obtiene un resultado de 1093590. En cambio, con el AG se obtiene una ganancia de 1058253. Esto es debido a que Lingo incluye 143 productos en tanto que el AG incluye 142 productos. Esta diferencia radica en un error de aproximación del 3.23%.

En la Tabla 3 se enlistan los productos a considerar de acuerdo a la solución proporcionada por el AG.

**Tabla 3.** Asignación generada por el AG. Fuente propia.

$j$	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25
$x_j$	1	0	0	1	0	0	1	1	1	0	1	1	0	0	1	1	0	0	0	0	0	0	0	1	1
$j$	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50
$x_j$	1	1	1	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	0	0	1	0	1	0	1
$j$	51	52	53	54	55	56	57	58	59	60	61	62	63	64	65	66	67	68	69	70	71	72	73	74	75
$x_j$	0	1	1	1	0	1	1	0	1	1	0	0	1	0	1	1	0	1	0	0	1	0	1	1	1
$j$	76	77	78	79	80	81	82	83	84	85	86	87	88	89	90	91	92	93	94	95	96	97	98	99	100
$x_j$	1	1	1	1	0	0	0	1	1	1	1	0	1	1	1	1	0	0	1	1	1	0	1	1	1
$j$	101	102	103	104	105	106	107	108	109	110	111	112	113	114	115	116	117	118	119	120	121	122	123	124	125
$x_j$	1	0	0	0	0	1	1	1	0	0	0	0	0	0	0	0	0	0	1	0	1	0	1	1	0
$j$	126	127	128	129	130	131	132	133	134	135	136	137	138	139	140	141	142	143	144	145	146	147	148	149	150
$x_j$	0	0	0	1	0	1	1	1	1	1	0	1	0	0	0	1	1	0	1	0	1	1	1	0	1
$j$	151	152	153	154	155	156	157	158	159	160	161	162	163	164	165	166	167	168	169	170	171	172	173	174	175
$x_j$	1	0	1	1	1	1	1	0	1	0	0	1	0	1	1	1	0	0	1	1	1	1	0	1	1
$j$	176	177	178	179	180	181	182	183	184	185	186	187	188	189	190	191	192	193	194	195	196	197	198	199	200
$x_j$	1	1	0	1	1	0	0	0	0	1	1	0	1	0	1	0	0	0	0	0	1	0	1	1	0
$j$	201	202	203	204	205	206	207	208	209	210	211	212	213	214	215	216	217	218	219	220	221	222	223	224	225
$x_j$	0	1	0	1	1	1	1	0	0	0	1	1	0	0	1	1	1	0	1	1	0	1	1	0	0
$j$	226	227	228	229	230	231	232	233	234	235	236	237	238	239	240	241	242	243	244	245	246	247	248	249	250
$x_j$	1	1	1	0	1	1	1	1	0	1	1	0	1	1	1	0	1	1	1	1	0	1	1	0	1

## 6. Conclusiones

El AG brinda una solución factible de manera libre con un error de 3.23% a un problema con amplio número de productos (250) y dos restricciones de capacidad. Debido a que el problema de la mochila o KP es del tipo NP-duro, con instancias mayores un software de optimización comercial puede ser más costo y requerir más tiempo computacional que un AG.

Dado el error mínimo obtenido por el AG se puede llegar a la conclusión de que es una herramienta factible para que empresas con recursos limitados lo puedan utilizar para la optimización de sus inventarios. Como trabajo a futuro aún existe área de oportunidad para continuar mejorando el desempeño del AG con distintos operadores y extenderlo para considerar diferentes tipos de productos con dimensiones de volumen.

## Referencias

1. Angel, E., Bampis, E., Gourvès, L.: Polynomial Approximation for Multicriteria Combinatorial Optimization Problems. 2nd Edition (ed V. Th. Paschos), Hoboken, NJ, USA: Wiley & Sons, Inc. (2014)

2. Avci, M., Topaloglu, S.: A Multi-Start Iterated Local Search Algorithm for the Generalized Quadratic Multiple Knapsack Problem. *Computers and Operations Research* 83, pp. 54–65 (2017)
3. Carraway, R. L., Schmidt, R. L., Weatherford, L. R.: An algorithm for maximizing target achievement in the stochastic knapsack problem with normal returns. *Naval Research Logistics* 40(2), pp. 161–173 (1993)
4. Disser, Y., Klimm, M., Megow, N., Stille, S.: Packing a knapsack of unknown capacity. In: *Proc. of the 31st International Symposium on Theoretical Aspects of Computer Science (STACS)*, pp. 276–287 (2014)
5. Gao, C., Lu, G., Yao, X., Li, J.: An iterative pseudo-gap enumeration approach for the Multidimensional Multiple-choice Knapsack Problem. *European Journal of Operational Research* 260(1), pp. 1–11 (2017)
6. Herrera, F., Lozano, M., Verdegay, J.: *Algoritmos Genéticos: Fundamentos, Extensiones y Aplicaciones*. Madrid: Arbor (1995)
7. Holland, J. H.: *Adaptation in Natural and Artificial Systems*. University of Michigan Press, Second Edition (1975)
8. Holland, J. H.: *Algoritmos Genéticos*. *Investigación y Ciencia*, pp. 38–45 (1992).
9. Iwama, K., Taketomi, S.: Removable online knapsack problems. In: *Proc. of the 29th International Colloquium on Automata, Languages and Programming (ICALP)*, pp. 293–305 (2002)
10. Kalaï, R., Vanderpooten, D.: The lexicographic  $\alpha$ -robust knapsack problem. *International Transactions in Operational Research* 18(1), pp. 103–113 (2011)
11. Karp, R.: Reducibility among combinatorial problems. *Proc. of Symposium on the Complexity of Computer Computations*, pp. 85–103 (1972)
12. Kellerer, H., Pferschy, U., Pisinger, D.: *Knapsack problems*. New York: Springer (2004)
13. Klamroth, K., Wiecek, M. M.: Dynamic programming approaches to the multiple criteria knapsack problem. *Naval Research logistics* 47(1), pp. 57–76 (2000)
14. MacGregor, J. N., Ormerod, T.: Human performance on the traveling salesman problem. *Perception & Psychophysics* 58(4), pp. 527–539 (1996)
15. Martello, S., Toth, P.: *Knapsack Problems: Algorithms and Computer Implementations*. England: John Wiley & Sons Ltd (1990)
16. Mencarelli, L., D'Ambrosio, C., Di Zio, A., Martello, S.: Heuristics for the General Multiple Non-linear Knapsack Problem. *Electronic Notes in Discrete Mathematics* 55, pp. 69–72 (2016)
17. Olivera, A.: *Heurísticas para Problemas de Ruteo de Vehículos*. Montevideo, Uruguay: Instituto de Computación, Facultad de Ingeniería (2004)
18. Pedemonte, M.: Estudio Empírico de Operadores de Cruzamiento en un Algoritmo Genético Aplicado al Problema de Steiner Generalizado. *RedUNCI-CACIC*, pp. 216–624 (2003)
19. Scenna, F., Anaut, D., Passoni, L. I., Meschino, G. J.: Reconfiguration of electrical networks by an Ant Colony optimization algorithm. *IEEE Latin America Transactions*, pp. 538–544 (2013)
20. Srinivas, M., Patnaik, L. M.: Genetic algorithms: a survey. *IEEE-Computer* 27(6), pp. 17–26, (1994)
21. Vergara-Canizales, V. G.: *Aplicación de Algoritmos Genéticos en el balanceo de líneas de producción*. Tesis, Reynosa, Tamaulipas, México: Universidad Autónoma de Nuevo León (2005)

# Clasificación de imágenes de redes mitocondriales asociadas al estudio del cáncer de mama

F.J. Vázquez-Vázquez, E. Castillo-Montiel, J.C. Chimal-Eguía,  
P. Maycotte-Gonzáles

Instituto Politécnico Nacional, Centro de Investigación en Computación,  
México

fvazquezv89@gmail.com, {ecastillom, jchmale}@ipn.mx, bisbenzimida@gmail.com

**Resumen.** Las mitocondrias son organelos dinámicos involucrados en diversos procesos celulares, sus funciones están relacionadas a su morfología. Las alteraciones en la dinámica mitocondrial se han asociado como biomarcadores a diversas enfermedades incluyendo el cáncer de mama. En este trabajo presentamos un método de clasificación para imágenes de redes mitocondriales extraídas de distintas líneas celulares (MCF10A, BT549, MDAMB23 y CMF) pertenecientes a distintos subtipos del cáncer de mama. El método se basa en tres etapas: en la primera se aplica un algoritmo de procesamiento morfológico y de segmentación, en la segunda se extraen características usando medidas empleadas comúnmente en redes complejas, finalmente se emplea el algoritmo no supervisado K-means para el proceso de clasificación. Los resultados indican una diferencia de las medidas obtenidas en cada subconjunto o clase, obteniendo una exactitud de 63.33% con 30 imágenes muestra.

**Palabras clave:** Morfología mitocondrial, grafos.

## Mitochondrial Morphology Images Classification for Assessing Breast Cancer

**Abstract.** Mitochondria are involved in a variety of cellular functions, which are related to their morphology. The disturbance in mitochondria dynamics has been associated to biomarkers for assessing breast cancer. In this work, we present a method for classification of mitochondrial morphology images, the set of images belongs to cell lines: (MCF10A, BT549, MDAMB23 y CMF) which are subtypes of breast cancer. The proposed method is based on the following image processing algorithms: segmentation (Otsu's method), features extraction (based on graph measures), and clustering algorithm (K-means). The study reveals a remarkable difference between the graph measures applied in features extraction. We obtain an accuracy of 63.33

**Keywords:** Mitochondrial Morphology, Graph Measures.

## 1. Introducción

Dentro de diversos procesos celulares, un incremento en la fisión (división de un solo organelo en dos o mas independientes estructuras) causa fragmentación mitocondrial y un aumento en la fusión (reacción opuesta la fisión) produce una red de mitocondrias que contraresta procesos metabólicos[1]. Un balance entre fisión y fusión define una morfología micondrial cuyo propósito es cumplir con las demandas metabólicas y asegurar la remoción de organelos dañados. Estos eventos se han asociado a la proliferación y redistribución de mitocondrias, permitiendo el estudio de distintos subtipos de cáncer de mama[1-3]. El objetivo de este trabajo es clasificar imágenes de redes mitocondriales(mas de una mitocondria), con base en los cuatro tipos de morfologías mostradas en la figura 1 (1.-completamente tubular, 2.-tubular con fragmentos, 3.-fragmentada con túbulos, 4.-completamente fragmentada) que son fueron definidas para el estudio del cancer de mama[1], este clasificador permite a los expertos en el campo agilizar el proceso de clasificación normalmente aplicado de forma visual, además se pretende disminuir el error que se pudiera derivar de factores relacionados a parámetros de percepción o condición humana. El conjunto de imágenes fueron extraídas de las siguientes líneas celulares (MCF10A, BT549, MDAMB23 y CMF), resultado de los estudios aplicados [1].

## 2. Método

La figura 2 representa el diagrama a bloques de las etapas de procesamiento. En la primera etapa se aplican dos algoritmos(segmentación y adelgazamiento) con objetivo de mantener la morfología de la red de mitocondrias en la imagen y tratar a esta morfología como un grafo. En la segunda etapa se obtienen medidas de redes complejas de la morfología obtenida en la etapa previa. Finalmente se define el algoritmo no supervisado que emplea las medidas obtenidas para utilizar el método K-means.

### 2.1. Etapa 1

Las imágenes empleadas se encuentran representadas por los canales RGB (red, green, blue). El primer paso es obtener una imagen en escala de grises, para esto se calcula el promedio del valor de cada canal para una unidad de pixel, la ecuación 1 determina la función  $f(x, y)$  que se obtiene al calcular el valor promedio de la intensidad de los canales  $r(x, y)$ ,  $g(x, y)$  y  $b(x, y)$ :

$$f(x, y) = \frac{r(x, y) + g(x, y) + b(x, y)}{3}. \quad (1)$$

**2.1.1. Segmentación** En esta sección el objetivo es aplicar un algoritmo que obtenga la región de interés de la imagen, es decir la red de mitcondrias. Este procedimiento es aplicado debido a que las imágenes contienen elementos(pixeles)

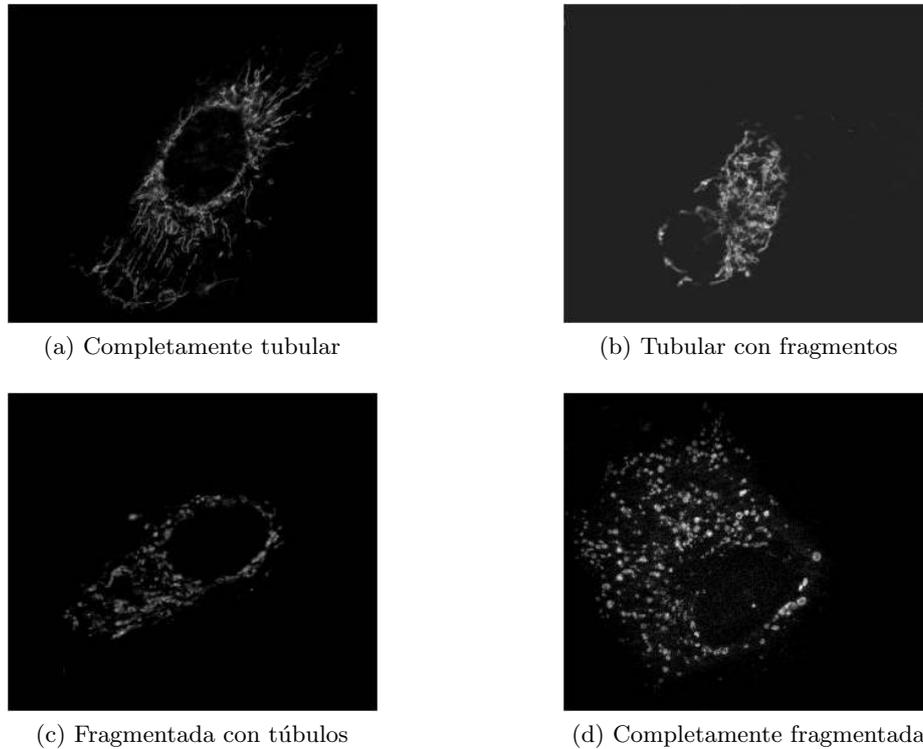


Fig. 1: Clases definidas para el proceso de clasificación.

que corresponden al núcleo de la célula, además representan parte de la reacción química con la cual se tiñe a la red de mitocondrias durante la captura de la imagen[1]. Esta segmentación se procesa sobre la función  $f(x, y)$ , para esto se emplea el método de segmentación Otsu [5].

El algoritmo Otsu consiste en obtener umbrales de segmentación dentro de un rango de intensidades 0-255, siendo el número de umbrales la cantidad de secciones entre las cuales se desea dividir el rango de escala de grises, para este trabajo se definen dos umbrales  $t_1$  y  $t_2$  que permiten extraer la geometría de la red de mitocondrias. Para encontrar los umbrales  $t_1$  y  $t_2$  se obtiene un histograma con las probabilidades de intensidad de cada nivel en la imagen, estos son separados obteniendo la probabilidad de tres clases con fronteras establecidas por  $t_1$  y  $t_2$ , con el objetivo de encontrar los valores  $t_1$  y  $t_2$  para los cuales la varianza de cada una de las tres clases es la máxima, estos umbrales definen la segmentación de las intensidades en el rango 0-255. En la ecuación 2 se muestra la función obtenida después de aplicar los umbrales encontrados usando el método Otsu, esta nueva imagen  $g(x, y)$  contiene solo la geometría de la red en una escala binaria.

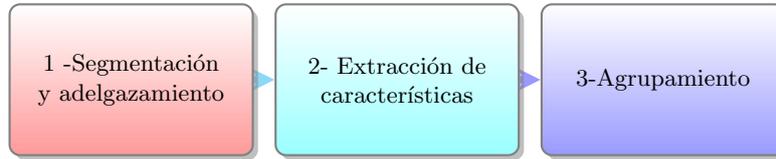
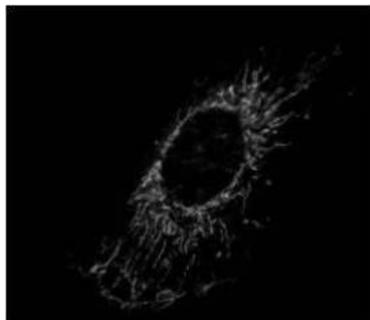


Fig. 2: Etapas del método de clasificación.

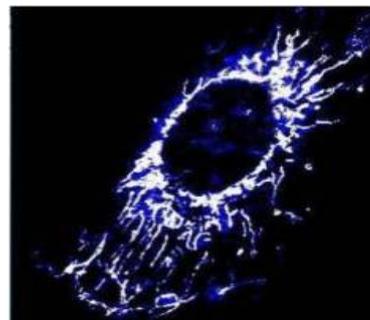
La figura 3 muestra la imagen obtenida después de aplicar el proceso de segmentación Otsu, la región de interés se define en la imagen con el nivel de intensidad blanco(255):

$$g(x, y) = \begin{cases} 0 & f(x, y) < t_1, \\ 0 & t_1 \leq f(x, y) \leq t_2, \\ 1 & f(x, y) > t_2. \end{cases} \quad (2)$$

En la figura 3 se muestra la imagen  $f(x,y)$ , y la imagen binaria  $g(x,y)$  obtenida en el proceso de segmentación, esta imagen es de nuestro interés ya que contiene la morfología de la red de mitocondrias.



(a) Imagen en escala de grises  $f(x,y)$ .



(b) Imagen  $g(x,y)$  obtenida mediante método Otsu

Fig. 3: Proceso de segmentación.

**2.1.2. Adelgazamiento** Previo a la extracción de características se emplea un algoritmo de adelgazamiento de la morfología de la imagen. El objetivo de este proceso es transformar la morfología de la red mitocondrial en un grafo, manteniendo las partes conexas. El método de adelgazamiento utilizado se fundamenta en la morfología matemática aplicada en imágenes, este tipo de morfología consiste en definir una imagen como un conjunto de elementos(pixeles) y aplicar

determinadas operaciones utilizando conjuntos establecidos [5], estos conjuntos establecidos son llamados elementos estructurales.

El algoritmo consiste en realizar una operación denominada acierto fallo  $g(x, y) \otimes b$  utilizando ocho elementos estructurales  $b$ , esta operación acierto o fallo se aplica sobre la imagen utilizando cada uno de los ocho elementos estructurales, y aplicando la operación sobre el resultado de operar el elemento  $n - 1$  sobre la imagen, tal como se define en la ecuación:

$$h(x, y) = (..((g(x, y) \otimes b_1) \otimes b_2 \dots \otimes b_8), \quad (3)$$

donde  $g(x, y)$  es la imagen obtenida en la etapa 1 y  $\{b_1 \dots b_8\}$  son los conjuntos o elementos estructurales que operan sobre la imagen, permitiendo obtener un adelgazamiento de la geometría sin afectar sus propiedades conexas. En la figura 4 se muestra la imagen  $h(x, y)$  obtenida a partir de  $g(x, y)$ , geometría que determina el grafo que representa a la red de mitocondrias en la imagen.

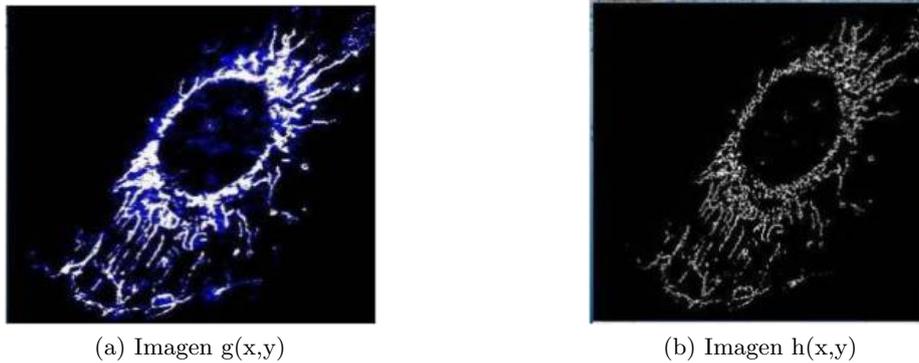


Fig. 4: Proceso de adelgazamiento.

### 3. Etapa 3

En esta etapa se propone un conjunto características que definen la morfología de la red mitocondrial, estas propiedades son normalmente aplicadas a redes complejas que topologicamente son grafos, con estas características se pretende establecer una diferencia cuantitativa en cada grupo de imágenes. Las medidas propuestas se obtienen estableciendo un nodo  $v$  como la unidad mínima dentro de la imagen(pixel), por otro lado las adyacencias de  $v$  con sus pixeles vecinos se definidas como las aristas  $e$ , un nodo  $v$  dentro de la imagen se toma en cuenta si el valor del pixel es igual al nivel de intensidad blanco en escala binaria. A continuación se describen las características:

### 3.1. Longitud promedio de la trayectoria

Se define a  $w$  como el promedio de pasos a través de la distancia (camino mas corto) de todos los posibles pares de nodos. Donde  $N$  es el número total de nodos  $v$  y  $d_{v_i v_j}$  es la distancia del nodo  $v_i$  al nodo  $v_j$ :

$$w = \frac{1}{N(N-1)} \sum_{i \neq j} d_{v_i v_j}. \quad (4)$$

### 3.2. Centralidad

Sea  $d_{ij}$  el camino mas corto posible entre los nodos  $v_i$  y  $v_j$  (también conocida como distancia geodésica). La centralidad  $x$  mide la distancia de un nodo, con respecto al nodo mas cercano:

$$x = \frac{N-1}{\sum_{i \neq j} d_{ij}}. \quad (5)$$

### 3.3. Excentricidad

La excentricidad  $y$  se define como la máxima distancia geodésica entre los nodos  $v_i$  y  $v_j$  donde  $i \neq j$ . En otras palabras es la máxima distancia de todos los pares de nodos:

$$y = \max_{v_i v_j} \{d_{v_i v_j}\}. \quad (6)$$

### 3.4. Densidad

Se puede definir a  $z$  como el radio de aristas en el grafo  $G$  con relación al número máximo de posibles aristas. Donde  $N$  es el número total de nodos en el grafo y  $E$  el número total de aristas:

$$z = \frac{2E}{N(N-1)}. \quad (7)$$

## 4. Clasificación

En este proceso se asigna el valor escalar obtenido en cada características  $w$ ,  $x$ ,  $y$ ,  $z$  a un vector de cuatro dimensiones  $(w, x, y, z)$  por cada imagen, posteriormente se definen cuatro grupos correspondientes a los tipos de morfologías definidas[1], después se aplica el algoritmo K-means para establecer el agrupamiento de las cuatro clases[6]. La ecuación 8 muestra la terminología establecida para el algoritmo.

$$\begin{aligned}
 &I(w, x, y, z) \text{ Vector imagen} \\
 &m_k \text{ centroide del cluster } k \\
 &r_{nk} \text{ Elemento r asignado al cluster } k \\
 &X_n \text{ Vector } X \text{ de la imagen } n \\
 &J \text{ Distancia de los vectores del grupo } k \text{ respecto a su centroide}
 \end{aligned} \tag{8}$$

Una vez establecido el vector  $(w, x, y, z)$ , el algoritmo busca minimizar el error  $J$  determinado por la suma de las distancias euclidianas de los vectores  $r$  con respecto al centroide  $m_k$ , pertenecientes a la clase  $k$ , obteniendo cada uno de los centroides por cada iteración, hasta que el error total sea menor a una condición de paro establecida, de esta forma se obtienen los cuatro grupos de imágenes que corresponden a los cuatro tipos de morfología (completamente tubular, tubular con fragmentos, fragmentada con túbulos, completamente fragmentada).

---

**algoritmo 1** Proceso de clasificación

---

```

1: while  $J > \text{error}$ 
2: for 1 to N
3:

$$r_{nk} = \begin{cases} 1 & k = \min ||x_n - m_k||^2 \\ 0 & \text{otro} \end{cases}$$

4:  $m_k = \frac{\sum_N r_{nk} x_n}{\sum_N r_{nk}}$ 
5:  $J = \sum_{n=1}^N \sum_{k=1}^K ||x_n - m_k||^2$ 
6: end for
7: end while

```

---

## 5. Análisis de resultados

Se utilizó un conjunto de 30 imágenes prueba correspondientes a los cuatro tipos de morfología de las redes mitocondriales, este conjunto fue clasificado por el expertos en el área despues de presentar los resultados obtenidos por el algoritmo propuesto. Para probar la eficiencia del clasificador y obtener algunas medidas de calidad (exactitud,error, sensibilidad, especificidad) se implementó el algoritmo y se compararon los resultados contra los resultados asignados por el experto: 5 completamente tubulares, 8 tubulares con fragmentos, 10 fragmentadas con túbulos y 7 fragmentadas, con estos datos se calculó la matriz de confusión mostrada en la tabla 1. Las filas de la tabla representan la clase a la cual pertenece la imagen, y las columnas representan la clase a la cual fue asignada la imagen por el clasificador. De esta forma se puede obtener el porcentaje de aciertos y fallas del conjunto de prueba.

Tabla 1: Matriz de confusión.

.	Tubular	Tubular con fragmentos	Fragmentada con túbulos	Fragmentada
Tubular	5	2	1	1
Tubular con fragmentos	2	2	1	0
Fragmentada con túbulos	1	4	6	0
Fragmentada	0	2	0	3

La figura 5 muestra la gráfica de la matriz de confusión, donde se puede ver mas a detalle en donde se encuentra acumulado la mayor cantidad de aciertos y fallos. Por ejemplo en la clase 1(Completamente tubular) y la clase 4(Fragmentadas) se obtienen los mejores resultados de clasificación obteniendo mayor número de aciertos. Por otro lado se observa que el mayor porcentaje de fallo en la clasificación se deriva de las clases 2(Tubular con fragmentos) y 3(Fragmentada con túbulos).

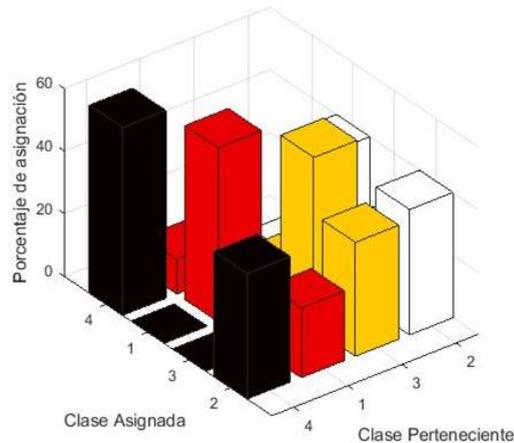


Fig. 5: Gráfica de la matriz de confusión.

$$Exactitud : 63,33\% . \tag{9}$$

Como se puede visualizar en la gráfica de la figura 5, el mayor porcentaje de aciertos corresponde a las clases 1 y 4, esto es debido a que algunas de las medidas

propuestas aplicadas en el grafo de la morfología presentan una diferencia notable las medidas propuestas, sin embargo las medidas como excentricidad, centralidad y longitud promedio, son vulnerables a escalabilidad en la imagen, lo que provoca una disminución o aumento en la magnitud obtenida, factor que influye en el porcentaje de aciertos.

$$\text{Error} : 36,66\%. \quad (10)$$

El mayor porcentaje de error es obtenido en la clasificación de grupos intermedios (2 y 3), debido a que presentan similitudes en las medidas propuestas, en estas circunstancias, el algoritmo de agrupamiento juega un factor importante, pues al aplicar el algoritmo K-means no se obtienen los clusters y el error no disminuye.

$$\text{Sensibilidad} : 58,12\%. \quad (11)$$

El número de aciertos por cada clase fue determinado a partir de la comparación de los resultados y la clasificación realizada por el experto para este conjunto de prueba, la sensibilidad obtenida se encuentra en su mayor parte definida por la clase 1 y 4, que es donde se obtuvo la mayor cantidad de positivos reales (elementos clasificados correctamente).

$$\text{Especificidad} : 84,09\%. \quad (12)$$

Este porcentaje se obtuvo debido a la incorrecta clasificación en las clases intermedias (2 y 3), las magnitudes en los escalares obtenidos para las características en estas clases, no presentan diferencia notable, por lo tanto un elemento puede ser mal clasificado con una mayor probabilidad.

## 6. Conclusiones

Las medidas propuestas para la caracterización de las redes mitocondriales son de gran apoyo para definir a las morfologías establecidas por los investigadores expertos en el área. Sin embargo el error obtenido (36.66%), deben disminuir con el objetivo de arrojar mejores resultados en los estudios realizados[1]. Las propiedades empleadas para la obtención de características son vulnerables a factores como escalabilidad en la imagen, tiempo de procesamiento y rotación, un ejemplo de esta afectación es la distancia entre pares de nodos, cuya magnitud puede ser afectada por escalabilidad o rotación de la imagen. Debido a estos factores, una propuesta del procedimiento es implementar un aprendizaje supervisado, en el cual los factores de afectación descritos, no jueguen un papel importante en el proceso de clasificación, además se pueda entrenar al sistema con un conjunto determinado para aumentar el porcentaje de exactitud y disminuir el error.

**Agradecimientos.** Agradecemos el apoyo al Consejo Nacional de Ciencia y Tecnología CONACYT y al Centro de Investigación en Computación del Instituto Politécnico Nacional.

## **Referencias**

1. Maycotte G. P. , Goyri A. M., Medina B. D., Reyes L. J, Anaya R. M, Cortes H. P: Estudio de parámetros asociados a la dinámica mitocondrial en cáncer de mama. Ctedras CONACYT (2016)
2. Rehman J., Zhang H., Toth P., Zhang Y., Marsboom G., Hong Z., Salgia R., Husain A., Wietholt C, Archer L. S: Inhibition of mitochondrial fission prevents cell cycle progression in lung cancer. *The FASEB Journal*, pp. 26–36 (2012)
3. Giedt R. J., Feruglio P. F., Pathania D., Yang K. S., Kilcoyne A., Vinegoni C., Mitchison T. J. , Weissleder R: Computational imaging reveals mitochondrial morphology as a biomarker of cancer phenotype and drug response. *Scientific Reports*, pp. 85–95 (2016)
4. Castillo M., E.: Estudio de neoplasias malignas utilizando dinámica no lineal. Master Tesis, pp. 32–40 (2009)
5. Gonzalez C. R., Woods R: *Digital Image Processing*. Prentice Hall, pp. 78–135 (2007)
6. Bishop C. M: *Pattern Recognition and Machine Learning*. Springer, pp. 423–428 (2006)

Impreso en los Talleres Gráficos  
de la Dirección de Publicaciones  
del Instituto Politécnico Nacional  
Tresguerras 27, Centro Histórico, México, D.F.  
noviembre de 2017  
Printing 500 / Edición 500 ejemplares

