

# EDUCACIÓN

SECRETARÍA DE EDUCACIÓN PÚBLICA



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

# Research in Computing Science

**Vol. 151 No. 8**  
**August 2022**

# Research in Computing Science

---

## Series Editorial Board

### Editors-in-Chief:

*Grigori Sidorov, CIC-IPN, Mexico*  
*Gerhard X. Ritter, University of Florida, USA*  
*Jean Serra, Ecole des Mines de Paris, France*  
*Ulises Cortés, UPC, Barcelona, Spain*

### Associate Editors:

*Jesús Angulo, Ecole des Mines de Paris, France*  
*Jihad El-Sana, Ben-Gurion Univ. of the Negev, Israel*  
*Alexander Gelbukh, CIC-IPN, Mexico*  
*Ioannis Kakadiaris, University of Houston, USA*  
*Petros Maragos, Nat. Tech. Univ. of Athens, Greece*  
*Julian Padget, University of Bath, UK*  
*Mateo Valero, UPC, Barcelona, Spain*  
*Olga Kolesnikova, ESCOM-IPN, Mexico*  
*Rafael Guzmán, Univ. of Guanajuato, Mexico*  
*Juan Manuel Torres Moreno, U. of Avignon, France*

### Editorial Coordination:

*Griselda Franco Sánchez*

*Research in Computing Science*, Año 21, Volumen 151, No. 8, agosto de 2022, es una publicación mensual, editada por el Instituto Politécnico Nacional, a través del Centro de Investigación en Computación. Av. Juan de Dios Bátiz S/N, Esq. Av. Miguel Othon de Mendizábal, Col. Nueva Industrial Vallejo, C.P. 07738, Ciudad de México, Tel. 57 29 60 00, ext. 56571. <https://www.rcs.cic.ipn.mx>. Editor responsable: Dr. Grigori Sidorov. Reserva de Derechos al Uso Exclusivo del Título No. 04-2019-082310242100-203. ISSN: en trámite, ambos otorgados por el Instituto Politécnico Nacional de Derecho de Autor. Responsable de la última actualización de este número: el Centro de Investigación en Computación, Dr. Grigori Sidorov, Av. Juan de Dios Bátiz S/N, Esq. Av. Miguel Othon de Mendizábal, Col. Nueva Industrial Vallejo, C.P. 07738. Fecha de última modificación 01 de junio de 2022.

Las opiniones expresadas por los autores no necesariamente reflejan la postura del editor de la publicación.

Queda estrictamente prohibida la reproducción total o parcial de los contenidos e imágenes de la publicación sin previa autorización del Instituto Politécnico Nacional.

*Research in Computing Science*, year 21, Volume 151, No. 8, August 2022, is published monthly by the Center for Computing Research of IPN.

The opinions expressed by the authors does not necessarily reflect the editor's posture.

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 of the IPN.

# Advances in Artificial Intelligence

Hiram Calvo Castro (ed.)



Instituto Politécnico Nacional, Centro de Investigación en Computación  
México 2022

## ISSN: in process

---

Copyright © Instituto Politécnico Nacional 2022  
Formerly ISSNs: 1870-4069, 1665-9899.

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

Electronic edition

## Table of Contents

	Page
Efecto del rezago en salud como un factor incidente en la pobreza en México, análisis a través de un enfoque en redes complejas y optimización .....	7
<i>Israel Santiago-Rubio, Roman Anselmo Mora-Gutiérrez, Juan Villegas-Cortez, Salomón Cordero-Sánchez, Jorge Juarez-Gómez</i>	
Agrupamiento basado en densidad para la detección automática de hot spots delictivos en la CDMX .....	19
<i>Jorge Carrasco-Islas, Edgar Moyotl-Hernández</i>	
Clasificación de los desórdenes del sueño a través de polisomnogramas usando un algoritmo de aprendizaje profundo .....	33
<i>Marco Antonio Rodríguez Castrejón, Leticia Ortega Máynez, Nelly Gordillo Castillo, José Manuel Mejía Muñoz, José David Díaz Román</i>	
Prototipo de robot manipulador para manejo de residuos peligrosos biológicos e infecciosos mediante detección de color .....	47
<i>Antonio-Alexander Benítez-Arévalo, Jesús García García, Guillermo-Rey Peñaloza-Mendoza</i>	
Propuesta de interfaz cerebro computadora realizada en una plataforma de bajo costo para la interpretación de la imaginación de movimiento .....	61
<i>Marcos Alejandro Garibay Campos, Antonio Alarcón Paredes, Jessica Cantillo Negrete, Rubén Isaac Cariño Escobar, Gustavo Adolfo Alonso Silverio</i>	
Análisis del comportamiento de estudiantes de programación durante su proceso de aprendizaje, utilizando árboles de decisión .....	77
<i>Víctor-Gonzalo Rivero-Martínez, Maricela Quintana-López, Asdrúbal López-Chau</i>	
Un método para simular lluvia mediante Motion blur y ruido sal y pimienta .....	89
<i>Roberto-Carlos Romero-Olmos, Sebastián Salazar-Colores, Hugo Moreno-Jiménez, Gerardo Flores</i>	
Extracción automática de frases clave para idioma inglés utilizando patrones léxicos .....	101
<i>Yanet Hernández-Casimiro, Yulia Ledeneva, René-Arnulfo García-Hernández, José-Luis Tapia-Fabela</i>	

Una revisión de chatbots en la salud.....	115
<i>Beatriz A. González-Beltrán, Miguel A. Vázquez-García, José A. Reyes-Ortiz, Raúl García-Ruiz</i>	
Control difuso para la mimetización del movimiento de un prototipo de prótesis de una extremidad canina .....	129
<i>Eduardo de Jesús García-Velázquez, Eugenia-Graciela López-Miranda, David Tinoco-Varela, Raúl-Dalí Cruz-Morales</i>	
Modelado del problema de composición musical polifónica automática como un problema de optimización biobjetivo.....	145
<i>Natan Vilchis, Adriana Lara</i>	
Aprendizaje automático para identificar episodios de esquizofrenias tratadas con clozapina.....	159
<i>Julieta G. Rodríguez-Ruiz, Carlos E. Galván-Tejada, Jorge I. Galván-Tejada, Hamurabi Gamboa-Rosales, José M. Celaya-Padilla</i>	
Uso de series temporales para el análisis y predicción de consumo de materia prima en la industria maquiladora.....	169
<i>Hugo Alberto Morales Casas, Francisco López Orozco, Vicente García Jiménez</i>	
Selección de atributos en microarreglos mediante un algoritmo wrapper metaheurístico.....	179
<i>Miguel Ángel Garrido-Castañeda, Antonio Alarcón Paredes, Oscar Camacho-Nieto</i>	

# Efecto del rezago en salud como un factor incidente en la pobreza en México, análisis a través de un enfoque en redes complejas y optimización

Israel Santiago-Rubio<sup>1</sup>, Roman Anselmo Mora-Gutiérrez<sup>2</sup>,  
Juan Villegas-Cortez<sup>2</sup>, Salomón Cordero-Sánchez<sup>3</sup>,  
Jorge Juarez-Gómez<sup>3</sup>

<sup>1</sup> Universidad Autónoma Metropolitana,  
Unidad Azcapotzalco, Maestría en Ciencias de la Computación,  
México

<sup>2</sup> Universidad Autónoma Metropolitana,  
Unidad Azcapotzalco, Departamento de Sistemas,  
México

<sup>3</sup> Universidad Autónoma Metropolitana,  
Unidad Iztapalapa, Departamento de Química,  
México

soporteisrael.isc@gmail.com,  
{mgra, juanvc}@azc.uam.mx,  
{scs, jjg85}@xanum.uam.mx

**Resumen.** En el presente trabajo se muestra un análisis sobre la pobreza en México, como consecuencia de la carencia en salud, en este análisis se involucra el enfoque de ciencia de la complejidad y la optimización. Se generó un modelo de red compleja donde se relacionaron los Estados de la República Mexicana con base a sus similitudes en salud; considerando para ello la información de los indicadores para el análisis de pobreza, propuestos por el Consejo Nacional de Evaluación de la Política de Desarrollo Social (CONEVAL). Los datos empleados se recuperaron del Instituto Nacional de Estadística y Geografía (INEGI), a partir de la Encuesta de Ingresos y Gastos en los Hogares en el año 2018 (ENIGH-2018), la cual desagrega a los indicadores de rezago educativo, acceso a los servicios de salud, acceso a la alimentación, acceso a la seguridad Social, calidad y espacios en la vivienda, y acceso a los servicios básicos en vivienda. El análisis de las comunidades en la red se hizo a través del problema de máximo clique, los resultados muestran una caracterización de la diferencia significativa en el sistema de salud y la pobreza de las provincias en México.

**Palabras clave:** Redes complejas, máximo clique, reconocimiento de patrones, análisis de pobreza.

## **Effect of Health Backwardness as an Incident Factor of Poverty in Mexico, Analysis through a Complex Network Approach and Optimization**

**Abstract.** This paper presents a series of analyses of multidimensional poverty in Mexico developed from a complex network and optimization approach. For this, network models were generated for each of the states of the Mexican Republic, based on the indicators provided by the National Council for the Evaluation of Social Development Policy (CONEVAL) for the analysis of multidimensional poverty. The used data were retrieved from the National Institute of Statistics and Geography (INEGI) from the Survey of Income and Expenses in Households in 2018 (ENIGH-2018) which disaggregates the indicators of educational backwardness, access to health services, access to food, access to social security, quality and spaces in housing, and access to basic services in housing. From this information, the maximum clique problem was used in each of the generated networks. The results show a significant difference of the healthcare system and the poverty of Mexican communities.

**Keywords:** Complex networks, maximum clique, pattern recognition, poverty analysis.

### **1. Introducción**

El CONEVAL define a la pobreza como la situación en la que se encuentra una persona al presentar al menos una carencia social, además que sus ingresos no son suficientes para cubrir sus necesidades alimentarias y no alimentarias. La pobreza es un fenómeno multidimensional, en México se emplea la metodología propuesta por el CONEVAL para su estudio y análisis. En esta metodología se emplea como insumo datos generados por el INEGI.

Estudios recientes en Latinoamérica han tomado como referencia esa metodología para llevar a cabo diferentes estudios sobre la pobreza multidimensional; sin embargo, el enfoque empleado para analizar este problema contempla técnicas de estadística clásica, análisis multivariable, análisis espacial, por mencionar algunos.

Por otro lado, cada vez se hace notorio la gran cantidad de datos que se generan de forma continua e incesante por entidades, usuarios, servicios o dispositivos, lo que ha implicado el desarrollo de nuevas técnicas, enfoques y métodos que permitan almacenar, procesar y analizar conjuntos de datos de gran tamaño (Big data). Este nuevo paradigma de tratar y analizar datos ha generado información y conocimiento en diversos sectores como la industria, la banca, las finanzas, el marketing, las redes sociales, el comercio electrónico, el sector salud, en el gobierno, las ciudades inteligentes, y en el ámbito académico [5].

Recientemente, varias aproximaciones han surgido del manejo de problemas de decisión complejos, como son: Algoritmos Genéticos (AG), Redes Neuronales Artificiales, Recocido Simulado, Búsqueda Tabú, Análisis de Objetivos y Búsqueda Dispersa, entre otros.

Estas heurísticas y metaheurísticas, tienen gran aceptación para proporcionar metodologías en el campo de la Inteligencia Artificial y la optimización [2]. En la actualidad, la investigación se ha dirigido hacia el diseño de buenas heurísticas, i.e., algoritmos computacionales eficientes, con respecto al tiempo de cómputo, y al espacio de memoria y con cierta verosimilitud, de entregar una solución “buena”, o relativamente cercana a la óptima mediante el examen de solo un pequeño subconjunto de soluciones del número total [2].

En este trabajo el estudio de comunidades se realiza a través de adaptar el problema de máximo clique, sobre la red de relaciones de salud generada. De acuerdo al CONEVAL, el problema de pobreza se cuantifica con base a un conjunto de indicadores, uno de ellos es el indicador de carencia por acceso a los servicios de salud, el cual es examinado.

La metodología propuesta consiste en lo siguiente: (1) recuperar los datos del indicador de carencia por acceso a la salud (2) caracterizar al indicador por Estados (las Entidades federativas de México), (3) generar el modelo de red compleja, (4) analizar el modelo obtenido mediante el uso de un AG para detectar el clique máximo en la red, (5) detectar las comunidades que se forman, y (6) comparar los resultados obtenidos para el indicador de salud con los generados en [6] para el indicador educación.

En la Sección 2 se presenta en resumen el estado del arte, en la Sección 3 se expone con mayor detalle la metodología propuesta y en la Sección 4 su implementación, presentando los resultados obtenidos en la Sección 5, para finalmente exponer nuestras conclusiones en la Sección 6.

## **2. Estado del arte**

En años recientes, se ha estudiado el problema de la pobreza multidimensional en México, desde diferentes enfoques estadísticos. Los análisis que se han aplicado toman como soporte al modelo desarrollado por el CONEVAL, con esta metodología se han llevado a cabo diferentes trabajos, por ejemplo: diagnósticos geográficos de pobreza y carencia alimentaria en México, que identifican las áreas del país donde se concentran éstos problemas, de tal manera que se prioriza a las regiones para el diseño de políticas de seguridad alimentaria [3].

Se ha aplicado la medición multidimensional de la pobreza a nivel intra-urbano en Ciudad Juárez, Chih. [4], año 2012; ahí se trabaja con análisis multivariado. Los resultados muestran la identificación de unidades de análisis territorial (UTA) en Ciudad Juárez, donde la pobreza se padece con una mayor incidencia, intensidad y profundidad en esa Entidad.

Hablando de otros trabajos relacionados, en [10] se presenta un enfoque basado en AG para detectar comunidades, dentro del método propuesto hace uso de la medida de coeficiente de clustering (CC-GA), que lleva a cabo el siguiente proceso: usa la medida de coeficiente de clustering para generar a la población inicial, la evalúa usando

modularidad, la cual también toma como medida para la función de aptitud y así evaluar a la calidad de la población inicial.

Aplica operadores genéticos de cobertura uniforme y mutación, que asegura que pequeñas comunidades sean unidas a comunidades mayores y de manera simultánea incrementa el valor de la comunidad.

En [9] se considera que los algoritmos convencionales para detectar comunidades, no toman en cuenta las interacciones existentes entre los nodos, por lo que su utilidad es limitada. Para superar esas limitaciones, propone un algoritmo de detección de comunidad atribuida multiobjetivo con análisis de importancia de nodo (MANIA), el cual toma en consideración lo siguiente:

Dos funciones objetivo para evaluar la idoneidad de las comunidades desde la perspectiva de la estructura y los atributos, incorpora la información de atributos de los nodos para beneficiarse de su mayor poder de discriminación, y estima la importancia de los nodos utilizando el grado de convergencia y el campo potencial de topología.

MANIA aporta tres características principales: detecta comunidades que son interpretables no solo desde la perspectiva de la estructura sino también desde la perspectiva de los atributos, caracteriza la confianza en la información de categorización de atributos, y brinda el análisis de la importancia de los nodos.

En [7] se menciona que algunos métodos de detección de comunidades, requieren que sea el usuario quien provea de un conocimiento previo sobre la estructura de la comunidad, tales suposiciones son poco razonables para la estructura de comunidad, lo que resulta en que los algoritmos de detección sean incapaces de revelar la estructura interna de la comunidad de manera eficiente. Por lo que propone detectar comunidades en redes complejas haciendo uso de la estrategia de enjambre de partículas discreta con diversidad y mutación (DPSO-PDM).

Para realizar la metodología propuesta, primero redefine: codificación, velocidad, y posición de la partícula, así como la operación evolutiva en desratización. Así también, toma en cuenta que el algoritmo tradicional de enjambre de partículas es fácil quedar atrapado en un óptimo local; por ello, hace uso de la estrategia de ajuste de peso de inercia de hibridación basada en la diversidad de partículas nuevas, de igual forma utiliza la estrategia de mutación adaptativa para evitar la convergencia local del algoritmo.

A pesar de que la teoría y los resultados experimentales muestran que DPSO-PDM puede efectivamente detectar la estructura de la comunidad de la red, la calidad de la clasificación de la comunidad y la convergencia global son estables, el algoritmo propuesto tiene el inconveniente de consumir demasiados recursos computacionales.

En [11] se diseña una medida para evaluar la robustez de una comunidad en redes interdependientes. Con base en esa medida propone un algoritmo de optimización mimético para mejorar la robustez de la comunidad en varios tipos de redes (reales y sintéticas).

Teniendo en cuenta que la solidez de la comunidad es importante en varias aplicaciones diarias, amplía la medida existente para evaluar la estabilidad de las comunidades en redes individuales aisladas a redes interdependientes. En el trabajo realizado, se valida que los nodos con mayor centralidad de intermediación sean clave para mantener

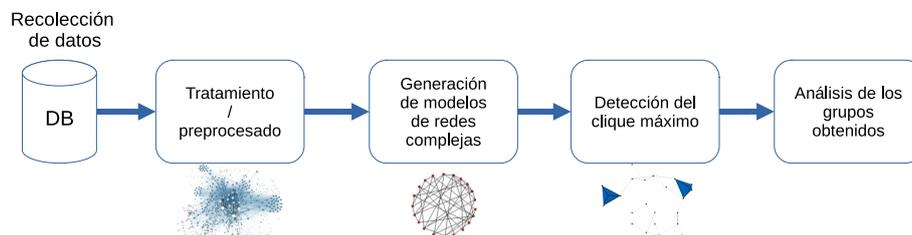


Fig. 1. Metodología propuesta.

la información de partición intrínseca en redes interdependientes, lo que es similar a la conclusión de varios estudios previos sobre redes individuales.

Además, para mejorar la invulnerabilidad de las comunidades en redes interdependientes a través de topologías de recableado, se ha diseñado un algoritmo mimético que contiene operadores dirigidos por red, MA-CR-inter, para manejar la tarea de optimización.

### 3. Metodología

En esta sección se describe la metodología propuesta en este trabajo; a saber, los pasos que se llevaron a cabo para la analizar al indicador de carencia por acceso a los servicios de salud, que mide junto a otros indicadores la pobreza multidimensional propuesta por el CONEVAL en México, los métodos que se utilizaron para generar la red compleja, de igual forma, se describen las características del AG utilizado para detectar comunidades mediante el problema de clique máximo para el modelo generado. La Figura 1 muestra el esquema general la metodología propuesta en este trabajo. Las etapas comprendidas se detallan a continuación.

**Recolección de los datos:** Para la generación de los indicadores de pobreza, se utilizaron las bases de datos provenientes del Modelo Estadístico 2018 para la continuidad del MCS-ENIGH del INEGI [1, 8]; los datos correspondientes a la MCS-ENIGH-2018 fueron recuperados de los levantamientos que realiza el INEGI cada dos años a nivel federal para este ejercicio los datos pertenecen al censo tomado en el año 2018. Con este conjunto de datos se generó el indicador de salud.

**Caracterización / preprocesado de los datos:** Se realizó un proceso para limpiar o adecuar los datos obtenidos de [1, 8], homogeneizando y estructurando la información. Los Estados de la República Mexicana se ubican en cada una de las filas y cada una de las variables que describen a cada indicador se ubican en las columnas, finalmente se realizó un resumen estadístico descriptivo sobre el nuevo conjunto de datos. Como resultado de esta etapa se generó un nuevo conjunto de datos, del cual se eligieron las tablas que contienen a los indicadores de pobreza multidimensional.

**Generación de modelos de redes complejas:** Para la construcción de los modelos de redes complejas, se analizó el conjunto de datos con la métrica de Mahalanobis. Se configuró un script que se ejecutó sobre un entorno de desarrollo integrado, el cual toma como parámetros de entrada al conjunto de datos, y retorna en su salida una matriz de

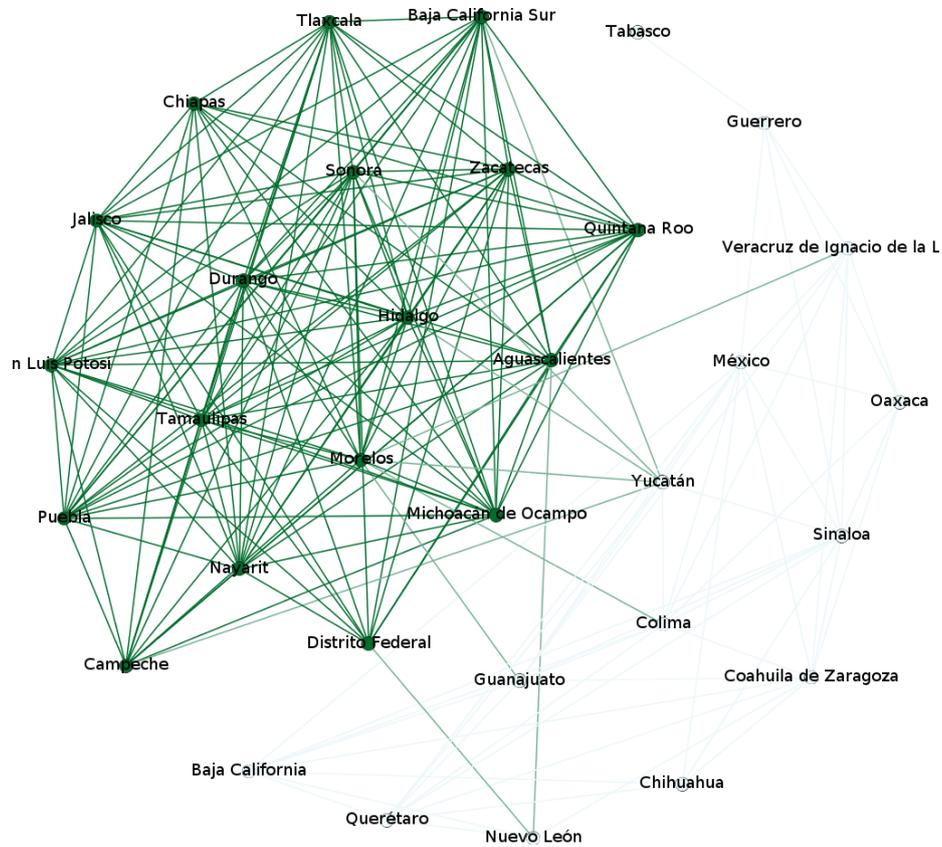
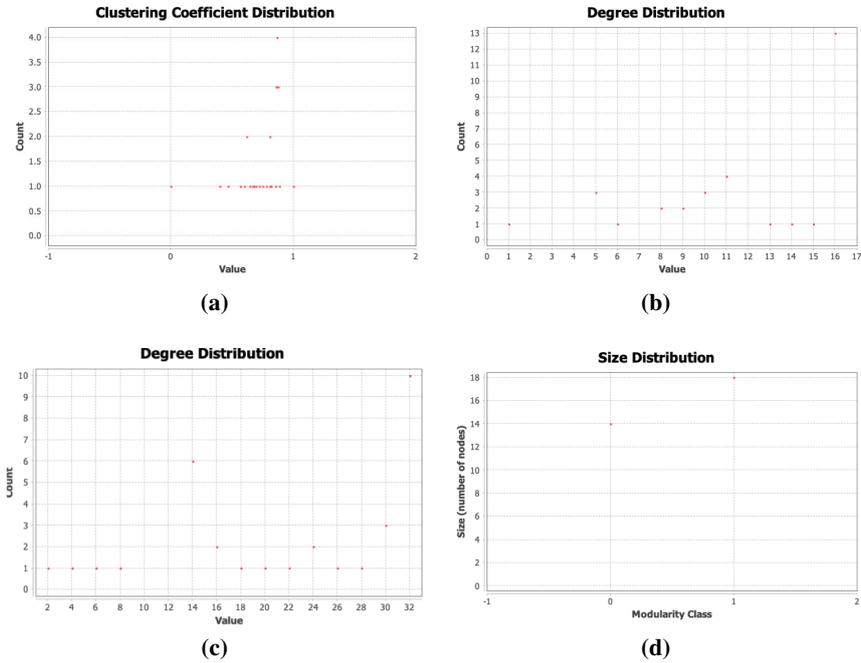


Fig. 2. Modelo de red compleja generado para el indicador de carencia por salud.

distancias y una matriz de adyacencia. De igual forma, se construyó un modelo de red general que retoma la información de salud.

**Detección del máximo clique:** Con el fin de detectar grupos de estados que compartan características similares, se configuró un AG, con el cual se analizaron los modelos de red obtenidos. Los parámetros considerados de entrada del AG son: una población inicial de 10 individuos, 500 generaciones, una tasa de sobrevivencia del 0.30 y la matriz de adyacencia. Con estas configuraciones el AG devuelve una matriz de soluciones,  $M5$ , con 20 aproximaciones por indicador.

**Análisis de los grupos obtenidos:** Se configuró un script el cual toma como entrada la matriz de soluciones, la matriz de adyacencia y la matriz donde se llevó a cabo la caracterización de los indicadores. Este proceso devuelve resúmenes estadísticos los cuales muestran a los grupos de Entidades que comparten características comunes, e.g., el promedio nacional por indicador, así como las Entidades que no comparten atributos en común.



**Fig. 3.** (a) Coeficiente de clustering por la red, (b) Número promedio de conexiones en la red, (c) Grado de conectividad de los nodos, y (d) Modularidad para la red compleja.

## 4. Implementación

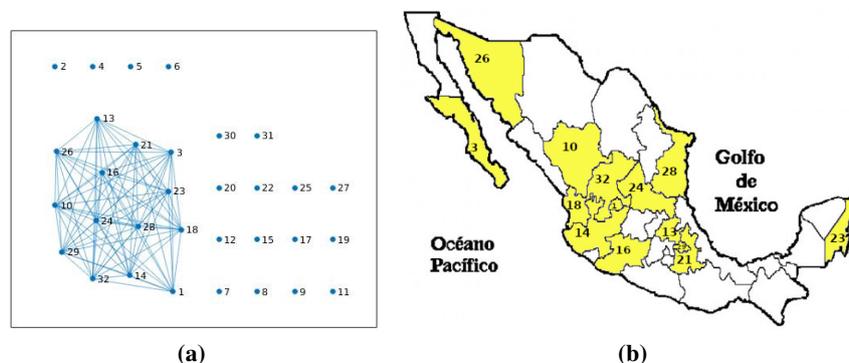
Toda la implementación de nuestra metodología propuesta se desarrolló en alto nivel, con scripts para MATLAB©2020 y R©<sup>1</sup>. Los scripts fueron ejecutados en una computadora con un procesador Intel Core i7 de dos núcleos, 3 GHz, con 8 GB RAM 1600 MHz DDR3.

### 4.1. Conjuntos de datos

Se utilizaron las bases de datos provenientes del Modelo Estadístico 2018 para la continuidad del MCS-ENIGH del INEGI [1, 8], siendo un aproximado de 16,000 registros, con 70 campos de información.

Los datos correspondientes a la MCS-ENIGH-2018 fueron recuperados de los levantamientos que realiza el INEGI cada dos años a nivel federal, para este ejercicio los datos pertenecen al censo tomado en el año 2018. Con este conjunto de datos se generó el indicador de salud.

<sup>1</sup> R: A Language and Environment for Statistical Computing. Vienna. <https://www.R-project.org/>



**Fig. 4.** (a) 2 comunidades creadas con base al problema de máximo clique. (b) Localización geográfica de las comunidades.

## 5. Resultados

En esta sección se presenta el modelo de red compleja obtenido luego de haber analizado al indicador de carencia por acceso a la salud propuesto por el CONEVAL, mediante la metodología propuesta. Para este indicador se construyó un modelo de red compleja donde los vértices o nodos corresponden a las Entidades federativas, y las aristas corresponden a la interacción entre ellas (los Estados).

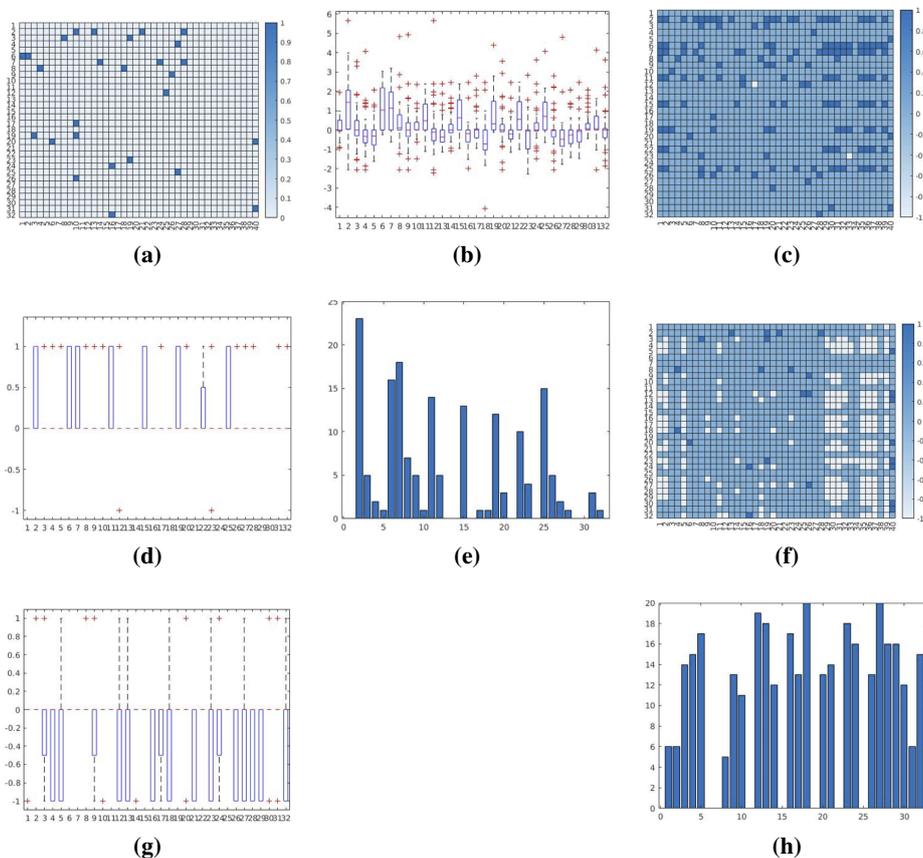
Primero se presenta el modelo de red compleja, posteriormente se muestran los resúmenes estadísticos que caracterizan al modelo de red, luego, se muestra al clique detectado mediante el AG, finalmente se presentan los resúmenes estadísticos de la comunidad detectada.

**Indicador de carencia por salud.** Está modelado por la red compleja mostrada en la Figura 2. Las métricas del modelo de red compleja para este indicador se muestran en la Figura 3, y son la siguientes: el número promedio de conexiones, 3(b), a través de la red es 11.062 (Average Degree). El grado de conectividad de los nodos en la red, 3(c), es 22.125 (Average Weighted Degree).

De acuerdo a la densidad del grafo (Graph Density), el 35 % de los nodos están conectados entre sí. Con base en la función de modularidad de la red compleja, el número de comunidades presentes en este indicador son 2. La longitud promedio de la trayectoria es 1.997 (Average Path Length) con un diámetro de 5, esto da una idea de que cualquier nodo está típicamente a menos de 2 grados de cualquier otro nodo, y ningún otro nodo está separado por más de 5 grados.

En la Figura 4a se presentan las 2 comunidades halladas. Éstas se crearon con base al problema de máximo clique, el cual fue resuelto mediante un AG. Una comunidad está integrada por los estados de Chihuahua (7), Coahuila de Zaragoza (7), Guanajuato (11), México (15), Nuevo León (19), Querétaro (22) y Sinaloa (25).

Una segunda comunidad está conformada por las Entidades Nayarit (18), Tabasco (27), Tamaulipas (28), Hidalgo (13), Jalisco (14), Puebla (21), Durango (10) y Tlaxcala (29). Michoacán de Ocampo (16), Veracruz de Ignacio de la Llave (30), Chiapas (5), Aguascalientes (1), Zacatecas (32), San Luis Potosí (24) y Morelos (17). Véase en la Figura 4b la localización geográfica de las comunidades.



**Fig. 5.** (a) Matriz de estados vs atributos, (b) Entidades por arriba y por debajo del promedio, (c) Entidades que integran a una comunidad, (d) Entidades por arriba del promedio, (e) Cantidad de atributos para las Entidades que no están en la comunidad, (f) Entidades que integran a la segunda comunidad,(g) Entidades que se ubican por debajo del promedio del indicador de carencia, y (h) Cantidad de atributos para cada una de las Entidades.

Existen algunas Entidades que no pertenecen a ninguno de los grupos formados, esto se debe a que en algunos casos los atributos examinados no cuentan con información suficiente para clasificar a la Entidad y en otros casos existen atributos donde hay un excedente de información lo cual hace que las Entidades queden fuera de los grupos.

En la Figura 5a, se observa que existen atributos donde la información que se captura para llevar a cabo la medición del indicador de carencia en salud que contienen un exceso de información, por otro lado, se puede observar que hay atributos donde la información capturada no está aportando relevancia al indicador de carencia en salud en nuestro caso de estudio, debido a que no se cuenta con la información suficiente para el indicador de carencia.

En la Figura 5b, se presenta la distribución de los Estados de la República Mexicana

con base a la media del indicador de carencia por salud. En la Figura 5c, se observa que las Entidades que forman una comunidad tienen el valor de cero en todos sus atributos, esto es, si se observa a la imagen por filas (cada fila representa a una Entidad) se puede apreciar al grupo formado, donde sus atributos son similares. Por otro lado, si se considera a la matriz por sus columnas, se puede observar a las Entidades que no forman parte de la comunidad debido a que sus atributos son diferentes, ya sea porque hay un exceso de información, lo que ocurre en la mayoría de las Entidades o porque carecen de información.

En la figura siguiente (Figura 5d), se observa que existen Entidades que se ubican por arriba del promedio, sin embargo, no pertenecen a la comunidad de la parte inferior de la Figura 4a, ya que como se mostró en la Figura 5c, éstas Entidades tienen atributos con información excedente, lo que implica que los atributos queden fuera de la comunidad conformada.

En la Figura 5e, se observan a las Entidades que no pertenecen a la comunidad inferior mostrada en la Figura 4a, debido al número de atributos que son diferentes de la comunidad. El eje vertical representa a los atributos y el eje horizontal corresponde a las Entidades. Ver Figura 5e.

Para el caso de la segunda comunidad que se conformó y que se muestra en la Figura 4a. Las descripciones estadísticas son las siguientes: En la Figura 5f, se observa a las Entidades que no forman parte de la comunidad de la Figura 4a, en este caso, se observa que existen Entidades donde la información que se captura para llevar a cabo la medición del indicador de carencia de salud resulta insuficiente, esto es debido al valor de “-1” (casilla en blanco) en cada atributo examinado para cada Entidad.

Existen también, aunque en menor medida, algunos atributos que cuentan con un excedente de información. Estos casos causan que las Entidades no pertenezcan al grupo formado. Por el contrario, las Entidades que comparten atributos similares, tienen un valor de cero (casilla en azul claro) son aquellas que integran una comunidad.

En la Figura 5g, se observa que existen Entidades que se ubican por debajo de la línea del promedio, que se puede apreciar en línea “0” horizontal de las Figuras 5b, 5d, 5g, como indicador de referencia para el promedio. Sin embargo, estas comunidades no pertenecen a la comunidad agrupada, debido a que en sus atributos son diferentes con respecto a aquellas Entidades que sí pertenecen a la comunidad agrupada.

En la Figura 5h, se observa a las Entidades que no pertenecen a la comunidad, debido a la cantidad de atributos que aportan poca información para clasificarlos en la comunidad agrupada. De igual forma el eje horizontal son las Entidades y el eje vertical son los atributos que miden a este indicador.

## **5.1. Análisis de resultados**

Como se observa en [6], al analizar el indicador de carencia por rezago educativo se forman dos comunidades, sin embargo, para el caso del indicador de salud analizado en este trabajo se forma una sola comunidad.

Se aprecia en ambos trabajos la existencia de Entidades que no se integran a las comunidades las cuales son: (31) Yucatán, (9) Distrito Federal, (12) Guerrero, (20) Oaxaca, (23) Quintana Roo, (2) Baja California, (4) Campeche y (8) Colima.

Si se compara lo presentado en [6] con lo presentado en este trabajo, se puede notar que existen Entidades aisladas en ambos trabajos, para el caso de este análisis estas Entidades no pertenecen a ninguna comunidad debido a que las variables que miden al indicador de carencia por acceso a la salud tienen sobrecarga de información.

Otro punto a destacar consiste en que al observar a la comunidad que se forma en este trabajo consiste en Entidades donde los atributos que miden al indicador de salud no aportan suficiente información o están por debajo del promedio, i.e., el conjunto de Entidades presentado en esta sección está contenido en el conjunto de Entidades mostrado en [6], y que sugiere que la comunidad se encuentra por debajo del promedio; ocurre lo mismo para el caso de las del indicador y de las Entidades que integran a la comunidad analizada en este trabajo.

## **6. Conclusiones**

En este trabajo se ha mostrado la forma en que los Estados de la República Mexicana se agrupan de acuerdo a las características o atributos que se consideran en uno de los indicadores de carencia de definidos por el CONEVAL. El indicador de salud lo hemos considerado para nuestro aporte aquí presentado, y muestra que se genera una comunidad entre los Estados de la República. Este comportamiento se refuerza con los resultados de [6]; lo que identifica que la diferencia de los grupos formados por las desigualdades de educación y salud han generado pobreza marcada en los grupos formados.

En los resultados se observa una comunidad que demuestra Entidades que están asociadas ya sea porque se ubican por arriba o por debajo del promedio de cada uno de los respectivos indicadores. De igual forma, se observa en los resultados que existen indicadores de carencia donde sus variables tienen una sobrecarga de información en algunos casos y en otros casos las variables contienen reactivos que no aportan información.

Por último, se ha presentado la forma de analizar a los indicadores de carencia como un único modelo de red compleja, esto arroja el resultado valioso de identificar algunas Entidades que no se integran a las comunidades. Como ejemplo de esto tenemos que los Estados de Durango, Hidalgo, Jalisco, Michoacán de Ocampo, Nayarit, Puebla, Sonora, Tamaulipas, Tlaxcala y Zacatecas, son Entidades que en cada una de los cliques siempre aparecen como nodos dentro de una comunidad. De igual forma, las siguientes Entidades Baja California, Colima, Guerrero, Oaxaca, y Yucatán, nunca se integran a las comunidades construidas.

Esperamos con nuestro trabajo poder aportar a las políticas de desarrollo públicas de nuestro país, en el apoyo de las decisiones de gobierno en salud y educación.

## **Referencias**

1. CONEVAL: Medición de la pobreza: Programas de cálculo y bases de datos 2008, 2010, 2012, 2014, 2016 y 2018. Consejo Nacional de Evaluación de la Política de Desarrollo Social. [https://www.coneval.org.mx/Medicion/MP/Paginas/Programas.BD\\_08\\_10\\_12\\_14\\_16\\_18.aspx](https://www.coneval.org.mx/Medicion/MP/Paginas/Programas.BD_08_10_12_14_16_18.aspx)

2. De los Cobos, S. G., Close, J. G., Gutiérrez, A. M. A., Martínez, L. A. E.: Búsqueda y exploración estocástica. Universidad Autónoma Metropolitana. (2010)
3. Espinoza-Ramos, J., Rodríguez-Gámez, L. I.: La geografía de la pobreza alimentaria en México. *Estudios sociales. Revista de alimentación contemporánea y desarrollo regional*, vol. 28, no. 52 (2018) doi: 10.24836/es.v28i52.574
4. Fuentes, C. M., Peña, S., Hernández, V.: La medición multidimensional de la pobreza a nivel intraurbano en Ciudad Juárez, Chihuahua (2012). *Estudios fronterizos*, vol. 19 (2018) doi: 10.21670/ref.1801001
5. García, J., Molina, J. M., Berlanga, A., Patricio, M.Á., Bustamante, Á. L., Padilla, W. R.: Ciencia de datos: Técnicas analíticas y aprendizaje estadístico. Alfaomega (2018)
6. Santiago-Rubio, I., Mora-Gutiérrez, R., Montes, O. E., Rincón, G. E. A., de los Cobos, S. G., Velazquez, P. L., Gutiérrez, A. M. A.: Analysis of poverty through educational lag using the maximum clique into the complex. In: Rodríguez García, M. D. P., Cortez-Alejandro, K.A., Merigó, J. M., Terceño-Gómez, A., Sorrosal-Forradas, M. T., Kacprzyk, J. (eds) *Digital Era and Fuzzy Applications in Management and Economy. XX SIGEF 2021. Lecture Notes in Networks and Systems*. Springer, Cham, vol. 384, pp. 97–108 (2022) doi: 10.1007/978-3-030-94485-8\_8
7. Li, X., Wu, X., Xu, S., Qing, S., Chang, P. C.: A novel complex network community detection approach using discrete particle swarm optimization with particle diversity and mutation. *Applied Soft Computing*, vol. 81 (2019) doi: 10.1016/j.asoc.2019.05.003
8. CONEVAL: Metodología para la medición multidimensional de la pobreza en México, 3rd ed. Consejo Nacional de Evaluación de la Política de Desarrollo Social. (2019)
9. Moayedikia, A.: Multi-objective community detection algorithm with node importance analysis in attributed networks. *Applied Soft Computing*, vol. 67, pp. 434–451 (2018) doi: 10.1016/j.asoc.2018.03.014
10. Said, A., Abbasi, R. A., Maqbool, O., Daud, A., Aljohani, N. R.: CC-GA: A clustering coefficient based genetic algorithm for detecting communities in social networks. *Applied Soft Computing*, vol. 63, pp. 59–70 (2018) doi: 10.1016/j.asoc.2017.11.014
11. Wang, S., Liu, J.: Community robustness and its enhancement in interdependent networks. *Applied Soft Computing*, vol. 77, pp. 665–677 (2019) doi: 10.1016/j.asoc.2019.01.045

# **Agrupamiento basado en densidad para la detección automática de hot spots delictivos en la CDMX**

Jorge Carrasco-Islas, Edgar Moyotl-Hernández

Benemérita Universidad Autónoma de Puebla,  
Facultad de Ciencias Físico Matemáticas,  
México

`jorge.carrascoi@alumno.buap.mx`, `emoyotl@fcfm.buap.mx`

**Resumen.** Este trabajo presenta la aplicación del agrupamiento basado en densidad a un conjunto de datos de delitos sexuales, cometidos en la Ciudad de México (CDMX) de 2019 a 2021, para detectar automáticamente zonas con incidencia delictiva elevada (hot spots delictivos). La metodología propuesta consiste en aplicar el método de agrupación OPTICS al conjunto de datos espaciales para localizar zonas con las agrupaciones más densas de puntos en los mapas y en estimar el área y la magnitud de incidencia delictiva encontrada en dichas zonas. Los resultados obtenidos indican que los hot spots no necesariamente coinciden con divisiones territoriales como podrían ser las alcaldías o las colonias de la ciudad. Así mismo, se observó que ciertos hot spots se repiten o presentan desplazamientos menores en distintos años.

**Palabras clave:** Datos espaciales, puntos críticos, mapas del delito, agrupamiento, OPTICS, patrones delictivos.

## **Density-Based Clustering for Automatic Detection of Criminal Hot Spots in Mexico City**

**Abstract.** This work presents a density-based clustering approach for automatic detection of criminal hot spots in Mexico City, focusing on sexual crimes committed during the 2019 – 2021 period. The proposed methodology consists of applying the OPTICS clustering algorithm to the data set in order to locate the regions with densest clusters of points on the maps and estimating the area and magnitude of criminal incidence in said regions. The results suggest that hot spots do not necessarily match with territorial divisions of Mexico City such as districts or neighborhoods. Moreover, it was observed that for different years, certain hot spots consistently reappeared or presented minor position shifts.

**Keywords:** Spatial data, hot spots, crime maps, clustering, OPTICS, crime patterns.

## 1. Introducción

La violencia y la inseguridad son un problema grave en México, de acuerdo con la última edición de la ENVIPE se estima que, en 2021, a nivel nacional el 58.9 por ciento de la población de 18 años y más considera la inseguridad como el problema más importante que aqueja actualmente, le siguen el desempleo con 41.5 % y la salud con 40.2 %.

Respecto a los crímenes sexuales, por cada 8 delitos sexuales cometidos contra mujeres, hay 1 delito sexual cometido a hombres [7]. La Unicef reconoce a la violencia sexual como una de las formas de violencia más extrema que sufren las personas en especial las mujeres y las niñas. En el mismo documento la define como cualquier acto sexual cometido en contra de la voluntad de otra persona [18].

Por otro lado, en la mayoría de las ciudades, el crimen está altamente concentrado en un número pequeño de lugares generalmente conocidos como puntos críticos o puntos calientes (hot spots) del crimen, es decir, mientras que algunos lugares apenas registran incidentes, en otros se acumulan los delitos. El estudio de [11] encontró que la mitad de los delitos de cinco grandes ciudades de Colombia se concentran entre el 3 % y 5 % de las calles.

Por esta razón, una de las estrategias para resolver el problema de la inseguridad es dirigir los recursos policiales hacia estos lugares. En general, el término hot spot hace referencia a un lugar donde existe alta concentración de un fenómeno, en este caso el fenómeno delictivo<sup>1</sup>. Un hot spot delictivo es una zona donde se comete un número de delitos superior a la media de una ciudad, o una zona donde el riesgo de ser víctima de un delito supera al promedio [6].

El análisis de hot spots mediante mapas del delito se ha convertido en una técnica analítica comúnmente utilizada por la policía para identificar visualmente los lugares donde el número de delitos tiende a ser más alto, facilitando así la toma de decisiones respecto a dónde dirigir recursos [4]. En el presente trabajo se emplea el agrupamiento basado en densidad con un conjunto de datos de crímenes sexuales, cometidos en la Ciudad de México (CDMX) de los años 2019 a 2021, para la detección de hot spots delictivos.

Los datos contienen la longitud y latitud de geolocalización lo que permite localizar el lugar donde se cometió el delito y visualizar geográficamente la actividad criminal en mapas. La metodología se centra, por un lado, en aplicar el método OPTICS a los datos espaciales para determinar los patrones en el tiempo y el espacio encontrando las zonas con agrupaciones más densas de puntos en los mapas; y por otro, en estimar el área y la magnitud de incidencia delictiva presentada en dichas zonas.

El artículo está organizado de la siguiente manera. En la sección 2 se exponen los trabajos relacionados. La sección 3 presenta la propuesta de solución donde se explica la metodología que se siguió. En la sección 4 se describen las pruebas con el modelo propuesto. Posteriormente, en la sección 5 se analizan y discuten los resultados obtenidos de los experimentos. Finalmente, la sección 6 muestra las conclusiones y el trabajo futuro.

<sup>1</sup> También existe el término cold spot o punto frío que se refiere a sitios con baja concentración de un fenómeno.

## **2. Trabajo relacionado**

Desde que se creó el primer mapa de hot spots criminales en [15] (citado por [1]) se concluyó que los crímenes no se distribuían de manera homogénea a lo largo de las ciudades, sino que, se concentraban en ciertas zonas; también afirmaron que el crimen está determinado por las características generales de la estructura social (factores socioeconómicos, espaciales y poblacionales).

Más tarde, la revisión sistemática de [3] que incluyó 19 estudios, 17 de los cuales se realizaron en Estados Unidos, uno en Australia y otro en Argentina, demostró una reducción general de la delincuencia cuando se implementaron acciones policiales en lugares de alta actividad criminal; la revisión también sugiere que invertir los recursos humanos limitados de las policías en el control de hot spots evitará la delincuencia en estas áreas y las áreas circundantes, reduciendo la delincuencia en su conjunto.

Sin embargo, los resultados del análisis en [12] evidenciaron diversos factores que evitaron que el modelo de seguridad ciudadana basada en los hot spots redujera los niveles de delincuencia en la ciudad de Bogotá, además, señala que los resultados obtenidos (buenos o malos) de la aplicación del modelo dependen en gran medida de la forma en que se implementen las tácticas policíacas en las zonas críticas.

En el caso de México, en [5] se realizó un análisis exploratorio de datos espaciales para conocer la distribución geográfica de los delitos pecuniarios<sup>2</sup> en la Zona Metropolitana del Valle de México donde explorando distintos factores se corroboró la relación de las variables socioeconómicas y urbanas con este tipo de crímenes.

Adicionalmente, los mapas comprobaron que los crímenes no se distribuyen de forma aleatoria en el espacio por lo que se pueden identificar patrones espaciales (hot spots y cold spots). Para verificar la significancia de los clústers encontrados se hizo un análisis de indicadores locales de asociación espacial (LISA) para la variable de delitos pecuniarios y un modelo espacial de regresión para conocer las variables que influyen en los delitos cometidos.

Por su parte, en [8] se identificaron hot spots de delitos callejeros en la CDMX mediante el uso de la técnica de análisis espacial densidad de kernel encontrando patrones espacio temporales del delito callejero; especialmente las zonas críticas, se asocian con la actividad comercial y laboral de la centralidad de la ciudad, pues en la periferia, en promedio, se presenta una menor actividad de este tipo de hechos.

De igual manera, los horarios de incidencia se relacionan con la salida y regreso a casa, es decir, por las mañanas y por la noche. Otro trabajo, es el de [9], el cual identificó patrones de agrupamiento espacial (hot spots) de las denuncias por violencia familiar en Ciudad Victoria (del estado de Tamaulipas en México) e ilustró su relación con las desventajas del vecindario.

También, desarrolló un análisis mediante las técnicas: densidad de kernel, promedio del vecino más cercano, indicadores locales de autocorrelación espacial y regresión logística; confirmando que la cantidad de ocupantes por vivienda y los hogares con jefatura femenina tienen una relación positiva con la probabilidad de estar en un hot spot de violencia familiar.

---

<sup>2</sup> Los crímenes pecuniarios son los que se realizan para obtener una retribución económica [5].

En contraste con los enfoques presentados en los trabajos citados, este trabajo, se centra en la aplicación del aprendizaje automático (machine learning, en Inglés) para la detección de hot spots, específicamente del aprendizaje no supervisado donde el algoritmo por sí mismo intentará aprender de los datos con el fin de encontrar patrones o tendencias.

Una categoría del aprendizaje no supervisado es el agrupamiento o clustering cuyo objetivo es encontrar cómo agrupar naturalmente los datos de forma que los elementos del mismo grupo sean más similares entre ellos que aquellos elementos de diferentes grupos [13]. Se experimenta con el método de agrupamiento OPTICS, basado en densidad, para determinar patrones de delitos sexuales: acoso sexual, abuso sexual y todos los tipos de violación sexual, cometidos en la CDMX en el periodo 2019-2021.

### 3. Propuesta metodológica

Entre los métodos de agrupamiento, los algoritmos basados en densidad enfocan el problema de dividir el conjunto de datos en grupos o clusters teniendo en cuenta la distribución de densidad de los puntos. Estos métodos son importantes debido a su capacidad para encontrar grupos de formas arbitrarias.

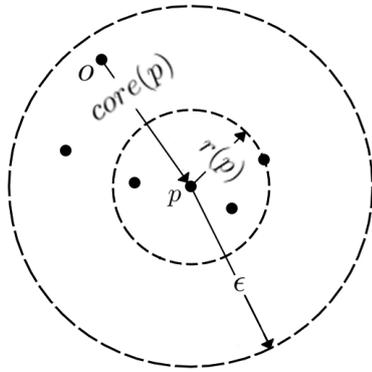
DBSCAN (Density Based Spatial Clustering of Applications with Noise) fue el primer algoritmo de agrupamiento basado en densidad [10], su idea básica es: dado un conjunto de puntos en algún espacio, agrupa los puntos que forman una región densa en un cluster, marcando como puntos de ruido los puntos que se encuentran en regiones de baja densidad. El método OPTICS es una generalización o mejora de DBSCAN que aborda una de sus principales debilidades que es el problema de detectar agrupaciones significativas en datos de densidad variable.

#### 3.1. Método de agrupamiento OPTICS

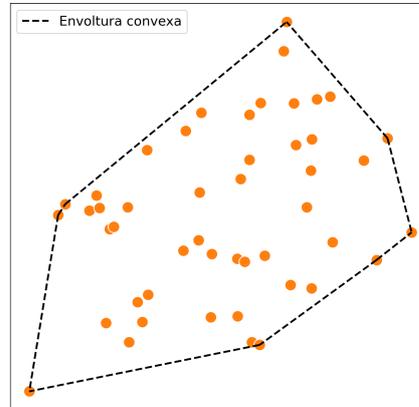
El algoritmo OPTICS (Ordering Points To Identify the Clustering Structure), presentado en [2], genera un orden de los grupos (clusters) basado en la densidad, en el cual los grupos con mayor densidad están completamente contenidos en otros grupos menos densos. Este algoritmo recibe dos parámetros,  $\maxEps$  y  $\minPts$ , los cuales representan la distancia máxima de generación de grupos y el número mínimo de puntos que debe contener la vecindad de radio  $\epsilon < \maxEps$  con centro en un punto  $p$ , como se puede ver en la figura 1, para que este tenga posibilidad de ser parte de un grupo.

Lo que diferencia a este algoritmo de otros métodos de agrupación es que no asigna etiquetas de grupo a los puntos, en cambio, almacena el orden en el que los puntos son procesados y dos características clave de cada punto  $p$ : la distancia a sus puntos vecinos ( $core(p)$ ) y la distancia de alcance ( $r(p)$ ), las cuales indican qué tan condensados están los vecinos de  $p$  y qué tan cerca está  $p$  de su predecesor  $o$ , respectivamente (ver figura 1). Esta información después puede ser utilizada para generar grupos de forma automática.

En [2] también se propone un método para agrupar los puntos de manera automatizada detectando fronteras tentativas de inicio y fin del cluster usando la distancia de alcance y el orden generado por OPTICS.



**Fig. 1.** Distancias a los vecinos y de alcance con  $\text{MinPts} = 4$ .



**Fig. 2.** Envoltura convexa de un conjunto de puntos.

En este algoritmo se introduce un nuevo parámetro,  $\xi$ , el cual controla la estrictez de las fronteras entre grupos y entre grupo y ruido. La implementación del lenguaje Python de OPTICS disponible en la biblioteca scikit-learn [16] implementa esta forma de agrupación y recibe los parámetros `min_samples`, `max_eps` y  $\xi$ , entre otros, donde `min_samples` es equivalente a  $\text{MinPts}$ .

En este trabajo, se eligió el algoritmo de agrupamiento OPTICS porque es muy versátil y eficaz al determinar los grupos, puesto que no asume que los clusters tengan una forma predefinida y toma en cuenta la posible existencia de ruido, el cual comúnmente se presenta en conjuntos de datos reales [2].

Así mismo, existen técnicas de visualización basadas en el orden generado por OPTICS y las distancias de alcance, las cuales permiten visualizar claramente los resultados obtenidos.

### 3.2. Concepto de hot spot

En [6] se define a un hot spot delictivo como una zona geográfica cuya incidencia delictiva es mayor a la del promedio, o una zona donde las personas tienen un riesgo de victimización más alto que la media. También sugiere la existencia de puntos fríos (cold spots) o zonas con menor cantidad de crímenes que el promedio y señala que algunos hot spots pueden ser más calientes que otros, es decir, varían en qué medida por encima de la media están.

El concepto de hot spot en el contexto de la detección de zonas altamente delictivas puede entenderse como una zona donde el número de casos por unidad de área sea mayor al número de casos por área promedio. En trabajos previos, este enfoque suele ser utilizado en conjunto con la división territorial de un mapa para determinar las zonas con mayor actividad criminal [5, 8, 9].

A pesar de que los hot spots obtenidos a partir de una división territorial predeterminada pueden ser útiles para determinar zonas de alta incidencia delictiva, estos están restringidos por la misma división de donde fueron producidos.

En teoría, es completamente posible que existan zonas geográficas de formas no convencionales con un mayor número de casos por área que el promedio, las cuales serían pasadas por alto si se presupone, explícita o implícitamente, que los hot spots tienen relación con una división predefinida del mapa.

Al aplicar OPTICS a un conjunto de coordenadas geográficas de casos de un cierto conjunto de delitos se obtienen grupos o clusters separados por zonas de ruido. Los grupos obtenidos mediante este procedimiento pueden tener una forma arbitraria dado que son detectados a través del cambio en la densidad de los puntos en el mapa.

Así mismo, se puede estimar el área que abarca cada grupo al calcular la envoltura convexa de dicho conjunto de puntos, esto es, al determinar el polígono que contiene a todos los puntos del cluster y cuyos vértices también pertenecen al cluster en cuestión, como se ilustra en la figura 2. Para efectuar el cálculo del área de la envoltura convexa se utilizó el módulo `scipy` [14] de Python, en especial la clase `ConvexHull`.

Al estimar el área de un cluster con un polígono, puede estimarse su número de casos por área y compararse con el número total de casos entre el área total del mapa para precisar si esta zona cumple con la definición convencional de hot spot anteriormente mencionada.

Desde esta perspectiva, si se define el número de casos por unidad de área del cluster como  $\alpha$  y al número total de casos entre el área de estudio total como  $\beta$ , el cociente  $\alpha/\beta$  es un indicador de la incidencia delictiva de la zona, puesto que si este es menor a uno, entonces la zona tiene una incidencia delictiva menor al promedio, si es igual a uno, entonces tiene una incidencia delictiva promedio y si es mayor a uno, entonces tiene una incidencia delictiva mayor al promedio, o dicho de otra manera, es un hot spot.

Además de ser un indicador de la incidencia delictiva, el cociente  $\alpha/\beta$  también indica la magnitud del hot spot, la cual puede interpretarse como el número de veces en que la incidencia delictiva del cluster supera al promedio. Esta medida no solo permite detectar los hot spots, también permite compararlos entre sí y clasificar su gravedad o prioridad.

## 4. Experimentación

### 4.1. Área de estudio

El área de estudio en el que se desarrolla este trabajo corresponde a la Ciudad de México (CDMX), la entidad federativa con la mayor densidad demográfica. La CDMX, capital del país, ocupa una superficie de 1495 km<sup>2</sup>, equivalente al 0.1 por ciento del territorio total nacional y se estima que en ella existen más de 2.6 millones de viviendas en las que residen casi 9 millones de personas, esto significa que aproximadamente 8 de cada 100 viviendas de las que hay en el país se ubican en esta ciudad y 7 de cada 100 habitantes residen también aquí [17].

La ciudad también cuenta con el número más alto de policías por habitante del país (9.6 policías por cada mil habitantes). La ciudad de México está dividida en dieciséis demarcaciones territoriales.

**Tabla 1.** Delitos sexuales geolocalizados cometidos por año.

Tipo de delito	2019	2020	2021	Total
Abuso sexual	3,451	2,758	3,162	9,371
Acoso sexual	831	934	1,219	2,984
Violación	1,263	1,366	1,462	4,091
Violación equiparada	239	388	516	1,143
Tentativa de violación	16	18	29	63
Violación tumultuaria	32	22	16	70
Violación equiparada por conocido	21	21	17	59
Violación equiparada y robo de vehículo	3	2	4	9
Violación tumultuaria equiparada por conocido	1	0	0	1
Violación tumultuaria equiparada	1	1	1	3

#### 4.2. Conjuntos de datos

Para los experimentos se utilizaron los datos de las víctimas de los delitos en las carpetas de investigación de la Fiscalía General de Justicia (FGJ) de la Ciudad de México<sup>3</sup>, datos con coordenadas geográficas (longitud y latitud) de crímenes cometidos en las 16 delegaciones de la CDMX en los años 2019, 2020 y 2021<sup>4</sup>.

De acuerdo con los datos, en dicho periodo se cometieron 683,129 delitos del fuero común y de estos hechos 20,742 fueron clasificados como delitos sexuales: acoso, abuso y violación sexual. Sin embargo, el número de crímenes sexuales con coordenadas geográficas disponibles, utilizados en los experimentos, es de sólo 17,794.

La Tabla 1 muestra que en todos los años el delito que más se cometió fue el abuso sexual con un 52.66 % del total, en segundo puesto se encuentran los delitos de violación con un 22.99 % y en tercer lugar los delitos de acoso sexual con un 16.77 %. Es importante señalar que el delito de violación se clasifica en varios tipos los cuales representan el 7.58 % del total de los crímenes de este tipo cometidos en la CDMX.

Respecto a las alcaldías, el número total de delitos de tipo sexual denunciados en cada una de las delegaciones en el periodo 2019-2021 es el siguiente: Iztapalapa (IZTP), primer lugar, con 2,892; luego Cuauhtémoc (CUAU) que tiene 2,341 casos; sigue Gustavo A. Madero (GAM) con 2,090; Álvaro Obregón (AO) 1,284; Tlalpan (TLAL) 1,194; Venustiano Carranza (VC) 1,124; Coyoacán (COYO) 1,073; Miguel Hidalgo (MH) 939; Benito Juárez (BJ) 932; Xochimilco (XOCH) 848; Iztacalco (IZTC) 724; Tláhuac (TLAH) 658; Azcapotzalco (AZ) 655; Magdalena Contreras (MC) 411; Cuajimalpa (CUAJ) que suma 355 y Milpa Alta (MA) registra 274.

La información de las carpetas de investigación también permite calcular la frecuencia o número de casos ocurridos, a nivel de demarcación, de los distintos tipos de crímenes sexuales.

<sup>3</sup> Actualización del 23 de marzo del 2022, disponibles en el Portal de Datos Abiertos de la CDMX en <https://datos.cdmx.gob.mx>

<sup>4</sup> Sólo se consideran estos años porque son los únicos que tienen un registro completo de actos delictivos, los años del 2018 hacia atrás tienen menos de 575 eventos y el 2022 es el año en curso.

En el mapa de calor mostrado en la figura 6b se puede observar la frecuencia de incidencia para el delito de abuso sexual, de acuerdo con la intensidad del color, para todo el periodo 2019-2021. Con este mapa se puede ver que la frecuencia es especialmente alta (polígonos color rojo y naranja) en Iztapalapa, Cuauhtémoc y Gustavo A. Madero, considerando que la ocurrencia promedio de este tipo de delito es de 585.6 casos por alcaldía.

Del otro extremo, en Magdalena Contreras, Cuajimalpa y Milpa Alta, la frecuencia es baja (polígonos color azul oscuro) e incluso es inferior al promedio. Cabe señalar que los datos utilizados poseen un sesgo, pues no reflejan la situación real del delito en la CDMX, ya que provienen de los delitos denunciados en los Ministerios Públicos y no tienen en cuenta los delitos cometidos que no se denuncian o denuncias que no derivan en el inicio de carpetas de investigación.

En [7] se estima que la cifra negra de delitos no denunciados a nivel nacional en 2020 fue de 93.3 por ciento, aproximadamente sólo se denuncian 7 de cada 100 delitos. Entre las razones para no denunciar destaca la pérdida de tiempo con 33.9 % y la desconfianza en las autoridades con 14.2 %.

### 4.3. Pruebas

Los resultados obtenidos al aplicar OPTICS al conjunto de datos de coordenadas geográficas filtrado por el tipo de delito pueden visualizarse en dos gráficas: un mapa en el que se puede observar la división territorial, los clusters encontrados y sus respectivas envolturas convexas y, la gráfica de alcance correspondiente, en la cual se resaltan los clusters con distintos colores para diferenciarlos.

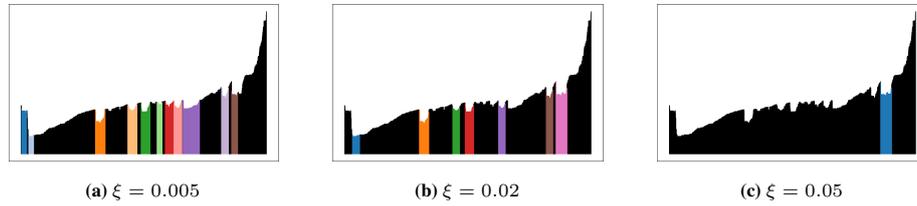
La figura 5 muestra los resultados para casos de acoso sexual en el año 2020. La gráfica de alcance representa la estructura generada por OPTICS en el orden en que se procesaron los puntos; el eje de las abscisas indica el orden de procesamiento de los puntos y el eje de las ordenadas representa la distancia de alcance.

De acuerdo con [2], en esta gráfica los clusters pueden visualizarse como muescas o depresiones, las cuales, mientras más pronunciadas sean, indican una mejor definición del cluster asociado. Del mismo modo, mientras más bajo sea el alcance de los puntos de un determinado cluster, este es más denso.

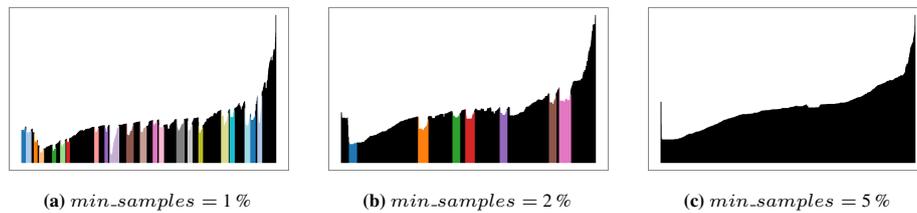
Para poder realizar pruebas con los datos se deben asignar valores específicos a los parámetros de OPTICS, es decir, se deben especificar los valores de  $\min\_samples$ ,  $\max\_eps$  y  $\xi$ . Se utilizará el valor por defecto de  $\max\_eps$  indicado en la documentación de [16], indefinido, el cual es el valor más adecuado según [2] y [16] si no se tiene un conjunto de datos muy grande.

Dado que  $\xi$  representa la estrictez de las fronteras entre clusters, se hicieron pruebas con diferentes valores para  $\xi$ : 0.005, 0.01, 0.02, 0.03, 0.04 y 0.05, en las cuales se encontró que un valor bajo de  $\xi$  detecta grupos pobremente definidos mientras que un valor alto solamente detecta grupos extremadamente bien definidos, situación poco común al tratar con datos reales.

Las observaciones anteriormente mencionadas, ilustradas en la figura 3 para delitos de acoso sexual en el 2020, coinciden con lo encontrado en [2], donde se muestra una aplicación de OPTICS en un conjunto de datos real en la que se utiliza el valor  $\xi = 0.02$  para encontrar clusters razonablemente bien definidos.



**Fig. 3.** Gráficas de alcance con distintos valores de  $\xi$  para acoso sexual en 2020.



**Fig. 4.** Gráficas de alcance con distintos valores de  $\text{min\_samples}$  para acoso sexual en 2020.

Por las razones anteriormente mencionadas se usará  $\xi = 0.02$  para obtener los resultados finales. En cuanto al valor de  $\text{min\_samples}$ , se cree que lo más adecuado es asignar un valor en proporción al número de observaciones en el conjunto de datos por analizar, de manera que se obtengan resultados comparables entre los diferentes conjuntos de delitos estudiados; también, se debe notar que el tamaño mínimo de cluster es mayor o igual al valor de  $\text{min\_samples}$  utilizado, por lo cual  $\text{min\_samples}$  no debe tener un valor muy alto si se quiere evitar pasar por alto algunos hot spots menores, es decir, zonas con una razón de delitos por área no muy alta.

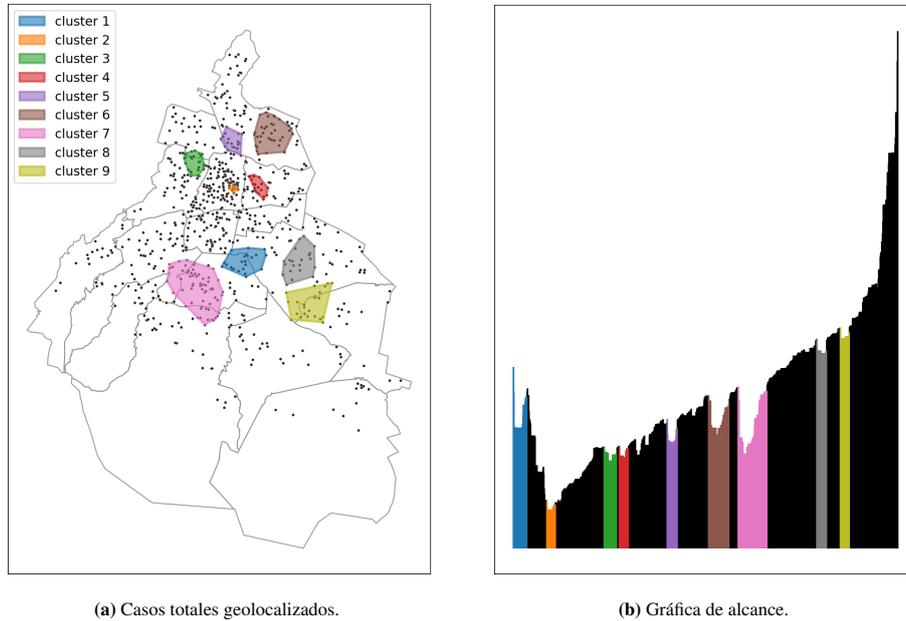
En este trabajo se usó el 2% del total de las observaciones como valor para  $\text{min\_samples}$  debido a que este valor fue el que dio mejores resultados entre los valores 1%, 2%, 3%, 4% y 5%, como se ilustra en la figura 4, de nuevo, para delitos de acoso sexual en el año 2020.

## 5. Análisis y resultados

En la figura 5a se puede apreciar la geolocalización de los casos de acoso sexual reportados en el año 2020 junto con las envolturas convexas de los clusters encontrados sobre el mapa de las alcaldías de la Ciudad de México, mientras que la gráfica de alcance correspondiente puede observarse en la figura 5b.

Los cocientes  $\alpha/\beta$  de los clusters 1 al 9 de la figura 5 son: 2.4037, 39.8669, 5.6725, 5.6116, 1.7247, 2.9072, 2.1555, 1.5048 y 1.2387, respectivamente, por lo que todos los clusters también son clasificados como hot spots.

Como puede verse en la figura 5a, a simple vista es difícil determinar la existencia de hot spots y delimitar sus respectivas fronteras. Por ejemplo, se puede observar que la región de puntos alrededor del cluster 2 de la figura 5 es más densa que cualquier otra zona del mapa, sin embargo, no toda la región es detectada como hot spot.



**Fig. 5.** Resultados de la metodología propuesta en casos de acoso sexual del 2020.

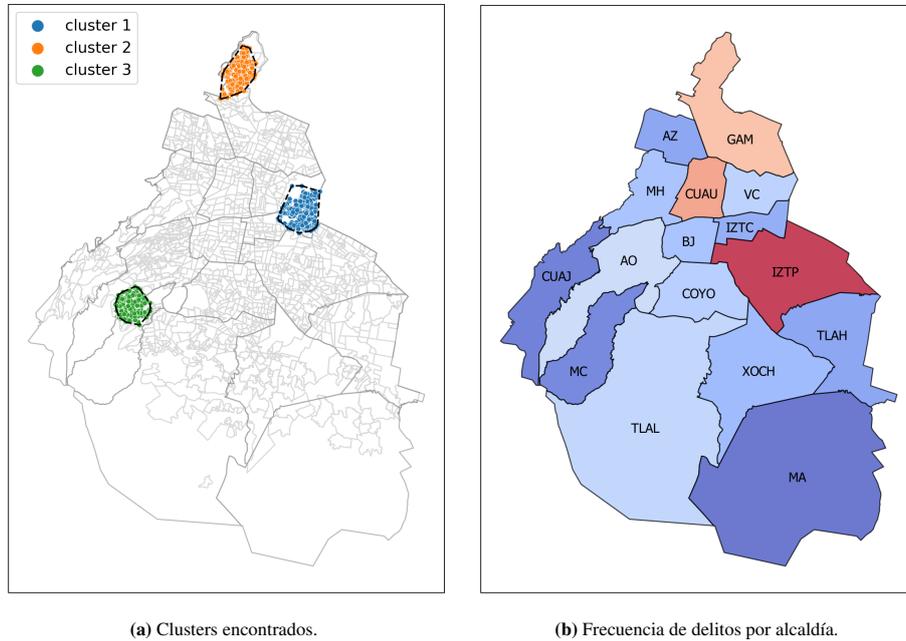
Esta situación ilustra otra característica de los hot spots encontrados mediante la metodología propuesta, los casos que componen la región del hot spot tienden a agruparse significativamente en comparación con sus alrededores, propiedad que no es posible detectar al usar solamente la noción clásica de hot spot.

Las figuras 5a y 5b muestran que aunque los datos sean difíciles de tratar y visualmente no se detecten posibles hot spots, mientras la estructura de los datos en la gráfica de alcance presente muescas relativamente bien definidas y del tamaño mínimo, entonces los datos tienden a agruparse en ciertas zonas del mapa.

De igual manera, si la estructura de la gráfica de alcance no presenta las características anteriormente descritas, entonces la metodología propuesta no detectará indiscriminadamente cualquier hendidura como hot spot.

Con relación a los casos de abuso sexual reportados en los años 2019, 2020 y 2021 se presentan las figuras 6a, clusters geolocalizados en un mapa de la CDMX, y 6b, un mapa de calor de la frecuencia de delitos por alcaldía. Los valores  $\alpha/\beta$  de los clusters 1, 2 y 3 son: 1.9818, 1.3877 y 1.5619, respectivamente, indicando que los clusters encontrados efectivamente son hot spots.

El cluster 1 (en color azul), correspondiente al 4.11 % de los casos, se presenta en la zona este de las alcaldías Venustiano Carranza e Iztacalco; el cluster 2 (en color naranja), correspondiente al 2.29 % de los casos, está completamente contenido en la alcaldía Gustavo A. Madero y localizado en la zona norte de dicha alcaldía; finalmente, el cluster 3 (en color verde) corresponde al 2.08 % de los casos reportados y la región abarcada por este atraviesa la frontera entre las alcaldías Álvaro Obregón y Magdalena Contreras.



**Fig. 6.** (a) Resultados de aplicar OPTICS y (b) Mapa de calor de delitos por alcaldía, ambos, para casos de abuso sexual reportados en 2019, 2020 y 2021.

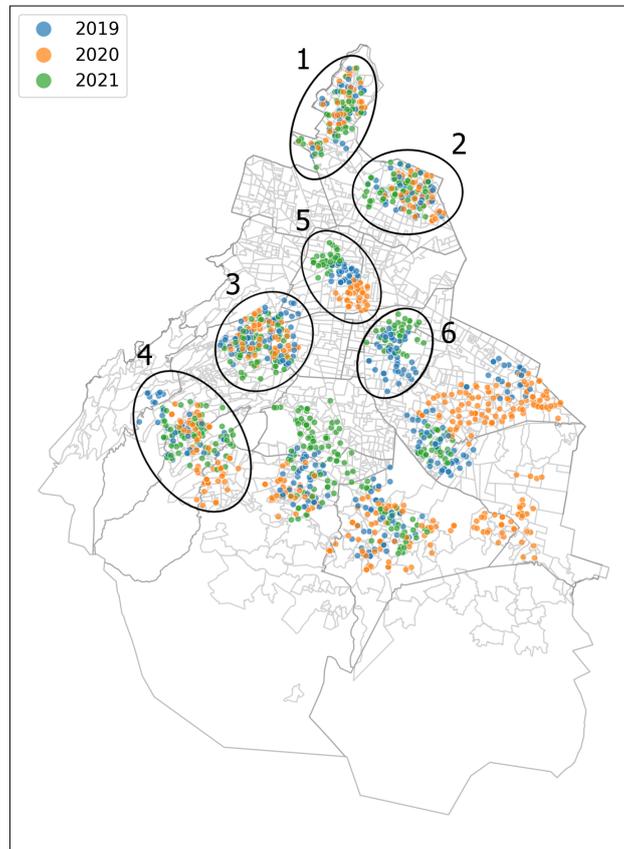
La figura 6 muestra un escenario en el cual la metodología propuesta podría no detectar hot spots pobremente definidos pero altamente densos, como lo son las alcaldías Cuauhtémoc e Iztapalapa, claramente reconocibles en la figura 6b por sus colores naranja oscuro y rojo, respectivamente.

Así mismo, un mapa de calor no es capaz de identificar las aglomeraciones de casos encontradas por OPTICS entre delegaciones, como los clusters 2 y 3, puesto que estas regiones son fragmentadas por la división territorial predefinida de las alcaldías.

Este resultado sugiere que el uso de OPTICS para detectar hot spots se beneficiaría de un contraste con un análisis tradicional para lograr considerar un panorama más amplio del contexto de los datos. De igual manera, la situación ilustrada por la figura 6 muestra que el enfoque menos restrictivo de la metodología propuesta al detectar hot spots ilumina zonas de interés posiblemente pasadas por alto al efectuar únicamente un análisis más tradicional.

Por otra parte, la figura 7 presenta una aplicación interesante del procedimiento propuesto en este trabajo, incorporando el factor temporal al problema de detectar hot spots y zonas de interés. En esta figura se muestran los clusters encontrados al aplicar OPTICS a los casos de violación y sus variantes en cada año por separado, tiñendo los hot spots de un color distinto para cada uno de ellos, azul para 2019, naranja para 2020 y verde para 2021.

Las parejas (min, max) de valores  $\alpha/\beta$  para los clusters encontrados en cada año son: (1.3802, 8.7994), (1.1024, 6.0884), y (1.2885, 6.6621) para 2019, 2020 y 2021, respectivamente, confirmando que todos los clusters encontrados son hot spots.



**Fig. 7.** Gráfica de dispersión con clusters sobrepuestos encontrados al aplicar OPTICS a casos de violación y sus variantes en 2019, 2020 y 2021 por separado.

Las regiones resaltadas y enumeradas en la figura 7 ejemplifican tres distintas evoluciones de interés que un hot spot podría exhibir al paso del tiempo. Los hot spots de las regiones 1, 2 y 3 son muy similares para los tres años de estudio, razón por la cual estos hot spots podrían considerarse recurrentes; la repetición de eventos en los mismos lugares indica que las zonas que concentran la mayor incidencia de violencia sexual en la ciudad permanecen cómo zonas peligrosas durante los tres años.

Las regiones 4 y 5 ilustran lo que se conoce como un desplazamiento, el cual puede ser más o menos pronunciado; en el caso del hot spot de la región 5, el desplazamiento es fácilmente distinguible incluso a simple vista, este desplazamiento significa que los crímenes de violación se han dispersado o distribuido a otras zonas de la CDMX que antes no registraban este tipo de delito.

La situación presentada en la región 6 es el resurgimiento de un hot spot mínimamente desplazado, este se detecta en el año 2019, en el año 2020 desaparece y en el año 2021 reaparece ligeramente más al norte.

Las interpretaciones de las tres situaciones anteriormente mencionadas pueden ser variadas dependiendo del contexto en el que se presenten, no obstante, posibles explicaciones para estos fenómenos podrían ser la falta de estrategias para asignar recursos a zonas cuya incidencia delictiva sea mayor al promedio, la adaptación de los criminales a la estrategia de seguridad pública vigente y el descuido de regiones previamente consideradas de atención después de reducir la actividad delictiva exitosamente.

## **6. Conclusiones y trabajo futuro**

Los resultados mostrados confirman que los delitos no se distribuyen uniformemente en el espacio y que, en el periodo de estudio, las denuncias por violencia sexual presentan un patrón de agrupamiento espacial.

Estos hallazgos son importantes porque el estudio se ha abordado desde la perspectiva del aprendizaje automático no supervisado a través de técnicas de agrupamiento basado en densidad. Aunque no sea un enfoque frecuentemente utilizado, en este trabajo se obtuvieron buenos resultados al aplicar el algoritmo de agrupamiento OPTICS con los valores de los parámetros elegidos.

Se comprobó que los hot spots encontrados mediante la metodología propuesta presentan las características deseadas por la definición clásica de hot spot así como una agrupación significativa de los casos en comparación con sus alrededores. También se encontró que un estudio híbrido entre los enfoques tradicional y de aprendizaje automático podría ser más informativo que cualquiera de los dos por separado, ya que cada uno obtiene resultados con una interpretación distinta y complementaria a los del otro tipo de estudio.

El trabajo actual se concentró en explorar las características espaciales de los delitos sexuales en la CDMX. No obstante, los factores que influyen en la espacialidad del fenómeno delictivo requiere un análisis más detallado en un trabajo futuro. Así mismo, a futuro, se buscarán más fuentes de información para considerar no sólo la cifra oficial de denuncias sino también la cifra negra.

Finalmente, la información obtenida podría ser evaluada por los departamentos de policía y servir como apoyo para tomar decisiones más informadas en materia de seguridad pública, permitiendo optimizar la asignación de recursos humanos y financieros; asimismo, la metodología propuesta se podría utilizar para detectar hot spots de interés con formas no convencionales y dar seguimiento a la evolución de dichas zonas al paso del tiempo.

## **Referencias**

1. Anderson, T. L.: Understanding deviance: connecting classical and contemporary perspectives. pp. 106–127 (2014) doi: 10.4324/9781315879635
2. Ankerst, M., Breunig, M., Kriegel, H. P.: Optics: Ordering points to identify the clustering structure. *ACM Sigmod Record*, ACM, vol. 28, no. 2, pp. 49–60 (1999) doi: 10.1145/304181.304187
3. Braga, A., Papachristos, A., Hureau, D.: Hot spots policing effects on crime. *Campbell Systematic Reviews*, vol. 8, no. 1, pp. 1–96 (2012) doi: 10.4073/csr.2012.8

4. Chaney, S., Tompson, L., Uhlig, S.: The utility of hotspot mapping for predicting spatial patterns of crime. *Security Journal*, vol. 21, no. 1, pp. 4–28 (2008) doi: 10.1057/palgrave.sj.8350066
5. Corzo, T. J. A.: Criminalidad y sus determinantes socioeconomicos en la Zona Metropolitana del Valle de México (2011-2014): Protocolo de Investigación. Facultad de Estudios Superiores Acatlán, UNAM (2018)
6. Eck, J. E., Chainey, S., Cameron, J. G., Leitner, M., Wilson, R. E.: Mapping crime: Understanding hot spots. National Institute of Justice (2005)
7. INEGI: Encuesta Nacional de Victimización y Percepción sobre Seguridad Pública (ENVIPE). Instituto Nacional de Estadística y Geografía (2021) <https://www.inegi.org.mx/programas/envipe/2021/>
8. Flores-Gutiérrez, S.: Análisis espacial del delito callejero en Ciudad de México, 2018. *Quivera Revista de Estudios Territoriales*, UNAM, vol. 23, no. 1, pp. 25–47 (2021) doi: 10.36677/qret.v23i1.15072
9. Jurado-Flores, V. D., Fuentes-Rios, F. J.: Concentración espacial de las denuncias por violencia familiar en Ciudad Victoria: un acercamiento mediante la detección de hot spots. *CienciaUAT*, vol. 16, no. 2, pp. 126–140 (2022) doi: 10.29059/cienciauat.v16i2.1551
10. Ester, M., Kriegel, H. P., Sander, J., Xu, X.: A density-based algorithm for discovering clusters in large spatial databases with noise. 2nd International Conference on Knowledge Discovery and Data Mining (KDD-96). AAAI Press, vol. 96, no. 34, pp. 226–231 (1996)
11. Mejía, D., Ortega, D., Ortiz, K.: Un análisis de la criminalidad urbana en Colombia. CAF Banco de Desarrollo de América Latina - Documento de trabajo, no. 06 (2015)
12. Ramírez, A.: Los hot spots: una estrategia fallida para Bogotá. *Desafíos*, vol. 28, no. 2, pp. 255–296 (2016) doi: 10.12804/desafios28.2.2016.07
13. Raschka, S., Mirjalili, V.: Python machine learning: Aprendizaje automático y aprendizaje profundo con Python, scikit-learn y TensorFlow. Marcombo (2019)
14. Virtanen, P., Gommers, R., Oliphant, T. E., Haberland, M., Reddy, T., Cournapeau, D., Burovski, E., Peterson, P., Weckesser W., Bright, J., et al.: SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. *Nature Methods*, vol. 17, no. 3, pp. 261–272 (2020) doi: 10.1038/s41592-019-0686-2
15. Shaw, C. R., McKay, H. D.: Juvenile delinquency and urban areas. University of Chicago Press (1942)
16. Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., et al.: Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830 (2011)
17. EVALÚA: Ciudad de México 2020: Un diagnóstico de la desigualdad socio territorial. Consejo de Evaluación del Desarrollo Social de la Ciudad de México (EVALÚA) (2020)
18. Unicef: ONU México hace un llamado a eliminar todas las formas de violencia sexual contra las mujeres y las niñas: Comunicado de prensa. Unicef México (2019)

# Clasificación de los desórdenes del sueño a través de polisomnogramas usando un algoritmo de aprendizaje profundo

Marco Antonio Rodríguez Castrejón, Leticia Ortega Máynez,  
Nelly Gordillo Castillo, José Manuel Mejía Muñoz,  
José David Díaz Román

Universidad Autónoma de Ciudad Juárez,  
Instituto de Ingeniería y Tecnología, Maestría en Ingeniería Eléctrica,  
México

al199048@alumnos.uacj.mx, {lortega, nelly.gordillo,  
jose.mejia, david.roman}@uacj.mx

**Resumen.** La presente investigación tiene como objetivo la clasificación y/o predicción de los siguientes trastornos del sueño: epilepsia nocturna del lóbulo frontal, apnea obstructiva del sueño, movimiento periódico de piernas y desorden de comportamiento REM) por medio de algoritmos de aprendizaje profundo con señales extraídas de polisomnogramas. Se consultaron las bases de datos públicas y privadas: CAP Sleep Database y Wisconsin Sleep Cohort (WSC), de las cuales se extrajeron polisomnogramas (PSG) de diez pacientes de cada trastorno además de 10 sin algún trastorno, de los cuales se utilizó las señales EEG, ECG y SaO<sub>2</sub>/SpO<sub>2</sub>. Primero se filtró el ruido de las señales por medio de un filtro FIR exceptuando la señal de oxígeno, posteriormente se segmentaron las señales en ventanas de 20 segundos para ser codificadas en imágenes por medio de funciones de transformación de series en el tiempo, las cuales alimentan a una red neuronal convolucional recurrente. Como resultado preliminar se obtuvo una precisión mayor del 90 %. Como conclusión podemos observar que esta investigación brinda una herramienta a los especialistas del área del sueño (psiquiatría, neurología, neurofisiología, etc.) para clasificar y diagnosticar los trastornos del sueño, ahorrándoles tiempo y automatizando el proceso con precisión sin embargo es necesario trabajar con más datos del PSG para obtener mejores resultados.

**Palabras clave:** Aprendizaje profundo, redes convolucionales recurrentes, apnea del sueño, epilepsia nocturna del lóbulo frontal, movimiento periódico de piernas, desorden de comportamiento REM, codificación series en el tiempo.

## Classification of Sleep Disorders through Polysomnograms Using a Deep Learning Algorithm

**Abstract.** This research aims to classify and/or predict the following sleep disorders: nocturnal lobe epilepsy frontal lobe, obstructive sleep apnea, periodic

leg movement, and REM behavior disorder) using deep learning algorithms with signals extracted from polysomnograms. Public and private databases were consulted: CAP Sleep Database and Wisconsin Sleep Cohort (WSC), from which polysomnograms (PSG) were extracted from ten patients with each disorder in addition to 10 without any disorder, of which EEG signals were used ECG and SaO<sub>2</sub>/SpO<sub>2</sub>. First, the noise of the signals was filtered by means of an FIR filter, except for the oxygen signal, then the signals were segmented into 20-second windows to be encoded in images by means of time series transformation functions, which feed to a recurrent convolutional neural network. As a preliminary result, an accuracy greater than 90% was obtained. In conclusion, we can observe that this research provides a tool for specialists in the area of sleep (psychiatry, neurology, neurophysiology, etc.) to classify and diagnose sleep disorders, saving them time and automating the process with precision. However, it is necessary to work with more PSG data for better results.

**Keywords:** Deep learning, recurrent convolutional networks, apnea of sleep, nocturnal epilepsy of the frontal lobe, movement leg newspaper, REM behavior disorder, time series coding.

## 1. Introducción

Día a día los trastornos de sueño se vuelven más frecuentes conforme el ritmo de vida se hace más acelerado, el estrés aumenta, la población crece, el clima cambia, nuevas enfermedades aparecen, se incrementan los déficits alimenticios, etc.

Este tipo de trastornos se pueden presentar desde la etapa neonatal hasta la adultez. El sueño es considerado un proceso fisiológico de vital importancia para la salud integral de los seres humanos. Además, la falta de sueño es detonante de alteraciones en el procesamiento y retención de la memoria.

Los especialistas del área se enfrentan a tener que diagnosticar diversos trastornos del sueño tales como: narcolepsia, insomnio, bruxismo, epilepsia nocturna del lóbulo frontal, apnea obstructiva del sueño, movimientos periódicos de extremidades, trastornos durante la fase REM, entre otros [1, 2].

Los cuales, para poder detectarlos y diagnosticarlos, es necesario recurrir a la obtención de polisomnografía, que es la representación de un estudio de señales de diversos canales conectados al paciente por electrodos en diferentes zonas del cuerpo mientras se está dormido, como es en la parte superior, frontal y lateral de la cabeza [3].

Todo esto en un cuarto controlado; este proceso conlleva horas de obtención de las señales y debe monitorearse en todo momento, siendo el tiempo una limitante en el estudio. Gracias al desarrollo de métodos de aprendizaje profundo [4], algunas tareas de la sociedad moderna que antes eran complejas ahora son fáciles de hacer, tales como: identificar objetos en imágenes, transcribir voz en texto, encontrar nuevos productos que simpaticen con los intereses de los usuarios, tareas de clasificación, etc.

Para realizar lo anterior, existen varias arquitecturas de redes neuronales: las multi-capas de perceptrones, que es la arquitectura más antigua y simple de redes neuronales, la CNN (convolutional neuronal network) basada en capas convolucionales, de uso muy

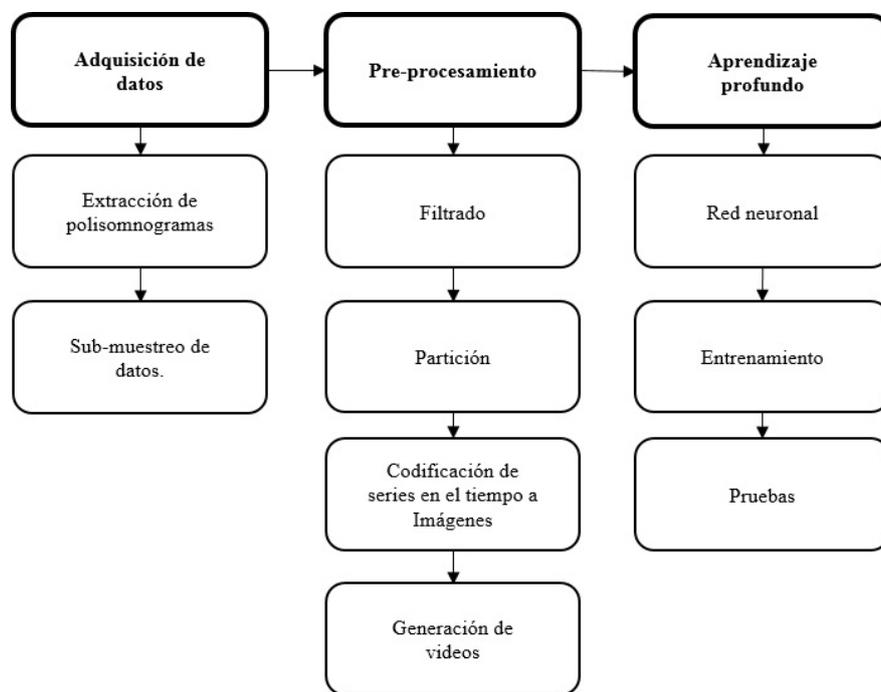


Fig. 1. Diagrama a bloques de la metodología.

común en procesamiento de imágenes y redes neuronales recurrentes usadas para datos secuenciales como texto o series en el tiempo [5].

En este proyecto se desarrollará una arquitectura de aprendizaje profundo con redes neuronales convolucionales para la clasificación de trastornos de sueño. Para el entrenamiento será necesario extraer los polisomnogramas de dos bases de datos públicas y privadas seleccionadas de la internet [6, 7, 8, 9].

Las señales digitales obtenidas serán pre-procesadas digitalmente por medio de filtros que nos ayudan a mejorar la señal final para a posteriori alimentar la red neuronal. Los propósitos de este proyecto son: brindar una herramienta a especialistas del área, para que puedan clasificar y diagnosticar los trastornos del sueño ahorrándoles tiempo y automatizando el proceso con alta precisión, así como avanzar en el conocimiento generado en el tema.

## 2. Trabajos relacionados

En 2019 Chowdhury, et al. [10] diseñaron una arquitectura de aprendizaje profundo para diagnosticar convulsiones por medio de señales EEG con una red convolucional 1-D. Está compuesta de una capa convolucional de tamaño kernel 173 con 23 filtros y pasos (strides) de 1, una capa de maxpooling de tamaño 173 y una capa densa con activación sigmoidea; la información fue dividida en segmentos representados con las

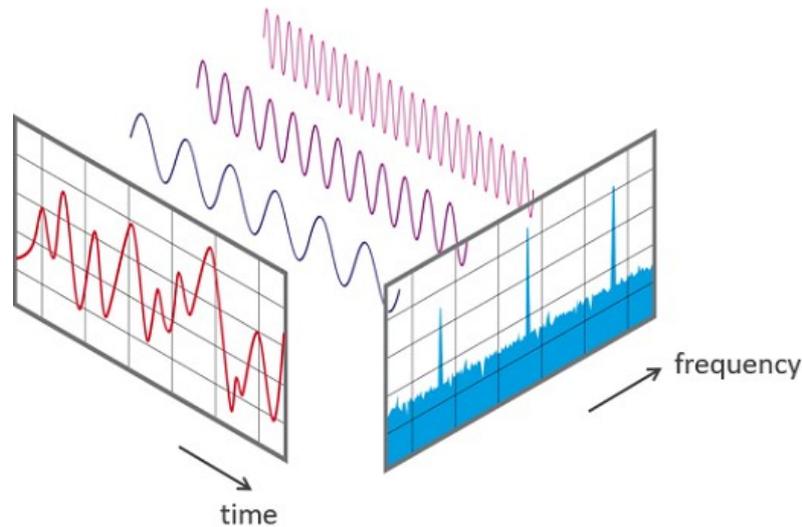


Fig. 2. Señal en el dominio del tiempo al dominio de la frecuencia [14].

primeras 5 letras del alfabeto donde A y B pertenecen a participantes con condiciones normales y C, D, E y F que presentan epilepsias en diferentes categorías.

Los resultados que obtienen para diferenciar cada grupo de clases siempre dan como resultado una precisión del  $99.4 \pm 0.52$ .

En 2020, Wickramaratne y Mahmud [?] propusieron un método de aprendizaje profundo para detectar esfuerzo respiratorio durante el sueño mediante polisomnogramas por medio de las señales EEG, ECG, EOG y SaO<sub>2</sub>, muestreadas a 200 Hz, las cuales se les aplicó un filtro FIR y fueron sub-muestreadas a 50 Hz antes de ser inyectadas a una red neuronal recurrente, la arquitectura de la red consta de capas bidireccionales LSTM con funciones de activación ELU y una capa densa distribuida en el tiempo al final.

Esta arquitectura fue aplicada en tres modelos diferentes, uno para la señal EEG y EOG, el segundo para las señales EMG y ECG y el tercero para las EEG y SaO<sub>2</sub> y al final los tres modelos fueron combinados para dar una mejor predicción. Para validar su clasificador utilizaron curvas AUPRC y AUROC las cuales dieron un resultado de 0.59 en el área debajo de la curva de precisión y 0.95 en el área debajo de la curva de recepción de características de operación.

En 2020, Pisano, et al. [12] construyeron una red neuronal convolucional para detectar epilepsia nocturna del lóbulo frontal por medio de señales EEG muestreadas a 256 Hz, la arquitectura de la red neuronal utilizada se basó en una red profunda residual compuesta de capas convolucionales, max pooling y average pooling.

Dentro de cada capa convolucional se agregó una capa de dropout de 0.20 para evitar sobreentrenamiento, y el algoritmo que se utilizó para optimizar el entrenamiento fue el gradiente descendiente estocástico con momento, obtuvieron resultados de precisión entre 0.94 y 0.96.

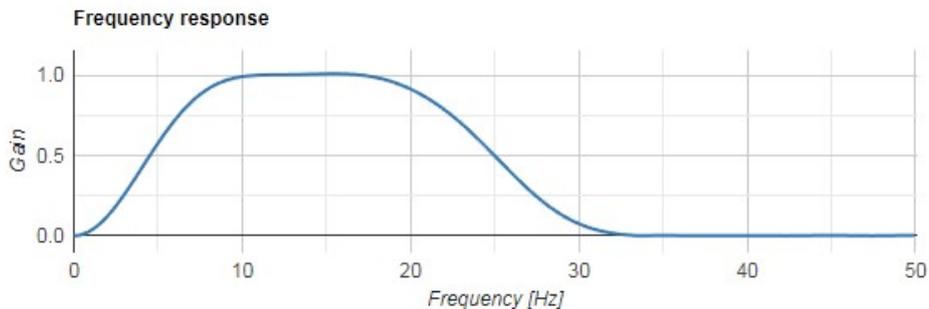


Fig. 3. Respuesta en la frecuencia del Filtro FIR para eliminar ruido en las señales.

En 2020, K. Palani, et al. [13] desarrollaron una red neuronal para detectar epilepsia por medio de imágenes generadas a partir de GASF (Gramian Angular Summation Field).

Utilizando una red convolucional pre-entrenada con los pesos de ImageNet con tres capas convolucionales con función de activación ReLU seguido de una capa de normalización lote (BN) y una capa de MaxPooling con, además, capas densas con activación sigmoidea y finalmente con una capa de clasificación SoftMax binaria para las clases normal y focal. Con este modelo ellos logran obtener una precisión de 0.885 en promedio y área bajo la curva de 0.92 los cuales son superiores a otras arquitecturas como AlexNet, VGG16, VGG19, entre otras.

### 3. Materiales y métodos

En este capítulo se mostrará la metodología empleada para la realización de esta investigación, así como las herramientas utilizadas.

#### 3.1. Modelado de los datos

Los pasos para el desarrollo de este proyecto fueron: la adquisición de señales de diferentes bases de datos públicas y privadas; pre-procesamiento de los datos para eliminar el ruido, así como convertirlos a imágenes; y el diseño de la red de aprendizaje profundo para la clasificación de los desórdenes del sueño. El diagrama a bloques de la Fig. 1 ilustra la metodología de este proyecto.

#### 3.2. Materiales

Para este proyecto basado en procesamiento digital de señales se utilizaron materiales de hardware y software.

**Hardware** La computadora que se utilizó para el desarrollo de esta investigación y con la que se realizó todo el pre-procesamiento de las señales adquiridas de las bases de datos, tiene las especificaciones que se muestran a continuación:

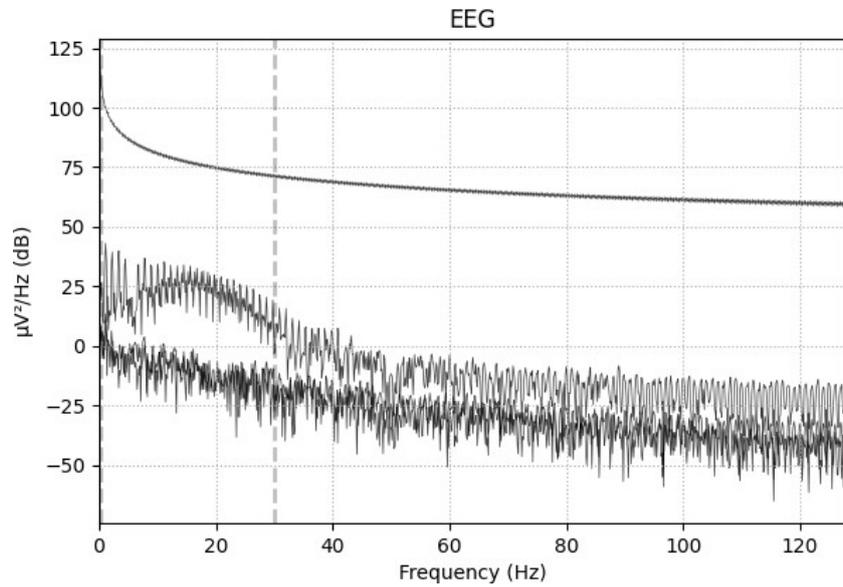


Fig. 4. Señales en el dominio de la frecuencia del polisomnograma.

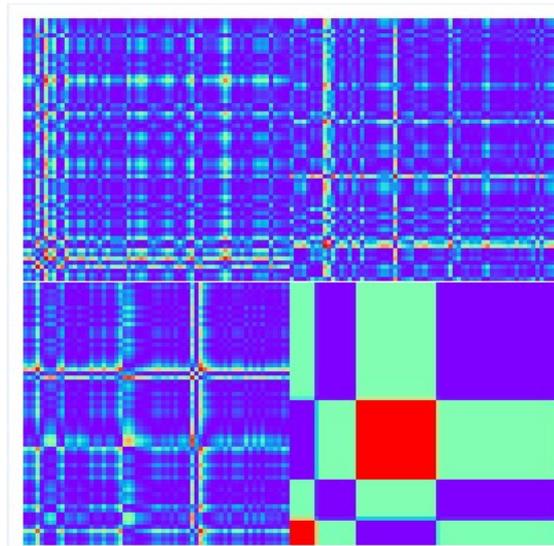


Fig. 5. Imagen resultante de la transformación GASF.

- PC Asus G75vw,
- Procesador Intel Core i7 a 2.4 GHz,
- Memoria RAM de 20 GB,
- Tarjeta de video NVIDIA GEFORCE GTX 660M de 2GB.

**Tabla 1.** Red neuronal propuesta.

Nombre	Tipo	Tamaño
Conv2D_1	Conv2D	(None, 224, 224, 64)
Conv2D_2	Conv2D	(None, 224, 224, 64)
BatchNorm_1	BatchNormalization	(None, 224, 224, 64)
MaxPool2D_1	MaxPooling2D	(None, 112, 112, 64)
Conv2D_3	Conv2D	(None, 112, 112, 128)
Conv2D_4	Conv2D	(None, 112, 112, 128)
BatchNorm_2	BatchNormalization	(None, 112, 112, 128)
MaxPool2D_2	MaxPooling2D	(None, 56, 56, 128)
GlobalMaxPool2D_1	GlobalMaxPooling2D	(None, 128)
TimeDist_1	TimeDistributed	(None, 5, 128)
GRU_1	GRU	(None, 5, 32)
GRU_2	GRU	(None, 5, 16)
Dense_1	Dense	(None, 1024)
Dense_2	Dense	(None, 512)
Dense_3	Dense	(None, 128)
Dense_4	Dense	(None, 64)
Dense_5	Dense	(None, 5)

**Software. Bases de datos** Toda la información se extrajo de bases de datos que se encuentran en la red las cuales se enumeran a continuación:

- Base de datos publica: CAP Sleep Database v1.0.0,
- Base de datos privada: Wisconsin Sleep Cohort (WSC).

**Python 3.7** El desarrollo de todo el código tanto del pre-procesamiento como el entrenamiento se realiza con el entorno de programación de Python, siendo este un recurso gratuito se ha ido popularizando en los últimos años, las librerías que se utilizaron para ello son:

- NumPy,
- Matplotlib Pyplot,
- MNE io,
- SciPy,
- YASA,
- TensorFlow,
- Keras,
- Scikit-Learn,
- Pandas.

**Google Colab** El entrenamiento de la red se vuelve una tarea que consume grandes recursos, así como tiempo de ser realizado en una computadora comercial, Google

**Tabla 2.** Parámetros de entrenamiento.

Número de participantes	10
Imágenes generadas por participante	14390
Duración del video por participante	3 min.
Videos generados por participante	959
Número de clases	5
Número de épocas	100
Índice de aprendizaje	1e-4
Optimizador	Adam
Función de costo	Categorical cross-entropy
CNN Layers/ VGG16	9
RNN Layers/GRU	3
Total de parámetros de entrenamiento	940,037
Tiempo de entrenamiento	6.58 horas

pone a disposición de los usuarios sus computadoras virtuales que cuentan con mayores recursos tecnológicos lo cual hace que la tarea de entrenamiento se vuelva más rápida, algunas de sus especificaciones de la versión Google Colab Pro son:

- GPU T4 y P100,
- Memoria RAM 25 GB,
- 24 horas continuas de entrenamiento.

### 3.3. Métodos

**Adquisición de datos** Se obtuvieron polisomnogramas de las bases de datos públicas y privadas: CAP Sleep Database v1.0.0 la cual contiene las clases: nfle (nocturnal frontal lobe epilepsy o epilepsia del lóbulo frontal), normal (sin trastorno), plm (periodic leg movement o movimiento periódico de piernas) y rbd (REM behaviour disorder o desorden de comportamiento REM); y Wisconsin Sleep Cohort (WSC) de la cual se extrajo la clase sdb (sleep disordered breathing o trastorno respiratorio del sueño).

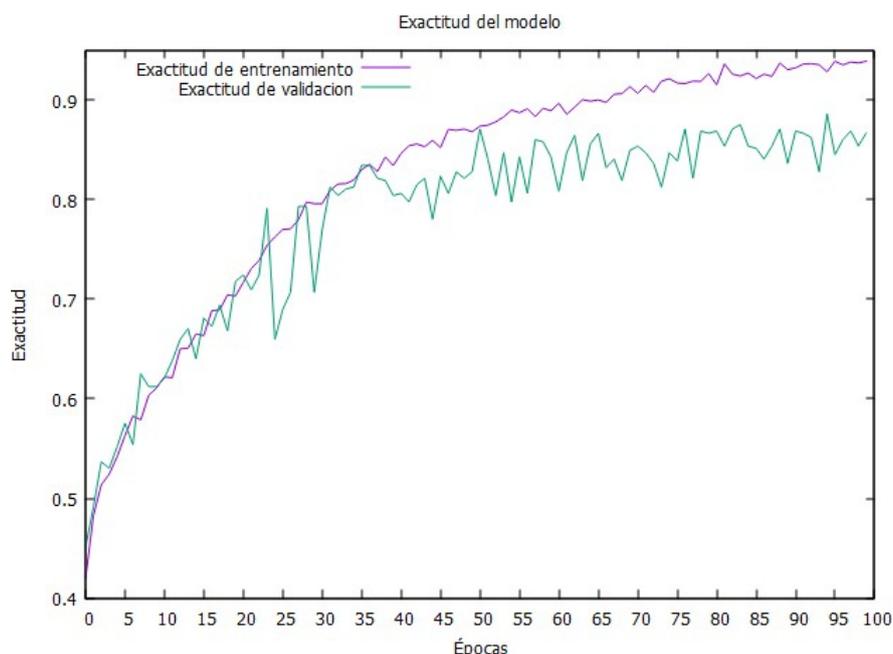
El formato para estos archivos es EDF, el cuál es un estándar europeo para almacenar polisomnogramas. De cada clase se obtuvo diez polisomnogramas de cuatro horas de sueño aproximadamente. Para acceder a ellos se usó la librería de mne de Python con la cual se seleccionó los canales EEG(dos), ECG y SaO2/SpO2 para su posterior procesamiento (Fig. 2).

**Pre-procesamiento** El pre-procesamiento se llevó a cabo de la siguiente manera:

1. Filtrado de ruido. Se analizaron las señales obtenidas en la sección anterior, pero en el dominio de la frecuencia por medio de la Transformada de Fourier rápida FFT, utilizando la función de numpy fft (ver Fig. 3), y se observa que las frecuencias más importantes están en el rango de 0.5 y 25 Hz. Se diseñó un filtro FIR con los siguientes parámetros:

**Tabla 3.** Métricas obtenidas para el modelo propuesto, promedio de precisión 0.91.

Trastorno de sueños	Precisión	Exhaustividad	Valor F	Muestras
nfle	0.86	0.90	0.88	97
normal	0.94	0.85	0.89	97
plm	0.81	0.88	0.84	97
rbd	0.93	0.90	0.91	97
sdb	0.99	1.00	0.99	97
Promedio	0.91	0.90	0.90	485

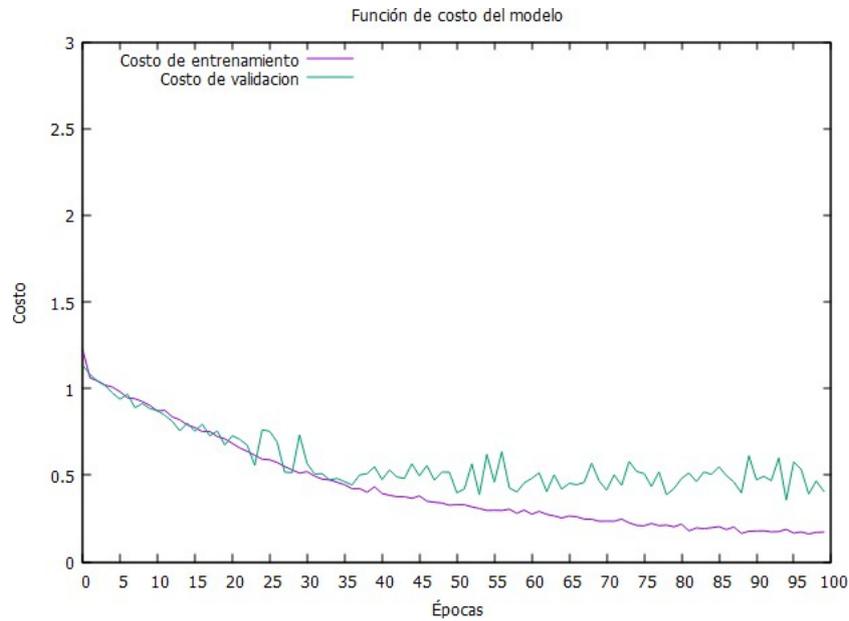


**Fig. 6.** Curva de entrenamiento, exactitud en entrenamiento de 0.93, exactitud en validación de 0.86.

- Frecuencia de corte baja: 0.5 Hz,
- Frecuencia de corte alta: 25 Hz,
- Número de coeficientes del filtro: 29,
- Tipo de ventana: Hamming.

En la Fig. 4, se observa el filtro que se diseñó para eliminar frecuencias fuera del rango deseado. Una vez creado el filtro se realizó la operación de la convolución con las señales del polisomnograma utilizando la función `signal.convolve` de la librería `scipy`.

2. Partición de datos en ventanas o épocas. Se dividieron los datos del arreglo de señales filtradas en ventanas de 20 segundos de longitud. La función con la que se generó estas ventanas fue `sliding_window` de la librería de `yasa`, con los siguientes parámetros:



**Fig. 7.** Curvas de la función de costo, 0.17 en entrenamiento y 0.40 en validación.

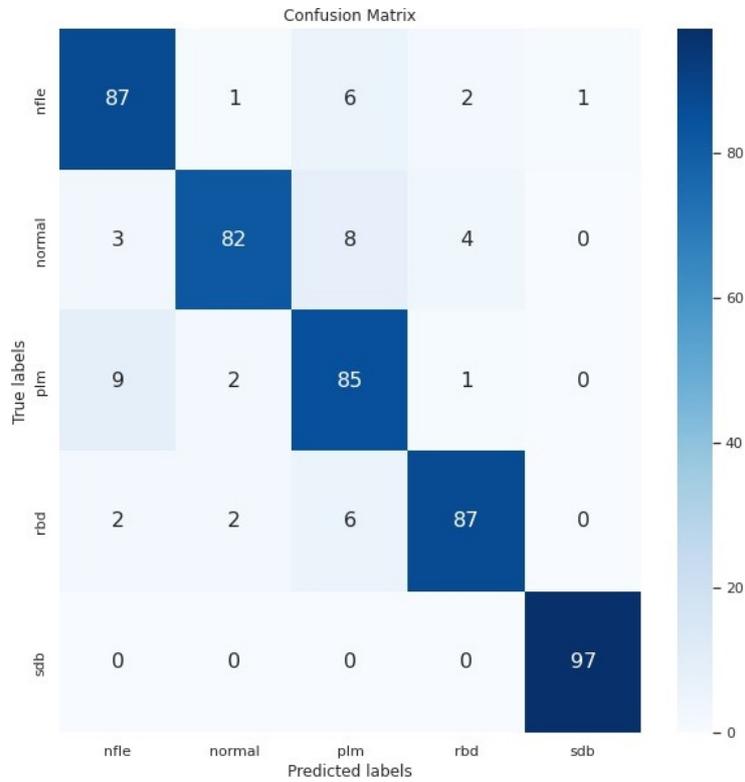
- Data = arreglo bidimensional de n canales por puntos en el tiempo,
  - Sf = Frecuencia de muestreo,
  - Window = 20.
3. Codificación de series en el tiempo a imágenes. Este proceso se realizó primero creando el objeto transformerGASF por medio de la clase GramianAngularField de la librería pyts.image con los siguientes parámetros:
    - Image\_size = tamaño del arreglo de la serie de datos en el tiempo,
    - Method = summation.

Una vez que se creó el objeto se llamó al método fit\_transform pasándole el arreglo bidimensional de la serie en el tiempo, lo que dio como resultado cuatro imágenes diferentes para cada época o ventana en el tiempo, estas fueron agrupadas en una sola imagen utilizando los métodos de la librería matplotlib.pyplot (ver Fig. 5), posteriormente se guardó en un directorio temporal para cada uno de los trastornos.

4. Generación de videos. Para este paso se usó la librería de Open CV de la cual se extrajo el método VideoWriter para escribir los videos a partir de las imágenes generadas en el paso anterior, se especificó el código de video \*mp4v el cual dio a la salida videos de extensión .mp4 que se usaron para alimentar la red neuronal.

### Aprendizaje profundo

1. Preparación de datos. Los datos fueron preparados antes de entrar en la red neuronal, por medio de la librería ImageDataGenerator y la clase VideoFrameGenerator, primero se realizó aumento de datos con las siguientes características:



**Fig. 8.** Matriz de confusión, la clase mejor clasificada es sdb mientras que la clase con menor desempeño es la normal.

- Rescale: 1. /255,
- Rotation\_range: 0.8,
- Width\_shift\_range: 0.2,
- Height\_shift\_range: 0.2,
- Shear\_range: 0.2,
- Zoom\_range: 0.2,
- Horizontal\_flip: True.

Posteriormente se creó el objeto generador de fotogramas(frames) a partir del video con los siguientes parámetros:

- Target\_size: 224, 224,
- Color\_mode: rgb,
- Batch\_size = 16,
- Shuffle: True,
- Nb\_frames: 5.

2. Arquitectura. Se diseñó una arquitectura basada en una red recurrente convolucional con capas convolucionales con su respectivo max pooling y activación ReLU, capas

**Tabla 4.** Comparativa entre el estado del arte actual.

Estudio	Señal	Clasificador	Trastornos del Sueño	Precisión	AUROC
Korkalainen, et al.[15]	EEG(x2), EOG	CNN(1D) LSTM	SDB	NA	NA
Palani, et al.[13]	EEG	Custom CNN	NFLE	0.88	0.92
Sharma, et al.[16]	EEG	Ensemble Bagged and Boosted trees	Normal, Narcolepsia, PLM, SDB, NFLE	0.89	0.99
Modelo propuesto	EEG(x2), ECG, SaO2/SpO2	CNN(2D) GRU	Normal, NFLE, SDB, PLM, RBD	0.91	0.98

GRU, capas densas y una capa densa de activación softmax de 5 unidades para la clasificación de las diferentes clases (Tabla 1).

3. **Compilación y Entrenamiento.** La compilación y el entrenamiento de la red se realizó con los siguientes datos:

- Optimizer: Adam,
- Learning rate: 1e-4,
- Loss: categorical-cross entropy,
- Épocas: 100.

## 4. Resultados

La red utilizó diez participantes con cinco clases: sdb, nfle, normal, plm y rbd; de cada uno se extrajeron cuatro horas del polisomnograma con cuatro canales de este (EEG, SaO2/SpO2, ECG) lo que generó 14390 imágenes de 20 segundos con traslape de 10 segundos entre cada una, las cuales fueron almacenadas en 959 vídeos de 3 minutos por clase. La red fue alimentada con 80 % del total de los vídeos para entrenamiento, 10 % para validación y 10 % para pruebas, se utilizaron 100 épocas de entrenamiento. Los parámetros de entrenamiento se pueden ver en la Tabla 2.

El resultado en entrenamiento después de 100 épocas alcanza una exactitud de 0.93, mientras que en validación alcanza 0.86 de exactitud (Fig. 6). La curva de la función de costo alcanza a decrecer hasta 0.17 en entrenamiento y hasta 0.40 en validación como se observa en la Fig. 7. Los resultados en las curvas de entrenamiento están por arriba del objetivo mayor a 0.80.

Las métricas de precisión (Precision), sensibilidad (Recall) y el valor F (F1-Score) obtenidas con este modelo se muestran en la Tabla 3, obteniendo un promedio de precisión de 0.91, siendo la mayor precisión en el trastorno sdb con 0.99 y la menor con la condición plm con 0.81.

La matriz de confusión (Fig. 8) muestra que clasifica mejor la clase sdb con 97 de 97, siendo la clase normal la que más confunde con la condición plm. La Tabla 4 muestra la comparación entre el trabajo presentado con otros trabajos realizados en la literatura actual.

## 5. Conclusiones

El modelo propuesto siendo una red neuronal recurrente, compuesto por capas convolucionales y de memoria (GRU) logró clasificar los cuatro trastornos de sueño y la condición normal propuestos para este trabajo con un promedio de 0.91 en precisión utilizando señales EEG(x2), ECG y SaO2/SpO2 del polisomnograma las cuales son convertidas a imágenes y posteriormente a vídeos de tres minutos de duración.

La red propuesta está diseñada usando capas convolucionales para la extracción de características de las imágenes y capas recurrentes para guardar la información en función del tiempo, el entrenamiento de la red se realizó con apoyo de la plataforma de Google colab. utilizando sus tarjetas gráficas este fue más rápido (7-8 horas).

La metodología propuesta basada en la codificación de señales en el tiempo en imágenes es un método novedoso que basándose en el principio que las redes neuronales han avanzado más en clasificación de imágenes, buscó competir con modelos basados en señales unidimensionales en el tiempo o capas de una dimensión.

El trabajo logra ser una herramienta confiable para la clasificación de trastornos del sueño para los especialistas del sueño ahorrándoles tiempo para analizar toda la señal del polisomnograma.

Esta investigación queda abierta para mejorar la precisión utilizando un mayor número de participantes de entrenamiento, así como también con arquitecturas más modernas. También un trabajo a futuro puede ser su implementación en una interfaz de usuario, llámese computadora o teléfono inteligente por medio de sensores de señales del polisomnograma más ergonómicos y que el usuario final pueda auto-evaluarse.

## Referencias

1. Kingman, P. S.: Apnea obstructiva del sueño. Case School of Medicine, Case Western Reserve University (2020)
2. García, J. , Williams, N.: Sleep terrors or nocturnal frontal lobe epilepsy? Sleep. A pediatric perspective. vol. 20, no.2, pp. 1–6 (2011)
3. Biggers, A.: Polysomnography. Healthline (2020)
4. LeCun, Y., Bengio, Y., Hinton, G.: Deep learning. Nature, vol. 521, no. 7553, pp. 436–444 (2015) doi: 10.1038/nature14539
5. Ghosh, A., Sufian, A., Sultana, F., Chakrabarti, A., De D.: Fundamental concepts of convolutional neural network. In Recent trends and advances in artificial intelligence and internet of Things. Springer, Cham, vol. 172, pp. 519–657 (2020) doi: 10.1007/978-3-030-32644-9\_36
6. Terzano, M. G., Parrino, L., Smerieri, A., Chervin, R., Chokroverty, S., Guilleminault, C., Hirshkowitz, M., Mahowald, M., Moldofsky, H., Rosa, A., et al.: Atlas, rules, and recording techniques for the scoring of cyclic alternating pattern (CAP) in human sleep. Sleep medicine, vol. 3, no. 2, pp. 187–199 (2002)
7. Goldberger, A. L., Amaral, L. A. N., Glass, L., Hausdorff, J. M., Ivanov, P. C., Mietus, J. E., Moody, G. B., Peng, C. K., Mark, R. G., Stanley, H. E.: PhysioBank, physioToolkit, and physioNet: components of a new research resource for complex physiologic signals. Circulation, vol. 101 no. 23, pp. e215—e220 (2000) doi: 10.1161/01.CIR.101.23.e215
8. Zhang, G. Q., Cui, L., Mueller, R., Tao, S., Kim, M., Rueschman, M., Mariani, S., Mobley, D., Redline, S.: The national sleep research resource: towards a sleep data commons. Journal of the American Medical Informatics Association, vol. 25 no. 10, pp. 1351–1358 (2018) doi: 10.1093/jamia/ocy064

9. Young, T., Palta, M., Dempsey, J., Peppard, P. E., Nieto, F. J., Hla, K. M.: Burden of sleep apnea: rationale, design, and major findings of the Wisconsin sleep cohort study. *WMJ: official publication of the State Medical Society of Wisconsin*, vol. 108 no. 5, pp. 246–249 (2009)
10. Chowdhury, T. T., Hossain, A., Fattah, S. A., Shahnaz, C.: Seizure and non-Seizure EEG signals detection using 1-D convolutional neural network architecture of deep learning algorithm. In: 1st International Conference on Advances in Science, Engineering and Robotics Technology (ICASERT), IEEE, pp. 1–4 (2019) doi: 10.1109/ICASERT.2019.8934564
11. Wickramaratne, S. D., Shaad-Mahmud, M. d.: Automatic detection of respiratory effort related arousals with deep neural networks from polysomnographic recordings. In: 42nd Annual International Conference of the IEEE Engineering in Medicine & Biology Society (EMBC). IEEE, pp. 154–157 (2020) doi: 10.1109/EMBC44109.2020.9176413
12. Pisano, F., Sias, G., Fanni, A., Cannas, B., Dourado, A., Pisano, B., Teixeira, C. A.: Convolutional neural network for Seizure detection of nocturnal frontal lobe epilepsy. *Complexity*, vol. 2020 (2020) doi: 10.1155/2020/4825767
13. Thanaraj, K. P., Parvathavarthini, B., Tanik, U. J., Rajinikanth, V., Kadry, S., Kamalanand, K.: Implementation of deep neural networks to classify EEG signals using gramian angular summation field for epilepsy diagnosis. arXiv:2003.04534 (2020) doi: 10.48550/arXiv.2003.04534
14. Wikipedia: File:FFT-Time-Frequency-View.png (2017)
15. Korkalainen, H., Aakko, J., Nikkonen, S., Kainulainen, S., Leino, A., Duce, B., Afara, I. O., Myllymaa, S., Töyräs, J., Leppänen, T.: Accurate deep learning-based sleep staging in a clinical population with suspected obstructive sleep apnea. *IEEE Journal of Biomedical and Health Informatics*, vol. 24, no. 7, pp. 2073–2081 (2019) doi: 10.1109/JBHI.2019.2951346
16. Sharma, M., Tiwari, J., Patel, V., Acharya, U. R.: Automated identification of sleep disorder types using triplet half-band filter and ensemble machine learning techniques with EEG signals. *Electronics*, vol. 10, no. 13, pp. 1531 (2021) doi: 10.3390/electronics10131531

## **Prototipo de robot manipulador para manejo de residuos peligrosos biológicos e infecciosos mediante detección de color**

Antonio-Alexander Benítez-Arévalo, Jesús García García,  
Guillermo-Rey Peñaloza-Mendoza

Instituto Tecnológico Superior de Pátzcuaro,  
División de Ingeniería Biomédica,  
México

guillermoreypm@hotmail.com

**Resumen.** El presente trabajo aborda la construcción y prueba de un prototipo de robot manipulador de 3 grados de libertad (GDL) para el manejo de Residuos Peligrosos Biológico e Infecciosos (RPBI), asistido por medio de visión artificial para reconocer objetos por color, y así poder clasificarlos evitando de esta manera que el personal médico y de enfermería estén en contacto con estos agentes, reduciendo los accidentes provocados por errores humanos en su manejo. El manejo del robot se realiza mediante comandos de voz empleando el lenguaje de programación Python y enlazándolo con la plataforma Arduino donde se lleva a cabo el control de las articulaciones del robot. El reconocimiento de objetos se realiza empleando la librería llamada OpenCV, de Python, con una escala de colores HSV para la calibración en la detección de objetos. Los colores que se utilizan para la clasificación son rojo, amarillo y verde, esto en base a la clasificación de los RPBI según la Norma Oficial Mexicana NOM-087-ECOL-SSA1-2002. Para el reconocimiento de voz se emplea la librería `speech_recognition` de Python que permite utilizar los servicios de Google. Los movimientos del prototipo se basan en el reconocimiento de la posición de los RPBI y la posición HOME del robot, mediante cinemática inversa se obtiene el ángulo de referencia para los servomotores, los cuales comienzan a girar y de esta manera los eslabones del robot van acomodándose hasta llegar al objeto. Como efector final, el robot cuenta con un electroimán que toma el RPBI y permite su transporte.

**Palabras clave:** Robot manipulador, RPBI, visión artificial.

### **Manipulator Robot Prototype for Biological and Infectious Hazardous Waste Management by Color Detection**

**Abstract.** The present work deals with the construction and testing of a manipulator robot prototype with 3 degrees of freedom (DOF) for the handling of Biological and Infectious Hazardous Waste (RPBI), assisted by means of

artificial vision to recognize objects by color, and thus be able to classify them, thus preventing medical and nursing staff from being in contact with these agents, reducing accidents caused by human error in handling them. The robot is handled by voice commands using the Python programming language and linking it to the Arduino platform where the control of the robot's joints is carried out. Object recognition is performed using the Python OpenCV library with an HSV color scale for object detection calibration. The colors used for the classification are red, yellow and green, based on the classification of the RPBI according to the Official Mexican Standard NOM-087-ECOL-SSA1-2002. For speech recognition, the Python speech\_recognition library is used, which allows the use of Google services. The movements of the prototype are based on the recognition of the position of the RPBI and the HOME position of the robot, through inverse kinematics, the reference angle for the servomotors is obtained, which begin to rotate and in this way the links of the robot are accommodated until you reach the object. As an end effector, the robot has an electromagnet that picks up the RPBI and allows it to be transported.

**Keywords:** Robot manipulator, RPBI, artificial vision.

## 1. Introducción

Un robot manipulador puede ser definido como mecanismo, que suele consistir en una serie de segmentos, articulados o deslizándose entre sí, con la finalidad de agarrar y mover objetos, controlado remotamente por una computadora o por un humano [1]. A lo largo de la historia, se ha buscado realizar tareas de manera automática, por lo que, el desarrollo de nuevas tecnologías ha evolucionado hasta el punto en el que hoy en día existen muchos tipos de robots que hacen diferentes tareas en beneficio de la humanidad.

En el ámbito de la salud se cuenta con un área destinada a la manipulación y control de los agentes residuos peligrosos biológico e infecciosos (RPBI), esta actividad se realiza a nivel hospitalario de forma manual. Debido a sus características infecto-contagiosas, estos residuos representan un riesgo para la salud tanto de las personas como del medio ambiente, su manejo y control adecuado representa uno de los temas con mayor controversia dentro del sector salud [2].

En la actualidad, debido a la contingencia ocasionada por el COVID-19, los desechos RPBI incrementaron exponencialmente, según datos del Instituto Nacional de Ecología y Cambio Climático se estimaba que hasta julio de 2020 habría un incremento del 3.3 al 16.5% en la cantidad de agentes RPBI por la contingencia [3]. Estos datos muestran un mayor riesgo para los trabajadores de la salud, por lo tanto, el usar la tecnología para disminuir el contacto directo resulta una gran ayuda.

La aplicación de la robótica para la clasificación de estos agentes RPBI es parte fundamental en su adecuado manejo y para reducir peligros por exposición. Tomando los depósitos de los agentes RPBI se puede realizar la detección de estos por medio de visión artificial, discriminando los objetos por el color del envase contenedor en un espacio de almacenaje, en base a esto, determinar el material que se contiene y por medio de un robot manipulador tomarlos del depósito primario o temporal y llevarlos al depósito principal de cada uno de los tipos de agente RPBI. Para esto se diseña y

construye un robot manipulador, el cual cuenta con un algoritmo de control, que recibe la información de la posición relativa del depósito primario y el tipo de agente RPBI detectado para determinar los movimientos necesarios para el posicionamiento del robot, desde su posición inicial hasta el agente RPBI y posteriormente transportarlo al contenedor final respectivo.

### **1.1. Clasificación de residuos peligrosos biológico-infecciosos**

Según la Norma Oficial Mexicana NOM-087-ECOL-SSA1-2002, Protección ambiental-salud ambiental-residuos peligrosos biológico infecciosos-clasificación y especificación de manejo, se deben clasificar estos agentes por su origen y su estado físico, indicando su color y tipo de envase [4].

## **2. Estado del arte**

Robot angular, antropomórfico o también conocido como robot manipulador tiene sus tres principales articulaciones de tipo rotacional, con lo cual emplea las coordenadas angulares para determinar las posiciones de su elemento terminal. Se llama antropomórfico por que simula los movimientos de un brazo humano [1]. Este robot posee gran accesibilidad y maniobrabilidad, es rápido y ocupa poco espacio en relación al campo de trabajo que abarca. Debido a sus características, lo hacen un modelo versátil con múltiples aplicaciones.

En el mercado existen diversos robots manipuladores que se usan de manera industrial, por ejemplo, el Robot paletizador de botellas RP40 de la marca Tecno el cual es flexible, fiable y sencillo [5], el Robot KUKA KR 4 AGILUS es un robot compacto ampliamente utilizado en células de automatización [6] y el Robot ABB IRB 1100 cuyo diseño es compacto y ligero [7].

Se consideran los ejemplos para visualizar el alto desempeño que brinda un robot manipulador; además se usan como base en la construcción del manipulador de estudio, reduciendo costos con el uso de software libre de alto desempeño, implementación de algoritmos inteligentes y hardware de bajo costo.

Así mismo, hoy en día, la robótica es una de las áreas más exploradas, debido a que nos permite automatizar procesos que nos brinda la posibilidad de reducir peligros por exposición, realizar tareas repetitivas, entre otras actividades. Por lo cual, existen trabajos similares a sobre el uso de robots manipuladores aplicados a la detección de color o al manejo de RPBI, ejemplos de estos desarrollos se enlistan a continuación.

- Jordán Gamito en su trabajo de fin de grado [8], desarrolla un sistema que, mediante un brazo robótico es capaz de clasificar objetos según el color de estos. Empleando una cinta transportadora prototipo, pasa objetos de distinto color, un detector de presencia detiene el avance de la banda transportadora para que el color sea determinado a través de un sensor de color que le indicará al robot hacia donde moverlo,
- En [9] se presenta un sistema teleoperado para la manipulación de residuos de riesgo biológico, este proyecto se basa en la tecnología del control teleoperado y se apoya en un brazo robótico para el contacto directo, que posee sensores

para la recolección, en este el operador desempeña su labor de manera segura y eficaz,

- El desarrollo de un robot manipulador de 4 grados de libertad con el objetivo de clasificar objetos de diferentes colores utilizando técnicas de filtrado RGB se puede revisar en el trabajo de Rosas Arias et al, [10]. En este se muestra la construcción de un brazo robótico y la programación de una cámara para detectar objetos de colores y poder realizar su separación a posiciones fijas.

### **3. Desarrollo propuesto**

Actualmente, la pandemia por el COVID-19 provoca la generación de una gran cantidad de residuos peligrosos biológico e infecciosos en la atención hospitalaria, por lo que es indispensable tomar medidas sanitarias que garanticen el correcto manejo y eliminación de los residuos para mantener mínimos riesgos de contagio para los profesionales de la salud y la población en general por estar expuesta a los RPBI que pudieran tirarse de forma clandestina, y así evitar mayor número de contagios accidentales.

#### **3.1. Planteamiento del problema**

A lo largo de aproximadamente dos años la humanidad se ha visto afectada por la pandemia generada por el COVID-19, este virus ha cobrado la vida de millones de personas, y siendo la primera línea de defensa, los que están en mayor riesgo, son los médicos y enfermeras.

Estos atienden a los pacientes que lo necesitan, por lo que, tienen contacto directo con residuos peligrosos biológico e infecciosos que se van generando en las salas de consulta y en las mismas áreas de COVID que se han implementado en los complejos hospitalarios.

Sin embargo, el principal problema radica en la gran cantidad de residuos que se están generando y manejando dentro de las instalaciones hospitalarias, esto provoca que en ocasiones el personal no sea suficiente, además trabajando bajo presión y la sobrecarga de trabajo a la que se ven expuestos, incrementa la probabilidad de cometer errores de manejo o separación y verse infectados por los residuos biológicos e infecciosos, así como contaminar el entorno en áreas no COVID.

Actualmente, a nivel hospitalario, no se cuenta con tecnología para el manejo de los agentes RPBI dentro de la zona de almacenaje, todo el proceso, desde la generación hasta el transporte externo, se realiza por personal de salud de apoyo.

Este proceso hasta su extracción del hospital para su manejo integral dura días (varía de acuerdo al tamaño del hospital de 7 hasta 30 días), durante los cuales los agentes RPBI siguen multiplicándose, lo que genera un riesgo mayor.

#### **3.2. Solución propuesta**

Como propuesta de solución al manejo de agentes RPBI, se propone la construcción de un robot manipulador automatizado, que sirva como herramienta para la

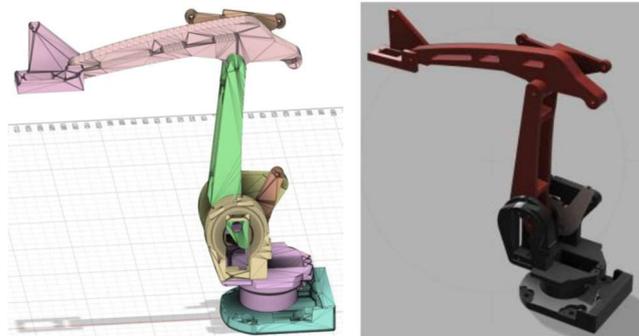


Fig. 1. Renderizado del ensamble del robot manipulador.

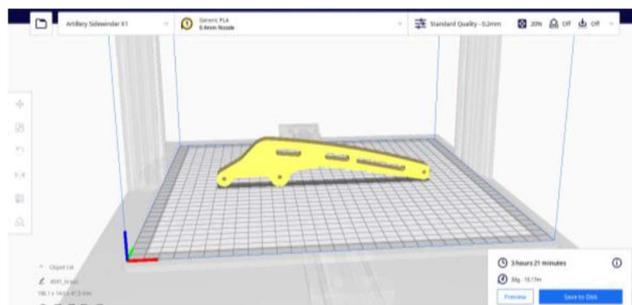


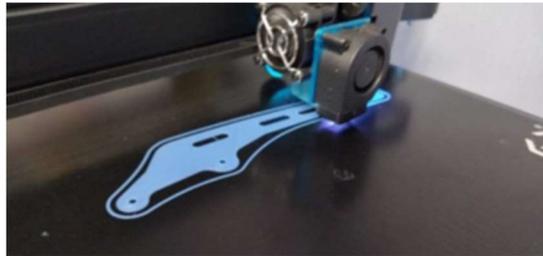
Fig. 2. Configuración de parámetros para las piezas mediante Cura Ultimaker.

clasificación y manejo de los agentes RPBI por medio de la detección del color de las envolturas de los agentes RPBI. Como metas principales se tienen:

- Construir un dispositivo robótico poliactuado, empleando tecnología 3D, con el cual se manipulen los agentes RPBI y se reduzca el riesgo de contagio en el personal de apoyo que maneja el almacenaje final de los agentes RPBI,
- Realizar la programación del robot que permita usar visión artificial para la detección y clasificación de los agentes RPBI,
- Control individual de los servomecanismos del robot para el posicionamiento en una localización de referencia (HOME) y en cada punto donde existan los agentes RPBI.

### 3.3. Diseño y construcción del robot manipulador

La implementación del prototipo se realizó mediante el diseño 3D del robot manipulador, primeramente, se realizó una investigación documental acerca de los diferentes tipos de robots manipuladores empleados en el manejo de objetos [1, 2, 6, 7, 9]. Con estos trabajos de referencia, se optó por emplear un robot manipulador básico de 3 grados de libertad, pero cuyos elementos de accionamiento no estuvieran montados sobre las articulaciones, esto para disminuir el peso del robot y permitir una mayor capacidad de carga en el efector final.



**Fig. 3.** Impresión de piezas mediante la impresora Artillery SideWinder X1.

Es importante mencionar que cada articulación está diseñada para ser controlada de manera independiente por un servomotor, el cual está conectado a una transmisión tipo biela – manivela para efectuar el movimiento a distancia de la articulación.

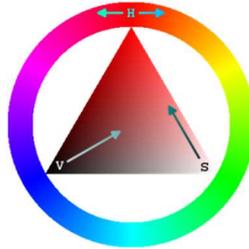
Como efector final se adapta un electroimán, el cual es un componente clave para el correcto funcionamiento del robot, ya que, se supone un elemento metálico en cada contenedor para su transporte, evitando así emplear una pinza que puede rasgar o romper los contenedores de plástico en el transporte, además un electroimán permite la limpieza de forma más sencilla que otros elementos porosos.

El diseño 3D de los elementos del robot manipulador se realizaron en el Software Fusion 360 de la compañía Autodesk, el cual es uno de los softwares más empleados para el diseño y modelado 3D. Fusion 360 es un software CAD, CAM y de circuitos impresos de modelado 3D basado en la nube para el diseño y la manufactura de productos. Cada eslabón del robot manipulador fue diseñado para que en extensión máxima alcanzara 27 cm (valor tomado con una escala 1:3 del robot ABB IRB 1100). En la Fig. 1, se presenta el renderizado del modelo del brazo robótico ensamblado completamente después del diseño de cada elemento, en este ensamble se verificó la interacción de cada elemento al hacer un movimiento, para evitar daños estructurales al momento de su implementación.

Para la implementación física, se debe configurar el tipo de material a emplear, el factor de relleno de la pieza, entre otros parámetros, esto para cada uno de los elementos a imprimir. Para la configuración de la impresión 3D se utilizó el software Ultimaker CURA, en la Fig. 2 se puede observar una pieza dentro del software de configuración para la impresión.

Una vez que las piezas fueron verificadas y configuradas para la impresión, se implementó una impresora Artillery SideWinder X1 para su fabricación, su tiempo de impresión total fue aproximadamente de 34 horas, gastando un aproximado de 150 gramos de material, en este caso al ser el primer prototipo se imprimió en PLA con un relleno del 20%. Parte de este proceso de fabricación se puede observar en la Fig. 3.

Una vez terminado el proceso de impresión, se optó por darle a las piezas un tratamiento para que su aspecto fuera mejorado, esto último debido a que el acabado final que entrega la impresora es rústico al usar un relleno del 20%. Este tratamiento fue lijar las piezas para remover las imperfecciones que quedaron en el proceso de impresión, posteriormente cada pieza fue sometida a un calor constante de 60 °C y a un baño de solvente, para crear un templado superficial que permita reformar la superficie y dejar un acabado pulido, por último, se agregan diversas capas de pintura vinílica para que resaltara su aspecto.



**Fig. 4.** Distribución del espacio de color HSV.

Al ensamblar físicamente el robot, el diseño prevé la colocación de los servomotores en la base del mismo, por lo tanto, es importante mencionar que, debido al diseño, los dos eslabones de extensión no pueden rotar de manera libre en todo el rango de movimiento que ofrecen los servomotores ( $0 - 180^\circ$ ), por lo que, el rango de movimientos viene dado de la siguiente manera:

- Primer GDL (Base):  $0 - 180^\circ$ ,
- Segundo GDL (Brazo):  $100 - 0^\circ$ ,
- Tercer GDL (Antebrazo):  $0 - 40^\circ$ .

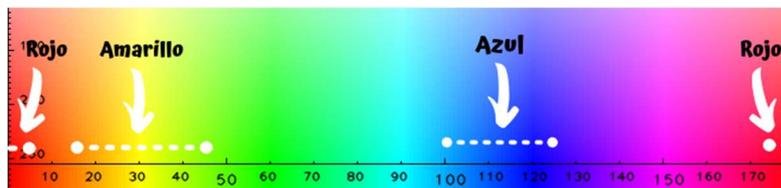
### 3.4. Programación para la detección por color de los agentes RPBI

Antes de iniciar el proceso de la detección del color de los objetos, es necesario recibir la orden del agente RPBI que se desea clasificar, esto puede ser por medio de botones, instrucciones o voz, en este caso, para disminuir el contacto físico de los usuarios, se realiza la orden por medio de voz usando los colores como referencias de las acciones a realizar.

El reconocimiento de voz, se realiza mediante Python utilizando la librería `speech_recognition`, la cual nos permite utilizar el micrófono integrado a una cámara web para adquirir la señal generada por las palabras, para posteriormente utilizar los servicios de google para el reconocimiento de voz mediante su diccionario, poder convertir el sonido a texto y este texto guardarlo como una variable que se utiliza como condicional en sentencias simples de casos `if` y `else if` donde mandamos llamar a las funciones que harán el reconocimiento por color.

Para la implementación de la detección de color se utiliza una cámara web de la marca Hisemy modelo ML2-MX-0095 que cuenta con las siguientes características: sensor de imagen de alta definición de 5 millones de píxeles, enfoque automático, calidad de cámara web 2K HD, capacidades de disparo de alta velocidad, grabación HD de  $2560 \times 1440$  a 30 fotogramas por segundo.

El código de programación se realiza mediante el lenguaje Python, en el cual se emplea el espacio de color HSV (Hue, Saturation, Value), como se ve en la Fig. 4, que significa matiz, saturación y brillo, de la librería de visión artificial de OpenCV, la diferencia con el espacio RGB, radica en usar un solo elemento, el matiz, para determinar el color a analizar en lugar de tres elementos. Los valores del HSV al igual que en el RGB, cuentan con rangos de trabajo, en este caso los canales de HSV en OpenCV son: H de 0 a 179, S de 0 a 255 y V de 0 a 255.



**Fig. 5.** Valores de matices (H) de componentes HSV. Imagen tomada del Blog Omes en el enlace <https://omes-va.com/deteccion-de-colores2/>.

Para determinar un color se toma el componente H que corresponde al matiz, que en este caso se puede ver en la Fig. 5, que este componente va cambiando de rojo, amarillo, verde, violeta hasta llegar nuevamente a rojo, siendo los matices que nos interesan el verde, rojo y amarillo. Bajo este esquema, se realizan 4 pasos generales para la detección del color, descritos a continuación.

Paso 1.- Tomar la imagen a procesar: Primeramente se debe leer la imagen que se desea procesar para detectar los colores, en este caso particular, se toma la imagen desde el video adquirido por la cámara web, esto se realiza con las siguientes instrucciones en Python. Para capturar el video se emplea la instrucción `cap = cv2.VideoCapture(0)`, posteriormente se debe capturar un fotograma para analizar el contenido, esto se hace con la instrucción `ret,frame=cap.read()`.

Paso 2.- Transformar de BGR a HSV: Al leer los fotogramas del video, por defecto se toma en valores BGR, por lo tanto, para hacer un manejo más simple de los colores, se procede a transformar el espacio a valores HSV, donde se discriminará por matices. Esta conversión se realiza con la instrucción `cv2.cvtColor(frame, cv2.COLOR_BGR2HSV)`, como primer argumento se tiene la imagen adquirida en el primer paso llamada "frame", esta corresponde a la que se desea convertir a valores HSV, el segundo argumento "cv2.COLOR\_BGR2HSV" realiza la transformación de valores RGB a HSV.

Paso 3.- Establecer rango de valores de los matices para los colores deseados: Usando como referencia la Fig. 5, el rango de valores del matiz (H) va de 0 a 179, el rango de saturación (S) va de 0 a 255 y el brillo (V) se mantiene constante en 255, sin embargo, este último parámetro también puede variarse de 0 a 255, considerando que los valores cercanos a 0 oscurecen el matiz cercano a negro. Para implementar la selección de los valores HSV en Python se debe crear un array con numpy, donde se especifiquen como argumentos separados los valores H, S y V, seguido de `np. uint8`, por lo que la instrucción base para la detección de cada color es `ColorBajo=np.array([Hcb, Scb, Vcb], np. uint8)` y `ColorAlto=np.array([Hca, Sca, Vca], np. uint8)`, donde una instrucción limita el valor mínimo del color y la otra el valor máximo.

Particularizando a este caso, los rangos de valores H para cada uno de los colores empleados en el etiquetado de agentes RPBI son: Rojo de 0 a 10 para el primer rango mostrado en la Fig. 6 y de 170 a 179 para el segundo rango de valores. Amarillo de 25 a 35 en el rango principal. Verde, este es el color de etiquetado neutro, se trabaja en valores de 50 a 70. Así mismo, los parámetros de saturación y brillo se ajustan para las condiciones de iluminación existentes en el área de trabajo, quedando estos en los límites, S de 100 a 255 y V de 20 a 255.

**Tabla 1.** Parámetros DH del robot manipulador diseñado.

Articulación	$\theta$	$d$	$a$	$\alpha$
1	$\theta_1$	$L_1 = 3cm$	0	$90^\circ$
2	$\theta_2$	0	$L_2 = 12 cm$	0
3	$\theta_3$	0	$L_3 = 12 cm$	0

Paso 4.- Visualización: La visualización de la imagen puede ser presentada de diversas maneras, esto debido a que la selección por color emplea una binarización de la imagen por matiz, por lo tanto, puede ser visualizada como imagen blanco y negro o en términos del color de selección. En este caso particular, se prefiere visualizar el color real de selección por lo que se emplea la instrucción `maskColorvis = cv2.bitwise_and (frame, frame, mask= maskColor)`, donde a la imagen se le aplica la máscara de color seleccionada. Una vez que se tiene la detección del color de los agentes RPBI, se requiere conocer la posición del color, es decir, del agente RPBI, para que el robot manipulador viaje a esa posición para recogerlo.

Esta actividad se realiza por medio de geometría, se identifican cuatro puntos en la imagen binarizada por color que se obtiene del proceso anterior, estos cuatro puntos corresponden a los píxeles con los valores mínimos y máximos de coordenadas (x,y) de la zona detectada con el color deseado. Utilizando estos cuatro puntos se encuentra el centro del objeto,  $(X_r, Y_r)$ , por medio de las expresiones de la ecuación 1. Esta información es referenciada al plano de la imagen, por lo que solamente generará el movimiento horizontal y vertical del robot, no el de profundidad o extensión del mismo:

$$\begin{aligned} X_r &= \frac{X_{m\acute{a}x} + X_{m\grave{i}n}}{2}, \\ Y_r &= \frac{Y_{m\acute{a}x} + Y_{m\grave{i}n}}{2}. \end{aligned} \quad (1)$$

Esta posición toma como referencia u origen la esquina superior izquierda de la imagen, lo cual es necesario referenciar posteriormente a la posición del robot manipulador.

### 3.5. Control del accionamiento de las articulaciones del robot manipulador

Para el funcionamiento del brazo robótico, se emplea el uso de su cinemática, la cual determina el movimiento del robot a partir de su geometría, este tema es ampliamente abordado en literatura de robótica [1, 8, 11, 12], por lo que, solamente se bosqueja la solución implementada en particular para este robot.

Empleando el algoritmo Denavit-Hartenberg (DH) se obtiene la cinemática directa del robot esto se reduce a encontrar una matriz homogénea de transformación T que relacione la posición y orientación del extremo del robot respecto del sistema de referencia fijo situado en la base del mismo, la ecuación 2 representa la matriz T de transformación, la cual puede ser descrita como el producto de matrices A referentes a las características físicas de cada eslabón:

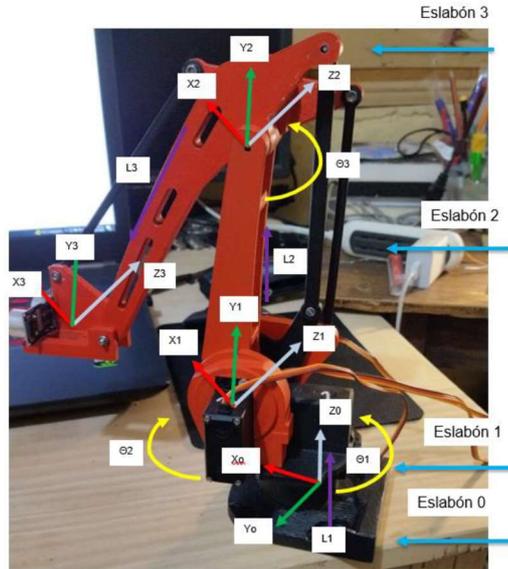


Fig. 6.- Representación de los etiquetados de Denavit-Hartenberg.

$$T = \begin{bmatrix} \text{Rotación} & \text{Traslación} \\ \text{Perspectiva} & \text{Escala} \end{bmatrix} = \begin{bmatrix} R_{3 \times 3} & P_{3 \times 1} \\ f_{1 \times 3} & w_{1 \times 1} \end{bmatrix} = \begin{bmatrix} n_x & o_x & q_x & p_x \\ n_y & o_y & q_y & p_y \\ n_z & o_z & q_z & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix}. \quad (2)$$

Con base en el algoritmo DH, se determinan las matrices A con los parámetros DH  $a_i$ ,  $\alpha_i$ ,  $d_i$  y  $\theta_i$  inherentes al robot manipulador propuesto. La ecuación 3 muestra la definición de las matrices A en términos de los parámetros DH, mientras que la ecuación 4, muestra la relación entre el producto de las matrices A y la matriz de transformación general. La Tabla 1 muestra los parámetros DH para cada eje de referencia en nuestro robot como se encuentran etiquetados en la Fig. 6:

$$A_i^{i-1} = \begin{bmatrix} \cos(\theta_i) & -\cos(\alpha_i) \sin(\theta_i) & \sin(\alpha_i) \sin(\theta_i) & a_i \cos(\theta_i) \\ \sin(\theta_i) & \cos(\alpha_i) \cos(\theta_i) & -\sin(\alpha_i) \cos(\theta_i) & a_i \sin(\theta_i) \\ 0 & \sin(\alpha_i) & \cos(\alpha_i) & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad (3)$$

$$T = A_1^0 A_2^1 A_3^2. \quad (4)$$

Definiendo todas las matrices A (ecuación 3) en términos de los parámetros DH, y conociendo la localización destino del extremo final del robot manipulador, que será la posición otorgada por el algoritmo de la detección del color, en términos de n, o, q y p de la matriz T (ecuación 2), se pueden manipular las 12 ecuaciones resultantes de la igualación de las ecuaciones 1 y 2, llamada cinemática inversa, para determinar el valor de rotación de cada articulación para lograr el posicionamiento deseado del robot, particularmente se utilizan las ecuaciones igualadas a cero o a la coordenada final deseada representada por  $(p_x, p_y, p_z)$ . Este proceso se realiza numéricamente mediante la ayuda del software MATLAB basándose en lo implementado en [1, 11, 12].

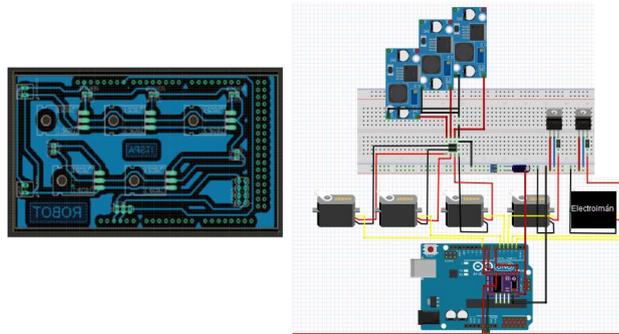


Fig. 7.- Circuito y diagrama electrónico implementado.



Fig. 8. Prototipo final.

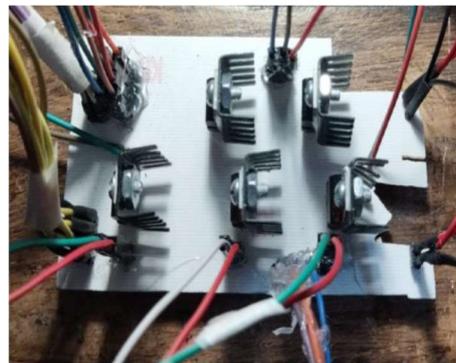
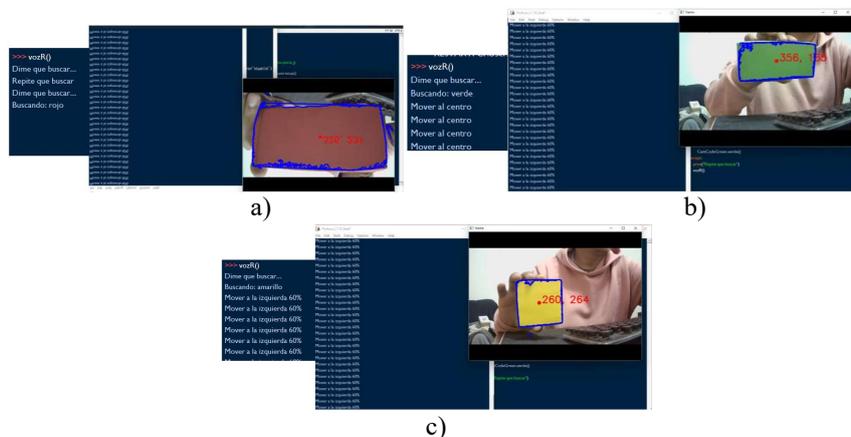


Fig. 9. Circuito electrónico.

La cinemática inversa se implementa dentro de la programación física de la plataforma Arduino, esto completa el ciclo de recolección de información, desde que la imagen es tomada y procesada en Python, el dato de posición del agente RPBI se manda por comunicación serial a Arduino implementando, por medio de las ecuaciones de cinemática inversa, el cálculo de los ángulos de rotación de cada articulación, y sea el mismo Arduino que controle los servomecanismos. La cinemática inversa permite conocer la localización cartesiana del robot y así determinar los ángulos que el robot debe girar, desde su posición actual, para que su elemento final se coloque en la coordenada deseada.

### 3.6. Implementación física de los componentes electrónicos

En este punto, se tienen todos los elementos necesarios para la detección del color, la programación para el movimiento del robot y la implementación física de los elementos del robot. Por lo tanto, solamente falta implementar los actuadores que le dan movimiento físico al robot.



**Fig. 10.** Pruebas de detección de color. a) Detección de Rojo, b) - Detección de Verde, c) Detección de Amarillo.

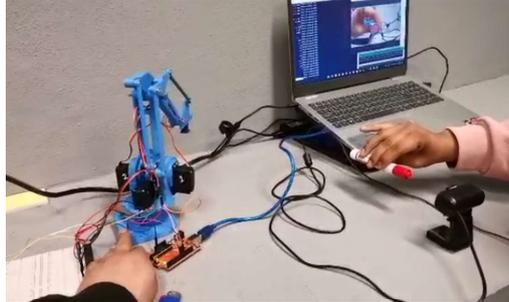
En este caso, por su costo se implementan servomotores TowerPro MG995 que son económicos, pero con alto torque de trabajo de hasta 11Kg-cm, estos para el movimiento del robot, como elemento final, se implementa un electroimán de bajo consumo de energía, todo ello controlado y programado a través de la tarjeta de desarrollo Arduino. Debido a que se trabaja con la tarjeta de desarrollo Arduino, la corriente que este otorga no es suficientes para operar todos los dispositivos electrónicos que requiere el robot. Para ello se diseñó un circuito impreso (PCB por sus siglas en inglés Printed Circuit Board), mostrado en la Fig. 7, el cual es alimentado con 9V y permite alimentar a la tarjeta Arduino, a los 4 servomotores implementados (uno para las articulaciones 1 y 3, dos para la articulación 2 del robot para otorgar mayor fuerza al extender el robot al máximo) y al electroimán implementado.

#### 4. Resultados

El diseño del robot manipulador se ensambló y se verificó antes de la impresión 3D, mediante simulación se encontró que no se presentaba ninguna interferencia en el mecanismo y que se podía mover libremente.

El resultado de la impresión mostró un prototipo viable para aplicarlo como demostración del funcionamiento de la clasificación por color. La fig. 8 muestra el prototipo final implementado. La escala que se emplea para el desarrollo del prototipo es la adecuada para simular el funcionamiento real de este, ya que trabajarlo como prototipo es más viable que hacerlo en escala real, para la validación de la estrategia. En la Fig. 9 se muestra la implementación de los elementos electrónicos necesarios para la alimentación de los servomotores empleados. Una vez finalizado el ensamblaje del robot, se procedió a programar cada uno de los motores que conforman al robot.

La programación se hizo con el software Arduino. Al momento de estar programando el robot se fueron realizando pruebas individuales para ir verificando que cada uno de los componentes del robot estuviera realizando de manera correcta la función con la que debería cumplir.



**Fig. 11.-** Pruebas de detección de color con el robot manipulador.

Con respecto a la identificación de los colores, las pruebas se realizaron con los comandos de voz, en este trabajo se emplearon solamente 5 comandos de voz, los cuales son:

- Inicio: Permite energizar todo el robot manipulador y activarlo para que realiza alguna actividad. Activa la cámara para que realice la adquisición del espacio de trabajo alrededor del robot,
- Detener: Este comando apaga el robot manipulador, dejando activo solamente los recursos mínimos para poder activarse en el momento deseado,
- Rojo, Verde y Amarillo: Son los comandos que permiten activar la detección de los colores empleados para la clasificación.

En la Fig. 10 a) se muestra el reconocimiento del comando “Rojo” y la detección del color realizada, con la cámara. En esta misma se sobrepone la coordenada central del contorno de color detectado. En las Fig. 10 b) y c), muestran lo mismo para los comandos “Verde” y “Amarillo”, respectivamente. La Fig. 11 muestra evidencia de la implementación del robot haciendo pruebas con el reconocimiento de color.

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

La programación de robot manipuladores no es una tarea sencilla, ya que se deben tener en cuenta muchos aspectos tanto mecánicos, como de programación. Sin embargo, el prototipo construido se desempeña de forma correcta, realizando las actividades de detección de color, movimiento a la posición de detección, toma de objeto y movimiento a la posición final de almacenaje. La propuesta, al generar un manejo a distancia y automatizado, disminuye significativa la interacción del usuario con la manipulación de los objetos, ya que el usuario funge como observador y no como manipulador de los objetos.

El trabajo presenta un prototipo a escala, por lo que actualmente cuenta con limitaciones tecnológicas para su implementación, sin embargo, como trabajo futuro se plantea la implementación de un robot manipulador de mayor capacidad para el manejo de objetos y su clasificación.

## Referencias

1. Sandler, B. Z.: Robotics: designing the mechanisms for automated machinery. 2nd Ed. Academic Press, (1999)
2. Morelos, R., Ramírez, M., Sánchez, G., Chavarín, C., Meléndez, E.: El trabajador de la salud y el riesgo de enfermedades infecciosas adquiridas. Revista de la Facultad de Medicina de la UNAM, vol. 57, no. 4, pp. 34–42 (2014)
3. Instituto Nacional de Ecología y Cambio Climático: Panorama de la generación y manejo de residuos sólidos y médicos durante la emergencia sanitaria por COVID-19. (2020)
4. Norma Oficial Mexicana NOM-087-ECOL-SSA1-2002. Obtenida del sitio <http://www.salud.gob.mx/unidades/cdi/nom/087ecolssa.html> (2003)
5. Tecnoco SA de CV. Obtenido en línea del enlace <http://manipulacion-ergonomica.tecnoco.com.mx/equipo/robot-paletizador-de-botellas/>
6. Kuka Robotics México. Obtenido en línea del enlace <https://www.kuka.com/es-mx/productos-servicios/sistemas-de-robot/robot-industrial/kr-4-agilus>
7. ABB Robotics. Obtenido en línea del enlace <https://new.abb.com/products/robotics/es/robots-industriales/irb-1100>
8. Jordán Gamito, A.: Diseño y realización de un sistema automatizado de clasificación por colores basado en brazo robótico y el TMS320F28335. TFG de Ingeniería en Tecnologías Industriales, Departamento de Ingeniería Electrónica Escuela Técnica Superior de Ingeniería Universidad de Sevilla, Sevilla, (2020)
9. Astete Rupay, J., Gallo Torres, D., Torres Hinostroza, A., Peña Romero, A., Sarmiento Cornejo, M.: Sistema teleoperado aplicado a la manipulación de residuos hospitalarios con riesgo biológico. Revista Ingenium, vol. 1, no. 2 (2016) ISSN en línea 2519-1403
10. Rosas-Arias, L., Vallejo-Meraz, J., Pérez-Bailón, W., Rojas-Cid, J. D.: Robot clasificador de objetos de color utilizando técnicas de filtrado RGB. Revista de Prototipos Tecnológicos, vol.3 no.10, pp. 50–59 (2017) ISSN-2444-4995
11. HengHui, S., MinZhou, L., Wei, L., Xiang, D.: Virtual link-lengths approximating method for forward kinematics of multiple motion coupled linkage mechanism. In: International Conference on Electronic & Mechanical Engineering and Information Technology, pp. 2318–2323 (2011)
12. Hendarto, H. Munadi, A., Setiawan, J. D.: ANFIS application for calculating inverse kinematics of programmable universal machine for assembly (PUMA) robot. In: The 1<sup>st</sup> International Conference on Information Technology, Computer, and Electrical Engineering, pp. 35–40 (2014)

## **Propuesta de interfaz cerebro computadora realizada en una plataforma de bajo costo para la interpretación de la imaginación de movimiento**

Marcos Alejandro Garibay Campos<sup>1</sup>, Antonio Alarcón Paredes<sup>1</sup>,  
Jessica Cantillo Negrete<sup>2</sup>, Rubén Isaac Cariño Escobar<sup>2</sup>,  
Gustavo Adolfo Alonso Silverio<sup>1</sup>

<sup>1</sup> Maestría en Ingeniería para la Innovación y Desarrollo Tecnológico,  
Chilpancingo, Guerrero,  
México

<sup>2</sup> Instituto Nacional de Rehabilitación Luis Guillermo Ibarra Ibarra,  
México

{10123701,aalarcon,gsilverio}@uagro.mx,  
jessica.cantillo.negrete@gmail.com, rubencarinoe@hotmail.com

**Resumen.** Las Interfaces cerebro computadora (ICC) son sistemas que permiten al usuario crear un canal artificial de comunicación entre el cerebro y un dispositivo periférico haciendo uso de la actividad cerebral, la imaginación motora consiste en incorporar el movimiento voluntario sin depender de la capacidad motora de la persona activando patrones neuronales similares a la ejecución de movimiento, en base a este principio se ha permitido el uso de las ICC como una herramienta auxiliar en el campo de la rehabilitación. Un inconveniente al realizar proyectos de este tipo es que el costo de los elementos necesarios para su desarrollo es muy alto. En el presente trabajo se propone una ICC para la interpretación de la imaginación de movimiento desarrollada en una plataforma de bajo costo que permita evaluar la potencialidad de una ICC de estas características.

**Palabras clave:** ICC, imaginación motora, bajo costo, interfaz cerebro computadora.

### **Brain Computer Interface Proposal Made in a Low-cost Platform for the Interpretation of the Movement Imagination**

**Abstract.** Brain Computer Interfaces (BCI) are systems that allow the user to create an artificial communication channel between the brain and a peripheral device making use of brain activity, motor imagery consists of incorporating voluntary movement without depending on the motor capacity of the person, activating neural patterns similar to the execution of movement, based on this principle has allowed the use of BCI as an auxiliary tool in the field of

rehabilitation. A disadvantage when carrying out projects of this type is that the cost of the necessary elements for its development is very high. In the present work we propose a BCI for the interpretation of the imagination of movement developed on a low-cost platform that allows evaluating the potential of a BCI of these characteristics.

**Keywords:** BCI, motor imagery, low cost, brain computer interface.

## 1. Introducción

Las interfaces cerebro computadora (ICC) son sistemas que permiten al usuario crear un canal artificial de comunicación entre el cerebro y un dispositivo externo mediante la traducción de las señales cerebrales (Lotte et al., 2018), la imagería motora (MI) o imaginación de movimiento consiste en la evocación de la persona de un movimiento o gesto para aprender o mejorar su ejecución, es decir se incorpora el movimiento voluntario sin depender de la capacidad motora de la persona y está activa patrones neuronales similares a la ejecución de movimiento (Fernández-Gómez & Sánchez-Cabeza, 2018) en el campo de la rehabilitación han sido una propuesta aceptada solo como herramienta auxiliar debido a la falta de evidencia a largo plazo para apoyar su relevancia clínica como se menciona en (Teo & Chew, 2014).

Para las ICC utilizadas en el campo de rehabilitación se busca identificar los fenómenos ocurridos en la actividad cerebral como los que se relacionan con la MI.

En trabajos como en (Khan et al., 2020) se han desarrollado y evaluado algoritmos para optimizar la interpretación de los patrones generados por la MI logrando obtener mejoras en los algoritmos de extracción de características que se reflejan en los rendimientos de clasificación, lo que implica que el peso de los algoritmos utilizados para buscar las características de interés complementan los rendimientos obtenidos por los sistemas de este tipo y acredita la exploración de electroencefalógrafos de bajo costo que utilicen algoritmos robustos para la interpretación de la MI.

La mayoría de los trabajos documentados han sido desarrollados en plataformas de licencia como Matlab como los presentados en (Roy et al., 2022) y a su vez, la mayoría de los *toolkits* especializados para sistemas ICC se han desarrollado para dicho entorno como por ejemplo EEGLAB, EEGVIS o Brainstorm. Los gastos que conllevan mantener una licencia para estas plataformas afectan directamente con la continuidad de los proyectos de ICC y con el desarrollo de nuevos algoritmos, de ahí la necesidad de migrar estos sistemas a plataformas de código abierto que puedan presentar la misma robustez. Esto aunado a que existen pocos dispositivos que sirvan como electroencefalógrafos monetariamente accesibles.

Aura® es una plataforma que permite utilizar su hardware para obtener señales de electroencefalograma (EEG) y es un dispositivo monetariamente accesible comparado con los equipos de grado médico existentes en el mercado, Aura® es compatible con la librería Lab Streaming Layer (LSL). LSL es una herramienta de uso libre que permite capturar las señales obtenidas por los electrodos de Aura®, Python es un lenguaje con

**Tabla 1.** Tabla de precios de dispositivos para ICC.

<b>Dispositivos de grado medico</b>				
<b>Fabricante</b>	<b>Modelo</b>	<b>Costo (USD)</b>	<b>Canales</b>	<b>Velocidad de muestreo</b>
g.tec	USB.Amp	15818	Hasta 16	38.4 Khz
g.tec	USB.Nutilus	7300	Hasta 64	1.024 Mhz
g.tec	USB.Hiamp	64386	Hasta 256	38.4 Khz
<b>Dispositivos comerciales</b>				
<b>Fabricante</b>	<b>Modelo</b>	<b>Costo (USD)</b>	<b>Canales</b>	<b>Velocidad de muestreo</b>
Emotiv	Emotiv Epoc x14	850	14	256 Hz
Mirai Innovation	Aura	3799	8	250 Hz
OpenBCI	CytonBiosensing Board	1543	8	250 Hz

una gran variedad de librerías de desarrollo, fácil de aprender, con una gran comunidad desarrolladores y foros, y es gratuito.

Adicionalmente Python es compatible con librerías robustas para el procesamiento matemático como Scipy y Numpy, para ciencia de datos como Pandas y para machine learning o Deep learning como Teano, Scikit-learn, TensorFlow y Keras por ejemplo, estas librerías pueden ofrecer una gran variedad de posibilidades de mejoramiento a los algoritmos existentes para la interpretación de patrones de MI en las ICC usadas en el campo de rehabilitación.

Por lo anteriormente descrito, en este trabajo se propone una ICC desarrollada en Python para la interpretación de la MI, utilizando la plataforma de Aura® para leer la actividad cerebral de los usuarios, dando como resultado una ICC de precio accesible y desarrollada en un lenguaje de código abierto, logrando evaluar la potencialidad de un electroencefalógrafo de bajo costo para su uso en sistemas de este tipo.

## 2. Trabajos relacionados

En (Martinez-Leon et al., 2016) habla sobre que ha habido poca discusión sobre el rendimiento de los dispositivos para EEG de bajo costo cuando se comparan con dispositivos de EEG profesionales y de investigación. El artículo evalúa un dispositivo de bajo costo, el Emotiv EPOC, en un problema de MI. Se logró concluir que, desde la perspectiva de una aplicación de trabajo final, es demostrado que el rendimiento de este dispositivo es comparable al que se encuentra en dispositivos profesionales cuando se usa el mismo número de sensores y posiciones de sensores.

Como se menciona en (McCrimmon et al., 2017) los sistemas ICC de bajo costo no presentan evidencia suficiente para evaluar si su diseño económico compromete su rendimiento, por lo tanto desarrollaron una ICC portátil de bajo costo y compararon su rendimiento con el de una BCI convencional.

El estudio demuestra que las plataformas ICC integradas basadas en un electroencefalógrafo de bajo costo pueden lograr un rendimiento similar a un sistema ICC convencional que tiene más canales y recursos computacionales. Se menciona que

los sistemas de bajo costo hacen que las ICC sean más accesibles para los usuarios, y aumentan la viabilidad de grandes ensayos clínicos que involucren el uso de las ICC.

Por otro lado, se menciona que los sistemas ICC actuales diseñados con dispositivos de alta calidad no son prácticos para su uso fuera de los laboratorios de investigación debido a su configuración/operación complicada, los costos prohibitivos y falta de portabilidad.

En (Peterson et al., 2020) se menciona que una de las principales limitaciones para el uso generalizado de los dispositivos ICC es el alto costo asociado a un equipo de alta calidad utilizado para capturar las señales biomédicas. Con el objetivo de acercar estos sistemas a los usuarios finales, han surgido diferentes alternativas de bajo costo para el consumidor. Se comenta que aún no se ha explorado cómo estas tecnologías de nivel de consumidor pueden integrarse y usarse para ICC basadas en MI. En (Peterson et al., 2020) Se realiza un análisis de la calidad de las señales adquiridas y la capacidad de detección de MI y también se menciona que encontraron que mediante un método basado en un banco de filtros, se pueden lograr incrementar los rendimientos de clasificación de las ICC construidas bajo dispositivos de bajo de costo logrando brindar evidencia de que se puede construir un ICC basada en MI de bajo costo.

En (García-Moreno et al., 2020) se utilizó una diadema BCI de bajo costo para detectar Imágenes motoras de electroencefalografía (EEG). Además, se discute acerca de la influencia del uso de datos sin procesar. También se discute sobre la influencia de la actividad de ciertas bandas de frecuencia sobre otras en la tarea propuesta. Se menciona que los resultados representan un descubrimiento prometedor para la adopción de tecnologías de bajo costo.

En base a lo anteriormente descrito se propone una ICC desarrollada en Python de bajo costo que pueda interpretar la imaginación de movimiento con una interfaz de usuario específica para dicha aplicación para explorar la potencialidad de una ICC de estas características.

### **3. Materiales y métodos**

#### **3.1. Selección del amplificador para el diseño de la ICC**

Existen diversos dispositivos con los cuales se pueden capturar las señales de electroencefalografía a continuación se presenta la tabla 1 con algunos de los dispositivos de grado médico que ofrecen las mejores prestaciones para poder ensamblar una ICC y a su vez se presentan las opciones comerciales de este tipo de dispositivos. Cabe destacar que el precio expuesto en los dispositivos de grado medico contemplan una configuración básica de componentes, el precio expuesto contempla el gorro y el amplificador, pero existen diversas opciones de electrodos, diversas herramientas para el desarrollo de la aplicación ICC y algunos accesorios extras que se pueden consultar en la página del fabricante.

Como se puede apreciar existe una diferencia significativa en cuanto al precio de los dispositivos comerciales con los dispositivos de grado médico, si bien los dispositivos comerciales no brindan la misma garantía de confiabilidad que los sistemas de grado médico y tampoco las mismas prestaciones son un buen punto de partida para realizar proyectos experimentales de este tipo para poder evaluar su potencialidad real.

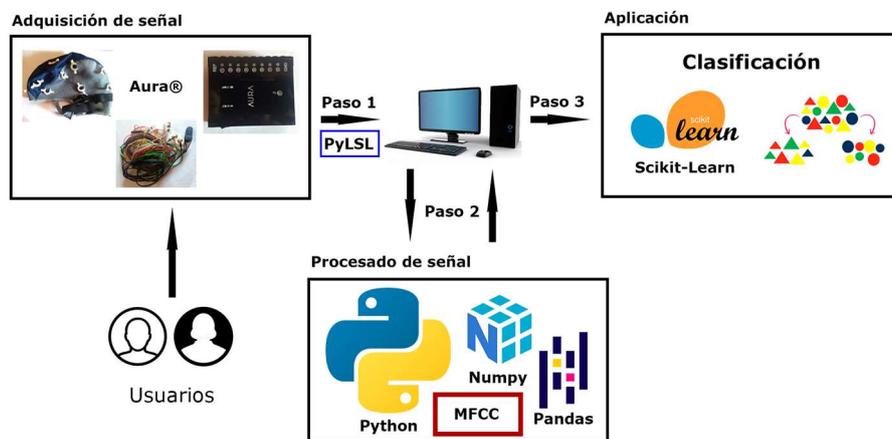


Fig. 1. Esquema propuesto de la ICC.

En general se contempla utilizar un dispositivo que sea capaz de leer la actividad cerebral y mediante software poder traducir las muestras extraídas para poder encontrar los patrones generados por la MI. El dispositivo seleccionado fue Aura, En primera instancia es porque tiene características similares a los demás productos de la misma gama, además de que Aura posee electrodos con un conector estándar como el DIN42802, teniendo así la posibilidad de tener refacciones en caso de que los electrodos dejaran de funcionar o se rompieran.

En segunda instancia porque es un dispositivo que se encuentra disponible en el Consejo de ciencia tecnología e innovación del estado de Guerrero (COCYTIEG) y se logró gestionar que se nos prestara el dispositivo.

### 3.2 ICC propuesta

Con el fin de conseguir el objetivo del presente trabajo se buscaron las herramientas más accesibles que pudieran conseguir buenos resultados de acuerdo con el estado del arte. La propuesta del presente trabajo se presenta con la metodología en la figura 1.

Como se menciona en (Mridha et al., 2021) la arquitectura básica de un sistema ICC consiste en la adquisición de la señal que es el proceso de tomar las muestras de las señales que miden la actividad cerebral, el pre procesamiento que es el proceso mediante él se quita el ruido y las señales indeseadas de las muestras captadas y ayuda a limpiar la señal con diferentes métodos y filtros, la extracción de características que implica analizar la señal y extraer los datos de utilidad empleando algoritmos de procesamiento que permitan la extracción de características del cerebro y la clasificación que ayuda a determinar el tipo de tarea mental que la persona está ejecutando. Otros ejemplos de arquitecturas para ICC bajo este esquema se pueden consultar en (Attallah et al., 2020; Salazar-Ramirez et al., 2019).

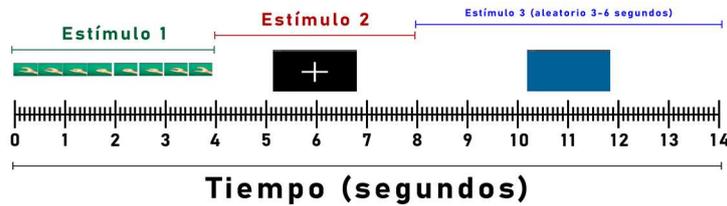


Fig. 2. Línea de tiempo de captura de ensayo.

### 3.3. Adquisición de la señal

La lectura de las señales de la actividad cerebral se tomó con Aura®, se utilizó PyLSL para poder comunicar el software del amplificador de Aura con Python, más información sobre aura en la sección 3.6, para más información sobre las instrucciones y forma en que se adquirieron los datos consultar la sección 3.7.

### 3.4. Procesado de la señal

La señal se procesó utilizando numpy y Pandas, esto debido a que pandas presenta muchas facilidades para exportar los conjuntos de datos a un formato estándar como es CSV, numpy es la librería de Python más popular para el procesamiento matemático, realiza las operaciones de una manera rápida y eficaz, y tiene métodos que ayudan a manipular matrices fácilmente, se utilizó Mel-frequency cepstral coefficients (MFCC) de librosa como extractor de características para encontrar la secuencia de MFCC captada por cada canal, más información acerca de MFCC se puede consultar en la sección 3.8. Adicionalmente se utilizaron filtros incluidos en el software del fabricante usando un pasabandas de 5-50 Hz y un elimina banda de 60 Hz esto es debido a que son los que el fabricante recomienda como óptimos.

### 3.5. Aplicación

El fin del presente proyecto tiene una parte de naturaleza exploratoria por lo que la fase de aplicación de esta versión consiste en evaluar el rendimiento de los clasificadores considerados para posteriormente conjuntar el sistema con los elementos que presenten los mejores rendimientos. Para más información sobre los clasificadores seleccionados consultar la sección 3.9.

### 3.6. Aura

Aura® es un producto de Mirai Innovation que puede ser utilizado como electroencefalógrafo, electromiógrafo, electrocardiógrafo y electrooculógrafo. Tiene dos formas de enviar las señales a la computadora que son por medio de un cable USB o por Bluetooth. También incluye un software que puede visualizar la actividad

cerebral en tiempo real con algunos filtros pasa bandas y elimina banda predeterminados en el sistema.

### **3.7. Adquisición**

Estudios como en (Batista et al., 2019) mencionan que existen fenómenos como los eventos relacionados a la sincronización (ERS) y eventos relacionados a la desincronización (ERD) que pueden ayudar a identificar los patrones generados por la MI. En base a esto se utiliza un filtro pasa bandas que tiene por defecto el software de Aura® (5-50 Hz) ya que como se describe en (Batista et al., 2019) el ritmo sensoriomotor (SMR) se encuentra entre el ritmo mu (8 – 12 Hz) y el ritmo beta (18 – 30 Hz) generalmente entre (12 – 17 Hz) aunque puede variar en algunas personas.

Se creó un conjunto de datos donde se puede diferenciar la MI de un estado de concentración que son las clases para evaluar por el clasificador y llamaremos clase 1 a la MI y clase 2 al estado de concentración.

Se pidió a los voluntarios estar lo más concentrados posible, que evitaran hacer parpadeos bruscos o muy constantes, evitar hacer movimientos que involucren a los músculos como los de las piernas o brazos, evitar respirar profundamente y tener una posición cómoda en la cual pudiesen estar lo más relajados posible. La captura del ensayo se ilustra en la línea de tiempo mostrada en la figura 2.

Los ensayos se capturan a una velocidad de 125 muestras por segundo y se usaron tres estímulos diferentes para mostrar en pantalla, cada uno indica lo que debe hacer el usuario. El primer estímulo muestra una animación de una mano que abre y cierra el puño, este estímulo indica al usuario que debe empezar a imaginar el movimiento para generar el patrón de MI, el segundo estímulo muestra una cruz que indica al usuario que debe dejar de imaginar el movimiento y concentrarse en observar la cruz en pantalla, el tercer estímulo muestra una pantalla azul que indica al usuario un descanso si es que necesita hacer un parpadeo prolongado para lubricar sus ojos, reacomodarse, rascarse o atender alguna incomodidad.

Para el primer y segundo estímulo se capturan 500 muestras respectivamente que toma un tiempo aproximado de 4 segundos para cada uno, el tercer estímulo dura un intervalo aleatorio de entre 3 a 6 segundos y sirve como señal para que el usuario pueda atender alguna incomodidad. Un ensayo se compone de las muestras obtenidas de un recorrido de estos 3 estímulos debidamente etiquetados con respecto al estímulo correspondiente capturado, se capturaron 25 ensayos por sujeto y se trabajó con 10 sujetos.

### **3.8. Extractor de características**

Se propone Mel Frequency Cepstral Coefficients (MFCC) como un extractor de características debido a que en (Medina et al., 2018) se menciona que se ha utilizado la transformada rápida de Fourier y la densidad de potencia espectral como extractores de características, y MFCC está basado en la transformada rápida de Fourier y tiene una base similar, además MFCC comúnmente se utiliza para el reconocimiento de patrones



Fig. 3. Interfaz gráfica para visualización de señales.



Fig. 4. Interfaz gráfica para visualización de señales.



Fig.5. ICC propuesta funcionando.

de voz con lo cual trabaja con señales de frecuencia al igual que la frecuencia se utiliza para poder interpretar la actividad cerebral.

Por defecto en diferentes scripts encontrados los coeficientes buscados van desde los 13 hasta los 20, una opción determinista para saber cuántos coeficientes encontrar sería probar por fuerza bruta con cuantos coeficientes da mejores resultados el clasificador,

pero para evitar excesivos consumos computacionales y para homogenizar los conjuntos de datos se propone usar uno de los valores por defecto que es 13.

### **3.9. Clasificador**

Como segunda fase de la etapa de procesamiento se evaluó con 4 diferentes modelos de clasificación además del Análisis discriminante lineal (LDA) que es el modelo más popular en ICC de este tipo. Se utilizaron algunos de los modelos de clasificación incluidos en la librería de Sklearn tal como Análisis discriminante lineal (LDA), Naive Bayes (NB), AdaBoost, vecinos más cercanos (KNN) y el clasificador de bosque aleatorio (RF), esto sustentado en que de manera general son los modelos más utilizados para la tarea de clasificación y existen algunos trabajos documentados donde utilizan estos clasificadores para ICC basados en la interpretación de MI.

En (Behri et al., 2018) se utilizan Naive Bayes, KNN y RF, específicamente, en (Miao et al., 2019) se utiliza Adaboost y en (Aggarwal & Chugh, 2019) utilizan LDA. Todos los clasificadores se utilizaron con sus valores por defecto en la librería de Sklearn.

### **3.10. Preparación del conjunto de datos y clasificación**

Una vez obtenidos los 25 ensayos requeridos se cuenta con los archivos CSV correspondientes a cada ensayo teniendo 9 columnas que corresponden a los 8 canales del amplificador y a la clase etiquetada de las muestras y 1000 filas que son las muestras obtenidas durante el ensayo que son 500 por cada clase.

Para cada archivo se utiliza Pandas para leerlo individualmente y transformarlo en matriz, se utiliza Numpy para transformar cada matriz de forma que se puedan extraer los 13 coeficientes de Mel en cada canal y se aplica el filtro MFCC.

Finalmente se obtienen dos filas, una para cada clase que llamaremos observaciones de 104 columnas que son los 13 coeficientes de Mel extraídos de cada canal. El filtro se aplica para cada uno de los ensayos, se obtienen las observaciones correspondientes a cada uno y se unen en un solo conjunto de datos.

El conjunto de datos final contiene un número de observaciones equivalente al doble del número de ensayos realizados ya que se obtiene una observación de la clase 1 y una de la clase 2 por cada ensayo. El conjunto de datos final es el que se utiliza para usar los 5 clasificadores considerados y obtener las métricas consideradas para evaluar el sistema mencionadas en la sección 3.11.

### **3.11. Validación**

Como parámetros de validación del sistema se consideraron dos enfoques. El primer enfoque consiste en la validación cruzada usando leave one out (LOO) ya que como se comenta en (Magnusson et al., 2019) es un enfoque para evaluar la generalización de un modelo y arroja datos más exactos del desempeño de un conjunto de datos. El segundo enfoque consiste en usar el método tradicional para obtener los conjuntos de prueba y entrenamiento haciendo 500 iteraciones usando el 25% de los datos aleatoriamente como conjunto de prueba, los resultados se guardan para después

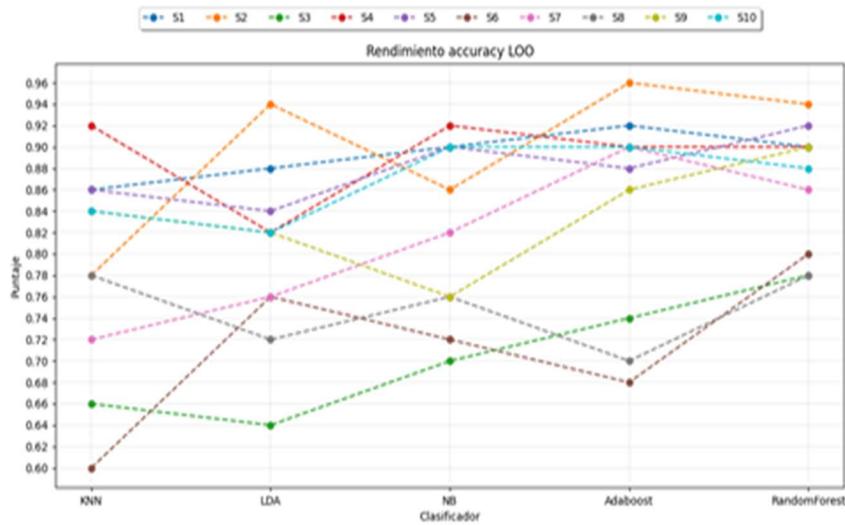


Fig. 6. Resultados de accuracy con LOO.

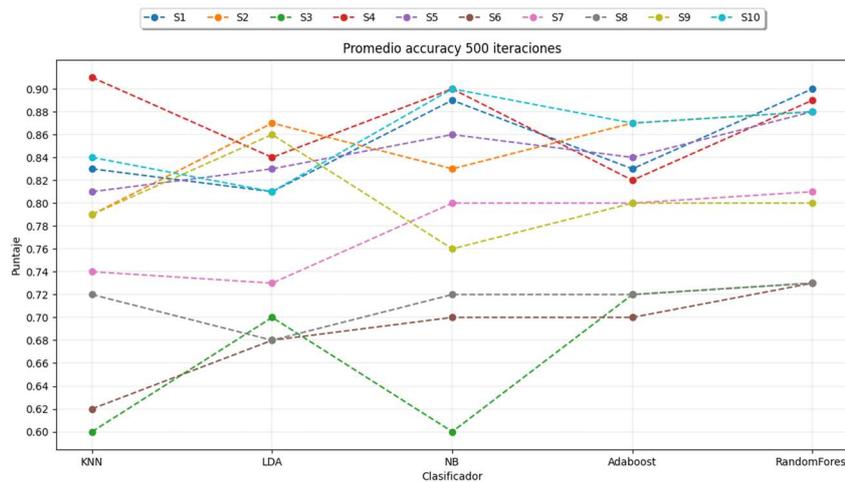


Fig. 7. Resultados de accuracy promedio con 500 iteraciones.

calcular el promedio de todas las iteraciones realizadas, se considera obtener el promedio de las 500 iteraciones de las métricas de accuracy, precision y puntaje f1.

AdaBoost (84.4%) y Naive Bayes (82.4%) fueron los clasificadores que mejor rendimiento promedio tuvieron.

Para el enfoque por promedio de los resultados obtenidos en 500 iteraciones usando el 25% de datos como validación se presentan las figuras 7, 8 y 9 que corresponden al resumen de los resultados obtenidos representados por gráficas para las métricas de accuracy, precision y puntaje F1 respectivamente.

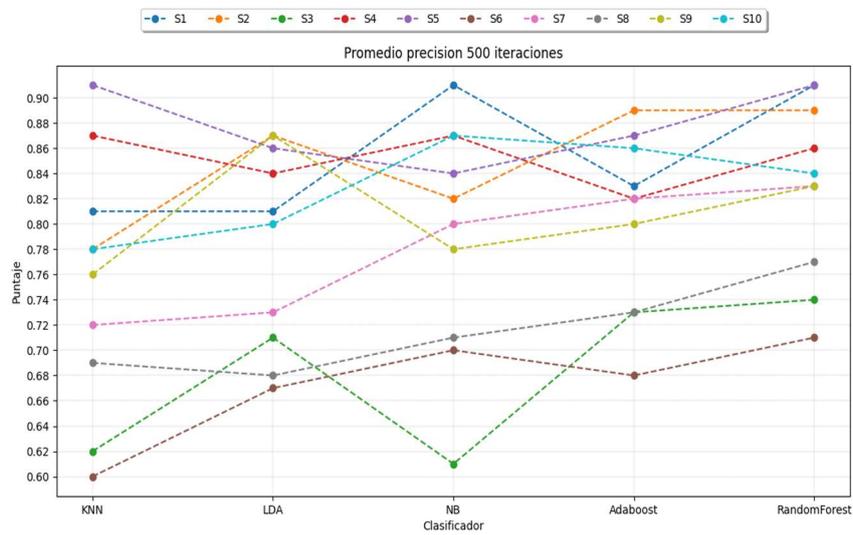


Fig. 8. Resultados de precisión promedio con 500 iteraciones.

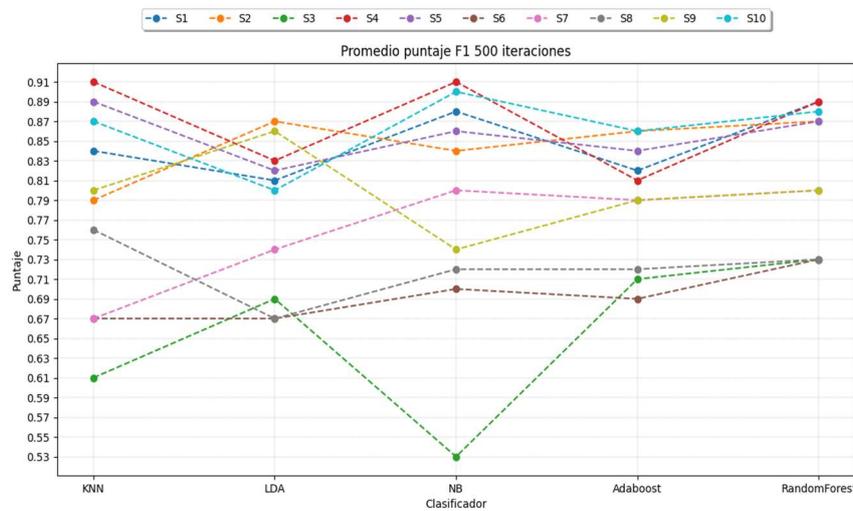


Fig. 9. Resultados de puntaje F1 promedio con 500 iteraciones.

Como se puede observar en las gráficas del porcentaje promedio de 500 iteraciones en los resultados de accuracy, precision y puntaje F1 ronda entre el 60% y 91% a excepción del puntaje obtenido por Sujeto 3 (S3) en puntaje F1 con Naive Bayes que fue de 53%. En estas tres métricas consideradas los clasificadores que tuvieron los mejores puntajes promedio generales utilizando la metodología propuesta fueron

RandomForest(82.3%, 82.9%, 81.9%), AdaBoost(79.7%, 80.3%, 78.9%) y Naive Bayes(79.6%, 79.1%, 78.8%).

#### 4. Resultados

A continuación, se presentan las ventanas realizadas para hacer funcionar el sistema.

Como se puede apreciar en la figura 6 se obtuvieron resultados desde el 60% hasta el 96% de clasificación correcta con los diferentes clasificadores utilizados. Los resultados por sujeto están representados por cada línea correspondiente a la letra "S" y el número que le correspondió al realizar sus ensayos, RandomForest (86.6%).

De acuerdo con las características consideradas en este estudio se observó que el clasificador que mejor rendimiento general obtuvo bajo estas condiciones particulares fue RandomForest con un promedio de 86.6% evaluándolo con LOO y a su vez fue el que mejor rendimiento general obtuvo usando el promedio de resultados obtenidos en 500 iteraciones que fue de 82.3%, 82.9% y 81.9% para accuracy, precision y puntaje F1 respectivamente.

En el aspecto individual de los sujetos evaluados el mejor resultado con LOO fue el obtenido por el Sujeto 2 con el clasificador de AdaBoost que fue de 96%, en el enfoque por iteraciones el mejor resultado en la métrica de accuracy fue el obtenido por el sujeto 4 con KNN, en precision el Sujeto 1 en Naive Bayes y Random Forest, así como el Sujeto 5 en KNN obtuvieron un puntaje de 91% y en puntaje F1 el mejor resultado lo obtuvo el Sujeto 4 con KNN y NB con un puntaje de 91% para ambos clasificadores.

En contraste con los resultados obtenidos en (Behri et al., 2018) donde el enfoque es utilizar WPD y MSPCA, se puede observar que en el trabajo de (Behri et al., 2018) los mejores clasificadores fueron KNN(94.57%), SVM(89.48%) y RandomForest(78.09%) utilizando validación cruzada de 10-folds para evaluarlos en la métrica de accuracy, en el presente trabajo los mejores clasificadores fueron RandomForest(86.6%), AdaBoost(84.4%) y Naive Bayes(82.4%) utilizando LOO para evaluarlos en accuracy.

Con respecto a los modelos de clasificación coincidentes entre ambos trabajos se observa que en el presente trabajo se obtuvieron resultados más bajos de clasificación con KNN que los presentados en (Behri et al., 2018), obteniendo 78.6% para esta propuesta y 94.57% para el trabajo de (Behri et al., 2018). Pero se obtuvieron mejores resultados para el clasificador RandomForest con un puntaje de 86.6% en nuestro trabajo y de 78.09% en el trabajo de (Behri et al., 2018).

En (Miao et al., 2019) se evalúa con dos conjuntos de datos públicos el rendimiento de AdaBoost además de otros dos clasificadores utilizando dos enfoques de extracción de características, el clasificador AdaBoost tuvo mejor rendimiento promedio que los otros dos clasificadores, alcanzando un puntaje de accuracy de 78.4 y 80.6% para los conjuntos 1 y 2 respectivamente utilizando validación cruzada de 10-Folds para evaluar su rendimiento. En este trabajo se obtuvo un rendimiento de AdaBoost del 84.4% que es mayor que el resultado expuesto en el trabajo de (Miao et al., 2019) y se obtuvo utilizando validación cruzada con LOO. Lo cual es evidencia que el proceso de extracción de características con MFCC funciona eficientemente para ciertos clasificadores y puede considerarse una ventaja con respecto a otros trabajos.

Es importante considerar que el rendimiento de la ICC propuesta depende mucho de que la forma en que se adquieren las muestras sea lo más controlada posible, el factor más importante es garantizar el contacto adecuado entre el cuero cabelludo y el electrodo, otro factor importante es el estado del usuario al usar el sistema ya que factores como el sueño, el desinterés y la desconcentración afectan negativamente el rendimiento que obtienen los datos al ser clasificados, otros factores que afectan negativamente el rendimiento son los movimientos involuntarios como los tics por nervios o ansiedad.

Los ambientes utilizados para la experimentación con los usuarios se controlaron de la mejor manera posible, pero factores fuera del alcance de controlarlos como los ruidos de los automóviles, puertas, voces de fondo, el sonido de los pasos, etc. No se lograron suprimir del todo. Además de que la experimentación se dio con voluntarios que quisieran utilizar el sistema, pero algunos mostraron notables signos de desinterés o falta de atención.

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

Se puede observar que la ICC obtuvo un rendimiento competitivo en los conjuntos de datos creados para la clasificación logrando el mejor puntaje promedio evaluado con leave one out de un 86.6%, a su vez en los promedios generales obtenidos por ejecutar 500 iteraciones usando el 25% aleatorio de los datos generados para validación del clasificador RandomForest presenta los puntajes más altos para la métricas evaluadas que son accuracy, precision y puntaje F1 con un puntaje de 82.3%, 82.9%, 81.9% respectivamente, es importante observar que AdaBoost y Naive Bayes mostraron rendimientos aceptables también, logrando un puntaje de 84.4% y 82.4% respectivamente en accuracy evaluándolos con leave one out y al ejecutar 500 iteraciones usando el 25% aleatorio de los datos generados para validación del clasificador AdaBoost obtuvo los puntajes de 79.7%, 80.3% y 78.9% para las métricas de accuracy, precisión y puntaje F1 respectivamente, y Naive Bayes 79.6%, 79.1% y 78.8% para dichas métricas.

Finalmente tenemos una propuesta de ICC de bajo costo desarrollada en una plataforma de código abierto que proporciona un rendimiento aceptable al interpretar y clasificar los patrones generados por la MI del usuario, siendo también flexible y escalable con la posibilidad de probar y desarrollar diferentes scripts para mejorar el rendimiento del sistema. El trabajo realizado es una primera versión de una ICC con estas características, se deben realizar más pruebas para explotar al máximo la potencialidad del sistema.

**Agradecimientos.** Agradecemos a Conacyt por financiar proyectos de ciencia como este, al Consejo de Ciencia y Tecnología del Estado de Guerrero (COCYTIEG) por habernos prestado el sistema Aura® utilizado para realizar este proyecto, al Ing. Leo Enver Garibay Campos jefe del departamento de fortalecimiento del COCYTIEG por intervenir como responsable del sistema Aura® a su cargo y también a todos los voluntarios que participaron en este experimento.

## Referencias

1. Aggarwal, S., Chugh, N.: Signal processing techniques for motor imagery brain computer interface: A review. *Array*, vol. 1-2 (2019) doi: 10.1016/j.array.2019.100003
2. Attallah, O., Abougharbia, J., Tamazin, M., Nasser, A. A.: A BCI system based on motor imagery for assisting people with motor deficiencies in the limbs. *Brain Sciences*, vol. 10, no. 11, pp. 864 (2020) doi: 10.3390/brainsci10110864
3. Batista, E., Serracín, S., Gómez, L., Moreno, R., Moreno, I.: Análisis y caracterización de señales eléctricas del cerebro utilizando interfaz cerebro computador. *Revista de Iniciación Científica*, vol. 5, pp. 110–113 (2019) doi: 10.33412/rev-ric.v5.0.2395
4. Behri, M., Subasi, A., Qaisar, S. M.: Comparison of machine learning methods for two class motor imagery tasks using EEG in brain-computer interface. In: *Advances in Science and Engineering Technology International Conferences (ASET)*, pp. 1–5 (2018) doi: 10.1109/ICASET.2018.8376886
5. Fernández-Gómez, E., Sánchez-Cabeza, Á.: Imaginería motora: revisión sistemática de su efectividad en la rehabilitación de la extremidad superior tras un ictus. *Rev. Neurol. (Ed. Impr.)*, vol. 66, no. 5, pp. 137–146 (2018)
6. Garcia-Moreno, F. M., Bermudez-Edo, M., Rodríguez-Fórtiz, M. J., Garrido, J. L.: A CNN-LSTM deep Learning classifier for motor imagery EEG detection using a low-invasive and low-Cost BCI headband. In: *16th International Conference on Intelligent Environments (IE)*, pp. 84–91 (2020) doi: 10.1109/IE49459.2020.9155016
7. Khan, M. A., Das, R., Iversen, H. K., Puthusserypady, S.: Review on motor imagery based BCI systems for upper limb post-stroke neurorehabilitation: From designing to application. *Computers in Biology and Medicine*, vol. 123, (2020) doi: 10.1016/j.compbimed.2020.103843
8. Lotte, F., Bougrain, L., Cichocki, A., Clerc, M., Congedo, M., Rakotomamonjy, A., Yger, F.: A review of classification algorithms for EEG-based brain-computer interfaces: a 10 year update. *Journal of Neural Engineering*, vol. 15, no. 3, 31005 (2018)
9. Magnusson, M., Andersen, M., Jonasson, J., Vehtari, A.: Bayesian leave-one-out cross-validation for large data. In: *International Conference on Machine Learning*, vol 97, pp. 4244–4253 (2019)
10. Martínez-Leon, J. A., Cano-Izquierdo, J. M., Ibarrola, J.: Are low cost Brain computer interface headsets ready for motor imagery applications? *Expert Systems with Applications*, 49, pp. 136–144 (2016) doi: 10.1016/j.eswa.2015.11.015
11. McCrimmon, C. M., Fu, J. L., Wang, M., Lopes, L. S., Wang, P. T., Karimi-Bidhendi, A., Liu, C. Y., Heydari, P., Nenadic, Z., Do, A. H.: Performance assessment of a custom, portable, and low-cost brain-computer interface platform. *IEEE Transactions on Biomedical Engineering*, vol. 64, no. 10, pp. 2313–2320 (2017) doi: 10.1109/TBME.2017.2667579
12. Medina, B., Sierra, J. E., Ulloa, A. B.: Técnicas de extracción de características de señales EEG en la imaginación de movimiento para sistemas BCI. *Revista ESPACIOS*, vol. 39, no. 22 (2018)
13. Miao, Y., Yin, F., Zuo, C., Wang, X., Jin, J.: Improved RCSP and AdaBoost-based classification for motor-imagery BCI. In: *IEEE International Conference on Computational Intelligence and Virtual Environments for Measurement Systems and Applications (CIVEMSA)*, pp. 1–5 (2019) doi: 10.1109/CIVEMSA45640.2019.9071599
14. Mridha, M. F., Das, S. C., Kabir, M. M., Lima, A. A., Islam, M., Watanobe, Y.: Brain-computer interface: Advancement and challenges. *Sensors*, vol. 21, no. 17, pp. 5746 (2021) doi: 10.3390/s21175746

15. Peterson, V., Galván, C., Hernández, H., Spies, R.: A feasibility study of a complete low-cost consumer-grade brain-computer interface system. *Heliyon*, vol. 6, no. 3, pp. e03425 (2020) doi: 10.1016/j.heliyon.2020.e03425
16. Roy, G., Bhoi, A. K., Das, S., Bhaumik, S.: Cross-correlated spectral entropy-based classification of EEG motor imagery signal for triggering lower limb exoskeleton. *Signal, Image and Video Processing*, vol. 16, pp. 1–9 (2022) doi: 10.1007/s11760-022-02142-1
17. Salazar-Ramirez, A., Martin, J. I., Martinez, R., Arruti, A., Mugerza, J., Sierra, B.: A hierarchical architecture for recognising intentionality in mental tasks on a brain-computer interface. *Plos One*, vol. 14, no. 6, pp. e0218181 (2019) doi: 10.1371/journal.pone.0218181
18. Teo, W. P., Chew, E.: Is motor-imagery brain-computer interface feasible in stroke rehabilitation? *PM&R*, vol. 6, no. 8, pp. 723–728 (2014) doi: 10.1016/j.pmrj.2014.01.006



## **Análisis del comportamiento de estudiantes de programación durante su proceso de aprendizaje utilizando árboles de decisión**

Víctor-Gonzalo Rivero-Martínez, Maricela Quintana-López,  
Asdrúbal López-Chau

Universidad Autónoma del Estado de México,  
Centro Universitario UAEM Valle de México,  
México

vriverom001@alumno.uaemex.mx,  
{mquintanal,alchau}@uaemex.mx

**Resumen.** El proceso de enseñanza-aprendizaje de la programación es fundamental para el desarrollo de la ciencia y la tecnología, desafortunadamente las estadísticas muestran altos índices de reprobación en materias de programación. Realizar un análisis sobre los factores que influyen en el comportamiento de los estudiantes de programación, permitiría detectar áreas de oportunidad con el fin de proponer mejoras en el proceso. El objetivo del presente trabajo es descubrir patrones, tendencias o comportamientos de estudiantes de programación mediante árboles de decisión. Para obtener los datos, se aplicó una encuesta a 200 estudiantes de programación, posteriormente se generó el árbol de decisión mediante el algoritmo J48 y el método de validación cruzada para las clases correspondientes a los niveles de programación principiante e intermedio, alcanzando en el mejor experimento un 75.5% de acierto y un 24.5% de error. De acuerdo con el árbol de decisión generado se obtuvieron reglas de comportamiento para los alumnos principiantes e intermedios. Los atributos que proporcionan más información son el número de semestres de programación cursados, la herramienta de apoyo que más utilizan, la resolución de dudas por parte del profesor y la ayuda proporcionada por una herramienta computacional.

**Palabras clave:** Programación, clasificación, arboles de decisión.

### **Analysis of the Behavior of Programming Students during their Learning Process Using Decision Trees**

**Abstract.** The teaching-learning process of programming is essential for the development of science and technology, unfortunately statistics show high failure rates in programming subjects. Carry on an analysis of the factors that have influence on the behavior of programming students will allow to detect opportunity areas in order to improve the process. The objective of this work is to discover patterns, trends, or behavior of programming students through decision trees. To obtain the data a survey was applied to 200 programming

students, later the decision trees were generated using the J48 algorithm and the cross-validation method to the classes corresponding to the beginner and intermediate programming levels, reaching in the best experiment 75.5% of success and 24.5% of error. According to the generated decision tree, behavioral rules were obtained for beginner and intermediate students. The attributes that provide more information are the number of semesters of programming coursed, the support tool they use the most, the resolution of doubts by the teacher and the help that a computational tool could provide.

**Keywords:** Programming, classification, decision trees.

## 1. Introducción

El progreso tecnológico influye en diversos sectores como el empleo, el hogar, la educación, el entretenimiento y la industria, entre otros. La humanidad se encuentra en un momento en el que los avances en la inteligencia artificial, el aprendizaje automático, ciencia de datos e internet de las cosas, entre otras, permiten innovar productos y servicios que utilizamos en nuestras actividades diarias. La tecnología forma gran parte de nuestras vidas cotidianas, por lo tanto, se requiere de personas con un conocimiento especializado que les permita modificarla ya sea con el fin de actualizarla o de crear una nueva tecnología.

De acuerdo con [1], la industria 4.0 requiere de la formación y actualización de profesionales que tengan la capacidad de resolver problemas a través de habilidades digitales, por ello es importante que desde sus primeros años en la universidad aprendan a programar.

La enseñanza de la programación usualmente consiste en impartición de conocimientos y realización de prácticas en una sala de cómputo. De esta forma, se les hace hincapié en que, para aprender a programar hay que programar de manera constante y aumentando la complejidad de manera progresiva tomando como base conceptos anteriores. Programar es crear la solución a un problema utilizando un lenguaje de programación, lo cual requiere un esfuerzo importante de razonamiento, capacidad de análisis y abstracción para así diseñar y proponer la solución más adecuada, por lo que se considera a la tarea de programar una tarea compleja [2].

Aunado a esto se encuentra el entorno de programación donde la función del compilador es detectar los errores léxicos, sintácticos y algunos errores semánticos, como los que pueden ocurrir al realizar la comprobación de tipos. Que un estudiante no pueda corregir este tipo de errores, implica un desconocimiento del lenguaje, así como la falta de interpretación para establecer la relación entre el error y el tema que lo causa.

Además, se ha identificado que un porcentaje importante del grupo se apoya en el docente para corregir los errores de compilación, pero debido al número de estudiantes, no se puede dar una atención personalizada en el tiempo que se tiene causando en ocasiones frustración en los estudiantes.

Por lo anterior, sería conveniente tener información acerca del comportamiento y opinión de los estudiantes en su proceso de aprendizaje, que se pueda analizar con el fin de descubrir patrones o tendencias para posteriormente detectar áreas de oportunidad y proponer líneas de mejora a dicho proceso. Para lograr lo anterior, en este trabajo se utilizará el algoritmo J48 utilizando ocho atributos y las clases

principiante e intermedio del nivel de programación. A partir del árbol generado se observarán los atributos que aportan mayor información al análisis y las reglas generadas, lo cual corresponderá a las tendencias y comportamientos de los estudiantes.

La presente investigación se organiza de la siguiente manera. En la sección 2 se describen los trabajos relacionados, en la sección 3, el desarrollo del clasificador. En la sección 4, se describen los experimentos realizados y sus resultados. Por último, en la sección 5 se presentan las conclusiones y trabajo futuro.

## **2. Trabajos relacionados**

La problemática de la enseñanza de la programación ha sido abordada en investigaciones realizadas principalmente en el nivel superior motivadas por los altos índices de reprobación en estas asignaturas.

Por ejemplo, en un estudio realizado en el año 2021 en una institución de educación superior en Veracruz, se utilizaron árboles de decisión para determinar el rendimiento académico de los estudiantes de carreras relacionadas con la programación. En el estudio se aplicó un cuestionario a 341 de los estudiantes, obteniendo como resultado que el 48.1% de ellos necesitan apoyo académico y que las variables de aprendizaje en el aula y tutorías tienen relación con la variable rendimiento académico [3].

En otro estudio realizado en el año 2019 para la Licenciatura en Ingeniería de Software de la Unidad Multidisciplinaria Tizimín en Yucatán, se tomaron muestras de los indicadores, de los años 2017 y 2018, de la asignatura de Algoritmia, donde en 2017 el 46.15% tuvo un desempeño entre Satisfactorio y Sobresaliente y el 53.84% entre Suficiente y No aprobado, concluyendo que el rendimiento en general de los grupos no era el óptimo y esto influía en los semestres posteriores [4].

Por otro lado, Fuentes y Moo determinan y clasifican las dificultades que tienen los estudiantes de las ingenierías en sistemas y electromecánica al momento de solucionar problemas mediante la programación básica. La clasificación de las dificultades y su propuesta de solución se presenta en la *¡Error! No se encuentra el origen de la referencia.* [5].

## **3. Metodología**

Para desarrollar el clasificador se utilizó la metodología para el descubrimiento de conocimiento (KDD) propuesta por Fayyad [9], la cual consta de las etapas de selección, preprocesamiento, transformación, minería de datos y, por último, interpretación y evaluación (ver Fig. 1). La explicación de cada etapa se presenta a continuación:

**Selección de datos.** Se reúnen los datos a utilizar en la investigación y se seleccionan a los más relevantes.

**Preprocesamiento y transformación de los datos.** Se seleccionan los atributos a utilizar y son convertidos en un formato computacionalmente apropiado para el análisis de minería de datos.

**Tabla 1.** Clasificación de las dificultades al programar.

<b>Dificultad</b>	<b>Descripción</b>	<b>Propuesta de solución</b>
Fobia a los problemas complejos	Los estudiantes tienden a no solucionar problemas que los impacten	Enseñar al estudiante a dividir el problema en pequeños subproblemas
Lógica incompleta	No consiguen establecer los pasos necesarios para llegar a una solución completa.	Fomentar la solución de acertijos en los que se apliquen los pasos para solucionar un problema.
Desconocen el lenguaje	Desconocen las palabras reservadas o bibliotecas del lenguaje	Elaborar material didáctico que sirva como guía
Desconocer las herramientas del entorno de desarrollo (IDE)	Los estudiantes escribían sus códigos, pero no lograban corregir los errores de lógica.	Enseñar a los estudiantes como correr el programa paso a paso empleando el entorno de programación

**Minería de datos.** Se aplican las técnicas y algoritmos que se encargan de extraer y representar el conocimiento de forma adecuada el propósito de identificar conocimientos, patrones o tendencias de los datos.

**Interpretación y evaluación del conocimiento descubierto.** Se analiza la consistencia de los resultados alcanzados.

## 4. Desarrollo del clasificador

A continuación, se presenta la aplicación de las etapas de la metodología con el fin de clasificar a los estudiantes en principiantes e intermedios, así como obtener las reglas de comportamiento durante su aprendizaje.

### 4.1. Selección de datos

Para este trabajo, la población a considerar fueron los estudiantes de los primeros semestres de programación de las carreras de Ingeniería en Sistemas y Comunicaciones e Ingeniería en computación del Centro Universitario UAEM Valle de México. Esta población actualmente está constituida por aproximadamente 500 estudiantes, por lo que con un nivel de confianza del 93% y un margen de error máximo de 5% y la ecuación para el muestreo aleatorio simple (Ecuación 1), se tuvieron que encuestar a 200 estudiantes mediante un cuestionario en línea consistente en 23 preguntas, 18 de opción múltiple y cinco preguntas abiertas:

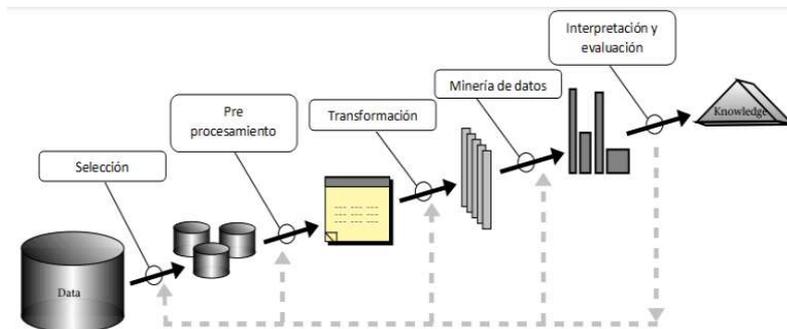


Fig. 1. Etapas del proceso de extracción del conocimiento.

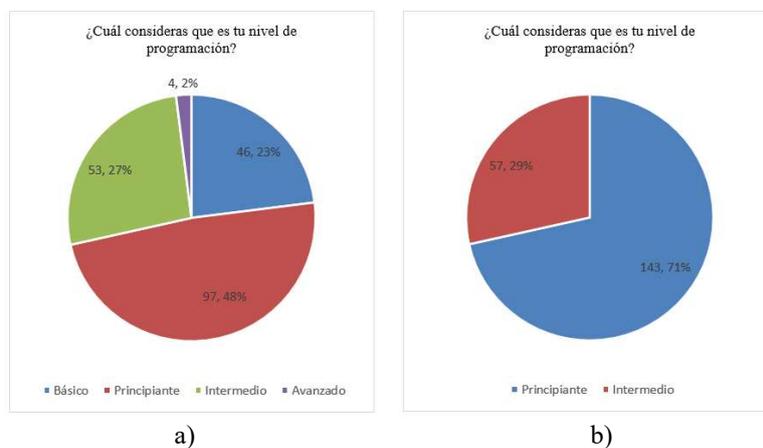


Fig. 2. Preprocesamiento de los resultados de las preguntas a) Preguntado original consistente en 4 opciones b) Reducción a dos opciones.

$$M = \frac{\frac{z^2 * p(1 - p)}{e^2}}{1 + \frac{z^2 * p(1 - p)}{e^2 N}}, \quad (1)$$

donde M es el tamaño de la muestra, N es el tamaño de la población, e es el margen de error y z es la cantidad de desviaciones estándar que una proporción determinada se aleja de la media.

#### 4.2. Preprocesamiento y transformación de datos

Debido a que algunas de las respuestas de las preguntas tenían un porcentaje muy bajo se optó por agrupar las opciones en dos. En la Fig. 2 se muestra un ejemplo de este proceso.

Además, en la encuesta aplicada se eliminaron aquellas preguntas de carácter personal ya que no aportaban información para el estudio, además de las preguntas

**Tabla 2.** Preguntas para el análisis y resultados.

Pregunta	Resultados
Semestres cursados de programación en cualquier lenguaje:	1: 9% 2: 32% 3: 25% 4: 13% 5 o más: 21%
Nivel de programación:	Principiante: 71.5% Intermedio: 28.5%
Tipo de material didáctico utilizado durante el curso:	Video: 60.5% Libro: 4.5% Otra: 1.5%
Si el profesor me dedicara más tiempo aprendería más:	Sí: 57.5% No: 42.5%
Cuando mis compañeros tienen dudas el profesor las resuelve oportunamente:	Sí: 87.5% No: 12.5%
Las clases son generales y no están adaptadas a los objetivos particulares de cada estudiante.	Sí: 57.5% No: 42.5%
Considero que una herramienta computacional ayudaría a mejorar mi aprendizaje.	Si: 89.5% No: 10.5%
De los errores del compilador, sé a qué se refieren y de que tema de estudio se trata:	Frecuentemente: 47% Algunas veces: 53%
Me cuesta trabajo entender los mensajes de error del compilador porque está en otro idioma y eso no me ayuda:	Si: 30.5% No: 69.5%
Considero que cuando no puedo realizar el ejercicio de programación que me dejaron es por qué:	No pude comprender el problema: 16.5% Comprendo el problema, pero no sé cómo resolverlo: 17% Comprendo el problema, ya sé cómo resolverlo, pero no sé cómo programarlo: 39% Comprendo, sé cómo resolverlo, lo programo, pero el compilador me marca errores: 23% Comprendo, sé cómo resolverlo, lo programo pero no hace lo que me pidieron: 4.5%

abiertas. Considerando finalmente un total de ocho preguntas como atributos para la clasificación del nivel de programación en intermedio o avanzado. En la Tabla 2 se muestra como quedaron las preguntas para el análisis y sus resultados.

**Tabla 3.** Experimentos en el Algoritmo J48 para el nivel de programación con 10 pliegues.

No.	Confidencia	Acierto	Error	Matriz de confusión	
				Principiantes	Intermedios
1	0.05	71.5%	28.5%	143	0
				57	0
2	0.1	70.5%	29.5%	141	2
				57	0
3	0.15	72%	28%	138	5
				51	6
4	0.20	75%	25%	131	12
				38	19
5	0.25	75.5%	24.5%	129	14
				35	22

### 4.3. Minería de datos

En esta investigación se utilizó el algoritmo J48 del software Weka [7], el cual genera un árbol de decisión que partición a el conjunto de entrenamiento con base en la selección de atributos que aportan más información para la determinación de la clase. Este proceso es recursivo para cada división que se realiza [8].

El factor de confianza es el parámetro que influye en el tamaño y capacidad de predicción del árbol construido y define la probabilidad de error que se permite en cada operación.

A menor probabilidad, se exige que la diferencia en los errores de predicción antes y después de podar sea más significativa para no podar. El valor por defecto es del 25%, si este valor baja se permiten más operaciones de poda [9].

### 4.4 Evaluación

El modelo que se utilizó en la investigación fue el de validación cruzada con n pliegues, en el cual los datos se dividen en n partes, utilizándose n-1 partes para entrenamiento y uno para prueba, en este caso, como se utilizaron 10 pliegues, 90% de los datos se utilizaron para entrenamiento y 10% para prueba. Este proceso se hace 10 veces y los errores calculados serán el promedio de todas las ejecuciones [9].

### 4.5 Interpretación y uso

Mediante el modelo generado por el algoritmo J48 se analizó el comportamiento de los estudiantes durante su proceso de aprendizaje de programación y para observar la influencia y relación que tienen en la determinación de su nivel de programación.

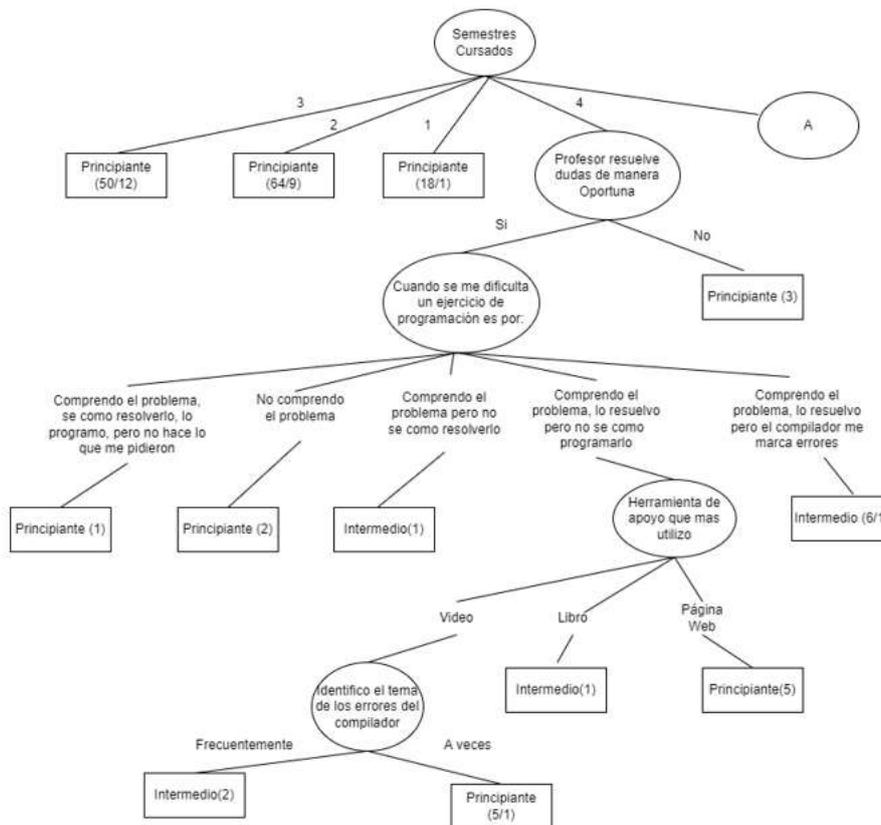


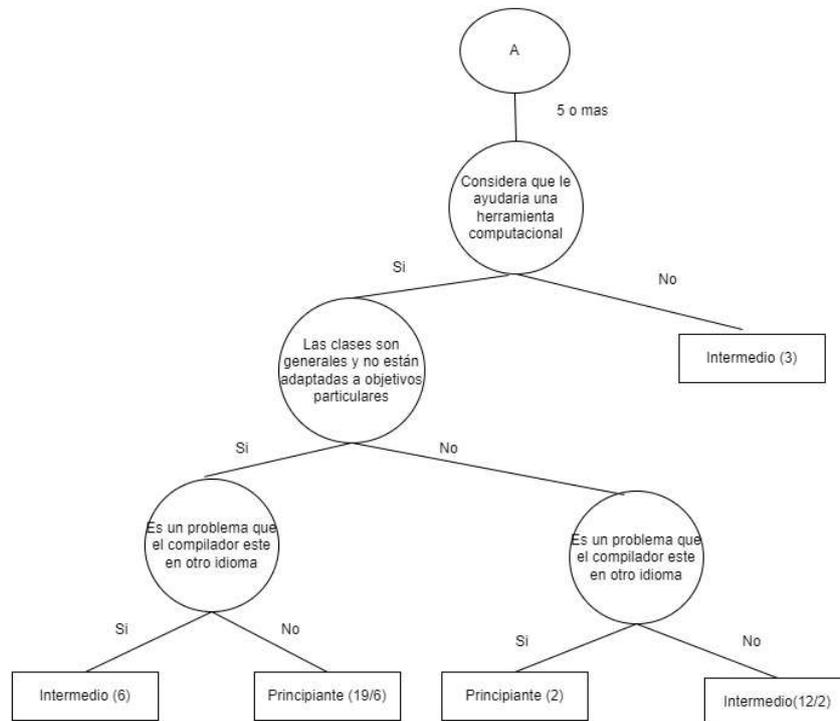
Fig. 3. Rama izquierda del árbol de decisión del nivel de programación.

## 5. Experimentos y resultados

Se generó el clasificador para las 200 instancias (143 principiantes y 57 intermedios) utilizando el algoritmo J48 y se realizó la prueba del modelo utilizando el método de validación cruzada variando el nivel de confianza. Los resultados obtenidos se presentan en la Tabla 3, en la cual se observa que el mejor resultado es alcanzado en el experimento 5 donde con una confianza de 0.25 se alcanza un porcentaje de instancias clasificadas correctamente de 75.5% y un porcentaje de instancias clasificadas incorrectamente de 24.5%.

Para el mismo experimento, la matriz de confusión, indica que, de 143 alumnos que se consideran principiantes, 129 son clasificados correctamente, y 14 incorrectamente como intermedios. Por otro lado, de los 57 intermedios, solo se logra clasificar correctamente a 22 e incorrectamente a 35.

El árbol de decisión generado para el experimento 5 fue dividido a fin de poder presentarlo y analizarlo. La parte izquierda del árbol se presenta en la Fig. 3 y la parte derecha en la Fig. 4. A continuación se realiza un análisis de cada parte.



**Fig. 4.** Rama derecha del árbol de decisión del nivel de programación.

En la Fig. 3. se observa que, si el número de semestres está entre uno y tres, el estudiante es clasificado como principiante, específicamente, 132 instancias se clasifican como principiantes (18 para uno, 64 para dos y 50 para tres), con 22 instancias mal clasificadas.

También que un estudiante se considera principiante cuando su programa no hace lo que le pidieron o no comprenden el problema que les solicitan resolver. Por otro lado, los intermedios comprenden el problema, saben cómo resolverlo, lo programan, pero el compilador les marca errores.

En el mismo árbol se pueden observar que los estudiantes cuando comprenden el problema y no saben cómo resolverlo, recurren a una herramienta de apoyo, los principiantes recurren a páginas web y video tutoriales, mientras que los intermedios a libros y video tutoriales.

En la Fig. 4 se observa a los estudiantes han llevado 5 o más cursos de programación, de los cuales se observa que son intermedios si consideran que no necesitan una herramienta computacional de apoyo para su aprendizaje, aunque se siguen considerando intermedios si consideran varios factores como la ayuda de la herramienta computacional, que las clases son generales y no están adaptadas a sus objetivos particulares y que es un problema que el compilador este en otro idioma, lo cual habla que consideran importante el entorno de programación.

Se puede destacar de los alumnos intermedio que, si no consideran sus clases generales, entonces no es un problema que el compilador este en otro idioma, lo que puede hablar de cierta independencia.

Por otro lado, se consideran principiantes si para ellos es importante la herramienta computacional, que las clases con generales, pero no consideran que sea un problema que el compilador este en otro idioma, esto puede ser debido a que no toman en cuenta los mensajes del compilador.

Las reglas de comportamiento que se generaron en el árbol de decisión son las siguientes:

### **Reglas de comportamiento de los estudiantes.**

#### **Los alumnos son principiantes si:**

1. Han cursado de uno a tres semestres de programación.
2. No consideran que su profesor resuelva dudas de manera oportuna.
3. Si al momento de realizar un ejercicio de programación han cursado cuatro semestres, consideran que su profesor revuelve las dudas oportunamente y cuando les dejan un programa para resolver un problema ellos:
  - Comprenden el problema, lo resuelven, pero al programarlo no hace lo que les pidieron.
  - No comprenden el problema a resolver.
4. Cuando les dejan un ejercicio de programación, comprenden el problema, saben cómo resolverlo, pero no programarlo, además la herramienta de apoyo que utilizan es una página web y a veces identifican el tema de estudio por medio de los errores del compilador.
5. Si han cursado 5 semestres o más de programación, les gustaría utilizar una herramienta computacional de apoyo y:
  - Consideran que los cursos son muy generales y no están adaptados a sus objetivos particulares.
  - No consideran un problema que el compilador este en otro idioma.

#### **Los alumnos son intermedios si:**

1. Si han cursado al menos 5 semestres de programación y no necesitan utilizar una herramienta computacional de apoyo.
2. Si al momento de realizar un ejercicio de programación han cursado 4 semestres, consideran que su profesor es oportuno en la resolución de dudas y cuando les dejan un programa para resolver un problema ellos:
  - Comprenden el problema, pero no saben cómo resolverlo.
  - Lo programan y el compilador les marca errores,
  - Cuando les dejan un ejercicio de programación, comprenden el problema, saben cómo resolverlo, pero no programarlo, la herramienta de apoyo que utilizan es un video tutorial o libro y frecuentemente identifican el tema de estudio por medio de los errores del compilador.

3. Si han cursado 5 semestres o más de programación, les gustaría utilizar una herramienta computacional de apoyo y:
  - No es un problema que los mensajes de error del compilador estén en otro idioma.
  - No consideran relevante que sus clases son generales y no estén alineadas a sus objetivos particulares.

## **6. Conclusiones y trabajo a futuro**

En el presente trabajo se realizó una encuesta a 200 estudiantes de los cuales 143 se consideran principiantes y 57 intermedios. Se aplicó el modelo J48 para obtener el modelo de árbol de decisión obteniéndose un 75.5% instancias clasificadas correctamente y 24.5% de instancias clasificadas incorrectamente. De árbol de decisión obtenido se observa que el atributo que proporciona más información para la clasificación de las clases es la cantidad de semestres que los alumnos han cursado programación, obteniendo que de las 200 instancias 132 alumnos que han cursado de uno a tres semestres, se consideran principiantes. Otro atributo relevante es la herramienta de apoyo que utilizan en su aprendizaje, donde se encontró que el video tutorial es la herramienta que más se utiliza y que si utilizan el libro de texto se consideran intermedios.

En el caso del entorno de programación se puede concluir que, si no comprenden el problema o no lo pueden resolver, entonces se consideran principiantes y si llegan a programarlo y el compilador les marca errores se consideran intermedios. Cabe resaltar que en la encuesta se les pregunto a los alumnos si les gustaría utilizar una herramienta computacional que sirva como apoyo a su aprendizaje, este atributo fue relevante para los estudiantes que ya estudiaron cinco o más semestres de programación, quienes seguramente ya tienen necesidades más específicas en su aprendizaje.

Se considera que el objetivo del presente trabajo se cumplió ya que se descubrieron patrones de comportamiento relevantes de los estudiantes durante su proceso de aprendizaje.

Las áreas de oportunidad que se pueden destacar en este estudio son, que, aunque los estudiantes ya hayan cursado cinco semestres o más, se consideran principiantes, por lo que se puede proponer que elaboren o participen en proyectos con aplicaciones reales o más avanzadas, por otro lado, las herramientas computacionales pueden jugar un rol importante en el soporte a estudiantes durante la codificación ya que manejo del entorno también resulto relevante en este estudio.

Como trabajo futuro queda analizar mediante minería de textos las respuestas de los estudiantes a las preguntas abiertas y verificar si hay alguna relación con el presente análisis.

## **Referencias**

1. Muñoz-La Rivera, F., Hermosilla, P., Delgadillo, J., Echeverría, D.: Propuesta de construcción de competencias de innovación en la formación de ingenieros en el contexto de la industria 4.0 y los objetivos de desarrollo sostenible (ODS). Formación Universitaria, vol. 14, no. 2 (2021) doi: 10.4067/S0718-50062021000200075

2. Compañ-Rosique, P., Satorre-Cuerda, R., Llorens-Largo, F., Molina-Carmona, R.: Enseñando a programar: un camino directo para desarrollar el pensamiento computacional. *Revista de Educación a Distancia.*, no. 46 (2015)
3. Díaz-Martínez, M. A., Ahumada-Cervantes M. A., Melo-Morín, J. P.: Árboles de decisión como metodología para determinar el rendimiento académico en educación superior. *Revista Lallista de Investigación*, vol. 18, no 2 (2022) doi: 10.22507/rli.v18n2a8
4. Narváez Díaz, L. E., López Martínez, R. E.: Propedéutico como estrategia de apoyo en una asignatura de la Licenciatura en Ingeniería de Software de la Unidad Multidisciplinaria Tizimín. *Revista Electrónica sobre Tecnología, Educación y Sociedad*, vol. 7, no. 14, pp. 1–18 (2020)
5. Fuentes-Rosado, J., Moo-Medina, M.: Dificultades de aprender a programar. *Educación en Ingeniería*, vol. 12, no. 24, pp. 76–82 (2017) doi: 10.26507/rei.v12n24.728
6. Osman, A. S.: Data Mining Techniques: Review. *International Journal of Data Science Research*, vol. 2, no. 1 (2019)
7. Weka 3: Machine learning software in java [En línea]. Available: <https://www.cs.waikato.ac.nz/ml/weka/> [Último acceso: 23 04 2022]
8. Cardona-Taborda, C. H., Gelvez-García, N., Palacios-Rozo, J.: Análisis de datos mediante el algoritmo de clasificación J48, sobre un cluster en la nube de AWS. *Redes de Ingeniería*, pp. 3–25 (2017). doi: 10.14483/udistrital.jour.redes.2016.3.a01
9. García-Jiménez, M., Álvarez-Sierra, A.: Análisis de Datos en WEKA – Pruebas de selectividad (2010) [En línea]. Available: <http://www.it.uc3m.es/jvillena/irc/practicas/06-07/28.pdf> [Último acceso: 23 04 2022]
10. Fayyad, U., Piatetky-Shapiro, G., Smyth, P.: The kdd process for extracting useful knowledge from volumes of data. *Communications of the ACM*, vol. 39, no. 11, pp. 27–34 (1996)

## Un método para simular lluvia mediante Motion Blur y ruido sal y pimienta

Roberto-Carlos Romero-Olmos<sup>1</sup>, Sebastián Salazar-Colores<sup>2</sup>,  
Hugo Moreno-Jiménez<sup>2</sup>, Gerardo Flores<sup>2</sup>

<sup>1</sup> Instituto Tecnológico de México campus León,  
Departamento de Ingeniería Eléctrica,  
México

<sup>2</sup> Centro de Investigaciones en Óptica, Guanajuato,  
México

robertocarlosromeroolmos@gmail.com

**Resumen.** La eliminación de los efectos climáticos adversos en imágenes de exteriores tales como neblina, nieve o lluvia es un tema de profundo interés dado el número de posibles aplicaciones en áreas como la conducción asistida y autónoma de autos y drones, teledetección y vigilancia, entre otras. En el desarrollo de métodos heurísticos y de Inteligencia Artificial son empleados datos simulados para diseñar y entrenar sus modelos, por lo que la calidad de la simulación repercute en el desempeño final de los métodos desarrollados. Mientras existen muchos trabajos de simulación de neblina, la simulación de lluvia y nieve han sido poco estudiadas. Por lo que, en este trabajo se propone un método para la generación de lluvia sintética en imágenes, con el fin de crear bases de datos realistas en un extenso número de escenarios, con el cual se podrán entrenar diferentes algoritmos de eliminación de lluvia (deraining) basados en Deep Learning. En los resultados mostrados se observa un mejor desempeño cualitativo en términos de realismo en relación con otros métodos similares, al incluir velos atmosféricos generados por la acumulación de trazas de lluvia, así como diferentes formas y direcciones de las trazas de lluvia.

**Palabras clave:** Simulación eventos climáticos, lluvia, deraining.

### A Method for Simulation of Rain Using Motion Blur and Salt and Pepper Noise

**Abstract.** The elimination of adverse weather effects in outdoor images such as haze, snow or rain is a very interesting topic given the number of possible applications in areas such as assisted and autonomous driving of cars and drones, remote sensing and surveillance, among others. In the development of heuristic and Artificial Intelligence methods, simulated data are used to design and train their models, so the quality of the simulation has an impact on the final performance of the developed methods. There are many works on fog simulation,

but rain and snow simulation have been poorly studied. Therefore, in this work we propose a method for the generation of synthetic rain in images, in order to create realistic databases in a wide number of scenarios, with which different rain elimination algorithms (deraining) based on Deep Learning can be trained. The results presented, show a better qualitative performance in terms of realism in relation to other similar methods, by including atmospheric veils generated by the accumulation of rainfall traces as well as different shapes and directions of the rainfall traces.

**Keywords:** Weather events simulation, rain, deraining.

## 1. Introducción

La simulación de fenómenos naturales como la lluvia, nieve o neblina, es un problema complejo y de gran interés, debido a las aplicaciones que estas pueden tener tanto en la industria, fotografía, juegos, exploración, procesamiento de imágenes, entre otras áreas. Actualmente, la simulación realista de lluvia ha sido poco explorada a pesar del impacto que podría tener para la generación de bases de datos extensas y realistas que impulsen a los algoritmos de Deep Learning para eliminación de lluvia (deraining), estos métodos se verán beneficiados al ser entrenados con grandes bases de datos, que contemplen diferentes escenarios y características, debido a que estas propiedades harán más sencillo que la red aprenda a generalizar el problema.

La lluvia es uno de los fenómenos meteorológicos más comunes, y está perjudica en gran medida la eficiencia de muchos algoritmos de visión, ya que disminuye la visibilidad de la imagen e introduce interferencias indeseables que pueden afectar gravemente el rendimiento de los algoritmos de visión por computador [1, 2, 3, 4]. Para abordar este problema se han propuesto varios métodos para la eliminación de los efectos de la lluvia en imágenes, entre los cuales se encuentran los algoritmos de aprendizaje automático basados en redes neuronales, como ya se mencionó estos métodos mejoran su desempeño al utilizar una base de datos extensa, ya que la precisión del modelo depende en gran medida de la cantidad de datos utilizados y la generalización del problema en diferentes escenarios, para el entrenamiento de la red.

Los métodos de deraining en imágenes singulares se pueden dividir en dos enfoques básicos: basados en modelos físicos (sin aprendizaje profundo) y basados en datos (aprendizaje profundo) [9]:

Los métodos basados en modelos emplean marcos de optimización para el deraining, En general estos métodos solo se ocupan de las trazas de lluvia e ignoran la presencia de acumulación de lluvia, el deraining se realiza mediante la descomposición de la imagen mediante un análisis de componentes morfológicos como en [10, 11, 12].

Los métodos basados en Deep Learning para deraining, emplean redes conjuntas que se encargan de la detección y eliminación de lluvia, estos métodos pueden soportar lluvias intensas, trazas de gotas superpuestas y acumulación de lluvia. En general estas redes pueden detectar las ubicaciones de lluvia al predecir una máscara de lluvia binaria y tomar un marco recurrente para eliminar las trazas de las gotas y aclarar progresivamente la acumulación de lluvia.



**Fig. 1.** Aplicación de lluvia sintética con Photoshop.

Este método logra buenos resultados en casos de lluvia intensa. Sin embargo, podría eliminar texturas verticales falsamente y generar subexposición [6,13,14]. Estos métodos se ven beneficiados al ser entrenados con bases de datos extensas y que contemplen diferentes escenarios, ya que esto permite generalizar el problema.

Actualmente, el método utilizado para la generación de lluvia en imágenes se realiza en gran medida con herramientas de Photoshop, este se basa en añadir una capa blanca y rellenarla con ruido uniforme monocromático, a este se le aplica un filtro de blur gaussiano, tras lo cual se usan niveles para reducir la cantidad de ruido, y aumentar el contraste en el ruido aplicado, al resultado de este proceso se le aplica un MotionBlur para ajustar la dirección en la que la lluvia caerá, al tener esto se reajustan los niveles de blancos y negros según sea necesario, finalmente se utiliza esta capa como "Pantalla" sobre la imagen original obteniendo una imagen con lluvia sintética, también se puede usar el "modo de fusión de pantalla" no lineal que incluye Photoshop en sus librerías. Este método es empleado en la creación de bases de datos de lluvia sintética con variación de la cantidad de lluvia [5].

El método actualmente utilizado, para generar bases de datos sintéticas, no es del todo realista y carece de efectos propios de la lluvia, ver Fig.1, este método claramente genera una superposición de líneas en la imagen original, además si el usuario quisiera tener trazas de lluvia en distintos ángulos tendría que realizar nuevamente el proceso debido a que este solo aplica líneas en un ángulo definido y con un tamaño promedio establecido.

En este trabajo se muestra un primer acercamiento y resultados iniciales al aplicar capas de trazado de lluvia con un método novedoso. El algoritmo propuesto se encarga de crear múltiples capas de lluvia e integrarlas con la imagen a la cual se aplique.

El resto del presente trabajo está distribuido de la siguiente forma. En la sección 2 se muestra la metodología empleada para realizar el proceso propuesto.

En la sección 3 y 4 se expone el uso del proceso y los resultados obtenidos. Por último, en la sección 5 se presentan las conclusiones y perspectivas de investigaciones futuras. El código fuente de este método se encuentra disponible<sup>1</sup>.

---

<sup>1</sup> <https://github.com/RobbRom1206/Rain-Layer>

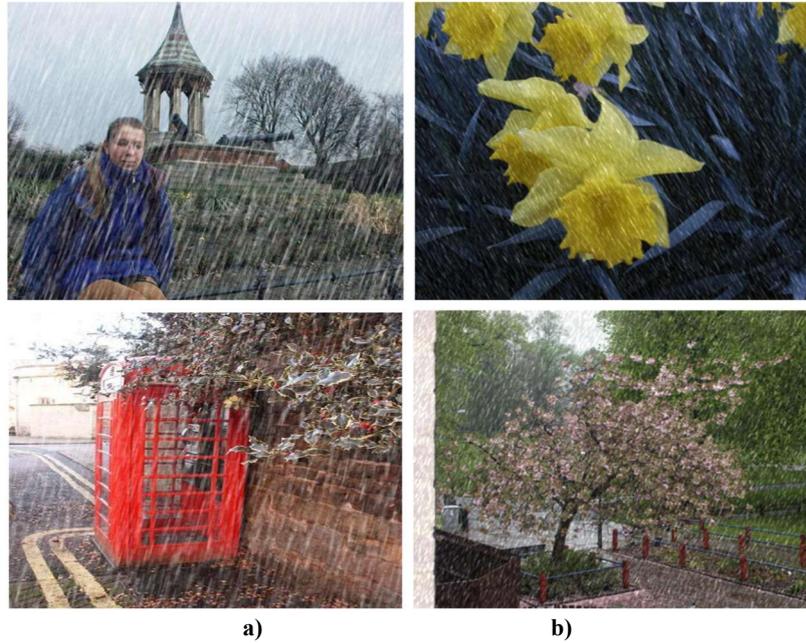


Fig. 2. Comparativa entre el método con Photoshop (a) y el método propuesto (b), al simular lluvia moderada.

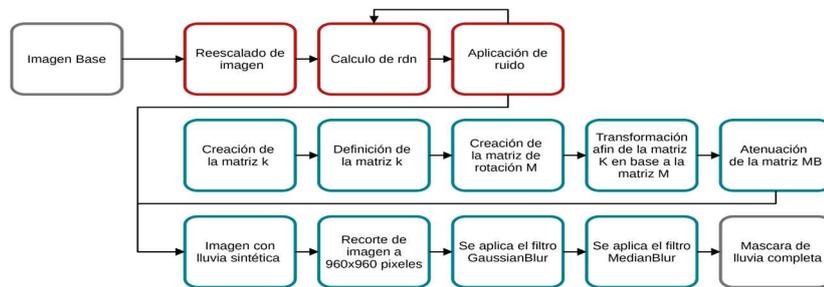


Fig. 3. Proceso de creación de máscara.

## 2. Método propuesto

Capturar las características de un fenómeno tan complejo como la lluvia es difícil y los métodos existentes generan imágenes que no lucen naturales, a continuación, se describe un método de creación de imágenes con lluvia sintética de manera automática. Este método está programado en Python, se encarga del procesamiento de imágenes para aplicar a estas una máscara de lluvia con la cual se podrán crear bases de datos que posteriormente se podrían usar en el entrenamiento de redes neuronales para la eliminación de los efectos de la lluvia (Deraining), ver Fig. 2.

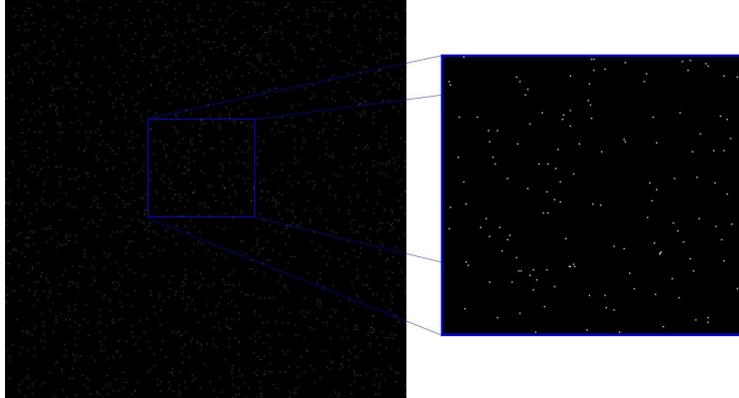


Fig. 4. Ruido de sal para la máscara de lluvia.

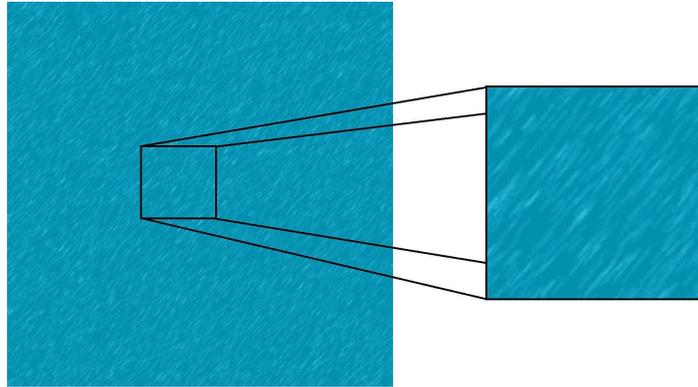


Fig. 5. MotionBlur aplicado a la imagen con ruido de sal (se cambió el fondo a color azul para una mejor muestra del efecto).

El algoritmo propuesto usa como base la aplicación de ruido de sal y pimienta, así como la aplicación de un efecto Motion Blur para simular el movimiento en una dirección de las gotas de lluvia, con ello se generaron capas (máscaras) con lluvia sintética, las cuales se combinarán con la imagen original, estableciendo con una menor intensidad el peso de la capa de lluvia, esto aplicará un efecto parecido a la transparencia en las trazas de lluvia.

Para ello se tuvieron en cuenta varias fases las cuales serán detalladas a continuación:

- 1 A partir de una imagen base (img) se crea la imagen lluvia y se genera ruido blanco para establecer la cantidad de gotas de lluvia que habrá en esta.
  - a. Escalar “img” a 1024 x 1024 píxeles.
  - b. Establecer el umbral “  $u$  ”, valor dado por el usuario donde  $\{u \in \mathbb{R}: 0 < u < 1\}$ .
  - c. Calcular un número aleatorio (“  $rdn$  ”) pixel a pixel, donde  $\{rdn \in \mathbb{R}: 0 < rdn < 1\}$ .



Fig. 6. Aplicación de máscara con el método propuesto.



Fig. 7. Proceso de aplicación de máscara de lluvia.

- d. Aplicar ruido de tipo sal (este se caracteriza principalmente por cubrir de forma dispersa toda la imagen con una serie de píxeles blancos), se genera una imagen “lluvia” llena de ceros con las mismas dimensiones de  $img[i,j]$ , y realiza un barrido de la misma, se compara  $rdn$  y  $u$  pixel a pixel, si  $rdn < u$  el valor del pixel se establece como  $lluvia[i,j]=255$ .
- 2 Aplicar *motion blur* (el cual se define como: “El rastro dejado por los objetos en movimiento en una fotografía o en una secuencia de imágenes” en este caso se aplica para simular el movimiento de la lluvia sintética) a la imagen “lluvia” con ruido de sal, y finalmente, se guarda este modelo como máscara de Lluvia.
    - a. Creación de una matriz  $k$  de ceros con el tamaño que el usuario estipulo “size” para la gota.
    - b. Se introducen unos en la fila central, -1 para asignar el centro de rotación de la imagen.
    - c. Se obtiene la matriz de rotación  $M$  a partir del tamaño de gota “size”, el ángulo y su escala que en este caso será 1.
    - d. Aplicamos una transformación afín a la matriz  $k$  con base en la matriz de rotación  $M$  y el tamaño de “size” dándonos como resultado la matriz  $MB$ .
    - e. Se atenúa el filtro de MotionBlur al hacer una división entre la matriz  $MB$  y 1 sobre la suma de los valores de esta.
    - f. Se aplica el filtro MotionBlur como un filtro 2D a la imagen previamente modificada con ruido de sal “lluvia”, estipulando que esta tendrá la misma profundidad que la imagen original.

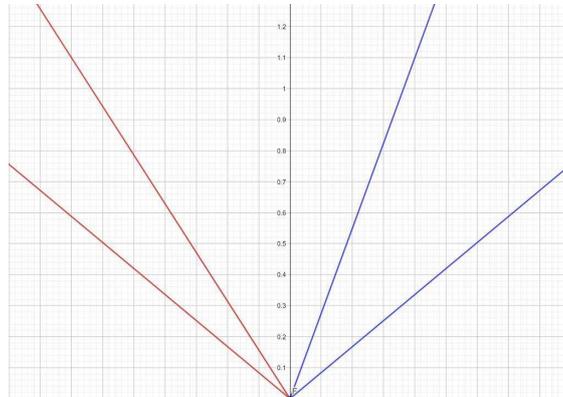


Fig. 8. Apertura del ángulo de selección para aplicación de MotionBlur.

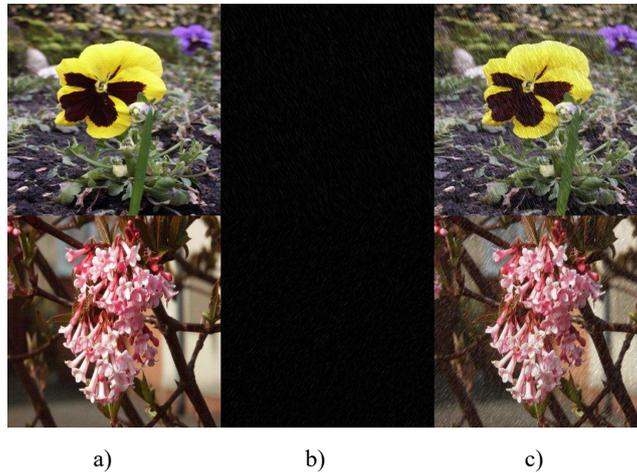


Fig. 9. Muestra de imagen. (a) GT(Ground-Truth) b) Máscara de lluvia c) Máscara de lluvia aplicada.

- g. Se recorta un área de la máscara para usarse como base (con un tamaño total de 960 x 960 píxeles).
- h. Se aplica un filtro GaussianBlur para eliminar un poco el ruido Gaussiano de la imagen.
- i. Se aplica después un filtro MedianBlur para realizar un difuminado de imagen y conseguir un efecto más realista.

### 3. Funcionamiento

En esta sección se muestra el funcionamiento del método, como primer paso se aplicará ruido de sal (puntos blancos en una imagen de fondo negro) para definir la cantidad de gotas de lluvia que nuestra máscara tendrá al final del proceso, ver Fig. 4.



**Fig. 10.** Comparativa entre imágenes creadas con nuestro método y creadas con métodos clásicos, en los 2 tipos de imágenes más usadas en bases de datos de Deraining.

La cantidad de gotas depende del umbral definido por el usuario, con base en el cual se establece la probabilidad de que un píxel de esta se convierta en los puntos blancos antes vistos. Después de ello se aplica un filtro motion blur con el cual se expanden los puntos blancos anteriores hacia un ángulo definido dándonos como resultado la máscara de lluvia que usaremos para la imagen final Fig. 5.

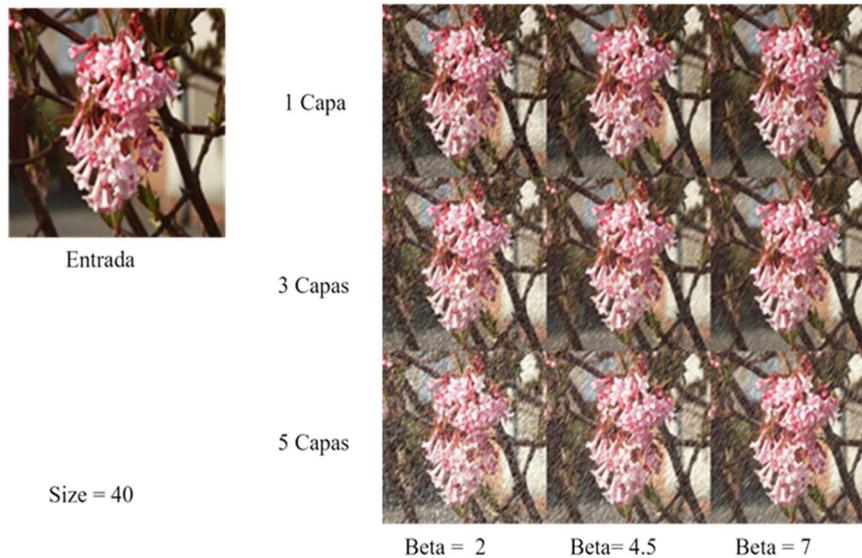
Después de esto se hace una adición de la máscara a la imagen original, realizando una suma pixel a pixel y limitando los valores a 255, para dar el efecto de falsa lluvia, ver Fig. 6, esto se puede realizar múltiples veces con ángulos ligeramente diferentes para crear imágenes más realistas.

#### 4. Uso

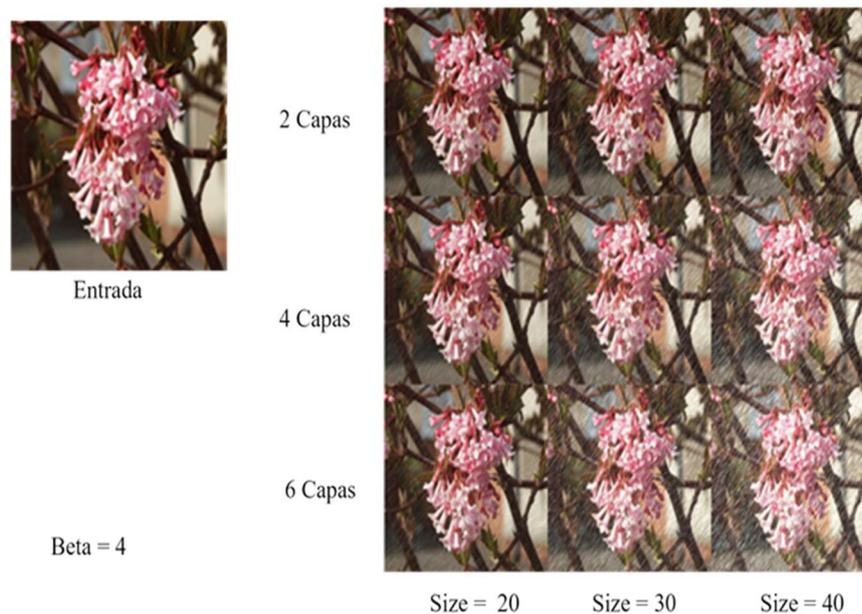
Para llevar a cabo este trabajo se realizó un programa que hace uso de este nuevo método, teniendo como input algunos parámetros que son necesarios, tamaño de la gota (size), umbral (u), ángulo (A), peso de la capa de lluvia (Beta), imagen original (img) y número de capas, este método de manera simple crea una máscara de lluvia con un solo tamaño de raya/gota, a continuación, se muestra un ejemplo el cual se encarga de realizar 3 máscaras de lluvia y las combina en una sola para una emulación más realista y con mejor efecto práctico.

El proceso mostrado anteriormente está diseñado para unir 3 capas de máscara de lluvia cada una con un valor de tamaño diferente, pero respetando el ángulo de estas el cual toma un valor de  $50^\circ$  a  $70^\circ$  y de  $110^\circ$  a  $140^\circ$  con respecto a la horizontal esto para dar una simulación de que las gotas caen de la parte superior de la imagen dándonos como base la siguiente apertura de ángulos para la caída de lluvia:

Ahora utilizando este algoritmo, se tomarán imágenes de pruebas de las cuales usaremos como base las imágenes aleatorias del apartado Ground-Truth del dataset Rain1400 [8] para tener un repertorio inicial de imágenes con las cuales trabajar y posteriormente comparar con su base de imágenes de lluvia sintéticas.



**Fig. 11.** imágenes modificadas por el método propuesto. En este ejemplo se muestran cambios de Beta que se refiere al peso que se usará en la capa de lluvia al unir las imágenes, Size (Tamaño en el cual se estira el ruido de sal aplicado), y una variación de capas lluvia aplicadas en el proceso.



**Fig. 12.** imágenes modificadas por el método propuesto. En este ejemplo se muestran cambios de Beta que se refiere al peso que se usará en la capa de lluvia al unir las imágenes, Size (Tamaño en el cual se estira el ruido de sal aplicado), y una variación de capas lluvia aplicadas en el proceso.

El programa mostrado con anterioridad solo trabajará una imagen por ejecución debido a que fue realizado con el propósito de realizar pruebas, a continuación, se muestran algunas aplicaciones de la máscara a imágenes aleatorias.

Con estas imágenes podemos observar el potencial de la aplicación de un software de este estilo, el cual tiene un tiempo de procesamiento de alrededor de 2 segundos por imagen para la creación de datasets específicos para Deraining, también debe ser tomado en cuenta el ahorro de esfuerzo y atención que el usuario necesita dar a la creación de estos en comparación a otros métodos los cuales necesitan que un usuario esté aplicando las máscaras el mismo una a una.

## **5. Resultados**

Durante el desarrollo de este programa se tomaron en cuenta 3 cosas, la imagen Ground-Truth a la cual se le aplicará una máscara de lluvia, la máscara de lluvia y la imagen resultante, En la figura 9 se muestra la transición de estas 3 fases realizadas durante el procesamiento.

Para terminar de comprobar la calidad de este proceso se comparará la aplicación de la máscara de lluvia contra el procesamiento de imágenes realizado con métodos anteriores basados en Photoshop para la aplicación de diferentes tipos de lluvia, que son usados en bases de datos actuales.

A continuación, se muestran algunos resultados variando los parámetros del método desarrollado Fig. 11 y 12.

## **6. Conclusiones**

En este trabajo se propuso un algoritmo de creación de lluvia sintética para la asistencia en el proceso de producción de bases de datos y simulación de lluvia en imágenes singulares. Obteniendo satisfactorios resultados de este primer acercamiento, lo que muestra la factibilidad del proceso propuesto, ya que el algoritmo es capaz de trabajar y elegir un tamaño y dirección de manera semiautomática y se reduce el tiempo empleado en la creación de estas máscaras.

Hemos observado que en términos generales la emulación de lluvia a través de este método se aproxima más a las características de lluvia real, sin embargo actualmente no se ha realizado un experimento cualitativo extenso sobre la percepción de las personas ante estas imágenes sintéticas, lo cual se trabajará en un futuro, a la vez que se refina el procedimiento y se implementaran algunas mejoras debido a que hemos detectado que las imágenes generadas presentan ciertas incongruencias al trabajar con imágenes con variados puntos de profundidad, por lo que como trabajo futuro esto será mejorado agregando un sistema de identificación de profundidad, también se implementará neblina en las imágenes, así como mejorar la selección de transparencia y tamaño de gota en la imagen, se contemplaran los efectos de la lluvia en diferentes entornos como lo son las marcas generadas en suelo seco o las ondulaciones en espejos de agua. Una vez conseguido lo anterior, se comenzará a realizar pruebas en imágenes más variadas, para un trabajo próximo de deraining en redes neuronales.

## Referencias

1. Girshick, R.: Fast R-CNN. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 1440–1448 (2015)
2. Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C. Y., Berg, A. C.: SSD: Single shot multibox detector. In: Leibe, B., Matas, J., Sebe, N., Welling, M. (eds.) Computer Vision – ECCV 2016. ECCV 2016. Lecture Notes in Computer Science, Springer, Cham, vol. 9905 (2016) doi: 10.1007/978-3-319-46448-0\_2
3. Ren, S., He, K., Girshick, R., Sun, J.: Faster R-CNN: Towards real-time object detection with region proposal networks. *Advances in neural information processing systems*, vol. 28, pp. 91–99 (2015)
4. Chen, Y., Li, W., Sakaridis, C., Dai, D., Van-Gool, L.: Domain adaptive faster r-cnn for object detection the wild. In: IEEE Conference on Computer Vision and Pattern Recognition, pp. 3339–3348 (2018)
5. Zhang, H., Patel, V. I.: Density-aware single image de-raining using a multi-stream dense net-work. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 695–704 (2018)
6. Zhang, H., Sindagi, V., Patel, V. M.: Image de-raining using a conditional generative adversarial network. *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 30, no. 11, pp. 3943–3956 (2020) doi: 10.1109/TCS.VT.2019.2920407
7. Zhu, J. Y., Park, T., Isola, P., Efros, A. E.: Unpaired image-to-image translation using cycle-consistent adversarial networks. In: IEEE International Conference on Computer Vision (ICCV), pp. 2223–2232 (2017)
8. Ren, D., Zuo, W., Hu, Q., Zhu, P., Meng, D.: Progressive image deraining networks: A better and simpler baseline. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 3937–3946 (2019)
9. Yang, W., Tan, R., Wang, S., Fang, Y., Liu, J.: Single image deraining: From model-based to data-driven and beyond. *Transactions on Pattern Analysis and Machine Intelligence*, vol. 43, no. 11, pp. 4059–4077 (2020). doi: 10.1109/TPA.MI.2020.2995190
10. Kang, L. W., Lin, C. W., Fu, Y. H.: Automatic single image-based rain streaks removal via image decomposition. *IEEE Transactions on Image Processing*, vol. 21, no. 4, pp. 1742–1755 (2012) doi: 10.1109/TIP.2011.2179057
11. Luo, Y., Xu, Y., Ji, H.: Removing rain from a single image via discriminative sparse coding. In: Proc. IEEE Int'l Conf. Computer Vision, 2015, pp. 3397–3405
12. L. Zhu, C. Fu, D. Lischinski, and P. Heng.: Joint bilayer optimization for single-image rain streak removal. In: Proceedings of the IEEE International Conference on Computer Vision (ICCV), pp. 2545–2553 (2017)
13. Qian, R., Tan, R. T., Yang, W., Su, J., Liu, J.: Attentive generative adversarial network for raindrop removal from a single image. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 2482–2491 (2018)
14. Li, R., Cheong, L. F., Tan, R. T.: Heavy rain image restoration: Integrating physics model and conditional adversarial learning. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 1633–1642 (2019)



## Extracción automática de frases clave para idioma inglés utilizando patrones léxicos

Yanet Hernández-Casimiro<sup>1,2</sup>, Yulia Ledeneva<sup>1</sup>,  
René-Arnulfo García-Hernández<sup>1</sup>, José-Luis Tapia-Fabela<sup>1</sup>

<sup>1</sup> Universidad Autónoma del Estado de México,  
Instituto Literario,  
México

<sup>2</sup> Cátedras COMECyT,  
México

{yhernandezcs, joseluis.fabela}@gmail.com  
yledeneva@yahoo.com, renearnulfo@hotmail.com

**Resumen.** En este artículo se presenta un método para la tarea de extracción automática de frases clave utilizando secuencias frecuentes maximales y patrones léxicos para el idioma inglés. El conjunto de datos utilizado para la evaluación del método es Inspect. Primeramente, se realiza el proceso de preparación de los textos para poder utilizarlos en el método de generación de patrones léxicos, ya que estos son obtenidos de los datos de entrenamiento y posteriormente convertidos en patrones de búsqueda, mediante el uso de expresiones regulares, permitiendo considerar secuencias de caracteres (signos de puntuación y *stopwords*) que determinen la forma en la que aparece una frase clave, sin depender de sus características sintácticas o semánticas y obtener un listado de posibles frases candidatas que puedan representar el contenido del documento. Además, para seleccionar las mejores, se consideraron únicamente aquellas que obtuvieron una ponderación alta. Se puede observar que los resultados por el método propuestos fueron satisfactorios, comparados con los métodos del estado del arte, obteniendo un rendimiento de 91.40% en precisión y 29.47% en recuerdo para el top 5, mientras que para el top 10 un 32.60% en precisión y 21.02% en recuerdo.

**Palabras clave:** Frases clave, patrones léxicos, secuencias frecuentes maximales.

### Automatic Keyphrases Extraction in English language Using Lexical Patterns

**Abstract.** In this paper we present a method for the automatic extraction of key phrases using maximum frequent sequences and lexical patterns for the English language. The data set used for the evaluation of the method is Inspect. Firstly, the process of preparing the texts is carried out in order to be able to use them in the method of generating lexical patterns, since these are obtained from the

training data and later converted into search patterns, through the use of regular expressions, allowing to consider sequences of characters (punctuation marks and stopwords) that determine the form in which a key phrase appears, without depending on its syntactic or semantic characteristics, and obtain a list of possible candidate phrases that can represent the content of the document. In addition, to select the best, only those that will obtain a high weight will be considered. The results for the proposed method were satisfactory, compared with the methods of the state of the art, obtaining a performance of 91.40% in precision and 29.47% in recall for the top 5, while for the top 10 a 32.60% in precision and 21.02% in recall.

**Keywords:** Key phrases, lexical patterns, maximal frequent sequences.

## 1. Introducción

En la actualidad, la generación de documentos digitales disponibles en la web ha crecido de forma exponencial debido al uso de la tecnología y digitalización, lo que ha provocado una necesidad en la búsqueda y gestión de información relevante para proporcionar un resumen del contenido principal de un documento.

Una manera de poder resolver este problema es realizar la asignación de frases, palabras o términos clave a una colección de documentos o un solo documento, que proporcionen una forma de identificar y caracterizar su contenido de manera concisa, para ayudar al lector a decidir si la información que encuentra es la adecuada para él [1, 2, 3, 4].

Una frase clave es un conjunto representativo de palabras, frases o términos que pueden resumir el contenido de un documento e identificar sus tópicos principales. Además, de acuerdo con la posición y tipo de texto en el que se encuentren, cumplen con un determinado objetivo; cuando son colocadas en la primera página de un artículo, estas apoyan en la generación de un breve resumen al lector y son asignadas por su autor, que van desde cinco hasta quince frases; si estas son encontradas dentro del índice de una revista, se puede considerar como un tema de indexación, para encontrar de forma rápida un artículo relevante.

Sin embargo, cuando se encuentran dentro de un motor de búsqueda, el objetivo es realizar “búsquedas” de documentos de forma precisas que coincidan con la frase.

Estos tres objetivos mencionados, cumplen con una característica en común, considerar una lista de frases que capturen los tópicos principales en los documentos [5, 6, 7, 8, 9, 10].

El proceso de asignación de frases clave es una tarea considerada como complicada, ya que es un proceso manual, el cual implica demasiados esfuerzos en tiempo, costo y recursos humanos; siendo esta última, llevada a cabo por un experto de determinado dominio, que implementa tesauros de acuerdo con la temática del documento, para poder normalizar las palabras que se asignan a cada texto. Por lo que algunas de las palabras asignadas al documento no se pueden encontrar en él y esto provoca confusiones o malas interpretaciones por parte del lector, de ahí la importancia de contar con una forma resumida que represente el contenido principal de un documento y un proceso automatizado para generarlas [11, 12].

Esta actividad es llamada Extracción Automática de Frases Clave o en inglés *Automatic Keyphrase Extraction (AKE)*, en la cual se generan frases a partir del texto de un documento fuente, que facilita la búsqueda e indexación de un gran volumen de colecciones de textos en formato digital cuyas mejoras en esta tarea podrían conducir a aumentar el rendimiento en algunas otras de la rama del Procesamiento de Lenguaje Natural (PLN) como: Extracción de información [13], clasificación [14], agrupación [15], traducción automática [16] y algunas otras tareas de procesamiento de información [2, 10, 17, 18, 19, 20].

Debido a su importancia y apoyo en tareas de PLN, se han propuesto diferentes enfoques para la AKE que van desde métodos supervisado y no supervisado; donde los primeros consideran la tarea como un problema de clasificación, mientras que los segundos usan otro tipo de técnicas como el agrupamiento [21, 22, 23].

En este artículo, se propone un sistema no supervisado para la extracción automática de frases clave, que comprende dos actividades principales; la primera es el proceso de descubrimiento de patrones léxicos, mediante Secuencias Frecuentes Maximales (SFM) y la segunda se encarga de la selección de frases clave, a partir de la implementación de una serie de pesos, derivados de la evaluación de cada patrón léxico. Los resultados de este artículo fueron comparados con los obtenidos por los métodos del estado del arte.

Las contribuciones de este artículo son la: propuesta de pesos para la selección de las mejores frases clave que representen el contenido principal de un documento. Así como la implementación del método en un conjunto de datos en idioma inglés.

El resto del artículo está organizado de la siguiente manera: sección 2, se realiza un análisis de los trabajos relacionados; sección 3, se da una descripción del método propuesto; sección 4, se presentan los resultados experimentales y, por último, en la sección 5, mostramos las conclusiones y trabajos futuros.

## **2. Trabajos relacionados**

La tarea de AKE se clasifica en dos enfoques principales: supervisado y no supervisado. En los cuales se implementan diferentes subprocesos y modelos.

Un proceso general para poder llevar a cabo esta tarea está constituido por 4 pasos principales: (1) pre-procesamiento, (2) selección de candidatas, (3) ponderación y (4) construcción de un listado de frases clave final [24, 25, 26]. A continuación, se abordan trabajos desde estos dos enfoques.

Dentro de los enfoques no supervisado se puede considerar el uso de la frecuencia del término [27, 28]; frases que coincidan con etiquetado POS [1] implementando medidas de centralidad.

También existen otros métodos como los basados en patrones léxicos [25, 29] en el cual se realiza una etapa de pre-procesamiento de los textos, para previamente obtener un conjunto de patrones léxicos que pueden identificar las frases clave de un documento, mediante la asignación de pesos que determinen la importancia de una palabra, generando un listado de palabras asignadas a un documento.

donde se demuestra que el uso de patrones léxicos no dependen de la característica de frecuencia para determinar que una frase es importante a diferencia de algunos otros

**Tabla 1.** Patrones léxicos de acuerdo con su longitud.

Longitud $l$	Frase clave	Patrón léxico
$l^1$	OF <u>RELATIONSHIP</u> .	OF<KP>.
$l^2$	FOR <u>WEBSIT DEVELOP</u> ,	FOR<KP><KP>.
$l^3$	THE <u>AVAIL CHEMIC DIRECTORI</u> ,	THE<KP><KP><KP>.
$l^4$	OF <u>MICROWEL POLYMERAS CHAIN REACTION</u> (	OF<KP><KP><KP><KP> (

trabajos, donde ésta es una de las características básicas para considerar si una palabra es clave [10, 15, 31, 30].

También existen trabajos basados en grafos [18, 21, 13, 32, 33] donde su principal característica es la construcción de grafos donde cada uno de los nodos (vértices) es considerado como una palabra o frase candidatas, mientras que los enlaces o aristas determina la relación que existen entre un nodo y otro.

Estas relaciones se basan en la ocurrencia o relación semántica. Además de los métodos anteriores, existen aquellos que pueden combinar técnicas como la implementación de patrones léxico-sintácticos con un modelo basado en grafos, unida a la perspectiva de lógica difusa [34], donde se requiere de medidas semánticas y sintácticas, además del operador OWA (*Ordered Weighted Averaging*).

Para trabajos con enfoques supervisados, la selección de frases clave se puede considerar como un problema de clasificación. Turney [5] implementa un sistema de aprendizaje automático llamado Genex que consta de dos secciones; un algoritmo genético llamado Genitor y el Extractor, constituido por doce parámetros que son ajustados por el algoritmo genético, para luego generar una lista de frases clave.

Otro de los trabajos reconocidos es KEA (*Keyphrase Extraction Algorithm*) [35], un algoritmo simple y eficaz que se basa en un aprendizaje automático implementando Naïve Bayes, para el entrenamiento y la extracción de frases clave que consta de dos fases; (1) Entrenamiento: se crea un modelo para identificar frases clave, utilizando documentos de entrenamiento donde se conocen las frases clave del autor, (2) Extracción: elije frases clave de un documento de prueba utilizando el modelo anterior.

### 3. Método propuesto

El método descrito en este artículo está constituido por cuatro fases importantes para la extracción automática de frases clave, donde se utilizan las SFM para el proceso de descubrimiento de patrones léxicos, que ayudarán en la generación de una lista de frases clave candidatas.

### 3.1. Pre-procesamiento

Esta fase es importante para la preparación de los datos, ya que el método propuesto se basa en el uso del algoritmo de SFM [36] que solo acepta, letras y el símbolo “@”, para ello es necesaria la eliminación de los caracteres especiales, puesto que no proporcionan información relevante en el texto, por ejemplo:  $\sqrt{\quad}$  (*raíz cuadrada*),  $\beta$  (*beta*),  $\Theta$  (*theta*),  $\Sigma$  (*sigma*), etc.; así como la limpieza, reestructuración y codificación de signos de puntuación, *stop words* y números, donde éstos últimos son transformados en una etiqueta, por ejemplo: “(”  $\rightarrow$  @PaA (paréntesis que abre); “:”  $\rightarrow$  @DP (dos puntos); “?”  $\rightarrow$  @PreC (signo de interrogación que cierra); etc., que ayudan en el proceso de la obtención de patrones léxicos. También se aplicó un proceso de lematización utilizando el algoritmo de Porter [37].

### 3.2. Descubrimiento de patrones léxicos

En esta fase, se crean tres conjuntos, partiendo del entrenamiento (*train*)  $D_{train} = \{d_1, d_2, d_3 \dots d_j\}$  y prueba (*test*)  $D_{test} = \{d_1, d_2, d_3 \dots d_j\}$ : El conjunto  $D_{formato}^{test|train}$  se obtiene de haber aplicado la fase de pre-procesamiento en  $D_{test|train}$ . Y los conjuntos  $D_{búsqueda}^{test}$  y  $D_{contexto}^{train}$ , parten de haber identificado un conjunto de frases clave generadas por un experto de acuerdo con cada documento  $d_j$  y longitud  $l^n$  de la frase,  $KP_{experto}^{l, test|train}_{d_j}$  en cada uno de los conjuntos  $D_{formato}^{test|train}$ . Donde  $KP_{experto}^{l, test|train}_{d_j}$  es transformado en frases clave de búsqueda  $KP_{búsqueda}^{l, test|train}_{d_j}$  mediante expresiones regulares.

Una vez creados estos conjuntos, se aplica el algoritmo de SFM en  $D_{contexto}^{train}$ . Para la creación de  $D_{contexto}^{train}$  se identificaron las  $KP_{búsqueda}^{l, train}_{d_j}$  consideradas como semillas  $KP_{semilla}^{l, train}_k$  y representadas con la etiqueta “<KP>” en el texto de cada documento de  $D_{formato}^{train}$ , de tal manera que  $D_{contexto}^{train}$  está constituido por un listado de las  $KP_{semilla}^{l, train}_k$  encontradas.

Para implementar el algoritmo de SFM, es necesario considerar un umbral  $\beta$  de aparición, en el que se toma un porcentaje aleatoria del 0 al 1 para asignar los valores de  $\beta$ , de acuerdo con la cantidad de  $KP_{semilla}^{l, train}_k$  identificadas en  $D_{contexto}^{train}$  y longitud de la frase.

Si se considera que la cantidad de semillas encontradas por  $l^n$  de frase en el conjunto es de  $l^{1train} = 28,350$ ;  $l^{2train} = 12,320$  y  $l^{3train} = 456$ ; tomando un umbral  $\beta$  al 0.250%, los valores serían:  $l^{1train} = 70.89 \approx 71$ ;  $l^{2train} = 30.8 \approx 31$  y  $l^{3train} = 1.14 \approx 3$ , este  $\beta$  representa el número de apariciones que debe presentar una  $kp_{semilla}^{l, train}_k$  para poder considerarla como un patrón léxico de acuerdo a la cantidad de semillas por  $l^n$  en el texto, es por ello que en  $l^{3train}$  no se puede considerar

---

**Algoritmo 1. EVALUACIÓN DE PATRONES LÉXICOS**

---

1: **procedimiento** EVALUACIÓN DE PATRONES LÉXICOS

---

2: **para cada**  $l^n \in D_{train}$  **hacer**

---

3:  $P_{búsqueda} \leftarrow$  ER (Expresiones Regulares) **aplicar**  $P_{léxicos}$

---

4:  $KP_{ident} \leftarrow P_{búsqueda}$  **aplicar**  $D_{format}$

---

5: **evaluar** resultado  $\leftarrow$  *gold estándar* **comparar**  $KP_{ident}$

---



---

**Algoritmo 2 SELECCIÓN Y EVALUACIÓN DE FRASES CLAVE**

---

1: **procedimiento** SELECCIÓN Y EVALUACIÓN DE FRASES CLAVE

---

2: **para cada**  $d \in D_{test}$  **hacer**

---

3: **para cada**  $p_{búsqueda} \in P_{búsqueda}$  **hacer**

---

4: **para cada**  $kp_{ident} \in KP_{ident}$  **hacer**

---

5: Precisión (P)  $kp_{ident} \leftarrow P_{búsqueda}$  Precisión

---

6: Recuerdo (R)  $kp_{ident} \leftarrow P_{búsqueda}$  Recuerdo

---

7: F-Measure (F-M)  $kp_{ident} \leftarrow P_{búsqueda}$  F-Measure

---

8: Booleano (B)  $kp_{ident} \leftarrow$  Booleano **donde** 1 = aparece o 0 = no aparece

---

9: **para cada**  $kp_{ident} \in KP_{ident}$  **hacer**

---

10: **sumar** peso  $\leftarrow kp_{ident}$

---

el valor de uno, porque encontraría todas las semillas del conjunto, por lo que en este trabajo el  $\beta$  mínimo de frecuencia es considerado como tres.

Para obtener  $P_{léxicos}^{l^{train}}$ , únicamente son elegidas las SFM que estén constituidas por al menos una  $kp_{semilla_k d_j}^{l^{train}}$  y que cumplan con la siguiente estructura:  $contexto_{izq} / < KP > / contexto_{der}$ .

El contexto es considerado como la secuencia de por lo menos 1 a 20 palabras a la derecha o izquierda de una  $kp_{semilla_k d_j}^{l^{train}}$  identificada, que posteriormente es considerada como un patrón léxico. Estas palabras pueden estar constituidas por signos de puntuación, números, *stop words* o palabras del texto. Las semillas serán colocadas en un listado de acuerdo con la longitud de la  $kp_{semilla_k d_j}^{l^{train}}$ , cabe señalar que la cantidad de etiquetas “<KP>” determina la longitud de la frase. En la tabla 1 se muestran algunos ejemplos de los patrones léxicos encontrados de acuerdo con su longitud.

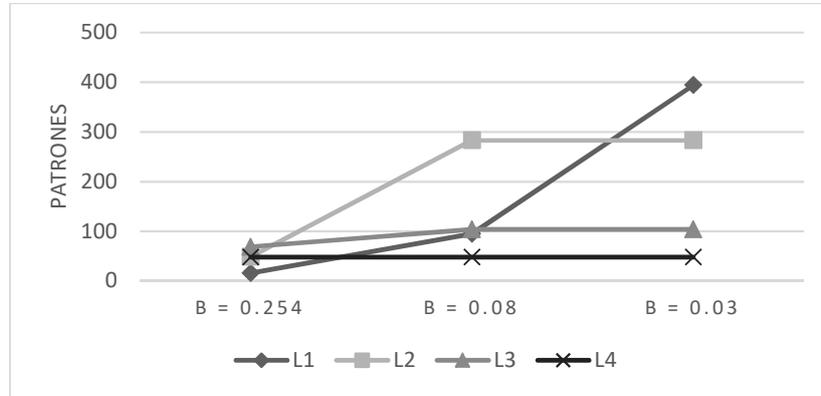


Fig. 1. Cantidad de patrones léxicos extraídos por cada  $\beta$ , de acuerdo con la longitud de la frase.

### 3.3. Evaluación de patrones léxicos

El algoritmo 1, representa el proceso realizado para poder evaluar cada uno de los patrones léxicos. El conjunto de  $P_{\text{léxicos}}^{l \text{ train}}$  es obtenido del conjunto  $D_{\text{train}} = \{d_1, d_2, d_3 \dots d_j\}$  de acuerdo con  $l^n$ . Un aspecto importante de considerar es que uno o más  $p_{\text{busqueda}_w}^{l \text{ train}}$ , pueden identificar la misma  $kp_{\text{ident}_w d_j}^{l \text{ test}}$  más de una vez. La evaluación se realiza para las  $kp_{\text{ident}_w d_j}^{l \text{ test}}$  de cada  $p_{\text{busqueda}_w}^{l \text{ train}}$  mediante las medidas de precisión (P), recuerdo (R) y F-Measure (F-M).

### 3.4 Selección y evaluación de frases clave

En el algoritmo 2, se muestra el proceso de selección y evaluación de frases clave. La identificación de las frases clave está considerada de tal manera que no dependa de la frecuencia que  $kp_{\text{ident}_w d_j}^{l \text{ test}}$  tiene en el documento del conjunto  $D_{\text{test}}$ , sino por el número de patrones por la que es extraída de acuerdo con el rendimiento del patrón, es por ello que se toman como pesos sus respectivos valores de acuerdo a su evaluación con P, R y F-M. Además de estos valores, se consideró agregar otro pesado, que en este trabajo es nombrado Booleano (B), donde el valor de 1 se asigna siempre que  $kp_{\text{ident}_w d_j}^{l \text{ test}}$  sea extraída por  $p_{\text{busqueda}_m}^{l \text{ train}}$ , en caso contrario el valor será  $\phi$ .

De esta fase, se obtiene  $KP_{\text{candidatas}_w d_j}^{l \text{ test}}$ , donde se considera el top 5 y 10 para cada documento. Por último, se evalúa  $KP_{\text{candidatas}_w d_j}^{l \text{ test}}$  mediante las medidas de Precisión (P) y Recuerdo (R).

**Tabla 2.** Rendimiento del método propuesto de los mejores  $\beta$  para el Top 5.

Top 5				
Umbral	NO CONTROLADAS		COMBINADAS	
	P	R	P	R
$\beta = 0.254$ -P	<u>86.20%</u>	<u>38.45%</u>	<u>91.40%</u>	<u>29.47%</u>
$\beta = 0.08$ -P	78.60%	35.06%	85.80%	27.66%
$\beta = 0.08$ -FM	71.40%	31.85%	77.60%	25.02%
$\beta = 0.254$ -FM	70.60%	31.49%	76.00%	24.50%
$\beta = 0.08$ -R	69.20%	30.87%	75.20%	24.25%
$\beta = 0.254$ -R	69.60%	31.04%	74.80%	24.12%
$\beta = 0.254$ -B	68.80%	30.69%	74.00%	23.86%
$\beta = 0.03$ -P	61.20%	27.30%	68.80%	22.18%
$\beta = 0.08$ -B	58.40%	26.05%	64.20%	20.70%
$\beta = 0.03$ -B	56.60%	25.25%	62.60%	20.18%

**Tabla 3.** Comparación de resultados del método propuesto con el estado del arte para el Top 5.

Top 5				
Umbral	NO CONTROLADAS		COMBINADAS	
	P	R	P	R
$\beta = 0.254$ -P	<u>86.20%</u>	<u>38.45%</u>	<u>91.40%</u>	<u>29.47%</u>
$\beta = 0.08$ -P	78.60%	35.06%	85.80%	27.66%
$\beta = 0.08$ -FM	71.40%	31.85%	77.60%	25.02%
Gallegos [29]	17.48%	17.79%	19.44%	13.56%
KEA [35]	9.58%	8.50%	7.82%	5.46%

## 4. Experimentación

### 4.1. Datos

El conjunto utilizado para evaluar el método propuesto es Inspec [38], está constituido por 2,000 resúmenes en inglés extraídos de artículos científicos de revistas. Cada resumen tiene dos conjuntos de frases clave de oro (*gold*): un conjunto de términos controlados, los cuales son restringidos al tesoro de la base de datos Inspec; y un conjunto de términos no controlados que puede ser cualquiera de los términos adecuados, asignados por algún especialista en las disciplinas.

Tanto los términos controlados como no controlados pueden o no estar presentes en los resúmenes. El conjunto de resúmenes se dividió en tres subconjuntos: el de entrenamiento (*train*) que consta de 1 000 documentos; validación (*validation*) con 500 documentos, y un conjunto de prueba (*test*) con 500 documentos. Para este trabajo el

rendimiento se evalúa con las medidas de Precisión (P) y Recuerdo (R), además se hizo la unión de las frases clave *gold* controladas y no controladas para evaluar el rendimiento de su combinación (conjunto combinado).

## 4.2. Resultados

Para Inspect, se evaluaron solo los conjuntos *gold* de frases no controladas y combinadas, para poder comparar el método propuesto, con los resultados reportados en el trabajo de Gallegos [29] y el método KEA [35]. El método se probó con tres  $\beta$ . En la figura 1, se muestra la cantidad de patrones extraídos para cada  $\beta$  y cada longitud de frase.

En esta gráfica se puede visualizar que  $\beta$  determina la cantidad de patrones léxicos encontrados.  $\beta$  considera la frecuencia del patrón, de acuerdo con la longitud de la frase, entre más pequeño sea  $\beta$ , mayor será la cantidad de patrones léxicos encontrados. En la tabla 2, se muestran los mejores diez rendimientos obtenidos de los diferentes  $\beta$ , en cada conjunto de frases *gold* (no controlado y combinado) y cada pesado (P, R, F-M, B), para el top 5.

Los resultados muestran que el rendimiento del método propuesto en  $\beta = 0.254$  con pesado P, es el más alto comparado con los demás  $\beta$ . En el conjunto no controlado se tuvo un rendimiento de  $P = 86.20\%$  y  $R = 38.45\%$ , mientras para el conjunto combinado  $P = 91.40\%$  y  $R = 29.47\%$ . Es por ello, por lo que el conjunto de palabras combinadas se considera con mayor énfasis en este trabajo, siendo éste la combinación de palabras que existen y no, dentro del documento. En la tabla 3, se realiza una comparación de resultados obtenidos en este trabajo, con los reportados en el trabajo de Gallegos [29] y el método KEA [35].

Como se muestra en la tabla anterior, los resultados reportados en este trabajo para el top 5 son superiores a los reportados en [29] y [35]. Para el conjunto combinado (unión de los conjuntos controlado y no controlado), en  $\beta = 0.254$  se obtuvo un rendimiento de  $P = 91.40\%$  y  $R = 29.47\%$ , quedando por un rango alto a los presentados en el estado del arte. En la tabla 4, se muestra otra comparación de resultados obtenidos para el top 10.

Para este top, el rendimiento obtenido en  $\beta = 0.08$ , es de  $P = 32.60\%$  y  $R = 21.02\%$ , superando el método de [29] y [35]. Los patrones extraídos por  $\beta = 0.254$  y  $\beta = 0.08$  fueron de mejor calidad en comparación con otros umbrales generados por el método. En la tabla 5, se muestran algunos ejemplos de estos patrones extraídos por el método propuesto, así como las frases clave extraídas por patrón.ue la fase de pre-procesamiento es importante a la hora de preparar los datos, ya que, si existe una buena limpieza, el método propuesto en este trabajo puede dar mejores resultados, como los mostrados en este trabajo.

Uno de los retos por abordar para este método es la implementación de patrones extraídos por un conjunto de datos, aplicados a otro para poder demostrar la independencia del dominio, ya que los patrones extraídos son del mismo conjunto de datos del conjunto *train*.

**Tabla 4.** Rendimiento de extracción de frases clave de los mejores  $\beta$  para el top 10.

Umbral	Top 10			
	NO CONTROLADAS		COMBINADAS	
	P	R	P	R
$\beta = 0.08$ -P	30.70%	27.39%	32.60%	21.02%
$\beta = 0.08$ -FM	28.60%	25.51%	30.40%	19.60%
$\beta = 0.08$ -B	27.90%	24.89%	29.60%	19.09%
Gallegos [29]	17.47%	17.79%	19.44%	13.56%
KEA [35]	9.58%	8.50%	7.82%	5.46%

**Tabla 5.** Patrones extraídos por el método propuesto y frases clave extraídas por patrón.

Longitud	Patrones léxicos	Frases clave extraídas
$l^1$	OF<KP>,	OF <i>TRANSFORM</i> ,
$l^1$	OF<KP>AND	OF <i>SYSTEM</i> AND
$l^2$	FOR<KP><KP>,	FOR <i>SPECIF APPLIC</i> ,
$l^2$	, <KP><KP>.	, <i>THE INTERNET</i> .
$l^3$	(<KP><KP><KP>)	(COMPRESS <i>GRAPHIC IMAG</i> )
$l^3$	OF<KP><KP><KP> (	OF <i>BUILD MANAG SYSTEM</i> (
$l^4$	OF<KP><KP><KP><KP> (	OF <i>ALGORITHM IN COMPUT</i> <i>ELECTROMAGNET</i> (
$l^4$	FOR<KP><KP><KP><KP>OF	FOR <i>A SOCIAL IDENT PERSPECT OF</i>

**Agradecimientos.** Trabajo realizado con apoyo del Gobierno Mexicano (COMECYT). Los autores agradecen a la Universidad Autónoma Estado del México.

## Referencias

1. Boudin, F.: A comparison of centrality measures for graph-based keyphrase extraction. In: Proceedings of the Sixth International Joint Conference on Natural Language Processing, pp. 834–838 (2013)
2. Alrehamy, H., Walker, C.: Exploiting extensible background knowledge for clustering-based automatic keyphrase extraction. *Soft Computing*, vol. 22, pp. 7041–7057 (2018) doi: 10.1007/s00500-018-3414-4
3. Alami-Merrouni, Z., Frikh, B., Ouhbi, B. Automatic Keyphrase extraction: a survey and trends. *J Intell Inf Syst*, vol. 54, pp. 391–424 (2020) doi: 10.1007/s108 44-019-00558-9
4. Asl, J. R., Banda, J. M.: GLEAKE: Global and local embedding automatic keyphrase extraction. arXiv preprint arXiv:2005.09740 (2020) doi: 10.48550/ar Xiv.2005.09740
5. Turney, P. D.: Learning to extract keyphrases from text. ArXiv Preprint ArXiv, cs/0212013 (1999) doi: 0.48550/arXiv.cs/0212013

6. Nguyen, T. D., Kan, M. Y.: Keyphrase extraction in scientific publications. In: Goh, D. H. L., Cao, T. H., Sølvberg, I. T., Rasmussen, E. (eds.) Asian Digital Libraries. Looking Back 10 Years and Forging New Frontiers, ICADL 2007, Lecture Notes in Computer Science, Springer, Berlin, Heidelberg, vol. 4822, pp. 317–326 (2007) doi: 10.1007/978-3-540-77094-7\_41
7. Kim, S. N., Medelyan, O., Kan, M. Y., Baldwin, T.: Automatic keyphrase extraction from scientific articles. *Language Resources and Evaluation*, pp. 723–742 (2013) doi: 10.1007/s10579-012-9210-3
8. Zhang, Q., Wang, Y., Gong, Y., Huang, X.: Keyphrase extraction using deep recurrent neural networks on twitter. In: *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pp. 836–845 (2016)
9. Ying, Y., Qingping, T., Qinzhen, X., Ping, Z., Panpan, L.: A Graph-based approach of automatic keyphrase extraction. *Procedia Computer Science*, pp. 248–255 (2017)
10. Shapira, O., Pasunuru, R., Dagan, I., Amsterdamer, Y.: Multi-document keyphrase extraction: A literature review and the first dataset. arXiv preprint arXiv:2110.01073 (2021) doi: 10.48550/arXiv.2110.01073
11. Beliga, S., Martinčić-Ipšić, S.: Network-enabled keyword extraction for under-resourced languages. In: Cali, A., Gorgan, D., Ugarte, M. (eds.) *Semantic Keyword-Based Search on Structured Data Sources. IKC 2016, Lecture Notes in Computer Science*, Springer, Cham, vol. 10151, pp. 124–135 (2016) doi: 10.1007/978-3-319-53640-8\_11
12. Johnny, S., Jaya-Nirmala, S.: Key phrase extraction system for agricultural documents. In: Gani, A., Das, P., Kharb, L., Chahal, D. (eds) *Information, Communication and Computing Technology. ICICCT 2019, Communications in Computer and Information Science*, Springer, Singapore, vol. 1025, pp. 240–252 (2019) doi: 10.1007/978-981-15-1384-8\_20
13. Mihalcea, R., Tarau, P.: TextRank: Bringing order into texts. In: *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing*, pp. 404–411 (2004)
14. Zhang, Y., Liu, H., Wang, S., Fan, W., Xiao, C.: Automatic keyphrase extraction using word embeddings, *Soft Comput*, vol. 24, pp. 5593–5608 (2020) doi: 10.1007/s00500-019-03963-y
15. Lahiri, S., Mihalcea, R., Lai, P. H.: Keyword extraction from emails. *Natural Language Engineering*, vol. 23, no. 2, pp. 295–317 (2017) doi: 10.1017/S1351324916000231
16. Haque, R., Penkale, S. Way, A.: TermFinder: Log-likelihood comparison and phrase-based statistical machine translation models for bilingual terminology extraction. *Lang Resources & Evaluation*, vol. 52, pp. 365–400 (2018) doi: 10.1007/s10579-018-9412-4
17. Perez-Guadarrama, Y., Simón-Cuevas, A., Hojas-Mazo, W., Olivas, J. A., Romero, F. P.: A fuzzy approach to improve an unsupervised automatic Keyphrase extraction process. In: *Proceedings of IEEE International Conference on Fuzzy Systems*, pp. 1–6 (2018) doi: 10.1109/FUZZ-IEEE.2018.8491487
18. Mothe, J., Ramiandrisoa, F., Rasolomanana, M.: Automatic keyphrase extraction using graph-based methods. In: *Proceedings of the 33rd Annual ACM Symposium on Applied Computing*, pp. 728–730 (2018) doi: 10.1145/3167132.3167392
19. Saxena, A., Mangal, M., Jain, G.: KeyGames: A game theoretic approach to automatic keyphrase extraction. In: *Proceedings of the 28th International Conference on Computational Linguistics*, pp. 2037–2048 (2020) doi: 10.18653/v1/2020.coling-main.184
20. Mahata, D., Kuriakose, J., Shah, R., Zimmermann, R.: Key2vec: Automatic ranked keyphrase extraction from scientific articles using phrase embeddings. In: *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, vol. 2, pp. 634–639 (2018) doi: 10.18653/v1/N18-2100
21. Bouguin, A., Boudin, F., Daille, B.: TopicRank: Graph-based topic ranking for keyphrase extraction. In: *Proceedings of International Joint Conference on Natural Language Processing*, pp. 543–551 (2013)

22. Xie, F., Wu, X., Zhu, X.: Document-specific keyphrase extraction using sequential patterns with wildcards. In: Proceedings of IEEE International Conference on Data Mining, pp. 1055–1060 (2014) doi: 10.1109/ICDM.2014.105
23. Awan, M. N., Beg, M. O.: Top-rank: A topical position rank for extraction and classification of keyphrases in text. *Computer Speech & Language*, vol. 65, pp. 101116 (2021) doi: 10.1016/j.csl.2020.101116
24. Lim, V. M. H., Wong, S. F., Lim, T. M.: Automatic keyphrase extraction techniques: A review. In: Proceedings of IEEE Symposium on Computers & Informatics, pp. 196–200 (2013) doi: 10.1109/ISCI.2013.6612402
25. Hernández-Casimiro, Y., Ledeneva, Y., García-Hernández, R. A., Ramos-Corchado, M. A.: Lexical patterns based on maximal frequent sequences for automatic keyphrase extraction. *Computación y Sistemas*, vol. 25, no. 1, pp. 153–163 (2021) doi: 10.13053/cys-25-1-3868
26. Xie, F., Wu, X., Zhu, X.: Efficient sequential pattern mining with wildcards for keyphrase extraction. *Knowledge-Based Systems*, vol. 115, pp. 27–39 (2017) doi: 10.1016/j.knosys.2016.10.011
27. Salton, G., Buckley, C.: Term-weighting approaches in automatic text retrieval. *Information Processing & Management*, vol. 24, no. 5, pp. 513–523 (1988) doi: 10.1016/0306-4573(88)90021-0
28. Sidorov, G.: Construcción no lineal de n-gramas en la lingüística computacional. *Sociedad Mexicana de Inteligencia Artificial* (2013)
29. Gallegos-Camacho, E. M., Ledeneva, Y.: Extracción de frases clave utilizando patrones léxicos a partir de resúmenes de artículos científicos. Tesis, Universidad Autónoma de Estado de México, pp. 1–119 (2016)
30. Wang, Q., Sheng, V. S., Wu, X.: Keyphrase extraction with sequential pattern mining. In: Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence, pp. 5003–5004 (2017)
31. Marsi, E., Skidar, U., Marco, C., Barik, B., Saetre, R.: NTNU-1@ScienceIE at SemEval-2017 Task 10: Identifying and labelling keyphrases with conditional random fields. In: 11th International Workshop on Semantic Evaluations SemEval17 (2017)
32. Wan, X., Xiao, J.: CollabRank: Towards a collaborative approach to single-document keyphrase extraction. In: Proceedings of the 22nd International Conference on Computational Linguistics, pp. 969–976 (2008)
33. Sterckx, L., Demeester, T., Deleu, J., Develder, C.: Creation and evaluation of large keyphrase extraction collections with multiple opinions, *Lang Resources & Evaluation*, vol. 52, pp. 503–532 (2018) doi: 10.1007/s10579-017-9395-6
34. Barreiro-Guerrero, M., Simón-Cuevas, A., Pérez-Guadarrama, Y., Romero, F. P., Olivas, J. A.: Applying OWA operator in the semantic processing for automatic keyphrase extraction. In: Nyström, I., Hernández Heredia, Y., Milián Núñez, V. (eds) Progress in pattern recognition, image analysis, computer vision, and applications. In: Proceedings of CIARP 2019, Lecture Notes in Computer Science, Springer, Cham vol. 11896, pp. 62–71 (2019) doi: 10.1007/978-3-030-33904-3\_6
35. Witten, I. H., Paynter, G. W., Frank, E., Gutwin, C., Nevill-Manning, C. G.: Kea: Practical automated keyphrase extraction. *Design and Usability of Digital Libraries: Case Studies in the Asia Pacific*, pp. 129–152 (2005)
36. García-Hernández, R. A., Martínez-Trinidad, J. F., Carrasco-Ochoa, J. A.: A new algorithm for fast discovery of maximal sequential patterns in a document collection. *Computational Linguistics and Intelligent Text Processing*, vol. 3878, pp. 514–523 (2006) doi: 10.1007/11671299\_53
37. Porter, M. F.: An algorithm for suffix stripping. *Program: electronic library and information systems*, vol. 14, no. 3, pp. 130–137 (1980) doi: 10.1108/eb046814

38. Hulth, A.: Enhancing linguistically oriented automatic keyword extraction. In: Proceedings of HLT-NAACL 2004, pp. 17–20 (2004)



## Una revisión de chatbots en la salud

Beatriz A. González-Beltrán<sup>1</sup>, Miguel A. Vázquez-García<sup>1</sup>,  
José A. Reyes-Ortiz<sup>1</sup>, Raúl García-Ruiz<sup>2</sup>

<sup>1</sup> Universidad Autónoma Metropolitana,  
Departamento de Sistemas,  
México

<sup>2</sup> Instituto Politécnico Nacional,  
Centro de Investigación y de Estudios Avanzados del IPN,  
México

{bgonzalez, al2201801758, jaro}@azc.uam.mx,  
rgarcia@cinvestav.mx

**Resumen.** Los chatbots actualmente se encuentran inmersos en la vida diaria de la personas, ya sea como sistemas para realizar pedidos o compras, resolver dudas de usuarios con respecto a un producto, apoyar a la educación, para información sobre un proceso administrativo y para dar asistencia en el área de la salud. Por otro lado, con el crecimiento de la población y el aumento en la esperanza de vida del ser humano, se estima a nivel mundial que los recursos médicos pueden resultar insuficientes para atender las necesidades de atención de la población. Este artículo presenta una revisión de los chatbots en el ámbito de la salud con el objetivo de realizar un estado del arte en este dominio e identificar áreas de oportunidad. El análisis de los trabajos se clasifica de acuerdo a las variables dominio del conocimiento, objetivo, servicio provisto, nivel de automatización, forma de comunicación, plataforma de desarrollo, procesamiento de entrada / método de generación de respuesta y el idioma. Del análisis realizado se encontró que la mayoría de los chatbots están basados en la conversación, son intrapersonales, la comunicación es por texto y están basados en reglas. Además, faltan chatbots en idiomas adicionales al inglés, falta desarrollar más chatbots para enfermedades crónicas como el cáncer y que estos utilicen métodos de aprendizaje automático o de aprendizaje profundo.

**Palabras clave:** Chatbots, chatbots en la salud, estado del arte de chatbots en la salud.

### A Review of Chatbots in Health

**Abstract.** Chatbots are currently immersed in people's daily lives, whether as systems to place orders or purchases, answer questions from users regarding a product, support education, for information on an administrative process and to give assistance in the area of health. Moreover, with the growth of the population and the increase in life expectancy, it is estimated that medical resources would be insufficient to meet the health care needs of the population. This article presents

a review of chatbots in the field of health with the aim of making a state of the art in this domain and identifying areas of opportunity. The analysis of the works is classified according to the variables domain of knowledge, goals, service provided, level of automation, form of communication, development platform, input processing / response generation method, and language. From the analysis carried out, it was found that most chatbots are based on conversation, they are intrapersonal, the communication is by text and they are based on rules. In addition, there is a lack of chatbots in languages other than English, and chatbots for chronic diseases such as cancer and that they use machine learning or deep learning methods.

**Keywords:** Chatbots, chatbots in health, state of the art of chatbots in health.

## 1. Introducción

Los chatbots son programas computacionales, también conocidos como asistentes inteligentes, que se encargan de entablar conversaciones de manera natural como lo haría una persona. Para lograr esta conversación se utilizan herramientas de procesamiento de lenguaje natural [2].

Los chatbots actualmente se encuentran inmersos en la vida diaria de la personas, ya sea como sistemas para entretener, realizar pedidos o compras, resolver dudas de usuarios con respecto a un producto, apoyar a la educación, para información sobre un proceso administrativo y para dar asistencia en el área de la salud.

Los chatbots se utilizan cada vez con más frecuencia dado que ofrecen ventajas para sus usuarios (e.g. servicio 24/7, facilidad de uso comparado con una aplicación, etc.). Los chatbots pueden ser integrados en redes sociales, aplicaciones web y en aplicaciones de mensajería móvil, permitiendo a sus usuarios tener un amplio abanico de posibilidades de usarlos, en cualquier momento y desde cualquier lugar.

Por otro lado, con el crecimiento de la población y el aumento en la esperanza de vida, se estima a nivel mundial que los recursos médicos serán insuficientes para atender las necesidades de atención de la población.

Aunado a esto, el difícil acceso a algunas zonas rurales y la existencia de estructuras deficientes dificultan el ofrecimiento de servicios de salud. Por esta razón, la existencia de chatbots en el área de la salud podrían funcionar como un primer filtro para satisfacer las necesidades básicas de atención médica.

Esta investigación busca oportunidades de mejora a los chatbots en el área de la salud, particularmente para atender a las personas con cáncer. En México, el cáncer es la tercera causa de muerte [10] y la segunda en Estados Unidos [22].

El resto del artículo está organizado como sigue: en la sección 2 se explica el método de selección de los trabajos a revisar.

En la sección 3 se muestran trabajos relacionados con revisiones de chatbots. En la sección 4 se describe la clasificación de los tipos de chatbots, misma que se utiliza para caracterizar a cada uno de los trabajos analizados.

En la sección 5 se describe una tabla comparativa de los trabajos incluidos en la revisión y en la sección 6 se presenta un resumen de las aportaciones del artículo así como el trabajo futuro.

## **2. Método**

La revisión se enfoca en artículos y memorias de congreso, revisados por pares. La búsqueda se realizó de marzo a junio de 2021 de artículos publicados en el período 2015-2021. Las bases de datos consultadas fueron: IEEE Xplore, Springer y ACM Digital Library. Se incluye un artículo disponible en Google Scholar, en idioma español. Cabe señalar que solo se consideraron artículos de acceso abierto o que estuvieran disponibles a través de la plataforma *bidi.uam.mx*.

Los términos de búsqueda fueron «chatbot» y «conversational agent». Cada uno de estos términos se combinó con los siguientes: «oncology», «cancer», «symptom» y «health».

Dado que Springer publica en disciplinas diversas y en diferentes idiomas, se seleccionaron los trabajos escritos en idioma inglés y de Ciencias de la Computación. Además, en IEEE Xplore se filtró el tópico de publicación con el término «chatbots». La selección final se realizó de manera manual, utilizando la lectura del resumen del artículo. Durante dicha lectura se seleccionaron solo los artículos que se enfocaban en el desarrollo de chatbots.

Se excluyeron los trabajos de chatbots con cuerpo y se analizaron solo chatbots sin cuerpo. También se excluyeron trabajos cuyo propósito era evaluar la eficiencia del chatbot.

De manera particular, el objetivo de esta revisión es identificar si existen chatbots en la salud que respondan al tema del cáncer y analizar sus aspectos tecnológicos. Se consideraron las siguientes variables de comparación: dominio del conocimiento, objetivo, servicio provisto, nivel de automatización, forma de comunicación, plataforma de desarrollo (i.e., permisos), procesamiento de entrada / método de generación de respuesta, e idioma.

Al inicio se contempló incluir las variables usuario y patología a atender. Sin embargo, el objetivo del chatbot descrito en la sección 4.2 incluye la patología y el usuario. Es conveniente hacer notar que al realizar un filtrado manual utilizando la lectura del resumen del artículo y considerando los elementos de inclusión y de exclusión anteriormente mencionados, los resultados de estudios equivalentes pueden ser diferentes al momento de repetirse.

## **3. Trabajo relacionado**

En [17], los autores desarrollaron una clasificación de los tipos de chatbots, de las diferentes herramientas y algoritmos que pueden ser utilizadas para el desarrollo de chatbots, proporcionan una arquitectura general para su construcción e identifican las áreas de investigación donde se necesita más atención. En este artículo se toma como marco de comparación la clasificación propuesta por estos autores.

Sin embargo, dado que la revisión se enfoca en la tecnología, no se enfoca en un dominio de aplicación específico.

[2] presenta una revisión histórica de los chatbots, las motivaciones de uso de los chatbots, identifica los conceptos relacionados con la tecnología de los chatbots y retoma la clasificación de [17] para explicar las diferentes técnicas. Sin embargo, este trabajo no presenta una tabla comparativa de los diferentes trabajos ni se enfoca en un dominio de aplicación en particular.

Por otra parte, en [14] se presenta una revisión de los agentes conversacionales con cuerpo. La similitud con esta investigación es que también se concentra en el área de la salud pero su estudio se concentró en las características de diseño enfocadas en el lenguaje, comportamiento, expresión emocional, cuerpo y ambiente. Sin embargo, el interés de este artículo es en los chatbots sin cuerpo.

En [8] se presenta una revisión de chatbots basados en texto y presentan una arquitectura en tres capas: 1) el procesamiento de lenguaje natural (por sus siglas en inglés, NLP); 2) el entendimiento del lenguaje natural (por sus siglas en inglés, NLU) y 3) la máquina de decisión. Este trabajo identifica los retos en el desarrollo de chatbots. Sin embargo, dado que la revisión se enfoca en la tecnología, no se enfoca en un dominio de aplicación específico.

Una revisión de los aspectos técnicos al desarrollar chatbot en la salud se presenta en [21]. Los módulos que componen una arquitectura general de un chatbot fueron las variables a comparar: entendimiento del texto, procesamiento de entrada / método de generación de respuesta, manejo del diálogo y repositorio de datos. Esta revisión se enfoca en chatbots en la salud y también analiza los aspectos técnicos. Sin embargo, se concentra en chatbot basados en texto y falta considerar una categoría más amplia que son los chatbots sin cuerpo, mismos que pueden incluir procesamiento de voz y de imágenes.

[26] presenta una revisión de aplicaciones oncológicas y chatbots para la salud. El trabajo se enfoca en la terapia de cáncer e incluye las siguientes variables: promoción de la salud, diagnóstico, tratamiento, monitoreo, soporte del paciente y eficiencia en el flujo de trabajo. Este trabajo se concentra en el objetivo (i.e., función) del chatbot pero no analiza los aspectos tecnológicos.

#### **4. Marco de comparación**

Esta investigación presenta un análisis del estado del arte de los chatbots en el área de la salud y está diseñada en base a la clasificación proporcionada por [2] quienes extendieron la clasificación de [17]. En [2] los tipos de chatbots se clasifican de acuerdo con las siguientes variables: dominio del conocimiento, objetivo, servicio provisto, nivel de automatización (i.e., ayuda humana), forma de comunicación (i.e., canal de comunicación), plataforma de desarrollo (i.e., permisos) y procesamiento de entrada / método de generación de respuesta. [17] no considera el nivel de automatización, ni la forma de comunicación, ni la plataforma de desarrollo. En este artículo se propone extender esta clasificación para tomar en cuenta el idioma.

#### 4.1. Dominio del conocimiento

La clasificación de los chatbots por dominio de conocimiento toma en cuenta el conocimiento al que puede tener acceso un chatbot. Los chatbots pueden ser de dominio abierto que consideran tópicos de carácter general o de dominio cerrado que se enfocan en un dominio de conocimiento. Alexa es un chatbot de dominio abierto, mientras que un chatbot de una aerolínea que atiende las reservaciones es de dominio cerrado.

#### 4.2. Objetivo del chatbot

Los chatbots se puede clasificar según su objetivo principal y pueden ser informativos, basados en la conversación y basados en la tarea. Los de tipo informativo tienen como objetivo proporcionar información para el usuario, tal como los chatbots de preguntas frecuentes. Por su parte, los chatbots basados en la conversación tienen como objetivo comunicarse con el usuario de manera natural y responder correctamente a las entradas del usuario. Por último, los chatbots basados en la tarea realizan una tarea en específico, tal como reservar algún servicio.

Medibot [24] tiene como objetivo diagnosticar a un paciente. Para ello utiliza los datos de la conversación para dar un diagnóstico personalizado en base al perfil y síntomas del paciente. iHelpr [9] está enfocado en los pacientes con problemas de salud mental y otorga ayuda en los temas de estrés, ansiedad, depresión, sueño y autoestima.

HealFavor [13] es un chatbot prototipo que identifica la condición de salud del paciente y proporciona asistencia en base a la conversación. El chatbot en [20] se enfoca en los mensajes provistos por el paciente, con el fin de detectar síntomas. HealthBuddy [12] tiene como objetivo atender problemas de salud mental de los jóvenes. Se enfoca en mantener un diálogo con el paciente para llevar el flujo de la conversación. C Bot [6] tiene como objetivo la detección de cáncer y selecciona el tipo de cáncer de los datos provistos durante la conversación con el paciente.

TELEBOT [25] es un chatbot informativo sobre el COVID-19 y si durante la conversación el chatbot determina que el paciente está enfermo de COVID, entonces le solicita que registre el caso y le proporciona información para su tratamiento. Dr.TARS [15] ofrece asistencia médica, proporcionando al paciente la descripción de los medicamentos y la dosis en base a su edad.

Este sistema también analiza los síntomas del paciente para dar una predicción de la enfermedad. MedChat [16] detecta los síntomas del paciente en base a la conversación con el usuario, quien previamente ha sido registrado. Disha [19] permite diagnosticar enfermedades potenciales en base los síntomas del paciente y mediante el registro de su estado de salud.

En [1] se propone un chatbot para atender consultas médicas como parte de la medicina ambulatoria a través de la evaluación de la salud del paciente, notificaciones de los procedimientos y resultados, de las citas, la toma de medicamentos o la necesidad de consultar a un médico. El chatbot puede transmitir los resultados de los pronósticos a solicitud del paciente, tomando en cuenta la actividad física o el esquema nutricional del paciente. Quoro [11] detecta una posible enfermedad, a partir de los síntomas que el paciente proporciona.

COOPERA [27] ayuda a prevenir la dispersión de COVID-19 a través de la recuperación de información del historial del paciente y de su estado de salud. En [3] proponen un chatbot que puede diagnosticar una enfermedad y dar información de dicha enfermedad antes de que el paciente consulte al médico. Medbot [7] está diseñado para dar educación de salud básica, medidas preventivas, remedios caseros para enfermedades comunes y recomendaciones de salud a pacientes crónicos. En [23] tiene como objetivo realizar un monitoreo de la salud del paciente basado en los síntomas del paciente.

El sistema puede clasificar e identificar el tratamiento y el médico apropiado para consultar los problemas de salud. Babylon [5] tiene un chatbot para detectar los síntomas y clasificar a los pacientes según un sistema de prioridades y determinar los que requieren atención médica inmediata. Dejal@bot [4] tiene como objetivo que la población adulta deje de fumar y proporciona información de los mitos asociados al tabaco, la prevención de recaídas, ejercicios de respiración o manejo de ansiedad.

Medibot y Fabi [18] son chatbots dirigidos a los profesionales de la salud. Medibot responde preguntas relacionadas con la estabilidad de los medicamentos y vacunas que son sensibles a la temperatura. Fabi incluye todo el portafolio de los productos de Pfizer y permite establecer una conversación con el usuario.

### **4.3. Servicio provisto**

El servicio provisto se refiere la cercanía que tiene el chatbot con el usuario desde el punto de vista de los sentimientos y depende de la tarea del chatbot. Los chatbots pueden ser interpersonales, intrapersonales e inter-agentes. Los chatbots interpersonales están enfocados en obtener información y otorgarla al usuario y no se espera que guarden información del usuario.

De manera general, los chatbots de preguntas frecuentes son interpersonales. En lo que concierne a los chatbots intrapersonales, pueden convertirse en compañero del usuario, conocer sus gustos y normalmente se encuentran dentro de aplicaciones de mensajería. Los chatbots inter-agentes se encuentran presentes en todo momento y requieren la comunicación entre asistentes inteligentes, tal como la integración de Alexa y Cortana [17].

Medibot [24] es una aplicación que intenta comunicarse con el usuario, mediante preguntas sobre el estado del usuario. iHelpr [9] está contenido en un sitio en línea y trata de entender los sentimientos del usuario. HealFavor [13] busca entender al usuario basándose en la respuesta actual y las anteriores para calcular la intención del usuario. El chatbot de [20] recolecta información para detectar los síntomas y problemas del usuario.

HealthBuddy [12] busca mantener una conversación con el usuario de manera natural y empática. C Bot [6] se comunica con el usuario mediante aplicaciones como Facebook o Whatsapp. TELEBOT [25] está contenido en una página web y permite mantener la comunicación a través de Facebook Messenger, SnatchApp, Skype, WeChat y Telegram Messenger utilizando la plataforma Snatchbot. Dr.TARS [15] es una aplicación que puede conectarse por diversas plataformas como PC (Windows) o móviles (iOs y Android), además de buscar entender las intenciones del usuario.

MedChat [16] es una aplicación para el sistema operativo Android y se comunica con el usuario para conocer sus intenciones.

Disha [19] puede tener una conversación básica para entender las intenciones del usuario.[1] puede funcionar con Telegram y cuenta con una base de usuarios para personalizar las respuestas. Quro [11] lleva la conversación a partir de las entradas del usuario. COOPERA [27] y [3] recuperan información del estado de salud del usuario para proporcionar sus respuestas.

Medbot [7] mantiene un flujo de conversación mediante respuestas cortas. [3] calcula los niveles de la enfermedad basado en las indicaciones del usuario. Babylon [5], es un chatbot dedicado a los profesionales de salud y solo busca clasificar a los pacientes por grado de urgencia. Dejal@Bot [4] es un chatbot “humanizado” que busca entender y adaptarse durante la interacción con el usuario. Finalmente Medibot y Fabi [18] responden a información de los productos Pfizer.

#### 4.4. Nivel de automatización

La variable nivel de automatización se refiere al nivel de intervención humana (del lado del chatbot) durante la conversación. En [2] esta variable se denomina «ayuda humana» y proponen dos tipos: mediados por el humano (i.e., semi-automatizados) donde un ser humano interviene en algún momento en la conversación y completamente automatizados donde no hay intervención humana. En este artículo se añade el tipo no automatizado donde un ser humano lleva la conversación de principio a fin.

Medibot [24], iHelpr [9], HealFavor [13, 20], HealthBuddy [12], C Bot [6], TELEBOT [25], Dr. TARS [15], MedChat [16], Disha [19, 1], Quro [11, 3], Medbot [7], Babylon [5], Dejal@Bot [4], y Medibot y Fabi [18] son chatbots completamente automatizados.

#### 4.5. Forma de comunicación

La forma de comunicación se refiere a la manera en que el usuario ingresa la información para comunicarse con el chatbot. En esta clasificación se considera la comunicación por texto o por voz.

iHelpr [9] puede utilizar como respuesta algunos GIFs pero el usuario no puede utilizar emojis o GIFs, solo texto. HealFavor [13], Medibot [24, 20], C Bot [6], TELEBOT [25], Dr. TARS [15], MedChat [16], Disha [19, 1], Quro [11, 3], Babylon [5], Dejal@Bot [4] se comunican con el usuario mediante texto. Medbot [7] se comunica con el usuario mediante texto y voz. Medibot y Fabi [18] utilizan un menú desplegable para que el usuario ingrese las respuestas. El usuario se comunica con HealthBuddy [12] principalmente a través de respuestas predefinidas y en algunas ocasiones se permite el uso de texto libre.

#### 4.6. Plataforma de desarrollo

La variable plataforma de desarrollo permite clasificar los chatbots de acuerdo a los permisos otorgados por la plataforma en que han sido desarrollados.

Para ello, las plataformas pueden ser de uso libre como RASA o cerradas. Estas últimas normalmente son comerciales y/o pertenecen a grandes compañías como Google o IBM y el código fuente no esté disponible.

Medibot [24] utiliza una plataforma de código abierto y emplea el lenguaje AIML (basado en XML) para el reconocimiento de patrones. iHelpr [9] se desarrollo utilizando una plataforma de código abierto llamada Microsoft Bot Framework, utiliza NodeJS como entorno de ejecución, se encuentra conectado a una base de datos MySQL y utiliza el servicio de Microsoft's Language Understanding Intelligent Service (LUIS) para el reconocimiento del lenguaje natural. LUIS está basado en la nube y es código cerrado.

Healfavor [13] utiliza una plataforma de código abierto llamada RASA. El chatbot [20] utiliza el lenguaje AIML y el API REST. HealthBuddy [12] utiliza Botsociety para diseñar las conversaciones y la plataforma Dialogflow (parte de Google Cloud) para el desarrollo del prototipo. C Bot [6] fue desarrollado utilizando Python con ayuda de la biblioteca Beautiful Soup que tiene una licencia MIT que es de tipo libre permisiva, de modo que el software puede ser redistribuido como libre o privado.

TELEBOT [25] se desarrollo utilizando la plataforma propietaria de Snatchbot aunque su uso es gratuito. Dr. TARS [15] solo menciona que utiliza un formato JSON y que funciona tanto en PC como en dispositivos móviles. Sin embargo, no especifica las tecnologías utilizadas. MedChat [16] fue desarrollado en Python con la biblioteca Natural Language Tool Kit (NLTK) disponible en GitHub. Disha [19] está desarrollado utilizando Python con la biblioteca Scikit-learn de software libre.

[1] está desarrollado en Java, utiliza el código cerrado y gratuito de API Telegram Bot, Akka Java como un kit de herramientas de ejecución gratuito y de código abierto y Elasticsearch de código abierto. COOPERA [27] fue desarrollado por Line Medbot y almacena sus datos en el servicio AWS. [3] desarrolla la interfaz web con el lenguaje Java. [7] utiliza la nube de Google y su API llamado Google Cloud Speech-to-Text, ambos de código cerrado.

Babylon [5] utiliza una plataforma basada en la web semántica, utiliza tanto los lenguajes estructurados (RDF/OWL) como semi-estructurados (XML, CSV/TSV) como fuente original de los datos, GraphDB-Free como herramienta para convertir los datos fuentes y un servicio llamado ClinicalKnowledge construido como servicio REST.

Quoro [11] recupera las entidades médicas utilizando múltiples ontologías, entre ellas UMLS que es de código cerrado pero cuyo uso es gratuito. Para manejar las tareas pregunta-respuesta complejas utilizaron un grafo de conocimiento médico realizado por los autores. Medibot y Fabi [18] no especifican la tecnología utilizada.

#### **4.7. Procesamiento de entrada / método de generación de respuesta**

En esta clasificación encontramos a los chatbots que están basados en reglas, los basados en la recuperación y los que tienen un modelo generativo. Los que están basados en reglas se encargan de responder las cuestiones del usuario usando un conjunto de reglas preestablecidas, mientras que los que están basados en la recuperación se encargan de analizar las intenciones del usuario de acuerdo a un índice que coincida con las posibles respuestas para el usuario y normalmente utilizan un API.

Por último, los modelos generativos responden al usuario dependiendo del contexto previo de la conversación, utilizan algoritmos de aprendizaje automático y de apren-

dizaje profundo y requieren entrenamiento previo. Medibot [24] es un chatbot basado en reglas debido a que sigue una serie de estructuras definidas para poder realizar un diagnóstico del paciente y utiliza el reconocimiento de patrones.

iHelpr [9] inicia la conversación con una serie de preguntas que guían la conversación y espera un tipo de oración como respuesta, por lo que es un chatbot basado en reglas. HealFavor [13] utiliza el reconocimiento de patrones, de acuerdo con la información introducida el usuario, para extraer las entidades clave de la oración.

El chatbot de [20] realiza un reconocimiento de patrones para proveer una solución basada en los síntomas provistos por el usuario, si éste no ingresa el nombre de la enfermedad. C Bot [6] realiza un reconocimiento de patrones para obtener las intenciones e identificar las entidades del mensaje. Una vez obtenidas las intenciones y entidades el sistema ajusta con los datos apropiados en la base de datos. TELEBOT [25] utiliza un árbol de toma de decisiones para llevar a cabo la conversación, por lo que es un chatbot basado en reglas.

Dr.TARS [15] utiliza las palabras clave para buscar coincidencias con un mensaje previo y mostrar una respuesta. Además, con otra función identifica los síntomas asociados a las enfermedades para mostrar la respuesta adecuada al usuario. MedChat [16] utiliza un algoritmo de aprendizaje automático para reconocer algunos patrones de conversación. Disha [19] busca los síntomas que coinciden en la base datos para dar información sobre alguna enfermedad. [1] para generar la respuesta a partir de una base de datos y cubre las preguntas que se realizan de manera frecuente.

Quro [11] utiliza un grafo de conocimiento para explorar las asociaciones entre las entidades, mediante una clasificación de los síntomas, y envía una respuesta al usuario mediante una plantilla. [3] utiliza reconocimiento de patrones para generar la respuesta. Medbot [7] utiliza dos algoritmos, uno de coincidencia gramatical basado en regla y coincidencia por aprendizaje automático.

Babylon [5] crea un grafo para generar las respuestas, donde cada nodo es asociado a una expresión y estos nodos determinan las siguientes preguntas que se le realizaran al usuario. Dejal@Bot [4] tiene un conjunto de escenarios para seguir un guión para responder al usuario. Medibot y Fabi [18] no especifican el método de generación de la respuesta para el usuario.

#### 4.8. Idioma

El idioma es una parte importante en la tecnología de los chatbots porque determina el enfoque apropiado de PNL a ser aplicado durante el análisis de las intenciones.

Medibot [24], iHelpr [9], C bot [6], Dr.TARS [15], MedChat [16], Quro [11, 3, 5] y Medibot [18] utilizan el idioma inglés. TELEBOT [25], Dejal@Bot [4] están en idioma español. HealthBuddy [12] está en noruego. Disha [19] está en idioma bengalí. [1] está en idioma ruso y Medbot [7] utiliza tanto el idioma inglés como el hindi y Fabi [18] en portugués. HealFavor [13, 20] no mencionan el idioma.

### 5. Análisis de resultados

En la tabla 1 se muestra una tabla comparativa de los trabajos incluidos en esta revisión. Dado que la clasificación propuesta en este artículo solo considera los chatbots

en el área de la salud, todos los trabajos analizados son de dominio cerrado. En cuanto a la variable objetivo del chatbot, solo Medibot [18] es un chatbot informativo, el resto de los trabajos analizados están basados en la conversación. Ninguno de los trabajos revisados está basado en la tarea.

Al momento de analizar el servicio provisto solo Medibot [18] es de tipo interpersonal. El resto de los trabajos analizados son intrapersonales y ningún chatbot es de tipo inter-agente. Respecto a la variable nivel de automatización todos los chatbots están completamente automatizados.

Sin embargo, [23] sugiere la especialidad médica a la que el paciente puede ir para atenderse. Por otro lado, si se considera la variable forma de comunicación se puede observar que solo Medbot [7] considera tanto el texto como la voz.

Si consideramos la plataforma de desarrollo, [20], Medibot [24, 13], HealFavor [13], C Bot [6], MedChat [16], Disha [19], Babylon [5] utilizan una plataforma de código abierto. HealthBuddy [12], TELEBOT [25], Medbot [7], Quro [11] son chatbots que utilizan plataformas de código cerrado.

Se asume que COOPERA [27] utiliza código cerrado porque el desarrollo fue realizado por la compañía LINE. iHelpr [9] y [1] utilizan tanto código abierto como cerrado. Por último, [3], Dr. TARS [15], Dejal@Bot [4], Medibot y Fabi [18] no especifican la plataforma de desarrollo.

Durante el análisis de la variable procesamiento de entrada / método de generación de respuesta Medibot [24], iHelpr [9], HealFavor [13, 20], HealthBuddy [12], C Bot [6], TELEBOT [25], Dr. TARS [15, 1], Quro [11, 3], Medbot [7], Babylon [5], Dejal@Bot [4] son chatbots basados en reglas. MedChat [16] es un chatbot cuyo método de respuesta es generativo. Por último, Medibot y Fabi [18] no especifican el método para generar la respuesta al usuario.

Durante el análisis del idioma se encontró que solo TELEBOT [25] y Dejal@Bot [4] están en idioma español. HealthBuddy [12] en noruego, Disha [19] en bengalí, [1] en ruso, Medbot [7] utiliza tanto el idioma inglés como el hindi. Fabi [18] en portugués. Salvo los trabajos que no indican el idioma, el resto de los trabajos están en inglés.

## **6. Conclusiones**

En este artículo se presenta una revisión de chatbots orientados a la salud. El análisis de los trabajos se clasificó de acuerdo a las siguientes variables: dominio del conocimiento, objetivo, servicio provisto, nivel de automatización, forma de comunicación, plataforma de desarrollo, procesamiento de entrada / método de generación de respuesta e idioma.

Entre los principales resultados se encontró que la mayoría de los chatbots están basados en la conversación, son intrapersonales, la forma de comunicación es por texto, están basados en reglas y la conversación es en inglés.

Todos los chatbots analizados son completamente automatizados. También se observa que faltan chatbots en idiomas diferentes al inglés. Se encontró que existen chatbots que están dedicados a atender problemas de salud mental y de salud en general, además de chatbots para detectar el COVID-19.

**Tabla 1.** Chatbots incluidos en el estudio.

Trabajo	Objetivo	Servicio Provisto	Nivel de automatización	Forma de comunicación	Plataforma de desarrollo	Procesamiento de entrada/ Método de generación de respuesta	Idioma
[20]	BC	Intrap	A	Texto	Código abierto (AIML, API REST, XML)	BReg	–
Medibot [24]	BC	Interp	A	Texto	Código abierto (AIML)	BReg	Inglés
iHelpr [9]	BC	Intrap	A	Texto	Código abierto (Microsoft Bot Framework, NodeJS, MySQL) y cerrado (Microsoft's Language Understanding)	BReg	Inglés
HealFavor [13]	BC	Intrap	A	Texto	Código abierto (framework RASA)	BReg	–
HealthBuddy [12]	BC	Interp	A	Texto	Código cerrado (Dialogflow)	BReg	Noruego
C Bot [6]	BC	Intrap	A	Texto	Código abierto (Python y Beautiful Soup)	BReg	Inglés
TELEBOT [25]	BC	Intrap	A	Texto	Código cerrado (plataforma Snatchbot)	BReg	Español
Dr.TARS [15]	BC	Intrap	A	Texto	–	BReg	Inglés
MedChat [16]	BC	Intrap	A	Texto	Código abierto (Python y NLTK)	Gen	Inglés
Disha[19]	BC	Intrap	A	Texto	Código abierto (Python y Scikit Learn)	BReg	Bangla
[1]	BC	Interp	A	Texto	Código cerrado (API Telegram Bot) y abierto (Akka Java,ElasticSearch)	BReg	Ruso
Quro [11]	BC	Intrap	A	Texto	Código cerrado	BReg	Inglés
COOPERA [27]	BC	Intrap	A	Texto	Código cerrado (LINE y AWS)	–	–
[3]	BC	Intrap	A	Texto	–	BReg	Inglés
Medbot [7]	BC	Intrap	A	Texto y Voz	Código cerrado (Google Cloud)	BReg y BRec	Inglés, Hindi
Babylon [5]	BC	Intrap	A	Texto	Código abierto (REST, Graph DB-Free)	BReg	Inglés
Dejal@bot [4]	BC	Intrap	A	Texto	–	BReg	Español
Medibot [18]	I	Interp	A	Texto	–	–	Inglés
Fabi [18]	BC	Intrap	A	Texto	–	–	Portugués

Sin embargo, falta desarrollar más chatbots para enfermedades crónicas como el cáncer y cuyo procesamiento de entrada / método de generación de respuesta sea generativo. Este estudio tiene las siguientes limitaciones: la investigación se realizó solo con los artículos de acceso abierto o disponibles a través de <https://bidi.uam.mx/>. El filtro se redujo a artículos escritos en el idioma inglés o español. Finalmente, cabe señalar que al no tener involucrados especialistas en el área de la salud, esta revisión se enfoca en los aspectos tecnológicos.

Como trabajo futuro el objetivo es proponer el diseño y desarrollo de un chatbot que permita atender dudas sobre el cáncer y que se encuentre en idioma español y cuya forma de comunicación incluya la voz.

## Referencias

1. Abashev, A., Grigoryev, R., Grigorian, K., Boyko, V.: Programming tools for messenger-based chatbot system organization: Implication for outpatient and translational medicines. *BioNanoScience*, vol. 7, no. 2, pp. 403–407 (2017) doi: 10.1007/s12668-016-0376-9
2. Adamopoulou, E., Moussiades, L.: An overview of chatbot technology. In: Maglogiannis, I., Iliadis, L., Pimenidis, E. (eds) *Artificial Intelligence Applications and Innovations (AIAI)*. pp. 373–383. *Advances in Information and Communication Technology (IFIP)*, Springer, Cham (2020) doi: 10.1007/978-3-030-49186-4\_31

3. Athota, L., Shukla, V. K., Pandey, N., Rana, A.: Chatbot for healthcare system using artificial intelligence. In: 8th International Conference on Reliability, Infocom Technologies and Optimization (Trends and Future Directions) (ICRITO). pp. 619–622. IEEE (2020) doi: 10.1109/ICRITO48877.2020.9197833
4. Avila-Tomás, J. F., Olano, E. E., Minué, L. C., Martínez, S. F. J., Matilla, P. B., Serrano, S. M. E., Güeto-Rubio, M. V.: Dejal@Bot: Un chatbot aplicable en el tratamiento de la deshabituación tabáquica. *Revista de Investigación y Educación en Ciencias de la Salud (RIECS)*, vol. 5, no. 1, pp. 33–41 (2020) doi: 10.37536/RIECS.2020.5.1.196
5. Barisevičius, G., Coste, M., Geleta, D., Juric, D., Khodadadi, M., Stoilos, G., Zaihrayeu, I.: Supporting digital healthcare services using semantic web technologies. In: Vrandečić, D., Bontcheva, K., Suárez-Figueroa, M. C., Presutti, V., Celino, I., Sabou, M., Kaffee, L. A., Simperl, E. (eds) *The Semantic Web – ISWC 2018*, vol. 11137, pp. 291–306. Springer, Cham (2018), *Lecture Notes in Computer Science*
6. Belfin, R. V., Shobana, A. J., Manilal, M., Mathew, A. A., Babu, B.: A graph based chatbot for cancer patients. In: 5th International Conference on Advanced Computing & Communication Systems (ICACCS). pp. 717–721. IEEE (2019) doi: 10.1109/ICACCS.2019.8728499
7. Bharti, U., Bajaj, D., Batra, H., Lalit, S., Lalit, S., Gangwani, A.: Medbot: Conversational artificial intelligence powered chatbot for delivering tele-health after COVID-19. In: 5th International Conference on Communication and Electronics Systems (ICCES). pp. 870–875. IEEE (2020) doi: 10.1109/ICCES48766.2020.9137944
8. Borah, B., Pathak, D., Sarmah, P., Som, B., Nandi, S.: Survey of textbased chatbot in perspective of recent technologies. In: Mandal, J. K., Mukhopadhyay, S., Dutta, P., Dasgupta, K. (eds) *2nd International Conference on Computational Intelligence, Communications, and Business Analytics, (CICBA, Part 2)*. vol. 1031, pp. 84–96. Springer (2019) doi: 10.1007/978-981-13-8581-0\_7, *communications in Computer and Information Science*
9. Cameron, G., Cameron, D., Megaw, G., Bond, R., Mulvenna, M., O’Neill, S., Armour, C., McTear, M.: Assessing the usability of a chatbot for mental health care. In: Bodrunova, S. S., Koltsova, O., Følstad, A., Halpin, H., Kolozaridi, P., Yuldashev, L., Smoliarova, A., Niedermayer, H. (eds) *Internet Science, INSCI International Workshop*, vol. 11551, pp. 121–132. Springer, Cham (2019), *Lecture Notes in Computer Science*
10. Ferlay, J., Ervik, M., Lam, F., Colombet, M., Mery, L., Piñeros, M., Znaor, A., Soerjomataram, I., Bray, F.: *Cancer Today: Data Visualization tool for exploring the global cancer burden in 2020* (2020), <https://gco.iarc.fr/today>, Global cancer observatory, International Agency for Research on Cancer, World Health Organization, January
11. Ghosh, S., Bhatia, S., Bhatia, A.: Quro: Facilitating user symptom check using a personalised chatbot-oriented dialogue system. In: Cummings, E., Ryan, A., Schaper, L. K. (eds) *26th Australian National Health Informatics Conference. Studies in health technology and informatics*. vol. 252, pp. 51–56 (2018) doi: 10.3233/978-1-61499-890-7-51

12. Hoiland, C. G., Folstad, A., Karahasanovic, A.: Hi, can I help? exploring how to design a mental health chatbot for youths. *Human Technology*, vol. 16, no. 2, pp. 139–169 (Aug 2020) doi: 10.17011/ht/urn.202008245640
13. Khilji, U. R. A. F., Laskar, S. R., Pakray, P., Kadir, R. A., Lydia, M. S., Bandyopadhyay, S.: HealFavor: Dataset and a prototype system for healthcare chatbot. In: *International Conference on Data Science, Artificial Intelligence, and Business Analytics (DATABIA)*. pp. 1–4 (2020) doi: 10.1109/DATABIA50434.2020.9190281
14. Loveys, K., Sebaratnam, G., Sagar, M., Broadbent, E.: The effect of design features on relationship quality with embodied conversational agents: A systematic review. *International Journal of Social Robotics*, vol. 12, no. 6, pp. 1293–1312 (2020) doi: 10.1007/s12369-020-00680-7
15. Madhu, D., Jain, C. N., Sebastain, E., Shaji, S., Ajayakumar, A.: A novel approach for medical assistance using trained chatbot. In: *International Conference on Inventive Communication and Computational Technologies (ICICCT)*. pp. 243–246 (2017) doi: 10.1109/ICICCT.2017.7975195
16. Mathew, R. B., Varghese, S., Joy, S. E., Alex, S. S.: Chatbot for disease prediction and treatment recommendation using machine learning. In: *3rd International Conference on Trends in Electronics and Informatics (ICOEI)*. pp. 851–856 (2019) doi: 10.1109/ICOEI.2019.8862707
17. Nimavat, K., Champaneria, T.: Chatbots: An overview. types, architecture, tools and future possibilities. *International Journal for Scientific Research & Development*, vol. 5, no. 7, pp. 1019–1024 (2017)
18. Pfizer: Meet the new digital assistants transforming patient medical information. Pfizer Inc. (2022)
19. Rahman, M. M., Amin, R., Khan Liton, M. N., Hossain, N.: Disha: An implementation of machine learning based bangla healthcare chatbot. In: *22nd International Conference on Computer and Information Technology (ICCIT)*. pp. 1–6 (2019) doi: 10.1109/ICCIT48885.2019.9038579
20. Rarhi, K., Bhattacharya, A., Mishra, A., Mandal, K.: Automated medical chatbot. *SSRN Electronic Journal*, (2017) doi: 10.2139/ssrn.3090881
21. Safi, Z., Abd-Alrazaq, A., Khalifa, M., Househ, M.: Technical aspects of developing chatbots for medical applications: scoping review. *Journal of Medical Internet Research*, vol. 22, no. 12 (2020) doi: 10.2196/19127
22. Siegel, R. L., Miller, K. D., Fuchs, H. E., Jemal, A.: Cancer statistics, 2022. *CA: A Cancer Journal for Clinicians*, vol. 72, no. 1, pp. 7–33 (2022) doi: 10.3322/caac.21708
23. Soewito, B., Limto, D., Yuanita, C., Vincent, Noprianto: Health monitoring with artificial intelligence. In: *2nd International Conference on Cybernetics and Intelligent System (ICORIS)*. pp. 1–6. IEEE (2020) doi: 10.1109/ICORIS50180.2020.9320793
24. Srivastava, P., Singh, N.: Automatized medical chatbot (medibot). In: *International Conference on Power Electronics & IoT Applications in Renewable Energy and its Control (PARC)*. pp. 351–354. IEEE (2020) doi: 10.1109/PARC49193.2020.236624

25. Stalin, E. W., Patricio, G. G., Carrión, B. C., Sánchez, S. I.: Chatbot implementation to collect data on possible COVID-19 cases and release the pressure on the primary health care system. In: 11th IEEE Annual Information Technology, Electronics and Mobile Communication Conference (IEMCON). pp. 0302–0307 (2020) doi: 10.1109/IEMCON51383.2020.9284846
26. Xu, L., Sanders, L., Li, K., Chow, J. C. L.: Chatbot for health care and oncology applications using artificial intelligence and machine learning: Systematic review. *JMIR Cancer*, vol. 7, no. 4 (2021) doi: 10.2196/27850
27. Yoneoka, D., Tanoue, Y., Kawashima, T., Nomura, S., Shi, S., Eguchi, A., Ejima, K., Taniguchi, T., Sakamoto, H., Kunishima, H., et al.: Large-scale epidemiological monitoring of the COVID-19 epidemic in Tokyo. *The Lancet Regional Health - Western Pacific*, vol. 3, pp. 1–8 (2020) doi: 10.1016/j.lanwpc.2020.100016

## Control difuso para la mimetización del movimiento de un prototipo de prótesis de una extremidad canina

Eduardo de Jesús García-Velázquez<sup>1</sup>,  
Eugenia-Graciela López-Miranda<sup>2</sup>,  
David Tinoco-Varela<sup>1</sup>, Raúl-Dalí Cruz-Morales<sup>1</sup>

<sup>1</sup> Universidad Nacional Autónoma de México,  
Facultad de Estudios Superiores,  
Ingeniería Mecánica Eléctrica  
México.

<sup>2</sup> Universidad Nacional Autónoma de México,  
Facultad de Estudios Superiores,  
Lic. en Medicina Veterinaria y Zootecnia,  
México

{albeitar, rdcruz}@comunidad.unam.mx,  
datival9@hotmail.com, jesus.g.velazquez@gmail.com

**Resumen.** En este artículo se presenta el desarrollo de un controlador difuso que se pretende sea implementado en una prótesis para un perro mestizo de raza grande, el cual carece del miembro torácico derecho, generando un movimiento inteligente en la prótesis. Este sistema consta de dos partes, una parte es el dispositivo que recaba los datos de movimiento de una pata canina sana y la otra parte que recibe tal información y la procesa para generar un movimiento natural en la pata amputada del can, ambas secciones del sistema propuesto comunicadas por medio de Bluetooth (con la intención de en un futuro integrarlo al IoT). Para el control difuso se recurre al uso de un sensor conocido como acelerómetro, el cual calcula en un tiempo definido de lectura y respuesta los movimientos que realiza el perro en su miembro sano; el comportamiento mimético de las extremidades al caminar permite crear un control que considera tanto la posición como la fuerza de los miembros traseros para formular las reglas que la lógica difusa necesita para replicar este comportamiento, aplicando el Método de Mamdani.

**Palabras clave:** Prótesis inteligente, lógica difusa, control difuso.

### Fuzzy Control for Motion Mimicry of a Canine Limb Prosthesis Prototype

**Abstract.** This paper presents the development of a fuzzy controller that will be implemented in a prosthesis for a large breed mongrel dog, which lacks the right thoracic limb, generating an intelligent movement in the prosthesis to be

developed. This system consists of two parts, a part that receives the movement data of a healthy canine paw and a part that receives such information and processes it to generate a natural movement in the canine's amputated paw, both sections of the proposed system communicated by Bluetooth medium. For fuzzy control, a sensor known as an accelerometer was used, which will calculate the movements made by the dog in its healthy limb at a defined reading and response time; the mimetic behavior of the limbs when walking will allow us to create a control that takes into consideration both the position and the strength of the hind limbs to formulate the rules that fuzzy logic (by Mamdani's Method).

**Keywords:** Intelligent prosthetics, fuzzy logic, fuzzy control.

## **1. Introducción**

El aspecto motriz es sumamente importante para la medicina veterinaria y la salud general de un perro, en este aspecto, obviamente, se ven involucradas las cuatro extremidades del can. Siendo que estas pueden sufrir traumatismos superficiales en algún momento por diversas causas, es importante considerar que estos traumatismos pueden provocar en él alguna hemorragia, infección o algún carcinoma que termine en una desafortunada amputación.

La amputación, en términos médicos, es una extirpación de una parte del cuerpo que se encuentre dañada y cuyo daño comprometa la salud general del animal. Los procedimientos más comunes respecto a esta intervención quirúrgica son el corte de la zona afectada y la desarticulación de la región donde el daño se desarrolla. Es decir, se puede cortar de tajo la zona o ubicar la región de la extremidad y extirpar la parte que se encuentre bajo una articulación.

Un perro puede llevar una vida normal con únicamente tres de sus extremidades, o incluso dos y ser tratado con una silla de ruedas que supla en ciertos aspectos el movimiento que sus miembros amputados le otorgaban.

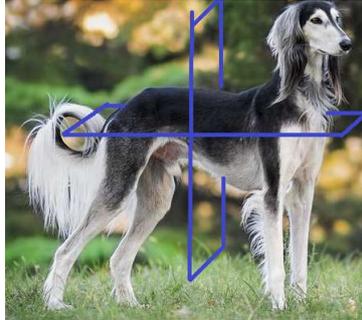
La problemática se presenta cuando el estado de cojera decae en problemas óseos en la extremidad de soporte al intentar caminar o trotar, provocando que su centro de gravedad, sobre la misma columna, cambie abruptamente causando daños en esta parte del cuerpo, parte sumamente delicada. La medicina veterinaria moderna, propone a los dueños como solución el implemento de prótesis mecánicas que suplan la o las extremidades perdidas.

En México las prótesis mecánicas para miembro torácico son escasas pues son pocas las organizaciones que realizan este tipo de prótesis, y aún al día de hoy, con costes entre \$1,350 y 1,750 dólares.

El propósito central del desarrollo presentado en este artículo es la creación de un control difuso que se pueda implementar en un aditamento mecatrónico que proporcione al perro una mejor calidad de vida.

## **2. Elementos de un sistema de prótesis canino**

En la práctica veterinaria existe una denominación para cada parte del cuerpo respecto a la geometría y su relación con los planos cartesianos, lo cual es apreciable



**Fig. 1.** Planos anatómicos en el perro. Imagen tomada y modificada a partir de <https://pixabay.com/es/photos/naturaleza-perro-greyhound-3047194/>.

en la figura 1; dada la universalidad de esta nomenclatura, nos serviremos de ella para denominar los elementos de un prótesis. La comunión entre términos de ingeniería y veterinaria es importante para futuros desarrollos.

Para el miembro torácico, se consideran los siguientes términos que son utilizados a lo largo del texto en relación con la ubicación de cada elemento. La nomenclatura se basa en la aceptada desde 1968 por la *Nómina Anatómica Veterinaria*: Dorsal: Cualquier entidad que se encuentre más cerca del dorso; Ventral: Cualquier entidad que se encuentre más cercana al vientre; Medial: Entidades cercanas al plano medio o dorsal; Distal: Cualquier entidad que se encuentre en dirección opuesta al plano medial; Proximal: Opuesto al distal, toda entidad que se encuentre en dirección hacia el plano medial.

El miembro torácico, como parte del conjunto que presenta el sistema locomotor en el perro, está dividido en cuatro regiones las cuales poseen huesos y músculos que proporcionan movimientos específicos que actuando en sinergia otorgan el andar característico de estos animales.

Se conoce como biomecánica al estudio de las interacciones entre fuerzas, velocidades y aceleraciones en los organismos vivos. Toda entidad biológica compleja, especialmente los animales, está sujeta a las mismas leyes de la mecánica clásica que los cuerpos inanimados, con la salvedad de que estos son capaces de “producir” fuerzas propias con la que sus elementos locomotores (huesos, articulaciones, músculos) generen desplazamiento.

## 2.1. Milo: paciente a la espera de su prótesis

Para poder hacer este proyecto realidad se ha encontrado un perro con la triste situación de no contar con su extremidad delantera derecha. Este es el caso de Milo.

Milo en la figura 2 (izquierda) es un perro mestizo adulto de aproximadamente 6 años. En una revisión de rutina con su médico veterinario, los datos anatómicos obtenidos de Milo fueron los siguientes: Masa total: aprox. 30.5 Kg; Altura hasta la cruz: 58 cm; Longitud normal de escápula izquierda: 19 cm; Longitud normal de húmero izquierdo: 23 cm; Longitud normal de ulna izquierda: 14 cm; Longitud normal de extremidad residual derecha: 11 cm.



**Fig. 2.** Milo, un cachorro con pérdida de miembro torácico delantero. A la derecha, prueba de sensor acelerómetro para comunicar al sistema difuso.

### 3. Lógica y control difuso

Se conoce a la inteligencia artificial como la capacidad que tiene las máquinas de interpretar e interactuar con el entorno de manera muy similar a como lo haría el ser humano. Estas interacciones se dan a través de dos características fundamentales, las cuales son atribuidas al concepto general de inteligencia: el razonamiento y el aprendizaje.

En un sistema inteligente, la toma de decisiones autónomas basadas en parámetros de entrada puede ser administrada de manera eficiente por la lógica difusa. A continuación, se presenta una explicación breve de los conceptos más generales de esta rama de la I.A. y cómo puede ser utilizado en el presente proyecto.

La lógica difusa es una rama de la inteligencia artificial que permite a una computadora analizar información del mundo real en una escala entre lo falso y lo verdadero [1]. Fue desarrollada a principios de la segunda mitad del s. XX por el matemático Lofti A. Zadeh [2].

Como es bien sabido, las máquinas poseen un lenguaje, el cual está basado en un sistema de numeración binaria; es decir, solamente acepta estados de niveles bajos o “ceros” y estados de niveles altos o “unos”, sin embargo, para un modelado más realista de un fenómeno físico, se utilizan valores reales entre 0 y 1, evaluación que nos permite realizar la lógica difusa.

Un ejemplo muy simple que se le puede dar a este concepto es el siguiente: “Juanito no gusta de la leche caliente porque le quema la lengua. A Juanito le gusta la leche tibia”.

El concepto que se está manejando en esta frase, la naturaleza de lo que se habla, es la temperatura de una bebida, la leche. Aquí se encuentran dos conceptos bien conocidos por las personas: “caliente” y “tibio”. Un valor dentro de la lógica formal sería 10 °C como una bebida fría y 60 °C como una bebida caliente.

Sin embargo, para una persona que no sea Juanito, 60 °C podría parecerle no tan caliente, o incluso los 10 °C le puede parecer “tibio”. El concepto de “tibio”, de una manera muy burda, se puede decir que se encuentra entre caliente y frío; es decir, si caliente es el máximo (uno) y frío es el mínimo (cero), el concepto de tibio no podría entrar en la lógica formal. Este solamente entraría en el de la lógica difusa como un 0.5, la pertenencia del valor de tibio es del 50% para caliente y 50% para frío.

Explicado esto, entra en el juego un segundo concepto de suma importancia para entender de manera completa la lógica difusa: grado de pertenencia. Este concepto refiere al punto que intersecta el valor de la variable de entrada a un conjunto difuso, es decir, que tanto el dato se encuentra dentro de un conjunto.

### **3.1. Proceso para la generación de un control difuso**

El controlador difuso, se compone de diversas secciones y cada una de ellas realiza un determinado tratamiento a los datos de entrada, para así obtener una conclusión “lógica” en la salida, a continuación, se describen los pasos a seguir:

1. Las características que describan al fenómeno a modelar, se deben interpretar en conjuntos difusos. Estos valores pueden tener un rango tan amplio como se quiera tomando en consideración la distribución de los datos dentro de los mismos,
2. Una vez seleccionada la naturaleza de los datos que se requieren, se procede al proceso de fusificación. Es decir, convertir los elementos de entrada en valores reales entre 1 y 0,
4. El siguiente paso es la asignación de reglas. Esta puede realizarse a través de una matriz de decisiones o reglas, bajo las cuales las entradas interactuarán e impactan a un conjunto difuso de salida,
5. El último punto del proceso es la defusificación. Es decir, convertir los valores difusos del conjunto de salida en valores nítidos o valores capaces de ser interpretados por una máquina. Para ello, existen diferentes métodos, entre ellos el método de Mamdani y el de Sugeno.

Estos sistemas han sido eje principal de desarrollo de una gran cantidad de aplicaciones e interfaces, entre las que se encuentran, el control de señales PWM para mejorar el rendimiento y la eficiencia en unidades de frecuencia variable (VFD) [3]; el control de la estabilización de un sistema de silla de ruedas [4]; control de sistemas de robots con llantas [5]; para el control de sistemas domóticos [6]; y en particular una aplicación interesante para los fines de este proyecto, es el control de diferentes tipos de prótesis [7, 8, 9].

## **4. Expectativa**

A través de la teoría desarrollada por M. Hildebrand sobre la simetría en los patrones de movimiento de los cuadrúpedos, el diseño de control inteligente en un perro de raza grande, el cual carece de su miembro torácico derecho, debe ser capaz de leer de manera inversa los patrones de movimiento de la extremidad delantera izquierda; a través de un control difuso se hace la lectura de todo un universo de datos de los ángulos de ataque sobre la articulación del hombro que es la que presenta un desplazamiento más significativo y marca el ritmo de la marcha.

Este controlador debe ser capaz de accionar un servomecanismo previamente diseñado por las teorías de síntesis de máquinas que emule de manera cercana el

movimiento natural de las extremidades, para otorgar al animal mejor movilidad y extender su calidad de vida.

La teoría de Hildebrand estipula que la marcha en los animales se compone de movimientos repetitivos y simétricos [10] entre cada una de las extremidades desplazándose a una secuencia claramente observable por métodos utilizados en el estudio veterinario.

## **5. Diseño de un sistema embebido de control electrónico**

Para que un sistema embebido se pueda considerar un elemento constitutivo y de control en un sistema mecatrónico, los elementos de entrada de datos deben ser sensores y las salidas deseadas, efectuadas por actuadores.

Considerando que un sistema mecánico posee como variables desplazamientos angulares para ejecutar su accionamiento, es necesario el uso de un sensor que sea capaz de interpretar los cambios angulares con precisión [11].

Dado que buscamos la capacidad del sistema a reproducir un movimiento de tipo “espejo” de la extremidad delantera derecha de un can frente a su opuesta, es necesario determinar las posiciones o los estados de posición de un punto determinado (en este caso la variación de cualquier punto del hombro en la línea normal), por lo que el sistema embebido debe estar unido a la pata sana del perro para poder detectar estas variaciones de posición y, por tanto, su tamaño debe ser pequeño, no tan pesado, que pueda ser adaptado a la pata del animal sin lastimarlo y sin que el sistema electrónico se maltrate o se caiga constantemente.

Con estas condiciones se puede determinar una característica discriminatoria para el sensor a utilizar: el tamaño. Ahora, con estos dos datos podemos encontrar un sensor de tamaño reducido y capaz de medir la variación del ángulo, es decir, un acelerómetro, en este caso se utiliza el MMA7361.

### **5.1. Sistema embebido de obtención de datos**

Siendo que, lo que se necesita de la extremidad sana, es solo la posición de esta con respecto a la normal, es decir, cuando la pata está en la posición de reposo. Se considera esta, la posición de aplomo, como el punto de partida. Siendo que los datos que se deben obtener de este sistema son únicamente datos numéricos, transducidos por el sensor a señales eléctricas, el control que se requiere debe tener un único parámetro de entrada y de salida. Sólo debe emitir un tipo de dato como respuesta.

Dada la simplicidad, el tamaño y el bajo coste, se ha elegido el controlador PIC16F88 de gama media para el diseño del sistema recolector de información, el cual cuenta con entradas analógicas.

En la figura 3 se muestra el diagrama de conexión del primer sistema embebido (sin el elemento de comunicación), en este caso se colocó la conexión con una pantalla LCD 16x2 con un módulo I2C para verificar los datos y la conversión obtenidos de la pierna de Milo.

## **5.2. Control difuso para sistema de potencia**

Para el apartado del control de movimiento del mecanismo, hay que recordar que la hipótesis del presente trabajo se basa en la hipótesis de movimiento planteada por el fisiólogo Milton Hildebrand: los movimientos simétricos son aquellos que poseen una relación espejo con su par colindante, es decir, en movimientos como la caminata o el trote lento, el animal cuadrúpedo respecto a sus extremidades delanteras envía una de estas hacia adelante mientras la opuesta se envía hacia atrás para mantener el equilibrio.

Es decir, mientras existe un posicionamiento para iniciar la fase de impulso y mantener el equilibrio durante desplazamientos simétricos, el animal debe enviar una pata hacia adelante y mantener la otra hacia atrás.

Considerando lo anterior se puede asumir que, si tomamos el plano dorsal o el medial como referencia, existe un desplazamiento angular sobre los ejes  $x$  o  $z$  dependiendo de la orientación que se quiera tomar para el acelerómetro.

En este caso se toma el plano dorsal o ventral como referencia ya que el MMA7361L presenta problemas al censar el eje  $z$  (en caso de colocar el sistema embebido en paralelo con el plano medial), por lo que se coloca en horizontal, iniciando el punto de referencia del movimiento en la posición de aplomo, con las dos extremidades delanteras y considerando el ángulo escapular, ya que aquí se realiza la medición. En la figura 4 se muestra un esquema de cómo podría quedar el primer sistema respecto a la anatomía del perro.

Con esto en mente se propone la generación de un soporte para el sistema embebido del sensor, de tal manera que el aplomo sea el punto “cero” o de grado cero, teniendo el circuito embebido en horizontal y, de acuerdo con el avance o retroceso del miembro sano, se producen ángulos relativos a la aceleración registrada por el sensor.

Se ha elegido un método de toma de decisión por software para el segundo sistema embebido (control de prótesis), siendo este el empleo de un controlador por lógica difusa. Para el diseño del control difuso se ha utilizado el aplicativo Fuzzy Logic Toolbox de Matlab®.

Lo primero en considerar son los valores que va a recibir el sistema de control. Estos valores son los ángulos que se obtuvieron del sistema recolector de datos. Estos valores serán enviados a través de comunicación inalámbrica. Ahora, los datos son grados, y su naturaleza es de índole numérica.

Para que pueda ser parte del control difuso debe ser convertido a un valor difuso, a un valor que entre dentro de una serie de conjuntos y que este mismo valor forme parte de uno o más de estos conjuntos.

Los ángulos tienen, en la geometría clásica, una clasificación de concepto que puede utilizarse para generar los conjuntos difusos, en los que puede entrar cada ángulo con un grado de pertenencia, incrementado dos etiquetas lingüísticas más para ajustar los datos: agudo, medio agudo, recto, medio obtuso y obtuso. Estos conjuntos, como se han descrito, pueden contener con un cierto grado de pertenencia los valores que serán recibidos del primer sistema.

Los valores de pertenencia para cada ángulo son los valores convertidos a difusos. En la figura 5 se muestran los datos fuscados con sus funciones de membresía en los conjuntos difusos.

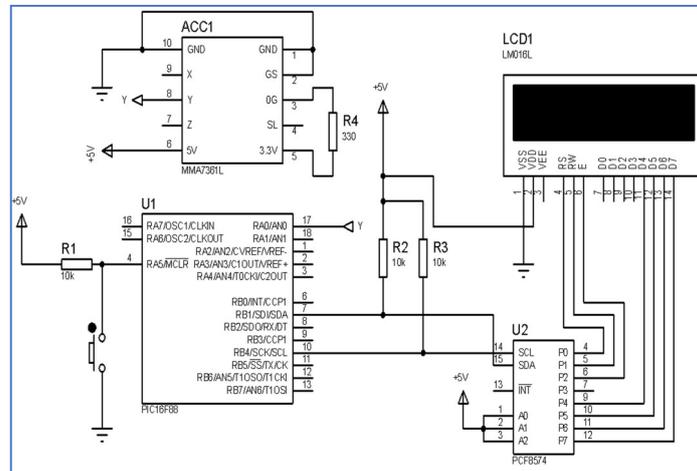


Fig. 3. Esquemático de prueba del acelerómetro MMA7361L.



Fig. 4. Localización del sensor acelerómetro en el cuerpo del perro. Imagen obtenida y modificada a partir de <https://pixabay.com/es/vectors/animal-canino-perro-naranja-1298936/>.

Ahora, ya definidos los conjuntos difusos, es importante replantear la teoría de Hildebrand: se requiere que el movimiento entre las extremidades sea simétrico y consecutivo, como si fuesen las cuchillas de unas tijeras.

Es así como se requiere que, si la extremidad sana avanza, generando por ejemplo un ángulo “agudo”, el mecanismo de potencia genere un giro que permita al prótesis realizar el retroceso para completar el movimiento.

Se ha seleccionado la secuencia de movimiento de la caminata como la secuencia de estudio para las reglas mandantes en el control difuso en la que la relación de los ángulos es contraria para cada evento, es decir, si el primer sistema embebido lee una lectura de un ángulo obtuso, la salida debe corresponder a un agudo; invirtiendo entonces también los conjuntos difusos a la salida, sin embargo, esto es tratado más

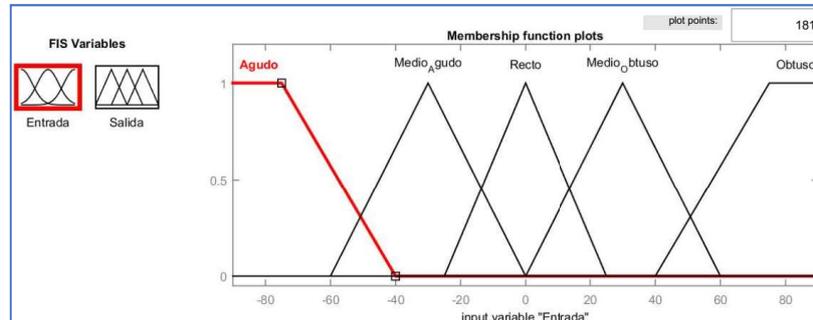


Fig. 5. Conjuntos difusos de entrada de los datos *fusicados*.

adelante. En la figura 6 se muestra el listado de reglas que se propuso para el control difuso.

Con estos datos se puede considerar la salida como un reflejo de los ángulos que se consideran agudos, rectos y obtusos dentro de los conjuntos definidos. En la figura 7 se muestra la distribución de los conjuntos de salida de acuerdo con sus funciones de membresía.

Siendo que ambos universos de conjuntos, tanto salidas como entradas, ya se encuentran en estado de variables difusas, es posible visualizar el comportamiento de las reglas en cualquier dato de entrada que el segundo sistema embebido reciba del primero.

El programa de reconocimiento evalúa el valor enviado por el primer sistema y lo compara dentro del valor asignado para la entrada (de acuerdo con su función de membresía) y sobre las mismas reglas, obteniendo por medio del método del centroide de Mamdani el valor correspondiente a la salida del controlador.

La figura 8 muestra un ejemplo de cómo el controlador difuso evalúa el dato y arroja un dato de salida *defusificado*, convirtiéndose en un dato numérico de la misma naturaleza que la entrada.

Con esto se puede visualizar que el controlador difuso es bidimensional, es decir, ya que solamente maneja una sola entrada se espera que la relación sea de tipo uno a uno, solamente habrá una entrada y la salida es proporcional a dicha salida.

Se ha propuesto el uso de un método de inteligencia artificial para mapear todos los datos posibles que registre el sensor. Ya que la sensibilidad del sensor tiende a inestabilidades y el diseño de un control más robusto está fuera de los alcances de este trabajo, el empleo de la lógica difusa ayuda a “amortiguar” posibles errores al ser capaz de detectar cualquier valor que envíe el acelerómetro y dictar su valor “espejo”. En la figura 9 se muestra la relación conjuntos de entrada y salida mediante una gráfica.

### 5.3. Servomotor como sistema de potencia mecánica

Para el impulso del sistema mecánico de potencia se utiliza un servomotor, el cual estará reaccionando en función de los valores de salida del controlador difuso. Cabe recordar que estos dispositivos son motores de CD, que poseen un reductor de velocidad



Fig. 6. Reglas del control difuso propuesto para la alternancia de ángulos.

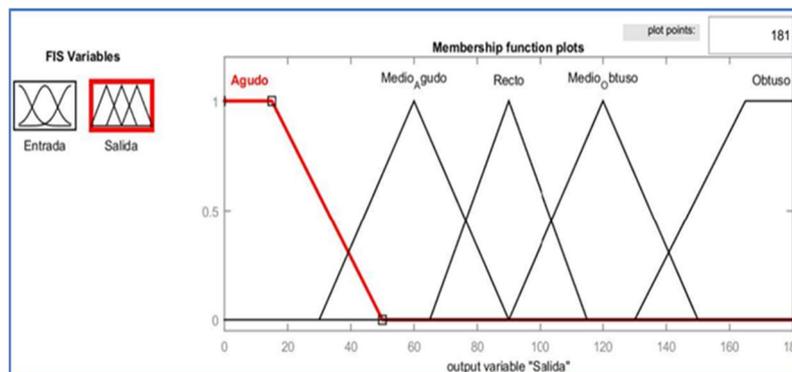


Fig. 7. Variables lingüísticas de la salida en conjuntos difusos.

un multiplicador de torque junto con un circuito de control. Este motor solamente puede dar revoluciones específicas en un rango de 0 a 180°.

Para poder controlar estos giros específicos es necesario aplicar un tren de pulsos de duración y frecuencia específicos. Dependiendo del ancho de las pulsaciones del tren, el sistema de control interno del servo lleva su eje a la posición correspondiente.

La manipulación o modulación de estos anchos de pulso o PWM (Pulse Width Modulation en inglés) consiste en generar una onda cuadrada en la que se varía el tiempo que el pulso está en nivel alto, manteniendo el mismo periodo, de tal manera que modifica la posición del eje de acuerdo con este ancho [12].

Ahora, es menester conocer el rango de ángulos y periodos del ancho de pulso en nivel alto del tren para identificar los posibles ángulos que se obtienen del control difuso cuando se evalué, es decir, el controlador difuso es el encargado de enviar el dato de salida al servomotor, pero como un valor de ángulo, es por ello que se requiere conocer el valor del pulso, que corresponde a cada ángulo, que envía el control difuso para que el microcontrolador lo envíe al motor.

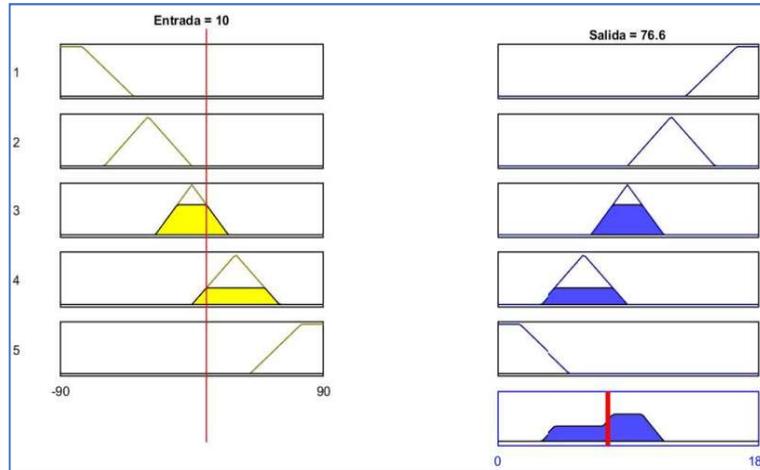


Fig. 8. Método de Mamdani para evaluación de datos de entrada por reglas, defusificado.

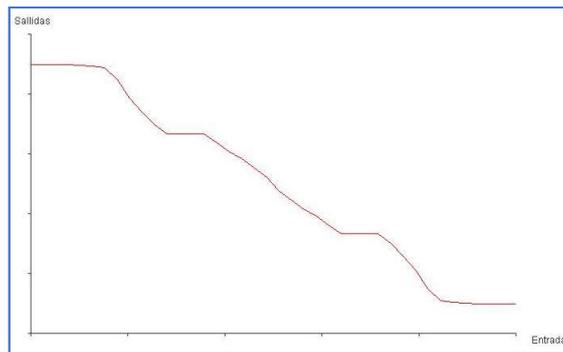


Fig. 9. Superficie de relación de entradas y salidas.

Tabla 1. Relación de ángulos del servomotor y el ancho de pulso del PWM.

Posición (°)	0	15	30	45	60	75	90	105	120	135	150	165	180
T. Alta (µs)	30	51	72	92	112	131	150	168	185	202	218	234	250
	0	4	2	5	2	4	0	1	6	5	9	8	1

En la tabla 1 se muestra una relación de ángulos a los cuales se les puede verificar la relación matemática para encontrar la ecuación que rige el comportamiento de los anchos de pulso.

#### 5.4. Sistema de comunicación vía bluetooth

Teniendo la base de acción de ambos sistemas (un emisor de datos o sistema sensor, y un receptor de datos o sistema actuador), el último paso del desarrollo es la comunicación.

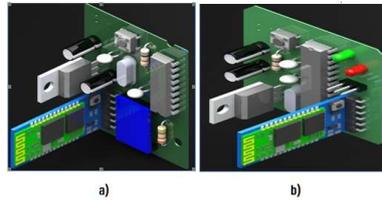


Fig. 10. a) Diseño CAD del circuito emisor; b) diseño CAD del circuito receptor.



Fig. 11. Sistema embebido del sensor con su contenedor: derecha, diseño renderizado; izquierda, producto final.

Se requiere que ambos sistemas trabajen en paralelo enviando y recibiendo información no bidireccionalmente sino unilateralmente, y que por ergonomía y comodidad del animal no se deben utilizar cables que, al final, sufrirán desgaste y quizá el perro llegue a morder o arrancar.

Es necesaria una comunicación inalámbrica y el medio más económico que puede encontrarse, sin recurrir a un módulo de radiofrecuencia es un módulo bluetooth, el cual ha sido utilizado en diferentes proyectos del Internet de las Cosas (IoT) [13, 14,15].

Los módulos con los que se trabaja para la comunicación entre ambos sistemas son los conocidos HC05.

## 6. Creación del circuito impreso

Con los esquemáticos ya creados, se genera el circuito impreso en placa fenólica para el montaje del prototipo de prótesis o PCB (*printed circuit board*). Es importante generar en primer lugar los sistemas embebidos para diseñar los contenedores mecánicos tanto para el dispositivo de transmisión de datos como el que va ligado al mecanismo ya diseñado.

Para esto se ha utilizado un software CAD para diseño de circuitos electrónicos, el mismo que se ha usado para generar los esquemáticos. En las figuras 10 a) y b) se muestran los circuitos generados por el programa CAD de diseño de circuitos electrónicos.

El resultado del ensamble para el sistema sensor con el acelerómetro MMA7361L y el microcontrolador PIC16F88 se muestra en la figura 11 junto con un renderizado del diseño original. Ahora, el diseño para el segundo sistema no solamente implica la adaptación del sistema electrónico de control, sino que además debe contener un espacio para ajustarse sobre el cuerpo del perro y el acople con el mecanismo motor.



**Fig. 12.** Adaptador del mecanismo para el muñón (modelo renderizado).

Una vez concluida la parte de control, así como el diseño del mecanismo del protésico, es necesaria la construcción de un aditamento que pueda colocarse a Milo en torno al muñón de su pata amputada, así como la posibilidad de colocarse de manera “embebida” tanto el sistema de control de potencia, el motor y la alimentación.

Durante las revisiones al perro se calcó el perfil del muñón para realizar un adaptador hueco.

Esto se realizó con la intención de crear una superficie para generar un sólido hueco o de tipo cascarón. Siendo el volumen de éste muy grande para el proceso de modelado se ha propuesto para este prototipo la división en dos del cascarón del sujetador, esto con el propósito de permitir al servomotor ajustarse sobre éste y permita las demás partes del protésico ser conectadas entre sí sin ocupar aditamentos mecánicos especiales para enlazar cada eslabón. En la figura 12 se muestra el modelo CAD de esta sección (sin las cavidades para insertar el circuito embebido de control y la batería).

## 7. Pruebas

A lo largo de este artículo se ha descrito el desarrollo de dos sistemas embebidos que forman parte de la prótesis en la cual se implementó el controlador difuso, uno de ellos es el sistema que capta los movimientos del pie sano de Milo, y el otro sistema, que se encuentra en la extremidad amputada.

Este último capta la señal del sistema de adquisición y los procesa por medio del controlador difuso, y de esta forma genera una salida que da como resultado el movimiento natural de la prótesis de la pierna amputada. Cuando se ha implementado en Milo, el sistema de transmisión ha funcionado adecuadamente, el sistema embebido uno, ha captado correctamente los datos de los ángulos del movimiento del pie sano, así mismo, el controlador difuso ha funcionado adecuadamente, generando ese movimiento de espejo deseado.

Hasta este punto, todo funciona adecuadamente, sin embargo, si ha existido un problema, y es el tiempo de respuesta del sistema, es decir, a pesar de que el procesamiento se está realizando de forma adecuada, los tiempos de respuesta no han sido óptimos, lo que genera un mal funcionamiento en la finalidad de la prótesis, tal que, buscamos que el can se pueda mover de forma natural y sin lastimar ninguna parte de su cuerpo.

Para fines de este artículo, se ha logrado la meta de lograr el funcionamiento adecuado del controlador difuso dentro de nuestra prótesis, sin embargo, actualmente se sigue trabajando en la manera de mejorar los tiempos de respuesta de interconexión y procesamiento de nuestros sistemas embebidos.

Esto, se deja en este momento como trabajo a futuro, así como la conexión con el sistema IoT. Adicionalmente, se está trabajando en el desarrollo mecánico de la prótesis, sin embargo, esta parte queda fuera del objetivo de este artículo.

## **8. Conclusiones**

En este artículo se ha descrito el desarrollo de un sistema de control difuso que es implementado en un prototipo de prótesis para un perro con la pierna derecha amputada.

Se diseñó un controlador que busca ajustarse a la hipótesis de Hildebrand, generando un comportamiento de espejo entre la pierna sana y la respuesta en la prótesis de la pierna amputada. Por medio de un acelerómetro, se han captado los movimientos de la extremidad sana, y estos han sido enviados a un controlador difuso, para que este genere el comportamiento de espejo antes mencionado.

El controlador ha dado los resultados esperados en su implementación, el movimiento de las extremidades del can se realiza de manera satisfactoria, sin embargo, se ha detectado un problema en la velocidad de respuesta, esto dado principalmente por la interconexión, comunicación y procesamiento de datos, lo cual tiene que ver con un retardo en el envío y recepción de datos. La resolución de este problema dentro del funcionamiento de los sistemas embebidos diseñados se deja como un trabajo a futuro.

**Agradecimientos.** Este artículo fue apoyado por los proyectos PAPIME PE103322, PAPIME PE105822 y Cátedra de investigación CI2254 de la UNAM y la FES-C.

## **Referencias**

1. Ponce-Cruz, P.: *Inteligencia artificial con aplicaciones a la ingeniería*. Ciudad de México: Alfaomega (2011)
2. Zadeh, L. A.: Fuzzy sets. *Information and Control*, vol. 8, no. 3, pp. 338–353 (1965) doi: 10.1016/S0019-9958(65)90241-X
3. Sharma, K., Agrawal, A., Bandopadhyaya, S.: Fuzzy Logic Controlled Variable Frequency Drives. In: Yadav, N., Yadav, A., Bansal, J., Deep, K., Kim, J. (eds) *Harmony Search and Nature Inspired Optimization Algorithms. Advances in Intelligent Systems and Computing*, Springer, Singapore, vol. 741, pp. 1153–1164 (2019). doi: 10.1007/978-981-13-0761-4\_107
4. Jamin, N. F., Ghani, N. M., Ibrahim, Z., Masrom, M. F., Razali, N., Almeshal, A. M.: Two-wheeled wheelchair stabilization using interval type-2 fuzzy logic controller. *International Journal of Simulation: Systems, Science & Technology*, vol. 19, no. 3, pp. 31–37 (2018) doi: 10.5013/IJSSST.a.19.03.03
5. Castillo, O., Aguilar, L. T.: *Fuzzy Control for Wheeled Mobile Robots. Type-2 Fuzzy Logic in Control of Nonsmooth Systems. Studies in Fuzziness and Soft Computing*, Springer, Cham. vol 373, pp.85–96 (2019) doi: 10.1007/978-3-030-03134-3\_6

6. Pau, V., Salerno, V. M.: Wireless Sensor Networks for Smart Homes: A Fuzzy-Based Solution for an Energy-Effective Duty Cycle, *Electronics.*, vol. 8, no. 2, pp. 131 (2019) doi: 10.3390/electronics8020131
7. Weir, R. F., Ajiboye, A. B.: A multifunction prosthesis controller based on fuzzy-logic techniques. In: *Proceedings of the 25th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (IEEE Cat. No. 03CH37439)*, vol. 2, pp. 1678–1681 (2003) doi: 10.1109/IEMBS.2003.1279711
8. Nordin, N. H. D., Muthalif, A. G. A., Razali, M. K. M., Ali, A., Salem, A. M. Development and implementation of energy-efficient Magnetorheological Fluid bypass damper for prosthetics limbs using a fuzzy-logic controller. In: *Proceedings of IEEE Access*, vol. 10, pp. 18978–18987 (2022) doi: 10.1109/ACCESS.2022.3149893
9. Ghazali, R., Saad, M. Z., Hussien, S. Y. S., Jali, M. H., Zohedi, F. N., Izzuddin, T. A. Intelligent controller design for multifunctional prosthetics hand. *International Journal of Mechanical Engineering and Robotics Research*, vol. 6, no. 6, pp. 495–501 (2017) doi: 10.18178/ijmerr.6.6.495-501
10. Hildebrand, M.: Analysis of the symmetrical gaits of tetrapods. In: Herman, R.M., Grillner, S., Stein, P.S.G., Stuart, D.G. (eds) *Neural Control of Locomotion. Advances in Behavioral Biology*, vol 18. Springer, Boston, MA. [https://doi.org/10.1007/978-1-4757-0964-3\\_9](https://doi.org/10.1007/978-1-4757-0964-3_9)
11. Bolton, W.: *Mecatrónica. Sistemas de control electrónico en ingeniería mecánica y eléctrica*. México D.F.: Alfaomega (2013)
12. Corona-Ramírez, L., Abarca-Jiménez, G.: *Sensores y Actuadores. Aplicaciones con Arduino*. Ciudad de México: Patria Education (2019)
13. Harris III, A. F., Khanna, V., Tuncay, G., Want, R., Kravets, R.: Bluetooth low energy in dense IoT environments. In: *Proceedings of IEEE Communications Magazine*, vol. 54, no. 12, pp. 30–36 (2016) doi: 10.1109/MCOM.2016.1600546CM
14. Collotta, M., Pau, G., Talty, T., Tonguz, O. K.: Bluetooth 5: A concrete step forward toward the IoT. In: *Proceedings of IEEE Communications Magazine*, vol. 56, no. 7, pp. 125–131 (2018) doi: 10.1109/MCOM.2018.1700053
15. Terán, M., Aranda, J., Carrillo, H., Mendez, D., & Parra, C.: IoT-based system for indoor location using bluetooth low energy. In: *Proceedings of IEEE Colombian Conference on Communications and Computing COLCOM*, pp. 1–6 (2017) doi: 10.1109/ColComCon.2017.8088211



# Modelado del problema de composición musical polifónica automática como un problema de optimización biobjetivo

Natan Vilchis, Adriana Lara

Instituto Politécnico Nacional,  
Escuela Superior de Física y Matemáticas,  
México

`nvilchist1300@alumno.ipn.mx`, `alaral@ipn.mx`

**Resumen.** En este trabajo se propone un modelo matemático que plantea la composición musical asistida por computadora como un problema de optimización. Se consideran dos funciones objetivo en conflicto para representar las características deseadas de la melodía a crear. Se muestran ejemplos de la aplicación de este modelo utilizando un algoritmo evolutivo multiobjetivo para aproximar las soluciones del problema biobjetivo. Los resultados de este trabajo pueden escucharse en la página web que proporcionamos para tal fin.

**Palabras clave:** Composición musical automática, algoritmo genético, optimización multiobjetivo.

## Modeling the Automatic Polyphonic Music Composition Problem as a Bi-Objective Optimization Problem

**Abstract.** This work proposes a mathematical model that states computer-assisted musical composition as an optimization problem. Two conflicting objective functions represent the desired characteristics of the melody to be created. Examples of the application of this model using a multi-objective evolutionary algorithm to approximate the solutions of the biobjective problem can be heard on the web page that we provide for this purpose.

**Keywords:** Automatic music creation, genetic algorithm, multi-objective optimization.

### 1. Introducción

La inteligencia artificial aplicada a la música ha explorado diversas herramientas como el deep learning, música generativa, computación cuántica, entre otros [17]. En el presente trabajo se modela el problema de la composición musical asistida por computadora con el lenguaje de la optimización matemática, teniendo como objetivo generar canciones con una alegría máxima, mientras se busca un minimalismo en

los adornos; para ello se utilizó la teoría musical emocional de Mauro de María [11], teoría musical clásica [6], estudio de patrones musicales [12, 5, 15, 14, 18, 7, 2, 13] y trabajos relacionados estado del arte como MetaCompose [20] y MorpheuS [10] que proporcionaron elementos valiosos para nuestro modelo.

El presente trabajo es una combinación de elementos existentes, tales como la definición 15 (basada en la ecuación 4 de [20]), las restricciones 17, 18, 19 y 20 del problema de optimización biobjetivo (basadas en [10] junto con combinación y simplificación de [12, 5, 15, 14, 18, 7, 2, 13]) y con aportes originales basados en teoría musical emocional [11] presentes en los demás elementos del problema de optimización.

## 2. Conceptos básicos y elementos fundamentales del modelo

Hemos representado una canción como un vector  $x \in \mathbb{Z}^{b+2bp}$ , donde las primeras  $b$  componentes denotan el índice del acorde para el correspondiente compás y las siguientes  $bp$  componentes representan las notas de la armonía; finalmente, las últimos  $bp$  componentes, representan las notas de la melodía.

De esta manera, cada melodía tiene  $b \in \mathbb{N}$  compases y a su vez, cada compás está dividido en  $p$  particiones, con  $p \in \{2, 4, 8, 16\}$ ; además, cada componente  $41 \leq x_j \leq 88$ , para  $j \in \{b+1, \dots, 2bp\}$  es una nota MIDI [1]; donde la nota MIDI 41 será tomada como una alargación de la nota anterior, mientras que la nota MIDI 42 se utilizará como silencio. El siguiente par de pentagramas muestra la representación de una canción correspondiente al vector  $x^0 \in \mathbb{Z}^{4+2(4)(8)}$  dado por:

$$x^0 = [1, 11, 11, 1, \\ 48, 48, 52, 52, 62, 64, 55, 42, 65, 65, 69, 69, 76, 77, 65, 42, \\ 53, 53, 57, 57, 67, 69, 60, 42, 60, 60, 64, 64, 76, 88, 84, 42, \\ 42, 41, 41, 71, 41, 83, 79, 88, 42, 41, 41, 81, 41, 84, 83, 84, \\ 72, 77, 41, 71, 77, 41, 41, 41, 83, 84, 41, 83, 88, 41, 41, 41]^T.$$



La definición 2 usa la definición de una nota MIDI [1], las definiciones 24, 25, 26, 27, 28 y 29 fueron construidas a partir de la teoría musical clásica de adornos musicales [19], mientras que las definiciones restantes fueron creadas a partir del modelo musical emocional [11] y adaptadas para el problema abordado del presente trabajo. A continuación se presentan definiciones necesarias para el modelo:

**Definición 1 (Nota musical)** Una nota musical  $w$  será un elemento del conjunto  $W$  definido por:  $W := \{C, C\#, D, D\#, E, F, F\#, G, G\#, A, A\#, B\}$ . La correspondiente

nota musical numerada de  $w$  está dada por la función:

$$\text{noteToNumbered} : W \rightarrow \mathbb{Z};$$

Descrita en la siguiente tabla:

Nota musical	$C$	$C\sharp$	$D$	$D\sharp$	$E$	$F$	$F\sharp$	$G$	$G\sharp$	$A$	$A\sharp$	$B$
Notación latina	Do	Do $\sharp$	Re	Re $\sharp$	Mi	Fa	Fa $\sharp$	Sol	Sol $\sharp$	La	La $\sharp$	Si
Nota musical numerada	0	1	2	3	4	5	6	7	8	9	10	11

**Definición 2 (Melodía)** El vector  $x = (x_1, x_2, \dots, x_m)^T \in \mathbb{Z}^m$  se dice ser una melodía, si cada  $x_i$  es una nota MIDI, para  $i = 1, \dots, m$ .

**Definición 3 (Melodía del compás  $j$ -ésimo de una melodía)** Sea  $x \in \mathbb{Z}^{bp}$  una melodía, con  $b, p \in \mathbb{N}$ . La melodía del compás  $j$ -ésimo  $x_{B(j)}$  corresponde al subvector de  $x$  dado como:  $x_{B(j)} := x_{(1+p(j-1):jp)}$ ,  $j = 1, \dots, b$ .

**Definición 4 (Acorde)** El vector  $x = (x_1, x_2, \dots, x_q)^T \in \mathbb{Z}^q$  se dice ser un acorde, si cada  $x_i$  es una nota musical numerada, para  $i = 1, \dots, q$  y además se verifica que  $x_i \neq x_j \quad \forall i \neq j$ .

**Definición 5 (Melodía con buen inicio)** Sea  $x \in \mathbb{Z}^k$  una melodía. La melodía  $x$  se dice tener un buen inicio si la función  $NS : \mathbb{Z}^k \rightarrow \{0, 1\}$  evaluada en la melodía  $x$  es igual a uno, donde la función  $NS$  se define como sigue:

$$NS(x) := \begin{cases} 1, & \nexists x_j = 41, j = 1, \dots, k, \\ 1, & \begin{matrix} \text{mín}(j) < \text{mín}(\ell), \\ x_j > 41 & x_\ell = 41 \\ 1 \leq j \leq k & 1 \leq \ell \leq k \end{matrix} \\ 0, & \text{en caso contrario.} \end{cases}$$

**Definición 6 (Melodía de notas reales)** Sea  $x \in \mathbb{Z}^k$  una melodía tal que  $\exists x_j > 42$ . La melodía de notas reales de la melodía  $x$  es el resultado de  $MR(x)$ , donde la función  $MR : \mathbb{Z}^k \rightarrow \mathbb{Z}^q, q \leq k$ , se define como sigue:

$$MR(x) := [x_1, x_2, \dots, x_j, \dots, x_{k-1}, x_k]^T, \quad x_j > 42, j = 1, \dots, k.$$

**Definición 7 (Total de notas reales en melodía)** Sea  $x \in \mathbb{Z}^k$  una melodía, el número total de notas reales de una melodía es el resultado de  $TN(x)$ , donde la función  $TN : \mathbb{Z}^k \rightarrow \mathbb{Z}$  está definida como sigue:

$$TN(x) := \text{número de elementos } x_j \text{ mayores a } 42, \quad j = 1, \dots, k.$$

**Definición 8 (Vector diferencia)** Sea  $x \in \mathbb{Z}^n, n \geq 2$ , el vector diferencia correspondiente a  $x$  está determinado por la función  $D : \mathbb{Z}^n \rightarrow \mathbb{Z}^{n-1}$ , donde:

$$D(x) := x_{(2:n)} - x_{(1:n-1)}.$$

**Algoritmo 1** Función repeatMelody para obtener la melodía repetición de una melodía.

---

```

1: Función repeatMelody(x)
2:   Sean  $m, j \in \mathbb{N}, \text{saveNote} \in \mathbb{Z}, y \in \mathbb{Z}^k$ .
3:   Hacer  $m \leftarrow \min_{\substack{x_j > 42 \\ 1 \leq j \leq k}}(j)$ .
4:   Hacer  $\text{saveNote} \leftarrow x_m$ .
5:   Hacer  $j \leftarrow m + 1$ .
6:   Hacer  $y \leftarrow x$ .
7:   Mientras  $j \leq k$  Hacer
8:     Si  $x_j == 41 \vee x_j == 42$  Entonces
9:       Hacer  $y_j \leftarrow \text{saveNote}$ .
10:    En caso contrario
11:      Hacer  $\text{saveNote} \leftarrow x_j$ .
12:    Terminar Si
13:    Hacer  $j \leftarrow j + 1$ .
14:  Terminar Mientras
15:  Devolver  $y$ .
16: Terminar Función

```

---

**Definición 9 (Máximo salto de una melodía)** Sea  $x \in \mathbb{Z}^k$  una melodía. El máximo salto de la melodía  $x$  es el resultado de  $MS(x)$ , donde la función  $MS : \mathbb{Z}^k \rightarrow \mathbb{Z}$  se define como sigue:

$$MS(x) := \begin{cases} 0, & \text{si } TN(x) < 2, \\ \|D(MR(x))\|_\infty, & \text{en caso contrario.} \end{cases}$$

**Definición 10 (Vector signo)** Sea  $x \in \mathbb{Z}^r$ , el vector signo correspondiente a  $x$  es el resultado de  $S(x)$ , donde la función  $S : \mathbb{Z}^r \rightarrow \mathbb{Z}^r$  se define como sigue:

$$S(x) := [\text{sgn}(x_1), \text{sgn}(x_2), \dots, \text{sgn}(x_r)]^T.$$

**Definición 11 (Melodía repetición de una melodía)** Sea  $x \in \mathbb{Z}^k$  una melodía tal que  $\exists x_j > 42$ . La melodía repetición  $y \in \mathbb{Z}^k$  de la melodía  $x$  es el resultado de  $\text{repeatMelody}(x)$ , donde la función  $\text{repeatMelody} : \mathbb{Z}^k \rightarrow \mathbb{Z}^k$  se describe en el algoritmo 1.

**Definición 12 (Conjunto de notas disonantes de una melodía)** Sea  $x \in \mathbb{Z}^k$  una melodía, tal que  $\exists x_j > 42$ . El conjunto de notas disonantes para la melodía  $x$  es el resultado de  $ND(x)$ , donde la función  $ND$  se define como:

$$ND(x) := \{\text{MIDIToNumbered}(x_i - 1) \forall x_i > 42\} \cup \{\text{MIDIToNumbered}(x_i + 1) \forall x_i > 42\}.$$

**Definición 13 (Nota adecuada para la melodía)** Sea  $x$  una nota MIDI, sea  $w \in \mathbb{Z}^k$  una melodía tal que  $\exists w_j > 42$ . Se dice que  $x$  es una nota adecuada para la melodía  $w$  si el resultado de  $MF(x, w)$  es igual a uno, donde la función  $MF$  se define como sigue:

$$MF(x, w) := \begin{cases} 1, & \text{si } x \notin ND(w), \\ 0, & \text{en caso contrario.} \end{cases}$$

**Definición 14 (Función SDR)** Sea  $x \in \mathbb{Z}^t$  una melodía con  $t \geq 2$ , tal que  $\exists x_j > 42$ . La función  $SDR : \mathbb{Z}^t \rightarrow \mathbb{Z}^{t-1}$  se define a continuación:

$$SDR(x) := S(D(\text{repeatMelody}(x))).$$

**Definición 15 (Primera nota de la melodía es nota fundamental del acorde)** Sea  $x \in \mathbb{Z}^k$  una melodía tal que  $\exists x_j > 42$ , sea  $w \in \mathbb{Z}^q$  un acorde. Se dice que la melodía  $x$  tiene como primer nota a la nota fundamental del acorde  $w$  si el resultado de  $\text{FirstNoteFC}(x, w)$  es igual a uno, donde la función  $\text{FirstNoteFC}$  se describe a continuación:

$$\text{FirstNoteFC}(x, w) := \begin{cases} 1, & \text{si } \text{MIDIToNumbered}(x_r) = w_1, \quad r = \underset{j=1, \dots, k}{\text{mín}}(x_j > 42), \\ 0, & \text{en otro caso.} \end{cases}$$

**Definición 16 (Nota MIDI en acorde)** Sea  $x$  una nota MIDI, sea  $w \in \mathbb{Z}^q$  un acorde. La nota MIDI  $x$  será una nota del acorde  $w$  si el resultado de  $\text{NInChord}(x, w)$  es igual a uno, donde la función  $\text{NInChord}$  se define a continuación:

$$\text{NInChord}(x, w) := \begin{cases} 1, & \text{si } \text{MIDIToNumbered}(x) = w_j, \text{ para algún } j = 1, \dots, p, \\ 0, & \text{en otro caso.} \end{cases}$$

**Definición 17 (Al menos el 50 % de las notas MIDI son notas del acorde)** Sea  $x \in \mathbb{Z}^k$  una melodía, sea  $w \in \mathbb{Z}^q$  un acorde. Se dice que al menos el cincuenta por ciento de las notas de la melodía  $x$  son notas del acorde  $w$  si el resultado de  $\text{FPN}(x, w)$  es igual a uno, donde la función  $\text{FPN}$  se define a continuación:

$$\text{FPN}(x, w) := \begin{cases} 1, & \text{si } \left( \sum_{x_j > 42} \text{NinChord}(x_j, w) \right) / \text{TN}(x) \geq 0.5, \\ 0, & \text{en caso contrario.} \end{cases}$$

**Definición 18 (Mayoría de las notas MIDI de la melodía son notas del acorde)** Sea  $x \in \mathbb{Z}^k$  una melodía, sea  $w \in \mathbb{Z}^q$  un acorde. Se dice que las notas MIDI de la melodía  $x$  son mayormente notas del acorde si el resultado de  $\text{MPN}(x, w)$  es igual a uno, donde:

$$\text{MPN}(x, w) := \begin{cases} 1, & \text{si } \sum_{x_i > 42} \text{NinChord}(x_i, w) \geq \text{TN}(x) - 1, \\ 0, & \text{en caso contrario.} \end{cases}$$

**Definición 19 (Pulsaciones de una melodía)** Sea  $x \in \mathbb{Z}^k$ , las pulsaciones de la melodía  $x$  será el resultado de  $\text{BT}(x)$ , donde la función  $\text{BT} : \mathbb{Z}^k \rightarrow \mathbb{Z}^k$  se define a continuación:

$$\text{BT}(x) := y, \quad y \in \mathbb{Z}^k,$$

donde  $y_j$ , para  $j = 1, \dots, k$ , está sujeto a:

$$y_j = \begin{cases} 1, & \text{si } x_j > 42, \\ 0, & \text{si } x_j = 42, \\ -1, & \text{si } x_j = 41. \end{cases}$$

**Definición 20 (Nota MIDI en la escala de Do Mayor)** Sea  $x \in \mathbb{Z}$  una nota MIDI, se dice que la nota  $x$  está en la escala de Do Mayor (denotada como  $\mathcal{S} = \{0, 2, 4, 5, 7, 9, 11\}$ ) si el resultado de  $NEC(x)$  es igual a uno, donde la función  $NEC$  se define como sigue:

$$NEC(x) := \begin{cases} 1, & \text{si } MIDIToNumbered(x) \in \mathcal{S}, \\ 0, & \text{en caso contrario.} \end{cases}$$

**Definición 21 (Total de notas de la melodía en la escala de Do Mayor)** Sea  $x \in \mathbb{Z}^k$  una melodía, el número total de notas de la melodía  $x$  que se encuentran en la escala de Do Mayor es el resultado de  $TNS(x)$ , donde la función  $TNS$  se define como sigue:

$$TNS(x) := \sum_{x_j > 42} NEC(x_j).$$

**Definición 22 (Notas pulsadas de forma simultánea)** Sean  $x, w \in \mathbb{N}$  dos notas MIDI, se dice que las notas  $x, w$  son pulsadas de forma simultánea si el resultado de  $PSN(x, w)$  es uno, donde la función  $PSN$  está definida de la siguiente manera:

$$PSN(x, w) := \begin{cases} 1, & \text{si } (x > 42) \wedge (w > 42), \\ 0, & \text{en caso contrario.} \end{cases}$$

**Definición 23 (Total de notas pulsadas simultáneamente de dos melodías)** Sean  $x, w \in \mathbb{Z}^{bp}$  dos melodías, el número total de notas pulsadas simultáneamente de las melodías  $x, w$  es el resultado de  $TNC(x, w)$ , donde la función  $TNC$  está definida de la siguiente manera:

$$TNC(x, w) := \sum_{j=0}^{b-1} \sum_{k=1}^p PSN(x_{(jp+k)}, w_{(jp+k)}).$$

**Definición 24 (Adorno bordadura)** Sea  $x \in \mathbb{Z}^3$  una melodía, sean  $u, w \in \mathbb{Z}^q$  acordes. Se dice que  $x$  es un adorno de bordadura para los acordes  $u, w$  si el resultado de  $esBordadura(x, u, w)$  es igual a uno, donde la función  $esBordadura$  se define como sigue:

$$esBordadura(x, u, w) := \begin{cases} 1, & \text{si } \begin{aligned} &(x_1 > 42) \wedge (x_2 > 42) \wedge \\ &(x_3 > 42) \wedge (1 \leq |x_1 - x_2| \leq 2) \\ &\wedge (x_1 = x_3) \wedge (MIDIToNumbered(x_1) = u_j, \text{ para} \\ &\text{algún } j=1, \dots, p), \wedge (MIDIToNumbered(x_3) = w_\ell, \text{ para} \\ &\text{algún } \ell=1, \dots, p), \end{aligned} \\ 0, & \text{en caso contrario.} \end{cases}$$

**Definición 25 (Adorno echapé)** Sea  $x \in \mathbb{Z}^3$  una melodía, sean  $u, w \in \mathbb{Z}^q$  dos acordes. Se dice que  $x$  es un adorno echapé para los acordes  $u, w$  si el resultado de  $esEchapee(x, u, w)$  es igual a uno, donde la función  $esEchapee$  se define como sigue:

$$\text{esEchapee}(x, u, w) := \begin{cases} 1, & \text{si } \begin{aligned} &(x_1 > 42) \wedge (x_2 > 42) \wedge (x_3 > 42) \\ &\wedge (1 \leq |x_1 - x_2| \leq 2) \wedge (|x_2 - x_3| > 2) \\ &\wedge ((x_2 - x_1)(x_3 - x_2) < 0) \\ &\wedge (\text{MIDIToNumbered}(x_1) = u_j, \text{ para} \\ &\quad \text{algún } j=1, \dots, p) \\ &\wedge (\text{MIDIToNumbered}(x_3) = w_\ell, \text{ para} \\ &\quad \text{algún } \ell=1, \dots, p), \end{aligned} \\ 0, & \text{en caso contrario.} \end{cases}$$

**Definición 26 (Adorno cambiata)** Sea  $x \in \mathbb{Z}^3$  una melodía, sean  $u, w \in \mathbb{Z}^q$  dos acordes. Se dice que  $x$  es un adorno cambiata para los acordes  $u, w$  si el resultado de  $\text{esCambiata}(x, u, w)$  es igual a uno, donde la función  $\text{esCambiata}$  se define como sigue:

$$\text{esCambiata}(x, u, w) = \begin{cases} 1, & \text{si } \begin{aligned} &(x_1 > 42) \wedge (x_2 > 42) \wedge (x_3 > 42) \\ &\wedge (1 \leq |x_2 - x_3| \leq 2) \wedge (|x_1 - x_2| > 2) \\ &\wedge ((x_2 - x_1)(x_3 - x_2) < 0) \\ &\wedge (\text{MIDIToNumbered}(x_1) = u_j, \text{ para} \\ &\quad \text{algún } j=1, \dots, p) \\ &\wedge (\text{MIDIToNumbered}(x_3) = w_\ell, \text{ para} \\ &\quad \text{algún } \ell=1, \dots, p), \end{aligned} \\ 0, & \text{en caso contrario.} \end{cases}$$

**Definición 27 (Adorno de paso)** Sea  $x \in \mathbb{Z}^3$  una melodía, sean  $u, w \in \mathbb{Z}^q$  dos acordes. Se dice que  $x$  es un adorno de paso para los acordes  $u, w$  si el resultado de  $\text{esDePaso}(x, u, w)$  es igual a uno, donde la función  $\text{esDePaso}$  se define como sigue:

$$\text{esDePaso}(x, u, w) = \begin{cases} 1, & \text{si } \begin{aligned} &(x_1 > 42) \wedge (x_2 > 42) \wedge (x_3 > 42) \\ &\wedge (1 \leq |x_2 - x_3| \leq 2) \wedge (1 \leq |x_1 - x_2| \leq 2) \\ &\wedge ((x_2 - x_1)(x_3 - x_2) > 0) \\ &\wedge (\text{MIDIToNumbered}(x_1) = u_j, \text{ para} \\ &\quad \text{algún } j=1, \dots, p) \\ &\wedge (\text{MIDIToNumbered}(x_3) = w_\ell, \text{ para} \\ &\quad \text{algún } \ell=1, \dots, p), \end{aligned} \\ 0, & \text{en caso contrario.} \end{cases}$$

**Definición 28 (Adorno apoyatura)** Sea  $x \in \mathbb{Z}^2$  una melodía, sea  $w \in \mathbb{Z}^q$  un acorde. Se dice que la melodía  $x$  es un adorno apoyatura para el acorde  $w$  si el resultado de  $\text{esApoyatura}(x, w)$  es igual a uno, donde la función  $\text{esApoyatura}$  se define como sigue:

$$\text{esApoyatura}(x, w) := \begin{cases} 1, & \text{si } \begin{aligned} &(x_1 > 42) \wedge (x_2 > 42) \wedge (1 \leq |x_1 - x_2| \leq 2) \\ &\wedge (\text{MIDIToNumbered}(x_2) = w_j, \text{ para} \\ &\quad \text{algún } j=1, \dots, p), \end{aligned} \\ 0, & \text{en caso contrario.} \end{cases}$$

**Algoritmo 2** Función countThree para obtener el total de adornos de tres notas dada una función  $J$  y una lista de acordes  $L$ .

---

```

1: Función countThree(w,x,J,L)
2:   Sean possible, j, found, k, ℓ, m1, m2 ∈ ℤ, s ∈ ℝ, y ∈ ℤ3, C1, C2 ∈ ℤq.
3:   Hacer found ← 0.
4:   Hacer possible ←  $\frac{pb-2}{2}$ .
5:   Hacer s ← 0.
6:   Si possible > 0 Entonces
7:     Hacer j ← 0.
8:     Mientras j < possible Hacer
9:       Hacer k ←  $\lfloor \frac{2j}{p} \rfloor + 1$ .
10:      Hacer ℓ ←  $\lfloor \frac{2j+2}{p} \rfloor + 1$ .
11:      Hacer m1 ← wk.
12:      Hacer m2 ← wℓ.
13:      Hacer C1 ← Lm1.
14:      Hacer C2 ← Lm2.
15:      Hacer y ← x(2j+1:2j+3).
16:      Si J(y, C1, C2) == 1 Entonces
17:        Hacer found ← found + 1.
18:      Terminar Si
19:      Hacer j ← j + 1.
20:    Terminar Mientras
21:    Hacer s ← found / possible.
22:  Terminar Si
23:  Devolver s.
24: Terminar Función

```

---

**Definición 29 (Total de adornos de anticipación)** Sea  $x, y \in \mathbb{Z}^{bp}$  dos melodías. El total de adornos de anticipación para la melodía  $x$  y la armonía  $y$  es el resultado de la función contarAnticipacion, donde la función contarAnticipacion se define como sigue:

$$\text{contarAnticipacion}(x, y) := \sum_{j=1}^{b-1} 1 \quad \text{tales que} \quad \begin{aligned} & (PSN(x_{\mathcal{B}(j)_p}, y_{\mathcal{B}(j+1)_1}) = 1) \\ & \wedge (x_{\mathcal{B}(j)_p} = y_{\mathcal{B}(j+1)_1}). \end{aligned}$$

**Definición 30 (Total de adornos de tres notas de una melodía)** Sea  $x \in \mathbb{Z}^{bp}$  una melodía, sea  $w \in \mathbb{Z}^b$  un vector con los índices de acordes para cada compás de la melodía  $x$ , sea  $J$  una función de adorno de tres notas, sea  $L$  una lista de acordes. El total de adornos de tres notas presentes en la melodía  $x$  es el resultado de countThree( $w, x, J, L$ ), donde la función countThree queda descrita en el algoritmo 2.

**Definición 31 (Adornos para la melodía)** Sean  $x, y \in \mathbb{Z}^{bp}$  dos melodías, sea  $w \in \mathbb{Z}^b$  un vector que contiene los índices de acordes para cada compás, sea  $L$  una lista de acordes. El total de adornos para la melodía  $x$  es el resultado de OrnamentsM( $w, x, L$ ), donde la función OrnamentsM se define de la siguiente manera:

$$\begin{aligned} \text{OrnamentsM}(w, x, y, L) := & \frac{1}{6} \left( \text{countThree}(w, x, \text{esBordadura}, L) \right. \\ & + \text{countThree}(w, x, \text{esEchapee}, L) \\ & + \text{countThree}(w, x, \text{esCambiata}, L) \\ & + \text{countThree}(w, x, \text{esDePaso}, L) \\ & + \text{contarApoyatura}(w, x, L) \\ & \left. + \text{contarAnticipacion}(x, y) \right). \end{aligned}$$

**Definición 32 (Melodía alargación)** Sea  $x \in \mathbb{Z}^k$  una melodía, se dice que  $x$  es una melodía alargación si  $\text{alargacion}(x)$  es igual a uno, donde la función  $\text{alargacion}$  se define como sigue:

$$\text{alargacion}(x) := \begin{cases} 1, & \text{si } (x_1 > 42) \wedge (x_j = 41), j = 2, \dots, k, \\ 0, & \text{en otro caso.} \end{cases}$$

**Definición 33 (Total de notas blancas en la melodía)** Sea  $x \in \mathbb{Z}^{bp}$  una melodía, la cantidad total de notas blancas en la melodía es el resultado de  $\text{Half}(x)$ , donde la función  $\text{Half}$  se define como sigue:

$$\text{Half}(x) := \sum_{j=1}^{bp - \frac{p}{2} + 1} \text{alargacion}(x_{(j:j + \frac{p}{2} - 1)}).$$

**Definición 34 (Alegría por compás de la canción)** Sea:

$$\begin{aligned} L = & \left( [0, 4, 7]^T, [2, 5, 9]^T, [4, 7, 11]^T, [5, 9, 0]^T, [7, 11, 2]^T, [9, 0, 4]^T, [11, 2, 5]^T, \right. \\ & [0, 4, 7, 10]^T, [2, 5, 9, 0]^T, [4, 7, 11, 2]^T, [5, 9, 0, 3]^T, [7, 11, 2, 5]^T, \\ & \left. [9, 0, 4, 7]^T, [11, 2, 5, 8]^T \right). \end{aligned}$$

Una lista con acordes de mayores y séptimos, sea:

$$M = [1, -2, -3, 2, 3, -1, -0.5, 1, -2, -3, 2, 3, -1, -0.5]^T.$$

Un vector con el nivel de alegría de cada acorde correspondiente a  $L$ . Sea  $w \in \mathbb{Z}^b$  un vector que contiene los índices del acorde correspondiente para el compás  $j$ , para  $j = 1, \dots, b$ . La alegría por compás de la canción es el resultado de  $\text{Happy}(w)$ , donde la función  $\text{Happy}$  se define como sigue:

$$\text{Happy}(w) := \frac{1}{b} \sum_{j=1}^b M_{w_j}.$$

**Algoritmo 3** Función contarApoyatura para obtener el total de adornos de apoyatura dada una lista de acordes  $L$ .

---

```

1: Función contarApoyatura(w,x,L)
2:   Sean possible, j, found,  $C_1, k, m \in \mathbb{Z}, s \in \mathbb{R}$ .
3:   Hacer found  $\leftarrow 0$ .
4:   Hacer possible  $\leftarrow \frac{bp}{2}$ .
5:   Hacer  $s \leftarrow 0$ .
6:   Hacer  $j \leftarrow 0$ .
7:   Mientras  $j < \text{possible}$  Hacer
8:     Hacer  $k \leftarrow \lfloor \frac{2j}{p} \rfloor + 1$ .
9:     Hacer  $m \leftarrow w_k$ .
10:    Hacer  $C_1 \leftarrow L_m$ .
11:    Hacer  $y \leftarrow x_{(2j+1:2j+2)}$ .
12:    Si esApoyatura(y,  $C_1$ )  $== 1$  Entonces
13:      Hacer found  $\leftarrow \text{found} + 1$ .
14:    Terminar Si
15:    Hacer  $j \leftarrow j + 1$ .
16:  Terminar Mientras
17:  Hacer  $s \leftarrow \frac{\text{found}}{\text{possible}}$ .
18:  Devolver  $s$ .
19: Terminar Función

```

---

**Definición 35 (Adornos para la armonía)** Sean  $y \in \mathbb{Z}^{bp}$  una melodía, sea  $w \in \mathbb{Z}^b$  un vector que contiene los índices de acordes para cada compás, sea  $L$  una lista de acordes. El total de adornos para la armonía  $y$  es el resultado de  $\text{OrnamentsA}(w,x,L)$ , donde la función  $\text{OrnamentsA}$  se define de la siguiente manera:

$$\begin{aligned} \text{OrnamentsA}(w, y, L) := & \frac{1}{5} \left( \text{countThree}(w, y, \text{esBordadura}, L) \right. \\ & + \text{countThree}(w, y, \text{esEchapee}, L) \\ & + \text{countThree}(w, y, \text{esCambiata}, L) \\ & + \text{countThree}(w, y, \text{esDePaso}, L) \\ & \left. + \text{contarApoyatura}(w, y, L) \right). \end{aligned}$$

### 3. El problema de optimización biobjetivo

La generación de las melodías deseadas se ha modelado mediante el siguiente problema:

$$\text{Minimizar } F(x) = [g_1, g_2]^T, \text{ con } x \in \mathbb{Z}^{b+2bp}.$$

Por comodidad el vector  $x$  se presenta como la concatenación de tres vectores  $w \in \mathbb{Z}^b$  y  $y, z \in \mathbb{Z}^{bp}$  como  $x = (w, y, z)$ . Las funciones  $g_1$  y  $g_2$  se minimizan de manera simultánea y están definidas como:

$$g_1(w, y, z) = 3 - \frac{TNS(y) + TNS(z)}{TN(y) + TN(z)} - \text{OrnamentsM}(z, y) - \text{OrnamentsA}(y) - \text{Happy}(w),$$

$$g_2(y, z) = 2 + \text{OrnamentsM}(z, y) + \text{OrnamentsA}(y) - \text{Half}(y) - \text{Half}(z) + \frac{TN(y) + TN(z)}{2bp}.$$

De esta manera, el vector  $w$  tendrá como componentes a los índices de los acordes de cada compás, el vector  $y$  tendrá las notas de la armonía y el vector  $z$  tendrá las notas de la melodía de la canción. El problema de optimización se presenta sujeto a las siguientes restricciones:

1.  $b \in \mathbb{N}, p \in \{2, 4, 8, 16\}$ ,
2.  $\exists y_j > 42$  para  $j = 1, \dots, bp$ ,  $\exists z_\ell > 42$ , para  $\ell = 1, \dots, bp$ ,
3.  $L = (L_1, \dots, L_r)$ , donde  $L_j$  es un acorde, para  $j = 1, \dots, r$ ,
4.  $w_1 = 1, w_b = 1$ ,
5.  $1 \leq w_j \leq r$ , para  $j = 2, \dots, b-1$ ,
6.  $41 \leq y_j \leq 88$ , para  $j = 1, \dots, bp$ ,
7.  $41 \leq z_j \leq 88$ , para  $j = 1, \dots, bp$ ,
8.  $y_j < z_j, \forall y_j, z_j$  tales que  $PSN(y_j, z_j) = 1$ ,
9.  $z_j - y_j \leq 24, \forall y_j, z_j$  tales que  $PSN(y_j, z_j) = 1$ ,
10.  $MS(y) \leq 12$ ,
11.  $MS(z) \leq 12$ ,
12.  $GB(y, z) = NS(y) + NS(z) - 2 = 0$ ,
13.  $\text{Motif}(y, z) = \sum_{j=0}^{b-1} \sum_{k=1}^p MF(y_{(j_p+k)}, z_{\mathcal{B}(j)}) - TN(z) = 0$ ,
14.  $\text{FirstNoteChord}(y, L) = \sum_{j=1}^b \text{FirstNoteFC}(y_{\mathcal{B}(j)}, L_j) - b = 0$ ,
15.  $\text{MajorityMelody}(z, L) = \sum_{j=1}^b \text{FPN}(z_{\mathcal{B}(j)}, L_j) - b = 0$ ,
16.  $\text{MajorityArmony}(y, L) = \sum_{j=1}^b \text{MPN}(y_{\mathcal{B}(j)}, L_j) - b = 0$ ,
17.  $\text{BTRhythm}(y) = \sum_{j=2}^b \|BT(y_{\mathcal{B}(j)}) - BT(y_{\mathcal{B}(1)})\| = 0$ , si  $b > 1$ ,
18.  $\text{SDRRhythm}(y) = \sum_{j=2}^b \|SDR(y_{\mathcal{B}(j)}) - SDR(y_{\mathcal{B}(1)})\| = 0$ , si  $b > 1$ ,
19.  $\text{BTMelody}(z) = \sum_{\substack{j=2 \\ j \text{ par}}}^b \|BT(z_{\mathcal{B}(j)}) - BT(z_{\mathcal{B}(j-1)})\| = 0$ , si  $b > 1$ ,
20.  $\text{SDRMelody}(z) = \sum_{\substack{j=2 \\ j \text{ par}}}^b \|SDR(z_{\mathcal{B}(j)}) - SDR(z_{\mathcal{B}(j-1)})\| = 0$ , si  $b > 1$ .

## 4. Resultados experimentales

Se utilizó el algoritmo evolutivo multiobjetivo NSGA-II [8] para resolver el problema de optimización biobjetivo con restricciones descrito en la sección anterior. Para los parámetros del algoritmo NSGA-II se usó una población de 100 individuos, una cruce de  $2b$  puntos [21] con probabilidad de cruce  $p_c = 0.75$ , una mutación con la distribución de cruce binaria simulada [3, 9], con una probabilidad de mutación de  $p_m = 0.05$  y  $\eta = 1$ .

Para la implementación del código se utilizó C++, Python 3 y la plataforma pymoo [4]. Cabe mencionar que las restricciones del problema mostradas en la sección anterior hacen difícil para varios algoritmos encontrar soluciones, dado que primero se debe llegar a la región factible para posteriormente optimizar  $g_1$  y  $g_2$ ; de esta manera, después de algunas pruebas con algunos algoritmos, se decidió utilizar el algoritmo NSGA-II por su poder de exploración, siendo el que llegaba más rápidamente a la región factible.

Para todos los resultados se utilizó la lista de acordes  $L$  presentada en la definición 34, donde  $L$  tiene los acordes mayores y séptimos de la escala de Do Mayor. Se realizaron distintas pruebas variando los parámetros  $b$  y  $p$ , los resultados obtenidos se han colocado en la siguiente página<sup>1</sup>, donde se muestran algunas de las soluciones encontradas en el frente de Pareto<sup>2</sup> obtenido por el algoritmo para cada ejecución, donde se logra apreciar la diferencia entre mayor alegría y mayor minimalismo descritas en las funciones  $g_1$  y  $g_2$  del problema de optimización.

En la página web se presentan cuatro ejecuciones diferentes. Por ejemplo, para la ejecución 2, se utilizaron los parámetros  $b = 10, p = 8$ , tardando aproximadamente 18 horas en completar el proceso de optimización. Por otra parte, para la ejecución 3, se utilizaron los parámetros  $b = 10, p = 4$  y se puede apreciar en los compases 7 y 8 que las notas son mayormente más graves que el resto de la canción, permitiendo tener mayor impacto en esos compases. Mientras tanto, para la ejecución 4, se utilizaron los parámetros  $b = 4, p = 8$ , a pesar que es una canción de cuatro compases, se logra apreciar que no acabó de forma abrupta y se puede identificar a la melodía.

Dado que la población del algoritmo evolutivo multiobjetivo es de 100, al terminar la ejecución se cuenta con 100 individuos óptimos aproximando el frente de Pareto. Cabe destacar que dadas las restricciones del problema, para ejecuciones distintas con los mismos parámetros  $b, p$  las canciones resultantes son distintas; esto es debido a que el algoritmo busca canciones que cumplan el mismo ritmo para la melodía y armonía (ver las restricciones 17, 18, 19 y 20 del problema de optimización biobjetivo), por lo que se pueden encontrar distintos patrones rítmicos para canciones con misma cantidad de compases  $b$  y divisiones de particiones  $p$ .

Finalmente, los resultados obtenidos muestran diversidad en las figuras de las notas, tal y como se muestra en la figura presentada como ejemplo  $x^0$  en la sección 2, donde tal figura fue generada por el algoritmo evolutivo multiobjetivo.

<sup>1</sup> [natanvilchis.org/comia2022/](http://natanvilchis.org/comia2022/)

<sup>2</sup> El frente de Pareto [16] es el conjunto que da solución a un problema multiobjetivo. Por razones de espacio estas definiciones se han dejado fuera de este texto.

## 5. Conclusiones

En este trabajo se modeló el problema de composición musical automática mediante optimización biobjetivo. El fin fue crear melodías donde se estableciera un conjunto de soluciones compromiso (frente de Pareto) respecto al nivel de alegría, como primera función objetivo, y del minimalismo o simplicidad de adornos como segunda función optimizar.

Para la creación el modelo se empleó teoría musical sobre emociones y trabajos relacionados, se obtuvieron resultados satisfactorios al encontrar elementos auditivamente diversos sobre el frente de Pareto en cada ejecución. La construcción del modelo matemático presentado en este trabajo puede servir como base para el abordaje de otras emociones y aspectos diversos de forma dentro de la composición automática de melodías. La composición musical involucra distintos aspectos que necesitan ser estudiados de forma particular y su interacción entre los mismos, tales como ritmo, melodía, arpegios, entre otros.

Para el presente trabajo se buscó simplificar los aspectos más importantes de la música, tratando de mantener un equilibrio entre la capacidad creativa de los resultados y evitar la complejidad en el problema; por ejemplo, para los patrones musicales (en la figura melódica y la figura armónica) se utilizaron las funciones BTRhythm, SDRRhythm, BTMelody, SDRMelody. Como trabajo a futuro se propone cambiar la lista de acordes utilizadas en el presente trabajo para otorgar emociones alternativas.

## Referencias

1. Association of Musical Electronics Industry AMEI and MIDI Manufacturers Association MMA: Universal midi packet (ump) format and midi 2.0 protocol (2020), version 1.0
2. Bertin-Mahieux, T.: Large-Scale pattern discovery in music. Ph.D. thesis, Columbia University (2013)
3. Blank, J.: pymoo - mutation (2020), [pymoo.org/operators/mutation.html](https://pymoo.org/operators/mutation.html)
4. Blank, J.: pymoo: Multi-objective optimization in python (2020), [pymoo.org/](https://pymoo.org/)
5. Collins, T. E.: Improved methods for pattern discovery in music, with applications in automated stylistic composition. Ph.D. thesis, Faculty of Mathematics, Computing and Technology, The Open University (2011)
6. Cooke, D.: The language of music. Oxford University Press, 1 edn. (1959)
7. Dannenberg, R. B., Hu, N.: Pattern discovery techniques for music audio. *Journal of New Music Research*, vol. 32, no. 2, pp. 153–163 (2003) doi: 10.1076/jnmr.32.2.153.16738
8. Deb, K., Pratap, A., Agarwal, S., Meyarivan, T.: A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation*, vol. 6, no. 2, pp. 182–197 (2002) doi: 10.1109/4235.996017
9. Deb, K., Sindhya, K., Okabe, T.: Self-adaptive simulated binary crossover for real-parameter optimization. In: *Proceedings of the 9th Annual Conference on Genetic and Evolutionary Computation*. pp. 1187–1194 (2007) doi: 10.1145/1276958.1277190

10. Herremans, D., Chew, E.: MorpheuS: generating structured music with constrained patterns and tension. *IEEE Transactions on Affective Computing*, vol. 10, no. 4, pp. 510–523 (2017) doi: 10.1109/TAFFC.2017.2737984
11. de María, M.: 18 dimensiones emocionales, decodificando las emociones de la música desde 18 dimensiones técnicas (2021)
12. Meredith, D.: COSIATEC and SIATECCompress: Pattern discovery by geometric compression. In: *International society for music information retrieval conference*. pp. 6. No. 14, International Society for Music Information Retrieval (2013)
13. Meredith, D.: Compression-based geometric pattern discovery in music. In: *2014 4th International Workshop on Cognitive Information Processing (CIP)*. pp. 1–6 (2014) doi: 10.1109/CIP.2014.6844503
14. Meredith, D.: Recur sia-rrt: Recursive translatable point-set pattern discovery with removal of redundant translators. In: *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*. vol. 1168, pp. 485–493 (2019) doi: 10.1007/978-3-030-43887-6\_42
15. Meredith, D., Lemström, K., Wiggins, G. A.: Algorithms for discovering repeated patterns in multidimensional representations of polyphonic music. *Journal of New Music Research*, vol. 31, no. 4, pp. 321–345 (2002)
16. Miettinen, K.: *Nonlinear multiobjective optimization*, vol. 12. Springer Science & Business Media (2012)
17. Miranda, E. R. (ed): *Handbook of artificial intelligence for music: foundations, advanced approaches, and developments for creativity*. Springer Nature (2021)
18. Ren, I., Volk, A., Swierstra, W., Veltkamp, R. C.: A computational evaluation of musical pattern discovery algorithms, (2020) doi: 10.48550/arXiv.2010.12325
19. Rodríguez-Alvira, J.: *Funciones armónicas : Notas de adorno* (2022)
20. Scirea, M., Togelius, J., Eklund, P., Risi, S.: Affective evolutionary music composition with metacompose. *Genetic Programming and Evolvable Machines*, vol. 18, no. 4, pp. 433–465 (2017) doi: 10.1007/s10710-017-9307-y
21. Umbarkar, A. J., Sheth, P. D.: Crossover operators in genetic algorithms: a review. *ICTACT journal on soft computing*, vol. 6, no. 1 (2015) doi: 10.21917/ijsc.2015.0150

## Aprendizaje automático para identificar episodios de esquizofrenias tratadas con clozapina

Julieta Guadalupe Rodríguez-Ruiz, Carlos Eric Galván-Tejada,  
Jorge Issac Galván-Tejada, Hamurabi Gamboa-Rosales,  
José María Celaya-Padilla

Universidad Autónoma de Zacatecas,  
Unidad Académica de Ingeniería Eléctrica,  
México

{jr.ruiz68, ericgalvan, gatejo,  
hamurabigr, jose.celaya}@uaz.edu.mx

**Resumen.** El aprendizaje automático ha sido bastante utilizado en el campo del cuidado de la salud para lograr asistencia en el diagnóstico, detección temprana de padecimientos, monitoreo de pacientes, monitoreo de tratamientos e incluso sub clasificación de enfermedades. Para la psiquiatría de precisión, este tipo de técnicas de inteligencia artificial se vuelven primordiales para dar un mejor estilo de vida a los pacientes. La esquizofrenia es uno de los desórdenes psiquiátricos que causa mayores efectos negativos tanto en los pacientes como en sus familiares y, al menos 30 % de ellos tienen algún tipo de resistencia al tratamiento. La clozapina es un medicamento neuroléptico que se ha establecido como la primera opción para este tipo de pacientes. Sin embargo, es necesario aumentar la cantidad de investigación sobre este tipo de pacientes y su tratamiento para corroborar su eficacia y descartar posibles efectos secundarios. Existen diversos enfoques sobre los datos que pueden utilizarse con pacientes psiquiátricos para lograr identificar tratamientos, eficacia y respuesta, entre ellos destaca el uso de imagen de resonancia magnética, datos clínicos y datos de actividad física. En este trabajo se presenta una clasificación de episodios de pacientes con diagnóstico de esquizofrenia que usan clozapina como tratamiento y pacientes que usan otro tipo de neurolépticos. La clasificación se realiza empleando una señal de actividad física recolectada a través de un acelerómetro minuto a minuto, el mejor modelo obtuvo una exactitud de 0.81 usando Regresión Logística y episodios de día completos. Este tipo de modelo puede hacer sinergia con otros análisis para una atención integral de pacientes con esquizofrenia.

**Palabras clave:** Clozapina, actividad física, clasificación, aprendizaje automático.

### Machine Learning for Identifying Episodes of Schizophrenias Treated with Clozapine

**Abstract.** Machine learning has been widely used in the field of health care to assist in diagnosis, early detection of conditions, patient monitoring, treatment

monitoring and even sub-classification of diseases. For precision psychiatry, this type of artificial intelligence techniques become essential to give patients a better lifestyle. Schizophrenia is one of the psychiatric disorders that causes the greatest negative effects on both patients and their families, and at least 30% of them have some type of resistance to treatment. Clozapine is a neuroleptic medication that has been established as the first option for this type of patient. However, it is necessary to increase the amount of research on this type of patients and their treatment to corroborate its effectiveness and rule out possible side effects. There are several approaches to data that can be used with psychiatric patients to identify treatments, efficacy, and response, including the use of magnetic resonance imaging, clinical data, and physical activity data. This paper presents a classification of episodes of patients diagnosed with schizophrenia who use clozapine as treatment and patients who use other types of neuroleptics. The classification is made using a signal of physical activity collected through an accelerometer minute by minute, the best model obtained an accuracy of 0.81 using Logistic Regression and full day episodes. This type of model can synergize with other analyzes for comprehensive care of patients with schizophrenia.

**Keywords:** Clozapine, physical activity, classification, machine learning.

## 1. Introducción

La esquizofrenia es una enfermedad grave que afecta alrededor del 1% de la población y a pesar de su baja prevalencia, se mantiene en las principales causantes de discapacidad tanto laboral como en el ámbito personal [19].

Entre los principales síntomas se encuentran alucinaciones, delirios, deterioro cognitivo, depresión, manía, trastorno compulsivo, entre otros, estos síntomas deben perdurar por al menos un mes para ser diagnosticados como causados por esquizofrenia [17]. Es un padecimiento que generalmente aparece durante la niñez o juventud a raíz de posibles cambios abruptos en la interacción cotidiana con su entorno [4].

Su descripción, técnicas de diagnóstico, clasificaciones y posibles tratamientos se encuentran en la Clasificación Internacional de Enfermedades (CIE) y El Manual Diagnóstico y Estadístico de los Trastornos Mentales (DSM por sus siglas en inglés), cuyos conceptos se han modificado y actualizado acorde a las ediciones publicadas y el conocimiento nuevo adquirido [17].

La clozapina se conoce como la primera opción de tratamiento para esquizofrenias con resistencia al tratamiento y cuyos síntomas no se ven afectados por los neurolepticos convencionales [10]. Sin embargo, es un tratamiento aún en etapa experimental y se requiere un monitoreo constante y eficiente para evitar efectos secundarios graves a mortales [5].

Es importante señalar que el promedio de psiquiatras por cada 100 mil habitantes alrededor del mundo es de 4.1, y en México, al menos en la mitad del país, tenemos una tasa de menos de dos psiquiatras por cada 100 mil habitantes [9]. La psiquiatría se mantiene en constante actualización y a la par del avance tecnológico, se han optado por nuevas técnicas para el mayor conocimiento de este y otros padecimientos, como el uso

de inteligencia artificial usando diferentes tipos de datos como imágenes, biomarcadores, entre otros.

Con el afán de aprovechar la gran cantidad de datos que se generan actualmente y la necesidad de incrementar la eficiencia en la atención médica para la población, se han introducido diversas técnicas para la asistencia en el diagnóstico, monitoreo de tratamiento, identificación temprana de enfermedades, entre otras, usando inteligencia artificial [7, 15].

Debido a la complejidad para diagnosticar y tratar la mayoría de las enfermedades, se requiere un esfuerzo conjunto para resolver los retos que se presentan en el sector de cuidados de la salud. Para ello se ha recomendado recolectar datos desde el hogar del paciente a través de dispositivos que puedan vestirse y combinarlos con datos clínicos recolectados por médicos y otros especialistas [8].

El uso de actígrafos han mostrado gran capacidad para reconocer patrones de comportfunamiento en la actividad física de pacientes con esquizofrenia y depresión, incluso se ha podido reconocer una reducción significativa de actividad durante las horas de sueño en pacientes con esquizofrenia usando clozapina [1, 13].

En otra revisión sistemática, Wee, Z. Y., Yong et al. incluyeron 38 estudios donde se utilizan satisfactoriamente distintos actígrafos para la recolección de datos de actividad física y, con ello, diferenciar entre pacientes con esquizofrenia y controles (personas sin enfermedades psiquiátricas) [18].

Los actígrafos están compuestos regularmente por un acelerómetro y algún dispositivo de almacenamiento, debido a los resultados que se han obtenido con este y otros padecimientos se ha catalogado como dispositivo médico capaz de monitorear pacientes [16].

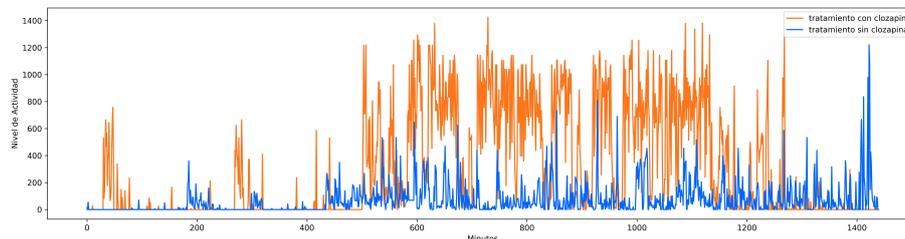
Existen otro tipo de investigaciones además del uso de actividad física, entre los más destacados está el trabajo de Bo Caol et al. donde usando imágenes de resonancia magnética en estado de reposo, obtienen un mapa de conexiones funcionales de diversas regiones corticales. Esto con objetivo de identificar los primeros episodios de pacientes esquizofrénicos sin tratamiento previo con un 78.6 % de precisión y 82.5 % para la predicción de la respuesta al tratamiento [2].

En este artículo se presenta la clasificación de esquizofrenias tratadas con clozapina y otro tipo de tratamientos. Usando actividad física diaria se pretende encontrar un modelo que logre identificar patrones entre los niveles de actividad física de un paciente que tiene en su tratamiento el medicamento clozapina y con ello diferenciarlos de aquellos episodios que correspondan a un paciente sin tratamiento de clozapina.

## **2. Materiales y métodos**

### **2.1. Recolección de datos**

El conjunto de datos utilizado para este experimento se titula *Psykose* y fue publicado bajo licencia exclusiva para investigación y enseñanza. Está conformado por archivos CSV (Comma Separated Values, por sus siglas en inglés) de 22 pacientes con diagnóstico de esquizofrenia y 32 personas saludables, controles. Cada archivo contiene tres columnas con la fecha, el minuto y la actividad del paciente. El valor de actividad



**Fig. 1.** Gráfica de comparación entra la señal de actividad física durante 24 h de un paciente con esquizofrenia y tratamiento con clozapina contra uno sin tratamiento con clozapina.

física se obtuvo usando un piezoeléctrico embebido en un actígrafo llamado Actiwatch, Cambridge Neurotechnology Ltd, England, model AW4, cuya frecuencia de muestreo es de 32Hz y contabiliza los movimientos mayores a 0.05g, de esta manera de cada minuto se tiene un conteo de movimientos proporcional a la cantidad de actividad motriz que se tuvo. El actígrafo se colocó en la muñeca derecha de cada paciente y cada control por 13 días en promedio [11].

Además de contar con la actividad de cada paciente, se incluye otro archivo CSV con información acerca de los pacientes incluidos en la recolección de los datos, donde se presenta información como rango de edad, tipo de esquizofrenia, padecimiento de migraña, calificación de la Escala Breve de Valoración Psiquiátrica (BPRS, por sus siglas en inglés), si se usa clozapina en el tratamiento, si se usa un estabilizador emocional, edad de la primera hospitalización, entre otros.

## 2.2. Preprocesamiento

Debido a que el objetivo es identificar el tratamiento entre los pacientes con padecimiento de esquizofrenia, solo se utilizaron sus datos de actividad física. Una vez identificados aquellos pacientes que tuvieron tratamiento con clozapina mientras se recolectaban los datos, se concatenaron los distintos archivos CSV en uno solo. Con ello se obtiene un solo archivo para los que tuvieron tratamiento con clozapina y otro para los que no.

Teniendo los datos agrupados en una sola matriz, se utilizó la traspuesta para obtener un solo arreglo de toda la señal de actividad física. Para segmentar la señal de actividad física y convertir una sola señal en episodios, se utilizó una ventana de 1440 minutos. De esta manera se generan observaciones de un día, cada una con un conteo de actividad por minuto, siendo esto último las características (columnas) de cada observación ( renglón).

Finalmente, los días incompletos, es decir, las observaciones que no alcanzaron los 1440 minutos de un solo día, fueron eliminadas para evitar datos faltantes, en total se eliminaron 86 observaciones. En la Figura 1 se puede observar la comparación de las señales de actividad física de un paciente esquizofrénico que usa clozapina y otro que no; es evidente las diferencias entre ambas señales, principalmente el aumento de la actividad física a través del uso de clozapina.

Los primeros 420 minutos corresponden al lapso de las 00:00 horas hasta las 07:00, el cual correspondería al tiempo de sueño. Berle, J. O. et al. utilizaron el mismo conjunto

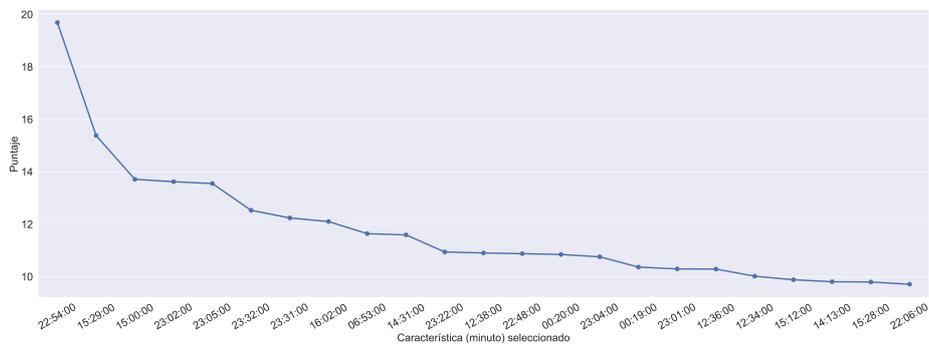


Fig. 2. Gráfica con las mejores veintitrés características seleccionadas a través del puntaje.

de datos para hacer un análisis estadístico de las señales y encontraron que los pacientes que usan clozapina tienen mayor estabilidad en la actividad física diaria y ciclos de sueño más estables y duraderos [1].

### 2.3. Reducción de dimensionalidad

A pesar de que diversos algoritmos de aprendizaje automático tienen la capacidad de clasificar usando datos complejos y de dimensiones elevadas, se recomienda simplificar los modelos para evitar redundancia o ruido en los datos y lograr mayor precisión [20]. En este caso se realizó una selección secuencial de características hacia adelante, este algoritmo toma una a una cada característica y decide si seleccionarla o eliminarla del conjunto dependiendo si la exactitud aumenta o disminuye.

El selector secuencial de características se implementó con Regresión Logística (RL) y una validación cruzada de 5 observaciones. Y, para definir si se queda o no la característica nueva seleccionada, se tomó en cuenta la exactitud en la clasificación de los episodios. Una vez realizada la selección se obtuvo que el mejor modelo para la clasificación está formado por veintitrés características y para obtenerlas se ponderó cada una usando el valor F de ANOVA entre ellas. En la Figura 2 se pueden observar los puntajes obtenidos de las mejores características obtenidas para formar el modelo de clasificación.

### 2.4. Aprendizaje automático

El aprendizaje automático puede tener dos objetivos; aprendizaje supervisado, algoritmos que basados en un entrenamiento identifican patrones en los datos para separarlos en clases, o aprendizaje no supervisado, cuyos algoritmos no tienen un entrenamiento previo y automáticamente intentan descubrir patrones que les permitan agrupar o clasificar los datos.

En este caso se optó por los algoritmos supervisados Random Forest (RF), Super Vector Machine (SVM), y Regresión Logística (RL), cuyos resultados en el campo de la psiquiatría han sido favorables [3].

Los tres algoritmos fueron entrenados con el 70 % de los datos, lo que corresponde a 140 instancias con 23 características. Y el entrenamiento se realizó con las 60 observaciones restantes.

**Evaluación.** Para llevar a cabo la evaluación de cada uno de los modelos es necesario calcular distintas métricas basadas en las principales salidas que obtenemos de clasificaciones binarias; verdaderos positivos (VP), verdaderos negativos (VN), falsos positivos (FP), y falsos negativos (FN). Estas medidas son los conteos de instancias clasificadas correcta o incorrectamente y de qué manera fueron clasificadas. Con estos conteos obtenemos métricas como precisión, exhaustividad, medida F1 y la exactitud.

La precisión es la proporción de verdaderos positivos correctos, es decir, del total de clasificaciones marcadas como positivas, cuántas realmente eran positivas, se puede calcular con la ecuación 1:

$$\text{precision} = \frac{VP}{VP + FP}. \quad (1)$$

La exhaustividad, ecuación 2, calcula el total de instancias clasificadas como positivas y que realmente son positivas:

$$\text{exhaustividad} = \frac{VP}{VP + FN}. \quad (2)$$

La medida F1, como se muestra en la ecuación 3 calcula el promedio entre la precisión y la exhaustividad, permitiendo conocer de manera general cómo se comporta el modelo hablando específicamente de las clasificaciones positivas.

$$F1 = 2 \times \frac{\text{precision} \times \text{exhaustividad}}{\text{precision} + \text{exhaustividad}}. \quad (3)$$

Y, finalmente la exactitud se define en la ecuación 4:

$$\text{exactitud} = \frac{TP + TN}{TP + TN + FP + FN}. \quad (4)$$

La exactitud obtiene la proporción de clasificaciones hechas correctamente sin importar si fuese una clasificación positiva o negativa. Nos da el comportamiento general del modelo teniendo en cuenta ambas clases. Es necesario retomar distintas métricas para observar qué tan bueno es un modelo para clasificar una clase o la otra.

### 3. Resultados

Del ranking de características seleccionadas para formar el modelo se puede observar que 12 de ellas, poco más de la mitad, son de los minutos entre las 10 pm y las 12:30 am. Y, el resto son minutos de entre las 12:30 pm y las 4:30 pm.

Estos horarios se pueden relacionar con horarios de poca actividad, ya sea por los momentos previos al sueño o por la tarde relacionarlo con horario de comida. Es importante recalcar esto porque entonces estos horarios de baja actividad física son los que están marcando una diferencia significativa en las señales.

**Tabla 1.** Resultados de la clasificación de episodios de pacientes que sufren esquizofrenia usando clozapina.

Model		Precisión	Exhaustividad	F1	Exactitud
<b>Regresión Logística</b>	0	0.82	0.93	0.87	0.81
	1	0.80	0.60	0.69	
<b>Random Forest</b>	0	0.78	0.95	0.85	0.78
	1	0.82	0.45	0.58	
<b>Super Vector Machine</b>	0	0.77	0.75	0.76	0.68
	1	0.52	0.55	0.54	

0 = tratamiento sin clozapina, 1 = tratamiento con clozapina

La clozapina funciona sobre sistema nervioso central y se recomienda su prescripción únicamente si se ha probado con al menos otros dos medicamentos. Este medicamento intenta regular los ciclos circadianos del paciente y reducir considerablemente los brotes psicóticos. Por lo tanto, el hecho de que estos momentos de menor nivel de energía nos describa mucho mejor la señal que el total de 1440 minutos, puede señalar que efectivamente los pacientes con uso de clozapina tienden a regular y tener actividades más estables que aquellos pacientes que no incluyen clozapina en su tratamiento. Una vez obtenido el conjunto con las actividades seleccionadas, se ejecutaron los algoritmos de aprendizaje automático RL, RF y SVM usando lenguaje Python y la librería de sklearn. Los resultados obtenidos por cada modelo se pueden observar en el cuadro 1.

Los algoritmos de regresión se basan en buscar relaciones entre las variables independientes para describir o identificar a la variable dependiente, en contraste con los algoritmos de árboles de decisiones, donde asemejando un diagrama de flujo utiliza los datos para tomar decisiones.

RF, es un algoritmo de aprendizaje automático supervisado que genera varios árboles de decisiones y selecciona los más eficientes para realizar la clasificación [6]. Y SVM, a su vez buscando las relaciones entre las variables independientes, realiza un análisis de regresión para construir un hiperplano con el mayor rango posible y con ello realizar la clasificación [12].

El mejor modelo para esta clasificación fue el de Regresión Logística, donde se obtuvo un 81 % de exactitud en la clasificación. Se debe recalcar que el 30 % con el que se probó cada modelo corresponde a 60 instancias, pero los valores de precisión y exhaustividad son calculados por el total de cada clase y son proporcionales.

De este modelo de RL y del resto se puede observar una mayor capacidad de reconocer episodios de pacientes sin clozapina en su tratamiento que aquellos que usan clozapina. Para el modelo de RL la exhaustividad de la clase con clozapina es de 60 % en contraste con la precisión de 80 % y la exhaustividad de pacientes sin clozapina en 93 %, esto indica que se están confundiendo más pacientes con clozapina y se les está clasificando como pacientes que no usan clozapina.

El segundo modelo con mayor exactitud es el de RF con 78 % y finalmente SVM con 68 %. Ambos tienen una mayor capacidad de reconocer actividad física de pacientes que no usan clozapina.

#### **4. Conclusiones y trabajo futuro**

Finalmente, se obtuvo un modelo con la exactitud adecuada para que a través de señal de actividad física se pueda reconocer el tratamiento de clozapina en pacientes esquizofrénicos con 80 %.

Este reconocimiento de tratamiento puede aportar en el análisis de eficacia del medicamento mediante la observación de los minutos específicos donde se está obteniendo mejor resultado, además de un posible seguimiento de los síntomas para observar si existe o no una diferencia a los pacientes que no tienen el tratamiento.

Algunas de las investigaciones previas mencionadas en la introducción se limitan a realizar un análisis estadístico de las señales. En este artículo proponemos un modelo de clasificación con un 81 % de exactitud al clasificar. Sin embargo, se puede ser más exhaustivo, para el trabajo futuro, un análisis univariado de esos veintitrés minutos que se seleccionaron puede dar más indicios sobre como modifica la actividad física el uso de clozapina.

También será interesante describir de otra manera la señal de actividad física a través de extracción de características. Esto pudiera darnos más nociones sobre el comportamiento total de cada paciente y obtener mejores resultados en la clasificación.

Además, por esta ocasión se decidió realizar segmentación por 24 h debido al modo de acción de la clozapina, pero se podrían hacer otro tipo de segmentación, ya sea por hora o por día y noche, para establecer mejores resultados y además identificar en qué momento existe mayor capacidad del modelo para diferenciar las señales, así como se ha hecho anteriormente con episodios de pacientes con depresión [14].

Finalmente, este tipo de investigación puede aportar en el campo de la psiquiatría de precisión y en el uso de inteligencia artificial para analizar la efectividad del tratamiento de clozapina y cuál es el comportamiento de este tipo de pacientes.

La identificación del primer episodio de esquizofrenia es vital para regresar la funcionalidad del paciente y con eso su integración a un ambiente familiar y laboral favorable. Para lograrlo se requiere atención psiquiátrica y tratamiento de inmediato, sin embargo, la tasa de psiquiatras por población en México impacta en la capacidad del sector salud para resolver esta problemática.

Implementar técnicas de inteligencia artificial a la par del sector salud podría mejorar tanto diagnóstico, tratamiento y reintegración de pacientes con desórdenes psiquiátricos.

#### **Referencias**

1. Berle, J. O., Hauge, E. R., Oedegaard, K. J., Holsten, F., Fasmer, O. B.: Actigraphic registration of motor activity reveals a more structured behavioural pattern in schizophrenia than in major depression. *BMC research notes*, vol. 3, no. 1, pp. 1–7 (2010) doi: 10.1186/1756-0500-3-149
2. Cao, B., Cho, R. Y., Chen, D., Xiu, M., Wang, L., Soares, J. C., Zhang, X. Y.: Treatment response prediction and individualized identification of first-episode drug-naive schizophrenia using brain functional connectivity. *Molecular psychiatry*, vol. 25, no. 4, pp. 906–913 (2020) doi: 10.1038/s41380-018-0106-5

3. Dwyer, D., Koutsouleris, N.: Annual research review: Translational machine learning for child and adolescent psychiatry. *Journal of Child Psychology and Psychiatry*, vol. 63, no. 4, pp. 421–443 (2022) doi: 10.1111/jcpp.13593
4. Enríquez-Sánchez, H., Ochoa-Madrigal, M. G.: Espectro de la esquizofrenia en niños y adolescentes. *Revista de la Facultad de Medicina (México)*, vol. 62, no. 4, pp. 9–23 (2019) doi: 10.22201/fm.24484865e.2019.62.4.02
5. Escamilla-Orozco, R. I., Becerra-Palars, C., Armendáriz-Vázquez, Y., Corlay-Noriega, I. S., Herrera-Estrella, M. A., Llamas-Núñez, R. E., Meneses-Luna, Ó., Quijada-Gaytán, J. M., Reyes-Madrigal, F., Rosado-Franco, A., Rosel-Vales, M., Saucedo-Uribe, E.: Tratamiento de la esquizofrenia en México. *Gaceta médica de México*, vol. 157, no. 4, pp. S1–S12 (2021) doi: 10.24875/gmm.m21000501
6. Espinosa-Zúñiga, J. J.: Aplicación de algoritmos random forest y XGBoost en una base de solicitudes de tarjetas de crédito. *Ingeniería, investigación y tecnología*, vol. 21, no. 3 (2020) doi: 10.22201/fi.25940732e.2020.21.3.022
7. Fabrissin, J.: ¿Los datos masivos serán el futuro de la psiquiatría? *V Xerte, Revista Argentina de Psiquiatría*, vol. XXIX, pp. 379 (2018)
8. Goecks, J., Jalili, V., Heiser, L. M., Gray, J. W.: How machine learning will transform biomedicine. *Cell*, vol. 181, no. 1, pp. 92–101 (2020) doi: 10.1016/j.cell.2020.03.022
9. Heinze, G., del Carmen-Chapa, G., Santisteban, J. A., Vargas, I.: Los especialistas en psiquiatría en México: su distribución, ejercicio profesional y certificación. *Salud mental*, vol. 35, no. 4, pp. 279–285 (2012)
10. Inchauspe-Aróstegui, J. A., Valverde-Eizaguirre, M. Á.: Creer en la clozapina: fe y evidencias. *Revista de la Asociación Española de Neuropsiquiatría*, vol. 38, no. 133, pp. 239–262 (2018) doi: 10.4321/s0211-57352018000100013
11. Jakobsen, P., Garcia-Ceja, E., Stabell, L. A., Oedegaard, K. J., Berle, J. O., Thambawita, V., Hicks, S. A., Halvorsen, P., Fasmer, O. B., Riegler, M. A.: Psykose: A motor activity database of patients with schizophrenia. In: 2020 IEEE 33rd International Symposium on Computer-Based Medical Systems (CBMS). pp. 303–308. IEEE (2020) doi: 10.1109/CBMS49503.2020.00064
12. Lugo-Reyes, S. O., Maldonado-Colín, G., Murata, C.: Inteligencia artificial para asistir el diagnóstico clínico en medicina. *Revista Alergia México*, vol. 61, no. 2, pp. 110–120 (2014) doi: 10.29262/ram.v61i2.33
13. Ransing, R., Patil, P., Khapri, A., Mahindru, A.: A systematic review of studies comparing actigraphy indices in patients with depression and schizophrenia. *Journal of Clinical and Diagnostic Research*, vol. 15, no. 8 (2021) doi: 10.7860/JCDR/2021/48760.15293
14. Rodríguez-Ruiz, J. G., Galván-Tejada, C. E., Zanella-Calzada, L. A., Celaya-Padilla, J. M., Galván-Tejada, J. I., Gamboa-Rosales, H., Luna-García, H., Magallanes-Quintanar, R., Soto-Murillo, M. A.: Comparison of night, day and 24 h motor activity data for the classification of depressive episodes. *Diagnostics*, vol. 10, no. 3, pp. 162 (2020) doi: 10.3390/diagnostics10030162
15. Rutledge, R. B., Chekroud, A. M., Huys, Q. J.: Machine learning and big data in psychiatry: toward clinical applications. *Current opinion in neurobiology*, vol. 55, pp. 152–159 (2019) doi: 10.1016/j.conb.2019.02.006

16. Strauss, G. P., Raugh, I. M., Zhang, L., Luther, L., Chapman, H. C., Allen, D. N., Kirkpatrick, B., Cohen, A. S.: Validation of accelerometry as a digital phenotyping measure of negative symptoms in schizophrenia. *Schizophrenia*, vol. 8, no. 1, pp. 1–6 (2022) doi: 10.1038/s41537-022-00241-z
17. Valle, R.: Schizophrenia in ICD-11: Comparison of ICD-10 and DSM-5. *Revista de Psiquiatría y Salud Mental (English Edition)*, vol. 13, no. 2, pp. 95–104 (2020) doi: 10.1016/j.rpsmen.2020.01.002
18. Wee, Z. Y., Yong, S. W. L., Chew, Q. H., Guan, C., Lee, T. S., Sim, K.: Actigraphy studies and clinical and biobehavioural correlates in schizophrenia: a systematic review. *Journal of Neural Transmission*, vol. 126, no. 5, pp. 531–558 (2019) doi: 10.1007/s00702-019-01993-2
19. World Health Organization: Esquizofrenia (2022), <https://www.who.int/es/news-room/fact-sheets/detail/schizophrenia>
20. Zhang, H.: Comparación entre dos métodos de reducción de dimensionalidad en series de tiempo. *Revista Colombiana de Estadística*, vol. 32, no. 2, pp. 189–212 (2009)

## Uso de series temporales para el análisis y predicción de consumo de materia prima en la industria maquiladora

Hugo Alberto Morales Casas, Francisco López Orozco,  
Vicente García Jiménez

Universidad Autónoma de Ciudad Juárez,  
Instituto de Ingeniería y Tecnología,  
Maestría en Cómputo Aplicado,  
México

hugomoca98@gmail.com, francisco.orozco@uacj.mx

**Resumen.** Los pronósticos de consumo de materia prima dentro de la industria maquiladora representan una de las etapas más importantes para el cumplimiento y satisfacción de las órdenes del cliente en tiempo y forma. Dentro de un esquema tradicional, las estimaciones se llevan a cabo con el uso de fórmulas matemáticas simples que van desde un promedio mensual hasta el incremento porcentual en tasas del 10% al 15% de un periodo a otro. Sin embargo, los resultados no siempre resultan satisfactorios. Por esta razón, la aplicación de técnicas de aprendizaje máquina podrían ofrecer una mejora de las estimaciones mensuales de consumo de energía eléctrica, producción agrícola y manufacturera, utilizando modelos de series temporales como “autoregressive integrated moving average”, (ARIMA). Su aplicación en modelos estacionales previamente demostrados por las pruebas de “Dickey-Fuller”, (DF) y “Kwiatkowski-Phillips-Schmidt-Shin”, (KPSS) para conjuntos de datos de al menos 3 años de registro histórico brinda apoyo para la reducción de incrementos y decrementos de demanda en la industria maquiladora. En este trabajo se propone el uso de series temporales (ST) basados en una mezcla del modelo tradicional de las ST y el modelo “Obtain – Scrub – Explore – Model – Interpret”, (OSEMN) para la estimación de consumo de materiales tales como resinas plásticas, rollos de metales dentro de la industria eléctrica doméstica, considerando solamente una parte del universo de números de parte que cuentan con características de estacionalidad.

**Palabras clave:** ARIMA; time series; forecasting.

## Use of Time Series for the Analysis and Prediction of Raw Material Consumption in the Maquiladora Industry

**Abstract.** Raw material consumption forecasts within the maquiladora industry represent one of the most important stages for the fulfillment and satisfaction of customer orders in a timely manner. Within a traditional scheme, estimates are carried out with the use of simple mathematical formulas that range from a monthly average to the percentage increase in rates of 10% to 15% from one period to another. However, the results are not always satisfactory. For this reason, the application of machine learning techniques could offer an improvement in the monthly estimates of electrical energy consumption, agricultural and manufacturing production, using time series models such as "autoregressive integrated moving average", (ARIMA). Its application in seasonal models previously demonstrated by the "Dickey-Fuller" (DF) and "Kwiatkowski-Phillips-Schmidt-Shin" (KPSS) tests for data sets of at least 3 years of historical record provides support for the reduction of increases and decreases in demand in the maquiladora industry. This work proposes the use of time series (ST) based on a mixture of the traditional model of the ST and the model "Obtain - Scrub - Explore - Model - Interpret", (OSEMN) for the estimation of consumption of materials such as plastic resins, metal coils within the domestic electrical industry, considering only a part of the universe of part numbers that have seasonal characteristics.

**Keywords:** ARIMA; timeseries; forecasting.

### 1. Introducción

El análisis del comportamiento de la demanda de materia prima en el sector maquilador representa una de las mayores áreas de oportunidad dentro de este rubro. El abastecimiento y el proceso que conlleva la logística para la cadena de suministros (Supply Chain) es uno de los principales problemas para lograr el cumplimiento de las entregas en tiempo para el cliente al 100% (On Time Delivery). Cabe mencionar también que esta situación es crucial en dicha industria orientada al cliente al momento de otorgar alguna certificación o una nueva oportunidad de negocios.

Los sistemas de mejora continua que resultan ser aplicados en la industria maquiladora convencional mencionan que dentro de los tópicos primordiales para un On Time Delivery al 100% es el abastecimiento óptimo de materia prima (Raw Material), así como la identificación de forma categórica de los números de parte, por ejemplo, A, B o C dependiendo de su uso temporal en al menos 52 semanas (1 año) [1].

Sin embargo, la metodología que es aplicada en la mayoría de las empresas al momento de asignar una categoría A, B o C a los números de parte de materia prima y producto terminado, consta del cálculo de un promedio de uso dentro de un rango segmentado por meses durante por menos un año, buscando tener un aprovechamiento

óptimo de la materia prima. De esta forma se asegura en todo momento disponibilidad, sin tener grandes excedentes ni la falta de el mismo. Esto en algunas ocasiones deriva en la falta de materia prima durante los periodos de temporada alta, y exceso de inventario en las temporadas bajas, ya que el consumo de materia se proyecta como un pronóstico similar o “estable” durante todo el año, el cual no siempre resulta ser así.

En la actualidad, la búsqueda de patrones y tendencias en los comportamientos diarios como sociedad, se puede ver reflejado en la implementación de técnicas de aprendizaje máquina (AM), en actividades cotidianas [4], con el fin de encontrar alternativas y soluciones que sean más simples a las tareas realizadas por el ser humano. Basta con dar un vistazo a algunas herramientas de uso diario, y en las que de alguna forma el AM se ve involucrado.

Desde sugerencias publicitarias apareciendo en redes sociales por medio de algoritmos que analizan conjuntos de búsquedas en el navegador, aplicaciones de compra y venta, palabras claves mencionadas a voz dentro de conversaciones cotidianas, e inclusive por medio de rutinas de traslado y ubicación GPS [5] que son analizadas por medio de técnicas para el procesamiento de datos.

Estas técnicas logran ser aplicadas en escalas aún mayores, como lo son el análisis de tendencias en temporadas electorales, logrando influenciar de forma crítica en las decisiones políticas de la población de un país [6].

Sumado al afán de identificar y establecer rutinas para cada día de la semana, con la intención de obtener una mayor administración para las tareas del hogar con ayuda de la domótica, y de esta forma evitar algún imprevisto a futuro gracias a que estas actividades logran ser identificadas por medio de patrones con múltiples similitudes, a veces aún más notorias de lo que se puede esperar [7].

Es aquí donde el aprendizaje máquina toma relevancia, identificando, descubriendo nuevas y significativas relaciones, modelos, así como tendencias al evaluar grandes cantidades de datos con la ayuda de tecnologías de reconocimiento de patrones, redes neuronales, lógica difusa, algoritmos genéticos, entre otros [8].

Por ejemplo, ciudades fronterizas donde se encuentran instaladas una gran cantidad de complejos maquiladores, por lo que se convierten en un gran campo de oportunidades para la integración de técnicas para el análisis de datos como las series temporales, tal y como se propone en este trabajo, el cual aplica técnicas basadas en aprendizaje máquina para el análisis y pronóstico del proceso de requisición de materia prima a proveedores del sector maquilador en un periodo de tres años.

Resultando que de esta forma más órdenes de producto terminado lleguen al cliente en tiempo, logrando un aprovechamiento óptimo de energías renovables actuales, e impulsando su correcta sustitución en hogares y automóviles donde aún no han sido implementadas.

## 2. Trabajos relacionados

En la actualidad existen herramientas teóricas basadas en la experiencia de grandes corporativos industriales como Toyota, basados en el concepto de mejora continua o “*Lean Manufacturing*”. Sin embargo, no resultan ser totalmente efectivas al momento de planear y realizar pronósticos (*forecast*) de consumos para la materia prima. El uso de herramientas de aprendizaje máquina o series temporales podría ayudar a obtener mejores resultados para el pronóstico de datos basados en los registros de usos

históricos con una mayor efectividad y fluidez, beneficiando a la industria maquiladora en el ramo eléctrico, como es el caso de Ciudad Juárez.

De esta manera se brindan pronósticos a proveedores con mayor efectividad y ofertando una reducción de costos ocasionados por tiempo extra de producción, costo de expeditado y compras con proveedores provisionales (*Spot Buy*), con quienes el material resulta ser más costoso de lo habitual al momento de intentar cubrir los requerimientos del cliente en tiempo.

Así más órdenes de producto terminado llegarían al cliente en tiempo y forma, logrando un aprovechamiento óptimo de energías renovables actuales, e impulsando su correcta sustitución en hogares y automóviles donde aún no han sido implementadas.

En el año 2016, Delgadillo [9], presentó su trabajo en el cual se emplearon diversas herramientas con la intención de realizar una comparativa en el pronóstico de series de tiempo aplicados a la producción de granos básicos en México

Esto, con la intención de predecir sus valores a corto plazo, aplicando modelos como *Autoregressive Integrated Moving Average* (ARIMA), el cual es una combinación de los parámetros de Auto Regresión (AR) e Integrado (I) a los modelos de medias móviles (MA) [10], modelo de Brown, suavización simple y modelo de Holt.

La integración de dichos modelos cuenta con antecedentes en trabajos realizados dentro de países como Canadá dentro de la predicción de producción de trigo y Bangladesh para la papaya, piña y mango utilizando modelos de series temporales ARIMA. Delgadillo utilizó registros de 112 años en segmentos mensuales, (1897 a 2009), provenientes de la organización federal SAGARPA.

Además de esto, se implementó el modelo ARIMA en sus cuatro etapas: Identificación, estimación, comprobación, pronóstico. Durante el proyecto, se realizaron pruebas con otros modelos, sin embargo, el modelo ARIMA fue el que presentó mejores resultados, brindándole de esta forma estimaciones hasta el año 2020, y 2050, siempre teniendo en cuenta que podrían verse afectadas por la gran distancia en años entre los datos originales a los estimados.

Por otro lado, dentro de la industria manufacturera de la India, Siregar [11] aplica la metodología ARIMA y Box – Jenkins para los pronósticos de ventas del año 2015. Los datos utilizados corresponden a las ventas de productos plásticos en el periodo mensual durante los años del 2012- 2014.

Además de esto, aplica métricas de error porcentual absoluto medio, (MAPE) para medir el nivel de error obteniendo resultados de entre el 68% y 74% de efectividad, planteando una posibilidad de aventajamiento ante la competencia industrial para ser más competitivos y ofrecer productos de calidad en tiempo y forma para los clientes.

Debido a que mantener un pronóstico débil o pésimo, orilla a las compañías a generar niveles bajos de inventario y un servicio ineficiente, es posible identificar de esta forma las métricas de ventas como un área de mejora.

Antes de comenzar con el análisis de datos, estos fueron preprocesados por medio de tres etapas: Limpieza, donde el autor removió los posibles puntos de ruido, se rellenaron los campos faltantes, y se identificaron “Outliers”, Integración de los datos, donde se combinaron los conjuntos de datos correspondientes a los años del 2012 a 2014, y la selección de datos, donde se escogieron las muestras relevantes y necesarias para el pronóstico.

En 2018 A. Rafael [12] realizó una comparativa de diversos modelos predictivos, esto debido a que no es posible aplicar la misma metodología de solución en diversos



**Fig. 1.** Adaptación del Modelo Obtain – Scrub – Explore – Model – Interpret (OSEM), presentado por Chen Han Lau, con inclusión de modelo tradicional en la sección de modelado e interpretación [14].

problemas. Su efectividad depende totalmente de los parámetros con los que se cuente dentro de los datos a analizar.

A. Rafael comparó resultados obtenidos al evaluar 95 diferentes conjuntos de datos para producción de chocolate en Australia, dentro de periodos mensuales durante los años de 1960 a 1990. Se aplicaron modelos como SARIMA, ARIMA, KNN y el modelo de Máquina de Soporte Vectorial (SVM).

Resultando el modelo SARIMA el que mejores resultados ofreció al término del análisis. Rafael utilizó la metodología de series de tiempo, la cual permite estimar los parámetros para posteriormente realizar predicciones de datos.

En algunas ocasiones, como parte de la etapa de preprocesamiento es necesario tratar la información antes de ingresarla al modelo seleccionado de evaluación. Llegando inclusive a aplicar técnicas de imputación de datos.

De hecho, Osman [13] lo identifica como uno de los principales problemas presentes en las series de tiempo. Por esta razón, en 2018 elaboró un estudio para identificar qué metodología de imputación resultaría ser la óptima dentro de los sistemas de distribución de agua. Utilizando desde metodologías básicas y tradicionales, hasta las más complejas.

Estos fueron clasificados en grupos de imputación por: media, valores aleatorios, proceso Gaussiano y regresión lineal. Siendo los primeros dos, parte de los modelos más simples de utilizar al momento de trabajar con información faltante dentro de las series de tiempo. Sin embargo, la principal desventaja que presentan corresponde a la pérdida de valores estadísticos de posible relevancia, como lo son incrementos estacionales.

Por otro lado, los últimos dos modelos pertenecen a la familia de aprendizaje máquina, resultando ser más sofisticados, un poco más costosos y para este caso de estudio presentando mejores resultados debido a la gran cantidad de datos disponibles para el estudio.

### 3. Metodología

En este trabajo se propuso como base de modelación un método tradicional del proceso de ciencia de datos (Data Science), también conocido como modelo O.S.E.M.N. (Obtain – Scrub – Explore – Model – Interpret), representado en la Figura 1, donde para el caso de la obtención de datos, se llevó a cabo un proceso de recopilación mensual de registros de consumo durante el periodo de enero 2019 a diciembre 2021 en una fábrica maquiladora perteneciente al sector electro en Ciudad Juárez, Chihuahua.

La base de datos original consta de 10,000 números de parte, correspondientes únicamente a materia prima obtenida directamente de proveedor, y números de parte enviados desde otras divisiones correspondientes a la misma compañía manufacturera (transferencias inter-planta).

Para el caso de la limpieza de datos, los registros como primera instancia se obtuvieron directamente en formato xls, con algunos vacíos de registros en semanas donde no se utilizó el material. Dichos espacios en blancos fueron rellenados con ceros y además de eso se estandarizaron a valores numéricos con valor decimal.

Al momento de seleccionar las muestras experimentales, se utilizaron números de parte que tuvieran movimientos en cada uno de los treinta y un meses de registros, no importando la unidad de medida.

En el caso de la exploración, los datos fueron dispuestos en tablas y gráficas de barras para la identificación de estacionalidad de uso. El modelado y resultados fueron basados en muestras previamente comprobadas como datos estacionales con la aplicación de las pruebas estacionalidad de Dickey-Fuller (DF) y Kwiatkowski-Phillips-Schmidt-Shin (KPSS), las cuales son pruebas utilizadas dentro del análisis estadístico para la verificación de la existencia o no de raíces unitarias dentro de una serie de tiempo [15], además de comprobar estadísticamente si dicha serie es estacionaria o no [16].

Después para cada uno de los números de parte analizados fueron aplicadas las pruebas de DF y KPSS, corroborando la estacionalidad. Las series estacionales series mantienen sus características estadísticas (valor medio, desviación estándar, etc.), a lo largo del tiempo analizado. Esto quiere decir que no es un comportamiento transitorio, por lo que tiene duración y largo infinito, al menos teóricamente [17].

### 3.1. Desarrollo

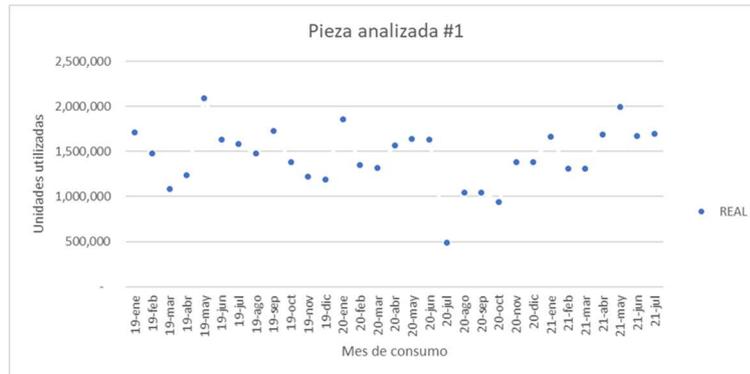
Para propósitos de este artículo, fueron seleccionados únicamente 5 diferentes números de parte (identificados como “Número de parte 1,2,3,4,5” en tablas posteriores) que cumplieran con las características anteriormente mencionadas, registro mayor a cero de consumo en cada uno de 31 meses estudiados en el presente trabajo, es decir, se hubieran utilizado al menos una vez en el mes registrado.

Se realizaron pruebas aplicando los modelos *Moving Average* (MA), *Weight Moving Average* (WMA) y ARIMA. Para los modelos de MA y WMA se consideraron parámetros de *lags* igual a tres.

Así mismo, los pesos para el modelo de WMA fueron asignados desde tres a uno, en forma descendiente. Finalmente, para la asignación de los valores P y Q de modelo ARIMA se utilizaron las funciones de Auto Correlación y Auto Correlación Parcial respectivamente. Los registros fueron separados en dos grupos, uno de entrenamiento con el 66% de los datos y 34% de prueba.

Para el primer número de parte se aplicaron los tres métodos de predicción. En la Figura 2, podemos observar el comportamiento que registro dentro de los 31 meses analizados, así mismo dentro de la Tabla 1, podemos observar los resultados obtenidos para cada uno de los modelos aplicados en este número de parte y visualmente en la Figura 3.

Este mismo procedimiento fue aplicado para el resto de las 5 piezas analizadas.



**Fig. 2.** Correspondiente a la pieza analizada #1 de parte seleccionado, en él se muestran los comportamientos de consumo a lo largo de 31 meses.

**Tabla 1.** Correspondiente a la pieza analizada #1, en ella se muestran los resultados obtenidos aplicando los 3 diferentes modelos, además de los resultados esperados o reales.

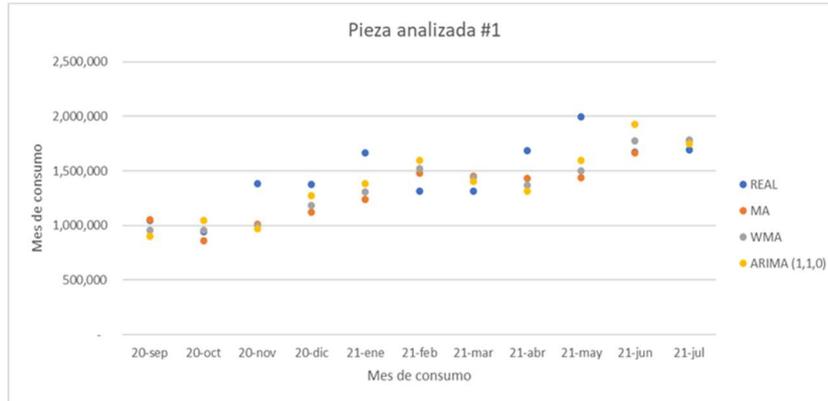
Mes	REAL	MA	WMA	ARIMA (1,1,0)
20-sep	1,046,833	1,053,865	956,554	904,666
20-oct	942,286	859,970	953,038	1,046,309
20-nov	1,384,952	1,011,257	994,196	967,369
20-dic	1,378,733	1,124,690	1,181,044	1,275,871
21-ene	1,666,592	1,235,324	1,308,065	1,380,224
21-feb	1,310,902	1,476,759	1,523,699	1,597,478
21-mar	1,310,902	1,452,076	1,440,771	1,400,490
21-abr	1,687,517	1,429,465	1,370,184	1,310,902
21-may	1,994,980	1,436,440	1,499,210	1,594,341
21-jun	1,673,653	1,664,466	1,778,479	1,925,291
21-jul	1,693,876	1,785,383	1,783,073	1,749,858

### 3.2. Validación de resultados

Como parte de la validación de resultados se aplicó el error MAPE (Error Porcentual Absoluto Medio o Mean Absolute Percentage Error por sus siglas en inglés), comparando los valores reales con los obtenidos en cada uno de los modelos predictivos. Otorgando resultados muy próximos, y en 3 de ellos teniendo un error menor para ARIMA, los cuales se encuentran resaltados en color verde.

También se aplicó el error MSE (Error Cuadrático Medio o por sus siglas en inglés mean squared error) como parte de validación de resultados, dentro de la Tabla 3, podemos observar los resultados obtenidos, siendo los remarcados en verde los mejores para dichos experimentos. Donde en esta ocasión solamente para uno de ellos se presentó una menor cantidad de error en la categoría de ARIMA.

Además de esto, se realizaron pruebas T. Student aplicado a la comparativa de cada uno de los resultados obtenidos al aplicar el cálculo de errores MAPE y MSE.



**Fig. 3.** Comparativa para los últimos 11 meses, correspondiente al 34% de los datos, donde se puede observar el comportamiento que presento cada uno de los modelos aplicados contra los registros reales de uso.

**Tabla 2.** Comparación de porcentaje de error MAPE para cada una de las piezas analizadas.

	MAPE / REAL-MA	MAPE / REAL-WMA	MAPE / REAL-ARIMA
Número de parte 1	13.9%	14.1%	15.4%
Número de parte 2	9.1%	8.5%	6.6%
Número de parte 3	16.8%	17.4%	16.7%
Número de parte 4	16.4%	17.1%	16.6%
Número de parte 5	31.5%	31.0%	30.6%

**Tabla 3.** Comparación de porcentaje de error MSE para cada una de las piezas analizadas.

	MSE / REAL-MA	MSE / REAL-WMA	MSE / REAL-ARIMA
Número de parte 1	75,587,017,701	68,741,895,279	68,813,906,564
Número de parte 2	174,723	163,444	125,331
Número de parte 3	63,650,560,805	62,946,768,760	65,218,856,112
Número de parte 4	59,016,581,648	58,724,258,348	61,906,662,175
Número de parte 5	47,658,823,669	48,574,991,531	57,194,017,714

En la Tabla 4, podemos observar que las diferencias no resultan ser significativas al momento de aplicar la prueba T. Student, debido a que los valores arrojados por esta prueba resultan ser muy próximos entre sí.

En la Tabla 5, podemos observar resultados similares para la prueba de error MSE, donde los resultados vuelven a no presentar significancia de resultados, debido a que se encuentran próximos entre sí.

#### 4. Conclusiones (ventajas y desventajas) y trabajo futuro

Se puede observar que el uso de metodologías como MA, WMA y ARIMA para la predicción de consumos de materia prima en la industria maquiladora, resulta ser muy

**Tabla 4.** Resultados obtenidos al aplicar T. Student a los resultados obtenidos por MAPE.

	MA-WMA	MA-ARIMA	WMA-ARIMA
Número de parte 1	0.484	0.359	0.365
Número de parte 2	0.357	0.114	0.172
Número de parte 3	0.460	0.493	0.457
Número de parte 4	0.451	0.489	0.467
Número de parte 5	0.470	0.447	0.473

**Tabla 5.** Resultados obtenidos al aplicar T. Student a los resultados obtenidos por MSE.

	MA-WMA	MA-ARIMA	WMA-ARIMA
Número de parte 1	0.430	0.426	0.499
Número de parte 2	0.355	0.104	0.160
Número de parte 3	0.492	0.484	0.477
Número de parte 4	0.497	0.468	0.466
Número de parte 5	0.470	0.309	0.328

variable y dependiente del comportamiento de los registros con los que se cuente al momento de analizar los datos.

Sin embargo, resulta ser una opción más apegada a la realidad que el uso de un promedio mensual aplicado al total de registros con los que se cuenta. Si bien los resultados al momento de les fuera aplicados la prueba de T. Student no presentaron significancia, los valores de ARIMA al momento de aplicar las pruebas de error resultaron ser menores, dando pie a una búsqueda de mayor efecto en los resultados para este modelo.

donde, las pruebas pudieran volver a repetirse con un mayor número de muestras, que contasen con más registros aplicados, cabe recalcar que 2 de los trabajos aquí citados utilizaron registros mayores a 30 años y obtuvieron como resultados valores favorables para metodología ARIMA, por lo que podemos deducir que dicho modelo funciona mejor cuando se cuenta con una base de datos más amplia.

Además de esto, una vez encontrado el rango de años adecuados para estudiar, el uso de técnicas de *clustering* podría ayudarnos a abarcar mayor cantidad de números sin profundizar en la búsqueda de parámetros P y Q para la utilización del modelo ARIMA.

## Referencias

1. Liker, J. K., Franz, J. K.: El modelo Toyota para la mejora continua: Conectando la estrategia y la excelencia operacional para conseguir un rendimiento superior. Profit Editorial (2020)
2. Monleón-Getino, A.: El impacto del Big-data en la sociedad de la información. Significado y utilidad. Historia y Comunicación Social, vol. 20, no. 2, pp. 427–445 (2015)
3. Brink, H., Richards, J. W., Fetherolf, M.: Real world - Machine learning. Shelter Island, NY: Manning Publications Co (2016)
4. León-Borges, J. A., Noh-Balam, R. I., Rangel-Gómez, L. Philip-Strand, M.: The machine learning in the prediction of elections. Computación e Informática, no. 2 (2015)

5. Herrera-Quintero, L. F.: Viviendas inteligentes. *Ingeniería e Investigación*, vol. 25, no 2, pp. 47–53 (2005)
6. Perez-Lopez, C., Montero-Lorenzo, J. M.: *Minería de datos. Técnicas y herramientas. Paraninfo* (2007)
7. Delgadillo-Ruiz, O., Ramírez-Moreno, P. P., Leos-Rodríguez, J. A., Salas-González, J. M., Valdez Cepeda, R. D.: Pronósticos y series de tiempo de rendimientos de granos básicos en México. *Acta universitaria*, vol. 26, no. 3, pp. 23–32 (2016) doi: 10.15174/au.2016.882
8. Gras, J. A.: *Diseños de series temporales: Técnicas de análisis. Universidad de Barcelona* (2001)
9. Siregar, B., Nababan, E. B., Yap, A., Andayani, U., Fahmi: Andayani and Fahmi, forecasting of raw material needed for plastic products based in income data using ARIMA method. In: *Proceeding of 2017 5th International Conference on Electrical, Electronics and Information Engineering (ICEEIE)*, no. 5, pp. 135–139 (2017) doi: 10.1109/ICEEIE.2017.8328777
10. Parmezana, A. R. S., Souza, V. M. A., Batista, G. E. A. P. A.: Evaluation of statistical and machine learning models for timeseries prediction: Identifying the state-of-the-art and the best conditions for the use of each model. *Information Sciences*, vol. 484, pp. 302–337 (2019) doi: 10.1016/j.ins.2019.01.076
11. Osman, M. S., Abu-Mahfouz, A. M., Page, P. R.: A survey on data imputation techniques: water distribution system as a use case. In: *Proceeding of IEEE*, vol. 6, pp. 63279–63291 (2018) doi: 10.1109/ACCESS.2018.2877269
12. Lau, C. H.: *Towards Data Science*. [En línea] Available: <https://towardsdatascience.com/5-steps-of-a-data-science-project-lifecycle-26c50372b492> (2019)
13. Hacker, S., Abdunnasser, H. J.: *The properties of procedures dealing with uncertainty about intercept and deterministic trend in unit root testing. Royal Institute of Technology, Centre of Excellence for Science and Innovation Studies* (2010)
14. Singh, A.: *A Gentle introduction to handling a non-stationary time series in Python. Analytics Vidhya* (2018)
15. Nava, A.: *Procesamiento de series de tiempo. Fondo de cultura económica* (2015)

## Selección de atributos en microarreglos mediante un algoritmo wrapper metaheurístico

Miguel Ángel Garrido-Castañeda, Antonio Alarcón-Paredes,  
Oscar Camacho-Nieto

Instituto Politécnico Nacional,<sup>1</sup>  
Centro de Investigación en Computación,  
México

{mgarridoc2021,aalarcon}@cic.ipn.mx, ocamacho@ipn.mx

**Resumen.** Las tecnologías de expresión genómica han permitido analizar una gran cantidad de información por medio del análisis de microarreglos, cuyos datos se conforman por unas cuantas observaciones y miles de atributos, muchos de ellos irrelevantes o redundantes, lo que puede llevar a un pobre desempeño en la clasificación. Si bien existen soluciones para la selección de atributos, como los wrappers, aunque son muy utilizados suelen ser computacionalmente costosos debido a su método de búsqueda. En este trabajo se presenta un algoritmo que realiza un pre-filtrado de atributos para su posterior uso en un wrapper metaheurístico. Se llevó a cabo una experimentación tomando en cuenta once conjuntos de datos de microarreglos utilizando cuatro metaheurísticas y diez algoritmos de clasificación en el estado del arte. Se demuestra que el desempeño del método propuesto ofrece mejoras en cuanto al desempeño de clasificación y/o tiempo respecto de utilizar el conjunto completo de datos y de los wrappers tradicionales.

**Palabras clave:** Wrappers, metaheurística, microarrays, clasificación.

### Feature selection in microarrays using a metaheuristic wrapper algorithm

**Abstract.** Genomic expression technologies have made it possible to analyze a large amount of information through microarray analysis, whose data is made up of a few observations and thousands of attributes, many of them irrelevant or redundant, which can lead to poor performance in gene expression classification. Although there are solutions for the selection of attributes, such as wrappers, although they are widely used, they are usually computationally expensive due to their search method. This paper presents an algorithm that performs a pre-filtering of attributes for later use in a metaheuristic wrapper. An experimentation was carried out taking into account eleven microarray datasets using four metaheuristics and ten state-of-the-art classification algorithms. It is shown that the performance of the proposed method offers improvements in terms of

classification performance and/or time compared to using the full data set and traditional wrappers.

**Keywords:** Wrappers, metaheuristics, microarrays, classification.

## 1. Introducción

Los microarreglos son una tecnología de amplio uso en la expresión genética; estos son estructuras de alta dimensionalidad que permiten agrupar un número muy grande de genes. Actualmente es posible encontrar conjuntos de datos (datasets) obtenidos a partir de esta tecnología, que comúnmente se conforman por pocas muestras (siendo  $m$  el número total de éstas) y muchos genes (con  $p$  como el número total de genes o atributos) representando los valores del dataset con tipo de dato numérico. Esta tecnología tiene la finalidad de monitorear y medir datos relevantes de la información biológica que permita comprender y analizar contextos específicos como el diagnóstico de enfermedades o clasificación de tumores [1,2,3]. Por tal motivo frecuentemente se une con el aprendizaje automático para realizar estas tareas. Sin embargo, han sido detectados algunos inconvenientes al tratar de aplicar algoritmos de clasificación con este tipo de datos específicos.

Dichos inconvenientes caen en la problemática de la maldición de la dimensionalidad, siendo el primero originado por las pocas muestras que se tienen en los conjuntos de datos ocasionando un deficiente entrenamiento del clasificador y posteriormente una pobre capacidad de predicción del mismo, y el segundo siendo ocasionado por presentarse un conjunto de datos que contiene un número muy grande de atributos que ocasionará que se tenga una probabilidad de que muchas de sus características estén correlacionadas o sean irrelevantes lo que al final ocasiona un pobre desempeño de clasificación acompañado de un costo computacional muy elevado [4].

Existen técnicas de pre procesamiento de datos en el estado del arte que ayudan a contrarrestar este tipo de problemáticas, como los métodos de selección de atributos que en concreto se encargan de seleccionar los atributos no correlacionados con la mayor información discriminativa con los cuales se forma un pequeño subconjunto el cual es mostrado al clasificador esperando que no degenera su desempeño de clasificación [4,5,6].

Dependiendo de la forma en que operen, existen 3 categorías principales de métodos para la selección de atributos: métodos de filtrado (conocidos como *filters* en inglés), métodos de envoltura (conocidos como *wrappers* en inglés) y métodos integrados (conocidos como *embedded methods* en inglés).

Los filters eliminan los atributos correlacionados antes de llegar a la fase de clasificación, normalmente evaluando y seleccionando por separado a cada uno de los atributos [7]. Por otro lado, los métodos wrappers se unen al clasificador para evaluar los distintos subconjuntos de atributos, siendo el elegido el de mayor desempeño de clasificación. Este método suele destacar con mejores resultados de clasificación comparado con los otros; sin embargo, tienden a requerir un mayor número de recursos computacionales y por lo tanto un mayor tiempo de procesamiento siendo éste un aspecto importante a tomar en cuenta cuando se trabaja con estos métodos [8,9].

Por último, en los métodos embedded la selección de atributos son parte de la construcción del modelo de clasificación, donde atributos fuertemente correlacionados están fuera o dentro del modelo por completo mediante algún tipo de penalización, donde comúnmente suelen utilizar la norma L1, L2, alguna combinación de ellas o bien un análisis discriminante disperso [10,11,12].

Otro tipo de complejidad que se suele encontrar al trabajar en conjuntos de datos de microarreglos es el desbalance de clases, para estos casos obtener el desempeño de un clasificador entrenado con este tipo de datos no basta con calcular el *accuracy* (número de aciertos entre total de datos) ya que la desproporción de las clases no permitiría que se refleje el verdadero desempeño del clasificador.

Para estos casos se hace una distinción entre las clases del conjunto de datos en clase positiva (P) y negativa (N) (para conjuntos de datos de dos clases). Dicha distinción permite crear la matriz de confusión conformada por los verdaderos positivos: el número de datos P clasificados como clase P (TP); los falsos negativos: número de datos P clasificados como clase N (FN); los verdaderos negativos: número de datos N clasificados como clase N (TN); y los falsos positivos: el número de datos N clasificados como clase P (FP).

Con la obtención de estos valores es posible implementar medidas de desempeño, que sean útiles en conjuntos de datos desbalanceados, como pueden ser: la sensibilidad (*sensitivity*:  $\frac{TP}{TP+FN}$ ), especificidad (*specificity*:  $\frac{TN}{TN+FP}$ ) y el *balanced accuracy* (promedio obtenido por las dos métricas anteriores).

## 2. Método propuesto

El presente método se propone con la finalidad de atacar el problema de la maldición de la dimensionalidad con un evidente objetivo primordial: disminuir la cantidad de atributos necesarios para la clasificación de conjuntos de datos altamente dimensionales, como son los microarreglos. Sin embargo, la correcta elección de un subconjunto de atributos lleva consigo una solución en dos vertientes: en primera instancia, se disminuye el tiempo requerido para su clasificación, y en segunda, los algoritmos de clasificación pueden obtener un mejor desempeño. Una vista general del algoritmo puede verse en la Figura 1.

El presente método retoma una idea propuesta en [13], donde se propone iniciar con un ranking de atributos y un pre-filtrado de forma que se tomen los primeros  $n < p$  atributos en el orden de importancia dado por el ranking.

Tras ese pre-filtrado, se propone un método wrapper metaheurístico (WMH), donde se llevan a cabo experimentos utilizando cuatro algoritmos, a saber: Bat Optimization (BAT) [14], Grey Wolf Optimization (GWO) [15], Firefly Optimization Algorithm (FFA) [16] y el conocido Particle Swarm Optimization (PSO) [17]. Las partes constituyentes de la propuesta se explican a detalle a continuación.

### 2.1. Pre-filtrado

Esta primera etapa es muy útil cuando se trabaja con datos altamente dimensionales. Su propósito es el de discriminar tempranamente aquellos atributos que son menos representativos, de forma similar a como trabaja un método filter de selección de

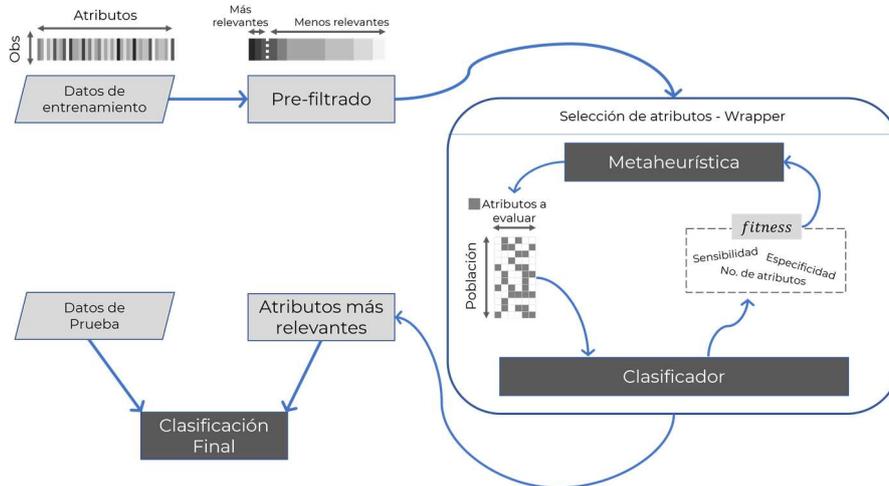


Fig. 1. Modelo propuesto.

atributos. Se lleva a cabo al evaluar individualmente la relevancia de cada atributo con respecto del vector de clase. Para este propósito, el criterio de evaluación de cada atributo consiste en clasificar el conjunto de datos con ese atributo únicamente; la métrica de rendimiento utilizada en este trabajo es el balanced accuracy.

Sea  $X = [f_1, f_2, \dots, f_p] = [X_1, X_2, \dots, X_m]^T$  una matriz de tamaño  $m \times p$  con  $m$  observaciones y  $p$  atributos, donde  $f_i$  y  $X_i$  son la  $i$ -ésima columna e  $i$ -ésima fila, respectivamente; y sea  $y = [y_1, y_2, \dots, y_m]^T$  su correspondiente vector de clases. Los atributos son evaluados al medir el desempeño del clasificador  $clf$  de tal forma que  $Balanced\ accuracy(i) = clf(f_i, y)$ , y se utiliza el método de validación leave-one-out [18]. Posteriormente, se ordena este resultado de mayor a menor en el vector  $SortedBAcc$ , y se toman los  $n$  mejores atributos, con  $n < p$ , tal que el nuevo conjunto de datos se construye como:

$$Xs = X[:, SortedBAcc[1:n]]. \quad (1)$$

## 2.2. Selección de atributos con un wrapper metaheurístico

Los métodos wrapper evalúan diferentes subconjuntos de atributos para encontrar aquel que obtenga el mejor desempeño en función de un algoritmo de clasificación. Es claro que para encontrar el óptimo sería necesario probar todas las combinaciones posibles, lo que es igual que aplicar fuerza bruta. Comúnmente estos algoritmos utilizan algún método de búsqueda voraz para encontrar dicho subconjunto.

En su lugar, se propone la utilización de una metaheurística como método de búsqueda. De forma general, las metaheurísticas son algoritmos de optimización que permiten encontrar una solución en un espacio continuo. Sin embargo, el problema de la selección de atributos es un problema en un espacio multidimensional binario, por ello, se consideró una modificación binaria de las metaheurísticas utilizadas.

**Tabla 1.** Datasets de microarreglos de cáncer, características y descripción.

Dataset	Atributos	Muestras	Clases	Distribución de clases (%)	Descripción
9 tumors	5,726	60	9	15 / 12 / 13 / 10 / 10 / 13 / 13 / 4 / 10	9 tipos de tumor
11 tumors	12,533	174	11	16 / 5 / 15 / 13 / 7 / 6 / 4 / 15 / 3 / 8 / 8	11 tipos de tumor
14 tumors	15,009	308	26	10 / 4 / 6 / 5 / 7 / 4 / 3 / 3 / 5 / 4 / 4 / 5 / 4 / 6 / 2 / 3 / 2 / 4 / 2 / 2 / 2 / 2 / 4 / 3 / 1 / 3	14 tipos de tumor y subclases
Braintumor 1	5,920	90	5	67 / 11 / 11 / 4 / 7	5 tipos de tumor cerebral
Braintumor 2	10,367	50	4	28 / 14 / 28 / 30	4 tipos de gliomas
DLBCL	5,469	77	2	75 / 25	Linfomas B difusos de células grandes
Leukemia 1	5,327	72	3	53 / 13 / 34	3 tipos de leucemia: mieloide aguda y linfocítica aguda de células B y T
Leukemia 2	11,225	72	3	39 / 33 / 28	3 tipos de leucemia aguda: mieloide, linfocítica y mixta
Lung Cancer	12,600	203	5	68 / 9 / 10 / 10 / 3	4 tipos de cáncer de pulmón, y tejido sano
Prostate tumor	10,509	102	2	49 / 51	Tumor de próstata o sanos
SRBCT	2,308	83	4	35 / 30 / 13 / 22	Tumores de células pequeñas y redondas

### 3. Experimentación

Para la fase experimental se tomaron en cuenta 11 datos de microarreglos [19], que pueden ser consultados en la Tabla 1. Además, 10 clasificadores del estado del arte (3-NN [20], 5-NN[20], Naive Bayes [21], Multilayer Perceptron [22], Random Forest [23], AdaBoost [24], Extra Trees [25], Decision Tree [26], SVM-lineal [27], SVM-RBF

[27]) fueron utilizados con las 4 metaheurísticas previamente mencionadas (PSO, GWO, FFA, BAT). En esta sección se presentan los detalles de la implementación, así como también los resultados obtenidos.

### 3.1. Datos

Se muestra en la Tabla 1 algunos detalles de los 11 datos de microarreglos usados para probar el método propuesto. Como particularidad de este tipo de datasets el número de atributos contenido en cada uno de estos supera ampliamente al número de observaciones, debido a la naturaleza de los microarreglos (matrices con miles de celdas) al obtenerse el dataset cada microarreglo queda representado como un vector de muy alta dimensionalidad y con valores numéricos representando el contenido de cada una de sus celdas. Además, suele ser común que algunos de ellos presenten la problemática del desbalanceo de clases, como puede verse la distribución de las mismas en la tabla descriptiva de los datos.

### 3.2. Condiciones experimentales

El algoritmo propuesto fue implementado en Python 3.8 utilizando la biblioteca scikit-learn para los algoritmos de Machine learning, y haciendo una modificación a las metaheurísticas propuestas en [28, 29]. Los parámetros de los clasificadores utilizados son los que se incluyen por defecto en scikit-learn.

Para proporcionar una válida experimentación, se dividió el conjunto de datos en dos particiones, de forma que el 75% de ellos se utilizó para llevar a cabo todo el procesamiento de selección de población en las metaheurísticas y dentro de ellas se usó un método de validación cruzada interno con 3 folds (por el desbalance de clases), mientras que el 25% restante de la primera división se utilizó para evaluar el resultado de los mejores subconjuntos de atributos. Debido al desbalance de clases, se eligió el balanced accuracy como métrica para evaluar el desempeño de los clasificadores, que de forma general puede ser expresada como sigue:

$$\text{Balanced Accuracy} = \frac{1}{K} \sum_{i=1}^K \left( \frac{TP_i}{P_i} \right), \quad (2)$$

donde  $TP_i$  y  $P_i$  representan las observaciones bien clasificadas de la clase  $i$ , y el total de observaciones de la misma clase  $i$ , respectivamente, mientras que  $K$  representa el total de clases del conjunto de datos.

Se usó también la prueba estadística de Friedman, que es una prueba no paramétrica basada en las evaluaciones promedio de los algoritmos de clasificación.

Ésta permite por un lado identificar si existen diferencias significativas en los resultados de clasificación y, por otro lado, establecer un orden de mejor clasificación. Se tomaron en cuenta diez algoritmos de clasificación, basados en diferentes enfoques y que son de los más utilizados en el estado del arte. Como se mencionó previamente, estos son: 3-NN, 5-NN, Naive Bayes, MLP, Random Forest, AdaBoost, Decision Tree, Extra Trees, SVM-Lineal, SVM-RBF.

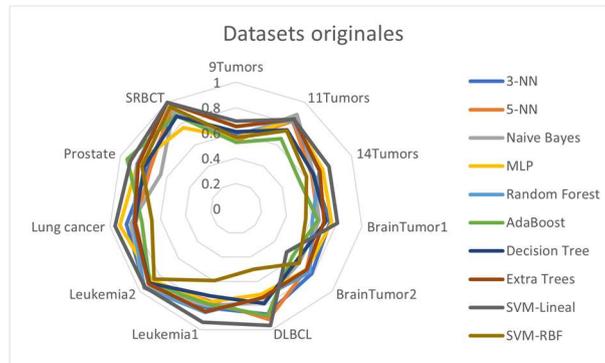


Fig. 2. Resultado datasets con todos los atributos.

Todos los experimentos se corrieron en una computadora con sistema operativo Windows 10, un procesador Intel Core i5 (2.40 GHz) de 11ª generación, con 8GB de memoria RAM instalada.

### 3.3. Modelo propuesto

#### 3.3.1. Tamaño de la población

Si bien las metaheurísticas referidas tienen un espacio de búsqueda continuo, se lleva a cabo una modificación en la generación de la población, estableciendo un umbral en los valores continuos con la finalidad de binarizar dicho espacio. Uno de los parámetros para utilizar las metaheurísticas es el tamaño de la población, que está íntimamente relacionado con la cantidad de atributos.

Por ejemplo, si un conjunto de datos tiene 3 atributos, existen  $2^3 - 2 = 6$  posibles combinaciones para seleccionar (con un 1) o descartar (con un 0) cada uno de esos tres atributos: (0,0,1), (0,1,0), (0,1,1), (1,0,0), (1,0,1), (1,1,0). Sin las combinaciones con todos 0 o todos 1 ya que no seleccionaría ningún atributo o se tomarían en cuenta todos ellos.

Así pues, para un conjunto de datos con 3 atributos, un tamaño de población mayor a 6 podría representar que tome en cuenta todas las combinaciones posibles de atributos. No obstante, esta situación es muy diferente para conjuntos de datos con más atributos, por ejemplo, uno con 10 atributos tendría  $2^{10} - 2 = 1022$  posibles combinaciones; el dataset de microarreglos más pequeño de los aquí analizados tiene 2308 atributos, lo que representaría una cantidad impensable de combinaciones.

Por lo anterior, el tamaño de población para datos con atributos  $p < 9$ , es  $2 \times p$ , y en cualquier otro caso el tamaño de población es igual a 200.

#### 3.3.2. Función fitness

Partiendo de que los datos de microarreglos presentan en su mayoría desbalanceo de clases, y que uno de los objetivos es seleccionar la menor cantidad de atributos posible,

para la creación de la función fitness se consideraron valores tales como la sensibilidad (Sen) y especificidad (Spe) además del número de atributos seleccionados ( $n_s$ ).

La penalización por seleccionar más atributos se considera como la cantidad total de atributos seleccionados  $n_s$  sobre los atributos  $n$  que ingresan a la metaheurística, y se le asocia un valor de ponderación  $W_n$  en caso de que se desee aplicar una mayor o menor penalización; por defecto aquí se tiene un esquema conservador con  $W_n = 0.1$ .

El hecho de buscar el mejor desempeño con una menor cantidad de atributos constituye un problema de maximización – minimización, y los métodos de optimización aquí usados corresponden a minimización, el desempeño del algoritmo se invierte al restársele al valor máximo si clasificara correctamente todo, que es 1.

De tal forma que la función fitness se propone como:

$$fitness = 1 - (Sen + Spe) + W_n \frac{n_s}{n}. \quad (3)$$

### 3.3.3. Propuesta general

El método propuesto funciona de la siguiente manera:

- 1) Se evalúa cada atributo del dataset usando como métrica el balanced accuracy, de modo que se realiza un pre-filtrado de atributos seleccionando los  $m$  mejores, de acuerdo con el algoritmo de clasificación seleccionado, como se explica en la subsección 2.1,
- 2) El dataset obtenido, con el número de atributos disminuido se analiza usando una de las metaheurísticas mencionadas, de manera que se genera una población de posibles soluciones iniciales; esto es, una población binaria con valores 0 y 1, indicando cuáles atributos habrán de seleccionarse en cada miembro de la población inicial, como se detalla en la subsección 3.1.1. Se divide el conjunto de datos en dos particiones: el 75% de ellos se dedica al procedimiento de las metaheurísticas para generar poblaciones más aptas, y el 25% restante se utiliza para evaluar el desempeño con la nueva población,
- 3) Se lleva a cabo el procedimiento de las metaheurísticas generando, en cada una de sus iteraciones, poblaciones más aptas, es decir, va generando soluciones donde se seleccionan diferentes atributos que mejoren el desempeño de la función fitness propuesta para este trabajo, ver subsección 3.1.2,
- 4) Finalmente se toma en consideración el subconjunto de atributos que obtiene un mejor valor en la función fitness; esto es, mejora el desempeño del balanced accuracy, y a su vez disminuye la cantidad de atributos utilizados en la clasificación del dataset.

## 4. Resultados y discusión

En esta sección se muestra una comparativa entre los resultados obtenidos con el método propuesto y los métodos wrapper convencionales que utilizan un método de búsqueda voraz para encontrar el mejor subconjunto de datos.

Selección de atributos en microarreglos mediante un algoritmo wrapper metaheurístico

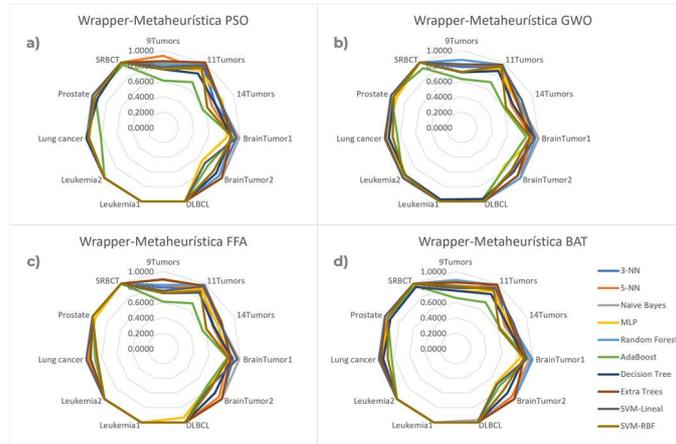


Fig. 2. Resultado datasets con todos los atributos.

Tabla 2. Mejores desempeños en clasificación para cada método.

Dataset	WMH-PSO	WMH-GWO	WMH-FFA	WMH-BAT	Wrapper base	Dataset Original
9 tumors	<b>0.92</b> (54)	0.876 (51)	0.896 (32)	0.892 (46)	0.75 (16)	0.691
11 tumors	<b>1.0</b> (118)	0.967 (91)	0.967 (103)	0.983 (123)	0.89 (15)	0.887
14 tumors	0.836 (131)	<b>0.849</b> (144)	0.824 (130)	0.799 (119)	0.763 (58)	0.802
Braintumor1	0.988 (49)	<b>1.0</b> (47)	0.988 (59)	<b>1.0</b> (55)	0.933 (10)	0.806
Braintumor2	<b>1.0</b> (115)	<b>1.0</b> (95)	<b>1.0</b> (83)	<b>1.0</b> (101)	0.98 (5)	0.786
DLBCL	<b>1.0</b> (33)	<b>1.0</b> (49)	<b>1.0</b> (42)	<b>1.0</b> (48)	<b>1.0</b> (4)	0.963
Leukemia1	<b>1.0</b> (42)	<b>1.0</b> (55)	<b>1.0</b> (41)	<b>1.0</b> (53)	<b>1.0</b> (4)	0.937
Leukemia2	<b>1.0</b> (103)	<b>1.0</b> (118)	<b>1.0</b> (108)	<b>1.0</b> (96)	<b>1.0</b> (6)	0.955
Lung cancer	<b>1.0</b> (125)	<b>1.0</b> (118)	<b>1.0</b> (132)	<b>1.0</b> (109)	0.985 (10)	0.962
Prostate tumor	<b>1.0</b> (84)	<b>1.0</b> (97)	<b>1.0</b> (116)	<b>1.0</b> (99)	0.951 (5)	0.941
SRBCT	<b>1.0</b> (18)	<b>1.0</b> (21)	<b>1.0</b> (17)	<b>1.0</b> (15)	<b>1.0</b> (7)	<b>1.0</b>
Promedio	<b>0.977</b>	0.972	0.970	0.970	0.932	0.885

Un método wrapper convencional inicia el subconjunto tomando en cuenta el primer atributo y aplica una búsqueda voraz para los atributos restantes agregándolos uno a uno y obteniendo el resultado de clasificación, posteriormente toma como inicio del subconjunto al segundo atributo solo y aplica una búsqueda voraz para los atributos restantes (del tercero hasta el último); este proceso se repite para el tercero, cuarto, quinto y todos los atributos restantes del dataset, por lo que en total evalúa un total de  $1 + \sum_{k=0}^{p-1} (p - k) = 1 + p(p - 1)/2$  subconjuntos [30].

En primera instancia se presentan los resultados de clasificación sobre el conjunto de datos con todos los atributos, como se ilustra en la Figura 1, donde los mejores

resultados fueron obtenidos por la SVM-Lineal para el 9 Tumors (0.692), 14Tumors (0.802), BrainTumor1 (0.806), DLBCL (0.964), Leukemia1 (0.937), y LungCancer (0.9624); el Naive Bayes obtuvo el mejor resultado para 11Tumors (0.887) y Leukemia2 (0.955); el 3-NN para BrainTumor2 (0.786); AdaBoost para Prostate (0.942); y un empate entre Random Forest, ExtraTrees y SVM-Lineal para SRBCT (1.0).

Para el caso de la propuesta usando PSO (Figura 3a), los mejores resultados fueron obtenidos por el 5-NN para el dataset 9Tumors (0.928), MLP para 14Tumors (0.836), Naive Bayes para BrainTumor1 (0.988); en todos los demás datasets se logró obtener un resultado 1.0 de clasificación: ExtraTrees en 11 Tumors, ExtraTrees y Naive Bayes para BrainTumor2; 3-NN, 5-NN, Random Forest y DecisionTree para LungCancer; Naive Bayes, Random Forest, ExtraTrees y SVM-Lineal para Prostate; y el resto de datasets todos los clasificadores obtuvieron 1.0 con excepción de AdaBoost en SRBCT (0.966).

Cuando se usa GWO (Figura 3b), Random Forest lidera en 9Tumors (0.876) y 11Tumors (0.967), SVM-Lineal en 14Tumors (0.849); en el resto de los datasets al menos tres clasificadores lograron obtener el 1.0 en balanced Accuracy.

Para el caso de FFA (Figura 3c), ExtraTrees obtiene el mejor resultado en 9Tumors (0.897), 11Tumors (0.967) y 14Tumors (0.824); Naive Bayes en BrainTumor1 (0.988), y en el resto de datasets al menos 4 clasificadores obtuvieron 1.0.

Finalmente, cuando se usa WMH-BAT (Figura 3d), Naive Bayes obtiene el mejor resultado para 9Tumors (0.892) y 14Tumors (0.799), Random Forest para 11Tumors (0.984) y BrainTumor1 (1.0); para BrainTumor1 y LungCancer, 3 algoritmos obtuvieron 1.0; y en el resto de los datasets cuando menos 5 clasificadores obtuvieron también la máxima evaluación de balanced Accuracy.

Aunado a lo anterior, también se presenta en la Tabla 2 una comparativa entre los mejores resultados de la propuesta y un método Wrapper tradicional, mostrando los resultados de balanced Accuracy, y la cantidad de atributos seleccionados entre paréntesis.

Hasta este punto es posible ver que los algoritmos reportan resultados competitivos. Sin embargo, en la Tabla 3 se muestra el resultado de la prueba estadística de Friedman con los rankings promedio para los algoritmos que están siendo comparados. Esta prueba, además arroja que los resultados difieren significativamente ( $p = 1.17e - 6$ ) con un 95% de confianza.

En la siguiente tabla se muestran los resultados de tiempo de ejecución para los métodos a comparar. En ella se demuestra que los métodos propuestos son más eficientes que el método wrapper tradicional con búsqueda voraz, con excepción del WMH-FFA cuyos tiempos de ejecución lo superan en todos los datasets exceptuando 11Tumors y 14Tumors.

## 5. Conclusiones

En este manuscrito se presenta un método wrapper metaheurístico que hace uso de cuatro metaheurísticas y diez algoritmos de clasificación sobre once datasets, para establecer una comparativa en cuanto a desempeño de clasificación, número de atributos seleccionados y eficiencia en tiempo de ejecución.

**Tabla 3.** Rankings promedio de la prueba de Friedman.

Dataset	Ranking promedio
WMH-PSO	<b>2.5</b>
WMH-GWO	2.7273
WMH-FFA	2.9091
WMH-BAT	2.8636
Wrapper tradicional	4.4091
Dataset Original	5.5909

**Tabla 4.** Tiempos de ejecución (en segundos) de los métodos de clasificación.

Dataset	WMH-PSO	WMH-GWO	WMH-FFA	WMH-BAT	Wrapper base
9 tumors	<b>49.79</b>	79.61	669.22	50.39	478.00
11 tumors	123.90	225.22	1692.32	<b>107.99</b>	1959.00
14 tumors	182.08	307.73	2070.12	<b>174.80</b>	18:56:21
Braintumor1	<b>71.09</b>	120.44	509.64	72.01	252.00
Braintumor2	98.55	196.11	695.55	<b>81.78</b>	231.00
DLBCL	66.69	112.22	302.53	<b>63.41</b>	102.00
Leukemia1	<b>61.45</b>	102.48	209.06	62.14	85.00
Leukemia2	97.44	178.94	331.53	<b>78.06</b>	278.00
Lung cancer	113.38	213.87	1048.80	<b>89.42</b>	903.00
Prostate tumor	106.64	202.74	424.15	<b>80.28</b>	296.00
SRBCT	<b>44.45</b>	62.63	185.94	57.05	72.00
Promedio	92.31	163.81	739.89	<b>83.39</b>	423.34

De acuerdo con los resultados obtenidos es posible ver que, aunque el método wrapper tradicional selecciona en general una menor cantidad de atributos en todos los conjuntos de datos, el método propuesto lo supera en cuanto el desempeño de clasificación, usando para ello el balanced Accuracy, siendo el método que usa PSO el que mejores resultados ofrece en este rubro.

Aunado a lo anterior, los tiempos de ejecución se disminuyen en gran medida con la propuesta, a excepción de aquel que usa el Firefly Optimization Algorithm cuyos tiempos son mayores; el mejor desempeño en tiempo se obtiene con el algoritmo Bat Optimization y en segundo lugar el Gray Wolf Optimization.

Es posible notar además que, si bien los métodos wrapper comúnmente se asocian a tiempos grandes de procesamiento, en este caso, el Wrapper Metaheurístico, depende totalmente del tipo de metaheurística que se elija. Por lo tanto, se piensa en un futuro mejorar algún algoritmo existente [31], probar en primera instancia con metaheurísticas que sean eficientes, o bien proponer una metaheurística simple que permita seleccionar los mejores atributos en un menor tiempo. Por otro lado, se sugiere también modificar la función fitness agregando una mayor penalización a la cantidad de atributos

seleccionados, o bien, realizar un pre-filtrado más estricto para tener un menor espacio de búsqueda y por lo tanto seleccionar también una menor cantidad de atributos.

## Referencias

1. Chan, W. H., Mohamad, M. S., Deris, S., Zaki, N., Kasim, S., Omatu, S., Al Ashwal, H.: Identification of informative genes and pathways using an improved penalized support vector machine with a weighting scheme. *Computers in biology and medicine*, 77, pp. 102–115 (2016) doi: 10.1016/j.compbiomed.2016.08.004
2. Chandrashekar, G., Sahin, F.: A survey on feature selection methods. *Computers & Electrical Engineering*, vol. 40, no. 1, pp. 16–28 (2014) doi: 10.1016/j.compenleceng.2013.11.024
3. Guyon, I., Weston, J., Barnhill, S., Vapnik, V.: Gene selection for cancer classification using support vector machines. *Machine learning*, vol. 46, no. 1, pp. 389–422 (2002) doi: 10.1023/A:1012487302797
4. Brezočnik, L., Fister, I., Podgorelec, V.: Swarm intelligence algorithms for feature selection: A review. *Applied Sciences*, vol. 8, no. 9, 1521 (2018) doi: 10.3390/app 8091521
5. Blum, A. L., Langley, P.: Selection of relevant features and examples in machine learning. *Artificial intelligence*, vol. 97, no. 1–2, pp. 245–271 (1997) doi: 10.1016/S0004-3702(97)00063-5
6. Pavlenko, T.: On feature selection, curse-of-dimensionality and error probability in discriminant analysis. *Journal of statistical planning and inference*, vol. 115, no. 2, pp. 565–584 (2003) doi: 10.1016/S0378-3758(02)00166-0
7. Bolón-Canedo, V., Sánchez-Marroño, N., Alonso-Betanzos, A.: A review of feature selection methods on synthetic data. *Knowledge and information systems*, vol. 34, no. 3, pp. 483–519 (2013) doi: 10.1007/s10115-012-0487-8
8. Guyon, I., Elisseeff, A.: An introduction to variable and feature selection. *Journal of machine learning research*, pp. 1157–1182 (2003)
9. Hancer, E.: An improved evolutionary wrapper-filter feature selection approach with a new initialisation scheme. *Machine Learning*, pp. 1–24 (2021) doi: 10.1016/j.mls.2021.05.009
10. Clemmensen, L., Hastie, T., Witten, D., Ersbøll, B.: Sparse discriminant analysis. *Technometrics*, vol. 53, no. 4, pp. 406–413 (2011) doi: 10.1198/TECH.2011.08118
11. Tibshirani, R.: Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society: Series B (Methodological)*, vol. 58, no. 1, pp. 267–288 (1996) doi: 10.1111/j.2517-6161.1996.tb02080.x
12. Tay, J. K., Narasimhan, B., Hastie, T.: Elastic net regularization paths for all generalized linear models. *arXiv preprint arXiv:2103.03475* (2021) doi: 10.48550/arXiv.2103.03475
13. Ventura-Molina, E., Alarcón-Paredes, A., Aldape-Pérez, M., Yáñez-Márquez, C., Adolfo Alonso, G.: Gene selection for enhanced classification on microarray data using a weighted k-NN based algorithm. *Intelligent Data Analysis*, vol. 23, no. 1, pp. 241–253 (2019) doi: 10.3233/IDA-173720
14. Tripathi, D., Ramachandra Reddy, B., Padmanabha Reddy, Y. C. A., Shukla, A. K., Kumar, R. K., Sharma, N. K.: BAT algorithm based feature selection: Application in credit scoring. *Journal of Intelligent & Fuzzy Systems*, vol. 41, no. 5, pp. 5561–5570 (2021) doi: 10.3233/JIFS-189876
15. Faris, H., Aljarah, I., Al-Betar, M. A., Mirjalili, S.: Grey wolf optimizer: A review of recent variants and applications. *Neural computing and applications*, vol. 30, no. 2, pp. 413–435 (2018) doi: 10.1007/s00521-017-3272-5

16. Fister, I., Fister Jr, I., Yang, X. S., Brest, J.: A comprehensive review of firefly algorithms. *Swarm and Evolutionary Computation*, vol. 13, pp. 34–46 (2013) doi: 10.1016/j.swevo.2013.06.001
17. Jain, N. K., Nangia, U., Jain, J.: A review of particle swarm optimization. *Journal of The Institution of Engineers (India): Series B*, vol. 99, no. 4, pp. 407–411 (2018) doi: 10.1007/s40031-018-0323-y
18. Kearns, M., Ron, D.: Algorithmic stability and sanity-check bounds for Leave-One-Out cross-validation. *Neural Computation*, vol. 11, no. 6, pp. 1427–1453 (1999) doi: 10.1162/089976699300016304
19. Cancer microarray data sets. Plymouth University (2005) [http://www.tech.plym.ac.uk/spmc/links/bioinformatics/microarray/microarray\\_cancers.html](http://www.tech.plym.ac.uk/spmc/links/bioinformatics/microarray/microarray_cancers.html)
20. Denceux, T.: A k-nearest neighbor classification rule based on Dempster-Shafer theory. *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 25, no. 5, pp. 804–813 (1995)
21. Yang, F. J.: An implementation of Naive Bayes classifier. In: 2018 International Conference on Computational Science and Computational Intelligence CSCI'18, pp. 301–306 (2018) doi: 10.1109/CSCI46756.2018.00065
22. Hush: Classification with neural networks: A performance analysis, In: IEEE 1989 International Conference on Systems Engineering, pp. 277–280 (1989) doi: 10.1109/ICSYSE.1989.48672
23. Pal, M.: Random forest classifier for remote sensing classification. *International Journal of Remote Sensing*, vol. 26, no. 1, pp. 217–222 (2005) doi: 10.1080/01431160412331269698
24. Schapire, R. E.: Explaining AdaBoost. In: Schölkopf, B., Luo, Z., Vovk, V. (eds) *Empirical Inference*. Springer, Berlin, Heidelberg pp. 37–52 (2013) doi: 10.1007/978-3-642-411366\_5
25. Safavian, S. R., Landgrebe, D.: A survey of decision tree classifier methodology. *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 21, no. 3, pp. 660–674 (1991) doi: 10.1109/21.97458
26. Sharaff, A., Gupta, H.: Extra-tree classifier with metaheuristics approach for email classification. Bhatia, S., Tiwari, S., Mishra, K., Trivedi, M. (eds) *Advances in Computer Communication and Computational Sciences. Advances in Intelligent Systems and Computing*, Springer, Singapore vol. 924, pp. 189–197 (2019) doi: 10.1007/978-981-13-6861-5\_17
27. Chauhan, V. K., Dahiya, K. Sharma, A.: Problem formulations and solvers in linear SVM: A review. *Artif Intell Rev* 52, pp. 803–855 (2019) doi:10.1007/s10462-018-9614-6
28. Khurma, R.A., Aljarah, I., Sharieh, A., Mirjalili, S. (2020). EvoloPy-FS: An open-source nature-inspired optimization framework in python for feature selection. In: Mirjalili, S., Faris, H., Aljarah, I. (eds) *Evolutionary Machine Learning Techniques. Algorithms for Intelligent Systems*, Springer, Singapore, pp. 131–173 (2020) doi: 10.1007/978-981-32-9990-0\_8
29. Hossam, F., Aljarah, I., Mirjalili, S., Castillo, P. A., Merelo-Guervós, J. J.: EvoloPy: An open-source nature-inspired optimization framework in python. In: *IJCCI (ECTA)*, pp. 171–177 (2016)
30. Tang, J., Alelyani, S., Liu, H.: Feature selection for classification: A review, *Data Classif, Algorithms Appl*, 37 (2014)
31. Bezdán, T., Cvetnić, D., Gajić, L., Živković, M., Strumberger, I., Bacanin, N.: Feature selection by firefly algorithm with improved initialization strategy. In: 7th Conference on the Engineering of Computer Based Systems, no. 8, pp. 1–8 (2021) doi: 10.1145/3459960.3459974



Electronic edition  
Available online: <http://www.rcs.cic.ipn.mx>



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



Centro de Investigación  
en Computación