ESTRATEGIAS MODERNAS BASADAS EN PROCESAMIENTO DE LENGUAJE NATURAL, APRENDIZAJE AUTOMÁTICO Y REPRESENTACIÓN DEL CONOCIMIENTO EN LA ERA DE LA INFORMACIÓN

Editores:
Mireya Tovar Vidal
María Teresa Torrijos Muñoz
Carlos Leopoldo Carreón Díaz de León
Daniel Marcelo González Arriaga
Jaime Alejandro Romero Sierra
Juan Carlos Conde Ramírez

Estrategias Modernas Basadas en Procesamiento de Lenguaje Natural, Aprendizaje Automático y Representación del Conocimiento en la Era de la Información

Estrategias Modernas Basadas en Procesamiento de Lenguaje Natural, Aprendizaje Automático y Representación del Conocimiento en la Era de la Información

Mireya Tovar Vidal
María Teresa Torrijos Muñoz
Carlos Leopoldo Carreón Díaz de León
Daniel Marcelo González Arriaga
Jaime Alejandro Romero Sierra
Juan Carlos Conde Ramírez
Editores

BENEMÉRITA UNIVERSIDAD AUTÓNOMA DE PUEBLA

Rectora: María Lilia Cedillo Ramírez • Secretario General: Damián Hernández Méndez • Vicerrector de Extensión y Difusión de la Cultura: Luis Antonio Lucio Venegas • Director General de Publicaciones: Mary Kriss Parra Górriz

Primera edición **2025**
ISBN: 978-968-9744-04-7

4 Sur 104, Centro Histórico, Puebla, Pue., CP 72000
Teléfono: 222 229 55 00
www.buap.mx

Dirección General de Publicaciones
2 Norte 1404, Centro Histórico, Puebla, Pue., CP 72000
Teléfono: 222 246 85 59
libros.dgp@correo.buap.mx | www.publicaciones.buap.mx

Diseño de portada: Adobe illustrator

Hecho en México
Made in Mexico

Prólogo

En el presente libro titulado **Estrategias Modernas Basadas en Procesamiento de Lenguaje Natural, Aprendizaje Automático y Representación del Conocimiento en la Era de la Información** se abordan diferentes áreas de investigación en procesamiento de lenguaje natural, ontologías, aprendizaje automático, ciencia de datos, entre otras de la Inteligencia Artificial. La obra incluye doce capítulos de investigación divididos en distintas áreas de conocimiento.

Los capítulos que forman parte de esta obra fueron revisados mediante el sistema de doble par ciego y aprobados para su publicación por expertos en el área de conocimiento a nivel nacional e internacional y fueron escritos por investigadores de distintas instituciones del país. De la misma manera, la obra en su totalidad fue revisada por dos expertos en el área de conocimiento. Cabe mencionar que se recibieron 18 trabajos de los cuales solo se aceptaron aquellos que cumplieron con los requerimientos, el contenido de acuerdo a la obra y con la actualización de las revisiones; obteniendo al final 12 capítulos aprobados.

A continuación se menciona la aportación de cada uno de ellos.

En el **Capítulo 1. Integración de Modelos de Incrustación de Palabras y Aprendizaje Profundo para el Análisis de Sentimientos en Español: Comparativa de Optimizadores** se presenta un enfoque para la clasificación de la polaridad emocional de textos en español, utilizando técnicas de Procesamiento de Lenguaje Natural (PLN) y aprendizaje profundo, se comparan distintos optimizadores aplicados a una red BiLSTM.

En el **Capítulo 2. Evaluación de modelos predictivos con inspiración biológica en Physarum polycephalum** se aborda el analisis del comportamiento del organismo unicelular Physarum polycephalum como modelo bioinspirado para aplicaciones en ciencia de datos y optimización. También se aplicaron técnicas de aprendizaje no supervisado (K-Means) para identificar patrones de agrupamiento y aprendizaje supervisado (regresión logística multinomial) para predecir el tipo de tratamiento. Se destaca el potencial de usar este organismo como modelo bioinspirado para algoritmos adaptativos y en redes logísticas.

En el **Capítulo 3. Análisis de Casos de Influenza en México mediante Algoritmos de Aprendizaje Automático** se ofrece un enfoque de aprendizaje automático supervisado (Random Forest, Multilayer Perceptron y SMO) y no supervisado (KMeans) para analizar más de 187,000 registros clínicos de casos de influenza en México, obtenidos del repositorio SECIHTI. El objetivo es identificar patrones clínicos, demográficos y geográficos para mejorar la

detección de casos confirmados, sospechosos y negativos.

En el **Capítulo 4. Identificación de características relevantes para la Insuficiencia Renal Crónica mediante Técnicas de Aprendizaje Automático** se aborda la importancia de detectar de forma temprana la Enfermedad Renal Crónica (ERC), una condición progresiva y relacionada con comorbilidades como la diabetes y la hipertensión. Para ello, se emplearon datos clínicos públicos con más de 20,000 registros y 43 variables, aplicando técnicas de selección de características como Chi-cuadrada, ANOVA F-score, Información Mutua y Random Forest.

En el **Capítulo 5. Creación de un modelo de incrustación léxica en español de propósito general basado en Word2Vec** se detalla el desarrollo y la evaluación de un modelo de incrustación de palabras, Word2Vec, para el español. La principal contribución del trabajo fue la implementación de un enfoque riguroso de evaluación realizando pruebas de analogías, sinónimos y similitud semántica entre oraciones.

En el **Capítulo 6. Clasificación de enfermedad celíaca mediante modelos de deep learning basados en datos clínicos** se presenta un estudio que aborda un problema de salud relevante y de impacto global, se plantean y comparan tres modelos de redes neuronales (modelo simple, modelo profundo con dropout y un ensamble por concatenación) para la clasificación binaria de la enfermedad celíaca, lo que permite un análisis comparativo sólido.

En el **Capítulo 7. MexaFood: una ontología para el manejo de dietas en el contexto mexicano** se describe una ontología diseñada para representar y gestionar información sobre dietas mexicanas, tomando como base el Sistema Mexicano de Alimentos Equivalentes (SMAE). Su desarrollo responde a la necesidad de contar con herramientas que integren datos nutricionales de manera estructurada, culturalmente pertinente y de bajo costo computacional.

En el **Capítulo 8. Diagnóstico automatizado de enfermedades autoinmunes cutáneas a través de imágenes médicas y aprendizaje profundo** se aborda la clasificación automática de cuatro condiciones cutáneas (liquen plano, psoriasis, vitíligo y piel normal) mediante redes neuronales profundas. Tema muy relevante en el ámbito médico, con impacto clínico y social. Se emplean arquitecturas de vanguardia (DenseNet121 y ResNet50), mostrando resultados relevantes en el estado del arte, lo que aporta solidez al estudio.

En el **Capítulo 9. Expedición Natura: Simulador educativo centrado en un ecosistema endémico del estado de Puebla** se presenta el desarrollo de un simulador educativo tridimensional que recrea un ecosistema endémico del estado de Puebla, centrado en el teporingo que es una especie en peligro de extinción. El simulador utiliza algoritmos de generación procedural con un agente

conversacional educativo que guía al usuario a través de misiones y diálogo. Los resultados muestran un prototipo funcional con coherencia ecológica, visualización 3D y un agente pedagógico.

En el **Capítulo 10. Adquisición de datos para el análisis de la locomoción de Gromphadorhina portentosa bajo distintas condiciones térmicas** se aborda un estudio estadístico que determinó que la temperatura influye de manera no lineal en la locomoción, se propone el desarrollo de biobots y se sugiere futuras líneas de investigación.

En el **Capítulo 11. Jaya y Lobo Gris aplicado a instancias de QAPLIB** se explora el uso de dos metaheurísticas, Jaya y Lobo Gris, para encontrar soluciones al problema de asignación cuadrática (QAP) y compara su desempeño con seis enfoques clásicos: algoritmo genético, recocido simulado, búsqueda tabú, optimización por enjambre de partículas, *fast ant system* y *firefly*. El texto incluye una breve revisión del QAP y de algunos métodos exactos y aproximados empleados en la literatura para abordar este problema.

Por último, en el **Capítulo 12. Detección de Parkinson mediante aprendizaje automático y señales biomédicas** se propone la detección de la enfermedad de Parkinson mediante señales de voz y técnicas de aprendizaje automático. Se implementaron distintos clasificadores: Decision Tree, Random Forest (RF), Bagging-KNN, así como Random Forest con técnicas de balanceo de datos (oversampling y subsampling) y con selección de características (algoritmos genéticos, Fisher's Score y PCA).

Finalmente, queremos agradecer a cada uno de los autores por su valiosa contribución, al comité revisor por su valiosa labor y cuidado en el proceso de revisión ciega, a la Facultad de Ciencias de la Computación de la Benemérita Universidad Autónoma de Puebla y a todos aquellos cuya participación contribuyó para la publicación de este libro.

Los editores,
Mireya Tovar Vidal
María Teresa Torrijos Muñoz
Carlos Leopoldo Carreón Díaz de León
Daniel Marcelo González Arriaga
Jaime Alejandro Romero Sierra
Juan Carlos Conde Ramírez

Índice general

Capítulo 1

Integración de Modelos de Incrustación de Palabras y Aprendizaje Profundo para el Análisis de Sentimientos en español: Comparativa de Optimizadores

Sebastian Guerrero Cangas, José A. Reyes-Ortiz, Josué Padilla Cuevas
Universidad Autónoma Metropolitana, Unidad Azcapotzalco, México
{al2203002842,jaro,jpc }@azc.uam.mx

Resumen. En este trabajo se presenta un enfoque computacional para analizar sentimientos a partir de la polaridad emocional, enfocado exclusivamente en textos en español, especialmente aquellos provenientes de redes sociales. Se utilizaron técnicas de Procesamiento de Lenguaje Natural (PLN), como lematización y tokenización, junto con modelos de aprendizaje automático para clasificar los textos según su polaridad: positiva, negativa o neutra. La propuesta se estructura en cinco fases: recolección de textos etiquetados, limpieza y procesamiento lingüístico, integración de representaciones vectoriales del lenguaje (embeddings), entrenamiento de modelos utilizando distintos algoritmos de optimización, y evaluación del rendimiento mediante métricas cuantitativas. Se emplearon dos conjuntos de datos: PolaridadV2, que contiene más de 250,000 textos clasificados en cinco niveles de polaridad (muy positiva, positiva, neutra, negativa y muy negativa); y tweetsDataset, con 1,451 textos distribuidos en tres categorías: positiva, negativa y neutra. Ambos conjuntos fueron procesados y convertidos mediante el uso de embeddings preentrenados. Posteriormente, se entrenó un modelo BiLSTM (Bidirectional Long Short-Term Memory) utilizando tres optimizadores: Adam, AdamW y SGD, evaluando su desempeño con dos tipos de embeddings: model_v64_w5_c5_senti y fasttext-sbwc. Aunque no se identificó un optimizador claramente superior en todos los casos, se observaron diferencias en cuanto a la precisión y la cantidad de épocas requeridas según el embedding utilizado, destacando el impacto del tipo de representación vectorial en la eficiencia del entrenamiento.

Palabras Clave: Red social, Análisis de sentimientos, Procesamiento de Lenguaje Natural (PLN), Redes neuronales profundas.

1 Introducción

En la actualidad, las redes sociales han provocado cambios estructurales en las formas de comunicación humana debido a sus múltiples aplicaciones. Uno de sus principales usos es facilitar la comunicación entre personas a distancia, sin necesidad de interacción cara a cara. Este tipo de interacción virtual ha producido grandes volúmenes de información en diversos idiomas y sobre múltiples temas.

Este fenómeno ha despertado un interés considerable por las opiniones expresadas en redes sociales, especialmente en ámbitos industriales, científicos y académicos. En el caso específico del idioma español, aún existe una gran cantidad de información no

estructurada en formato textual que requiere análisis, particularmente en el aspecto emocional (polaridad). Dada la magnitud de estos datos, realizar este análisis manualmente resultaría excesivamente laborioso y demandaría un tiempo considerable. Por ello, se propone un enfoque computacional que combine técnicas de PLN y algoritmos de aprendizaje automático para identificar automáticamente el sentimiento presente en los textos.

Este proceso se denomina *análisis de sentimientos* y consiste en clasificar textos de manera automática según su polaridad, es decir, determinar si presentan una connotación positiva, negativa o neutra basándose en su contenido. Para ello, se emplean técnicas de PLN y algoritmos de aprendizaje automático, apoyados en embeddings.

El enfoque computacional permite analizar de manera eficiente textos provenientes de redes sociales en español, lo que posibilita extraer información relevante sobre los sentimientos expresados. Estos resultados pueden aplicarse en diversos sistemas que requieran comprender las emociones de los usuarios.

El presente proyecto propone realizar una clasificación de polaridad a partir de textos recopilados, utilizando técnicas de PLN para extraer características estadísticas a nivel léxico. Dichas características servirán como entrada para algoritmos de aprendizaje automático, optimizando así la clasificación de sentimientos. Además, se llevará a cabo una comparativa entre distintos optimizadores y dos tipos de embeddings, con el fin de determinar cuál ofrece mejores resultados bajo las mismas condiciones de prueba, procesamiento y entrenamiento.

Este artículo abarca la implementación del análisis de sentimientos e incluye la descripción de las siguientes secciones: en la Sección 2, se revisa el estado del arte relacionado con el análisis de sentimientos mediante algoritmos de aprendizaje automático; en la Sección 3, se describe el enfoque propuesto y se detallan las etapas implementadas; en la Sección 4, se presentan los resultados obtenidos y su análisis comparativo; y finalmente, en la Sección 5, se exponen las conclusiones alcanzadas y se plantean posibles líneas de investigación futura.

2 Estado del arte

En esta sección se presenta una revisión del estado del arte en relación con el análisis de sentimientos utilizando textos de redes sociales en diferentes idiomas. Se hace énfasis en los algoritmos de clasificación utilizados y los resultados obtenidos en cada trabajo.

El artículo (Bo Pang, 2002) es uno de los primeros en investigaciones sobre el problema del análisis de sentimientos, quienes propusieron la clasificación de opiniones utilizando reseñas de películas como enfoque de estudio, en donde los autores exploraron 3 métodos de aprendizaje supervisado: Naive Bayes (NB), Maximum Entropy (ME) y Support Vector Machines (SVM), a través de un conjunto de 1,400 reseñas de películas (700 eran positivas y 700 negativas), llegando a utilizar unigramas y bigramas, los cuales no sirvieron mucho para mejorar los resultados. Como resultados principales consiguieron que Naive Bayes (NB) alcanzó una precisión del 81.0 %

usando presencia de unigramas como características. Máxima Entropía (ME) obtuvo una precisión del 80.4 % con el mismo conjunto de características. SVM, que mostró el mejor desempeño, alcanzó un 82.9 % de precisión usando presencia de unigramas.

En el artículo de (Liu, 2010), se hace una pequeña reflexión sobre las problemáticas que hay en el análisis de sentimientos, haciendo énfasis en que, antes de la era digital del internet, había una escasez de textos de opiniones para el estudio computacional. Y aun con el esplendor de las redes sociales, que vienen con miles de reseñas publicadas a diario, Liu hace un realce en que el análisis de sentimientos sigue siendo un problema bastante complejo que involucra problemas como detección de ironía, subjetividad, ambigüedad léxica, entre otros.

El artículo de (Mikolov et al., 2013) introdujo un enfoque efectivo sobre la generación de vectores continuos para la captura de similitudes semánticas y sintácticas entre las palabras. A través de Word2Vec se presentó el análisis en dos tipos de arquitecturas, las cuales son: CBOW (Continuous Bag-of-Words) y Skip-gram. Ambas fueron diseñadas para ser entrenadas por grandes cantidades de texto. Estos modelos se entrenaron con 783 millones de palabras y se evaluaron usando el conjunto Semantic-Syntactic Word Relationship, dando estos resultados: el modelo CBOW con vectores de 300 dimensiones logró una precisión del 15.5 % en tareas semánticas, 53.1 % en sintácticas y un rendimiento total del 36.1 %. Por su parte, el modelo Skip-gram, también con vectores de 300 dimensiones, obtuvo una precisión del 50.0 % en tareas semánticas, 55.9 % en sintácticas y un rendimiento total del 53.3 %, superando ampliamente al CBOW. Finalmente, el modelo de red neuronal de lenguaje (NNLM) con vectores de 100 dimensiones alcanzó una precisión del 34.2 % en tareas semánticas, 64.5 % en sintácticas y un rendimiento total del

50.8 %. El uso de estos embeddings se llegó a utilizar en análisis de sentimientos. En tareas como SemEval y TASS, modelos que integraron Word2Vec con LSTM, CNN o incluso SVM lograron mejores métricas que los métodos clásicos. De hecho, dicho por Mikolov et al., Word2Vec fue capaz de superar a modelos previos como NNLM y RNN en tareas semánticas, ofreciendo una alternativa eficiente.

Durante la pandemia de COVID-19, el análisis de sentimientos se volvió una parte crucial para poder entender las emociones públicas a distancia que se llegaban a publicar en las redes sociales, especialmente en páginas como Twitter, por su gran facilidad de dejar comentarios de todo tipo emocional. Por lo tanto, el artículo de (Samuel et al., 2020) se enfocó en analizar más de 170,000 tweets relacionados con la pandemia para ver su estado emocional de la época. En esos análisis realizados encontraron que el modelo Naive Bayes logró una precisión del 91 % en tweets cortos (77 caracteres); en tanto, el rendimiento se redujo en textos largos (120 caracteres). Con regresión logística se obtuvo un rendimiento del 74 %, pero, como se mencionó, en los textos largos ambos métodos redujeron su eficiencia, pues al ser más largos se volvían más ambiguas las respuestas. Este comportamiento resaltó una de las limitaciones fundamentales de los clasificadores lineales: su escasa capacidad para capturar relaciones semánticas profundas o necesidades contextuales.

En el artículo de (Raheja y Asthana, 2021), como en el anterior, realizaron una clasificación de comentarios de Twitter en el contexto de la pandemia, pero de origen indio, en la cual se hizo una clasificación de categorías de polaridad (positivo, negativo

o neutro), con el apoyo del análisis de emojis para identificar el sentimiento. Para la clasificación de estos sentimientos usaron palabras clave como ("COVID", "coronavirus", "COVID-19"). Predominaban los sentimientos neutros (alrededor del 50 %), seguidos por los positivos y en menor proporción los negativos. Este planteamiento demostró que las personas mantenían una actitud más neutral que de otro tipo. Además, se observó la importancia del uso de técnicas de PLN para datos de redes sociales, generando procesos de limpieza como la eliminación de URLs, menciones, emojis y símbolos, mejorando cada vez más el entendimiento textual.

Mientras que en el artículo (Chintalapudi et al., 2021) se inició un profundo avance en las redes neuronales, ya que utilizaron un modelo de BERT con 12 capas con la capacidad de clasificar emociones de tweets en 4 categorías emocionales: miedo, tristeza, enojo y alegría. Este modelo logró una precisión del 89 %, superando ampliamente a otros modelos más tradicionales como SVM (precisión de 74.75 %), regresión logística (75 %) y LSTM (65 %). Con este gran avance se pudo observar que los modelos basados en transformers para análisis de sentimientos, y con un contexto ideal, tienen una gran ventaja sobre los modelos más antiguos, ya que, como se ha demostrado, su capacidad de relaciones bidireccionales es esencial para entender mejor frases cortas o ambiguas. Esto es clave para lograr el reconocimiento de emociones complejas o incluso mixtas, que no pueden representarse mediante etiquetas binarias.

Y el artículo de (Witanto et al., 2022) presentó en el *JELIKU Journal* un estudio comparativo sobre el uso de dos optimizadores, los cuales son ADAM y RMSprop, aplicándolos a un modelo LSTM para clasificar reseñas de películas como positivas o negativas. En este estudio, los autores limpiaron y procesaron el conjunto de datos, que en este caso estaba en idioma indonesio, y los evaluaron con entropía cruzada y activación Softmax. Los resultados mostraron que: LSTM + Adam alcanzó una precisión de 77.11 %, mientras que LSTM + RMSprop alcanzó una precisión de 80.07 %, superando a Adam.

3 Enfoque propuesto

En esta sección se expone el enfoque propuesto para llevar a cabo el análisis de sentimientos en textos en español, los cuales han sido clasificados previamente según categorías emocionales obtenidas de fuentes bibliográficas. Este enfoque consta de cuatro etapas, representadas en la Figura 1, las cuales se describen inicialmente de forma general y, posteriormente, con mayor detalle.

Figura 1. Esquema del enfoque propuesto

Fase 1: Textos etiquetados. En esta etapa se recopiló información proveniente de diversos conjuntos de textos extraídos de la literatura especializada. A partir de dicha

recopilación, se seleccionaron dos conjuntos de datos denominados *PolaridadV2* (2023) y otro denominado *tweetsDataset*, obtenido de la plataforma Kaggle.

Fase 2: Limpieza y procesado. Ya definidos los conjuntos de textos, se aplicaron técnicas de PLN. Estas tienen el objetivo de normalizar, limpiar y reducir el ruido presente en los datos. En primer lugar, se realizó la normalización Unicode utilizando el esquema NFD, seguida de una conversión a codificación ASCII para eliminar caracteres diacríticos, con lo cual se estandarizan acentos y otros signos gráficos. A la vez, se normalizaron los emojis para poder determinar las emociones. Los textos fueron procesados utilizando lematización y tokenización con la librería *spaCy*, y al final todas las letras del texto fueron convertidas a minúsculas.

Fase 3: Carga de los embeddings. En esta etapa se cargaron distintos modelos de *embeddings* con el objetivo de analizar su comportamiento al aplicarlos sobre los conjuntos de datos y evaluar su eficiencia. Los modelos utilizados fueron: *model_v64_w5_c5_senti* y *fastText-SBWC* (Rodríguez, Espinoza, & Aguilar, 2021).

Fase 4: Aprendizaje automático. Se emplearon tres diferentes optimizadores para el entrenamiento de un modelo BiLSTM, los cuales fueron Adam, AdamW y SGD, con el fin de evaluar su eficiencia bajo el mismo algoritmo. El objetivo cambia dependiendo del texto: el primero fue clasificar los textos en sentimientos positivos, negativos y neutros; mientras que el segundo los clasificó en positivos, muy positivos, negativos, muy negativos y neutros.

Fase 5: Evaluación. En la fase final se llevó a cabo la evaluación de los algoritmos implementados con el propósito de cuantificar su rendimiento, utilizando la métrica de precisión.

A continuación, se presentan las descripciones detalladas de cada fase del enfoque propuesto.

3.1 Primera subsección

El propósito de esta etapa es recopilar dos conjuntos de textos en español provenientes de la literatura, con el fin de emplearlos en la aplicación de algoritmos de aprendizaje automático orientados al análisis de sentimientos.

Se seleccionaron dos conjuntos de textos: *tweetsDataset y PolaridadV2* incluye grupos de mensajes de texto en un solo idioma, que es el español. El conjunto de textos completo consta de 251,702 mensajes, y fue obtenido de (Alexcom, 2023).

A continuación, la Figura 2 presenta una muestra de los datos contenidos en el archivo *PolaridadV2*.

	text	label
0	Un agradable paseo por el casco antiguo de la ciudad Es imprescindible visitar esta parte de la ...	5
1	Gran llegar entrada procedente Gran ciudad en estilo colonial original. las buenas tiendas y res...	4
2	Sencillamente Espectacular Para Caminarlo o en carruaje, disfrutar, su paisaje, recorrer sus edi...	5
3	Monserrate y sus bellas vistas Es una visita obligada si se va a Bogota. Subirlo en funicular o ...	5
4	Un paraiso muy cerca de la ciudad de Mérida El Hotel y el restaurante es digno de visitarlo. El ...	5
5	Hotel Cartagena Plaza Súper recomendado, excelente servicio, instalaciones, el personal es muy a...	5
6	Visita obligada si vas a Riviera Maya Excelente zona arqueológica para visitar, una de las nueva...	5
7	#Paisaje #Espectacular Realizar un tour en el cable, nos lleva a visualizar un paisaje de ensueñ...	5
8	Buen hotel para negocios La ubicacion del hotel al lado de malecon es buena, pero para llegar a ...	4
9	Información poco clara. Lugar agradable Cierran a las 3 pm, cobraron $85 adultos y niños (por lo...	3
10	Escapada en pareja y agradable estadia Excelente servicio, muy amables. Primera vez en cartagena...	5
11	Buena comida, pero ten paciencia Este fue el primer restaurante que visitamos en el Mayan Resort...	2
12	1,034 escalones a 3200 metros de altura . . . Si deseas probar tu estado físico, deberás subir a...	5

Figura 2. Muestra de datos del archivo *PolaridadV2*

Para el conjunto de textos de *PolaridadV2* el total de registros por cada etiqueta para el idioma español se muestra en la Tabla 1.

Tabla 1. Textos por clase del idioma español en el archivo *PolaridadV2*

Clasificación	Total de textos
Muy Positivos	141385
Positivos	54204
Negativos	6257
Muy Negativos	5195
Neutros	19490

El segundo conjunto de textos es tweetsDataset, el cual contiene solo textos en idioma español. Este archivo cuenta con una cantidad mucho menor de textos: 1,451 en total, divididos en tres categorías de polaridad: positivos, neutros y negativos, como se muestra en la Tabla 2. Este conjunto fue obtenido de un concurso en la plataforma Kaggle.

Tabla 2. Textos en cada tipo de *tweetsDataset*

Clasificación	Total de textos
Positivos	264
Neutro	457
Negativo	730

En la Figura 3 se observa una muestra de los textos originales del archivo y cómo este los clasifica. En el lado izquierdo se muestran las categorías "labels", que corresponden a las etiquetas en el mismo orden, cuyos valores son: n para negativo, neu para neutro y p para positivo, los cuales en el procesamiento se convierten en números.

Figura 3. Textos por etiquetas en el conjunto de textos de *tweetsDataset*

	label	text
0	N	No mames este pinche dolor que pedo? ya mejor llévame Diosito.
1	NEU	@leomall2018 Según yo era como aviso, pero ahora sí ya es oficial
2	N	@benshorts a juzgar por mis comportamientos autodestructivos en las relaciones, aún quiero serlo
3	P	#BuenosDias mundo Twittero ya desperté y estoy listo para vivir un dia mas #ExcelenteMartes
4	N	No pude resolver el rompecabezas en Los rios de Alicia y ahora muero de tristeza
5	N	o sea ... me urge un Dr. @Rocktor101 (escuchó un programa de males digestivos y así)
6	NEU	@natyamezcua hay pues hay que ver que hacemos para vernos, te extraño, no quiero que te alejes

Para el conjunto de textos del tweetsDataset el total de registros tomando en cuenta lo mencionado con anterioridad, para manejar con mayor facilidad el archivo, se les asignó un número identificado el tipo de polaridad.

Tabla 3. Textos por etiquetas en el conjunto de textos de *Cardiff*

Clasificación	Tipo	Total de Textos
1	Positivos	264
2	Neutro	457
0	Negativo	730

Este conjunto de textos ajustado facilita una mejor clasificación de la polaridad. Incluye dos columnas: una llamada "texto", que almacena el contenido del mensaje, y otra denominada "etiqueta", que contiene la clase asignada a dicho mensaje.

3.2 Procesamiento de los textos

Esta fase tiene como objetivo preparar los textos para los algoritmos usando el lenguaje python y la cual consiste como se mencionó con anterioridad en la limpieza de los datos, aplicación de las técnicas lematización, usando las librerías spacy y nltk respectivamente, y la el acomodo de los datos para el entendimiento en especial del modelo.

3.2.1 Limpieza

Esta tarea se llevó a cabo con el propósito de preparar los textos para las etapas posteriores del proyecto, con el objetivo de eliminar el ruido y facilitar la comprensión por parte del modelo, eliminando aquellos datos que no contribuyen significativamente a la tarea de clasificación. A continuación, se describen los pasos realizados durante el proceso de limpieza.

- Eliminación de URLs (enlaces que inician con http, https o www).
- Eliminación de guiones aislados al inicio o final de palabras, conservando los guiones internos.
- Eliminación de caracteres especiales, permitiendo letras (con o sin acento), números, espacios y algunos emojis básicos.
- Normalización de palabras con letras repetidas, reduciendo repeticiones mayores a dos letras consecutivas.
- Tokenización del texto: se divide en palabras (tokens) en minúsculas.
- Eliminación de stopwords y palabras innecesarias (palabras muy cortas o sin valor semántico).

La normalización a minúsculas busca la coherencia y uniformidad de los datos, para facilitar su conexión con los embeddings.

Por último, se aplica un método para eliminar palabras vacías que no tienen relevancia dentro del texto, utilizando la biblioteca Stopwords de NLTK. Esta estrategia permite identificar y excluir términos frecuentes como artículos, pronombres y preposiciones, los cuales no contribuyen de manera significativa al análisis del contenido.

Para ambos conjuntos de textos, *PolaridadV2* y *tweetsDataset*, se realiza esta limpieza. La Figura 4 muestra un ejemplo del texto antes del procesamiento de limpieza y la Figura 5 muestra el texto después de la limpieza.

	label	text
0	N	No mames este pinche dolor que pedo? ya mejor llévame Diosito.
1	NEU	@leomall2018 Según yo era como aviso, pero ahora sí ya es oficial
2	N	@benshorts a juzgar por mis comportamientos autodestructivos en las relaciones, aún quiero serlo
3	P	#BuenosDias mundo Twittero ya desperté y estoy listo para vivir un dia mas #ExcelenteMartes
4	N	No pude resolver el rompecabezas en Los rios de Alicia y ahora muero de tristeza
5	N	o sea ... me urge un Dr. @Rocktor101 (escuchó un programa de males digestivos y así)
6	NEU	@natyamezcua hay pues hay que ver que hacemos para vernos, te extraño, no quiero que te alejes

Figura 4. Muestra del conjunto de textos *tweetsDataset* original

N	no mames este pinche dolor que pedo ya mejor llevame diosito
NEU	segun yo era como aviso pero ahora si ya es oficial
N	a juzgar por mis comportamientos autodestructivos en las relaciones aun quiero serlo
P	buenosdias mundo twittero ya desperte y estoy listo para vivir un dia mas excelentemartes
N	no pude resolver el rompecabezas en los rios de alicia y ahora muero de tristeza
N	o sea me urge un dr escucho un programa de males digestivos y asi
NEU	hay pues hay que ver que hacemos para vernos te extrano no quiero que te alejes
N	yo igual y ni asi este ano sere tipo marasalvatrucha para ver si asi me traen algo
N	no te puedo llevar en mi bicicleta me cai y esta descompuesta happy to walk with you tho

Figura 5. Muestra del conjunto de textos *tweetsDataset* limpiados

3.2.2 Lematización

La lematización es un proceso lingüístico que consiste en transformar las palabras a su forma base, considerando su contexto verbal y gramatical. Esta técnica reduce la variabilidad léxica y permite un análisis semántico más profundo de los textos a procesar.

Tras completar la etapa de limpieza, se lleva a cabo el proceso de lematización y tokenización. Para el conjunto de textos tweetsDataset, la Figura 6 muestra el resultado de este proceso, donde cada palabra ha sido transformada a su forma base.

```
Tweet 1:
  Original: No mames este pinche dolor que pedo? ya mejor llévame Diosito.
  Tokenizado: ['mames', 'pinche', 'dolor', 'pedo', 'mejor', 'llevame', 'diosito']
  Índices: [2744, 3453, 1499, 3332, 2831, 2641, 1446]
```

Figura 6: Muestra del conjunto de textos de tweetsDataset después del proceso de lematización

3.3 Carga de embedding

Antes de iniciar la clasificación de emociones con los datasets obtenidos, se cargan los modelos de embeddings que sirven como apoyo en el proceso que llevarán a cabo los

modelos BiLSTM. Para ello, se utiliza un modelo preentrenado en formato binario, específicamente model_v64_w5_c5_senti.bin, el cual ha sido entrenado con vectores distribucionales ajustados al dominio sentimental.

A partir de este modelo, se construye el vocabulario, asignando a cada palabra un índice único para permitir el posterior proceso de padding. Las secuencias resultantes son rellenadas (padded) hasta alcanzar la longitud del texto más largo, generando así una matriz uniforme adecuada para su procesamiento por redes neuronales.

Finalmente, se construye la matriz de embeddings, en la cual cada fila contiene el vector correspondiente a una palabra del vocabulario. Cabe destacar que esta matriz se inicializa con ceros y se completa con los vectores preentrenados del modelo cargado.

3.4 Aprendizaje Automático

Con el objetivo de realizar la clasificación de sentimientos mediante técnicas de PLN, como la lematización, se emplearon distintos optimizadores de aprendizaje automático. Estos modelos fueron implementados utilizando la biblioteca scikit-learn de Python, configurados con los siguientes parámetros:

El optimizador Adam se utilizó con una tasa de aprendizaje inicial (lr=0.001) para mantener una convergencia estable, mientras que AdamW, su variante mejorada, incluyó un decaimiento de peso (weight_decay=1e-2) para lograr un mejor desempeño y mayor generalización del modelo.

Finalmente, se empleó el optimizador SGD con una tasa de aprendizaje de lr=0.01 (debido a su procesamiento más lento), momentum=0.9 para facilitar las actualizaciones, y weight_decay=1e-4.

En todos los casos, se integró un planificador de tasa de aprendizaje (ReduceLROnPlateau), el cual reduce automáticamente la tasa de aprendizaje en un factor de 0.5 si la precisión del modelo no mejora durante tres épocas consecutivas.

El modelo implementado se basa en una red BiLSTM, la cual procesa la información en ambas direcciones de la secuencia para capturar de forma más completa los datos contextuales. La arquitectura incluye una capa de *embeddings* preentrenados, seguida de las capas BiLSTM y, finalmente, una capa de activación *softmax* para la clasificación. Cabe mencionar que la arquitectura no modifica el funcionamiento de los optimizadores, sin embargo, debido a la mayor complejidad y al número de parámetros de la BiLSTM en comparación con modelos más simples, algunos optimizadores, como AdamW, pueden ofrecer una convergencia más estable que SGD.

3.5 Evaluación

Para la evaluación de cada algoritmo es usada la métrica de precisión, con la siguiente fórmula.

$$\text{Precisión} = \frac{TP}{TP + FP}$$

donde TP = Verdaderos Positivos; FP = Falsos Positivos; FN = Falsos Negativos.

4 Resultados experimentales

A continuación, se presenta una comparativa con los resultados obtenidos en el experimento para diferentes algoritmos por clase y épocas que tardó por optimizador. La Tabla 4 y 6 muestra este comparativo para el conjunto de textos tweetsDataset y la Tabla 5 y 7 para el conjunto de textos PolaridadV2, además que cada tabla representa al mejor fold de la validación cruzada de 5. Durante este proceso también se calcularon métricas adicionales como F1-score, precisión y recall, lo cual permitió obtener un análisis más completo del rendimiento de cada modelo. Estos resultados ofrecen una visión más clara de cómo varía el desempeño según el dataset utilizado y el optimizador aplicado.

Tabla 4. Tabla comparativa de precisión del conjunto de textos *tweetsDataset* y embedding model_v64_w5_c5_senti

Método	Negativo	Positivo	Neutro	Precisión	Recall	F1-score	Épocas
ADAM	68%	58%	31%	58%	57%	57%	19
ADAMW	68%	55%	49%	60%	61%	60%	20
SGD	66%	52%	36%	58%	58%	57%	32

Tabla 5. Tabla comparativa de precisión del conjunto de textos PolaridadV2 y embedding model_v64_w5_c5_senti

Método	Muy Negativo	Negativo	Neutro	Positivo	Muy Positivo	Precisión	Recall	F1-score	Épocas
ADAM	43%	28%	36%	37%	78%	62%	61%	61%	68
ADAMW	52%	32%	42%	42%	78%	64%	65%	65%	75
SGD	59%	39%	47%	46%	78%	66%	66%	67%	200

Tabla 6. Tabla comparativa de precisión del conjunto de textos tweetsDataset y embedding fasttext-sbwc.

Método	Negativo	Positivo	Neutro	Precisión	Recall	F1-score	Épocas
ADAM	68%	58%	33%	58%	58%	58%	12
ADAMW	65%	65%	29%	58%	58%	58%	12
SGD	65%	65%	37%	62%	62%	60%	47

Tabla 7. Tabla comparativa de precisión del conjunto de textos PolaridadV2 y embedding fasttext-sbwc

Método	Muy Negativo	Negativo	Neutro	Positivo	Muy Positivo	Precisión	Recall	F1-score	Épocas
ADAM	53%	30%	38%	39%	77%	62%	64%	63%	20
ADAMW	50%	28%	36%	37%	78%	62%	62%	62%	15
SGD	59%	39%	49%	49%	79%	67%	68%	67%	90

Se realizó una comparación entre diferentes métodos de optimización (Adam, AdamW, SGD) para evaluar la precisión en conjuntos de textos usando dos tipos de embeddings: model_v64_w5_c5_senti y fasttext-sbwc, aplicados a dos datasets distintos (tweetsDataset y PolaridadV2). El objetivo fue comparar la precisión obtenida por cada método en las distintas categorías de polaridad y observar diferencias en el rendimiento según el optimizador y embedding utilizado.

En los resultados presentados, se observan variaciones en la precisión para las diferentes clases y métodos, sin embargo, dichas diferencias no muestran una tendencia consistente tan significativa en cuanto a la superioridad de un optimizador sobre otro, ni entre los embeddings probados, sólo respecto a la cantidad de épocas, que en esa sección sí varía bastante.

5 Conclusiones y trabajo a futuro

Este trabajo propuso un modelo BiLSTM para análisis de sentimientos en español, evaluando combinaciones de tres optimizadores (Adam, AdamW, SGD) y dos embeddings (model_v64_w5_c5_senti y fasttext-sbwc) sobre los conjuntos PolaridadV2 y tweetsDataset. Los mejores resultados generales sobre PolaridadV2 se obtuvieron con ambos embeddings: con *model_v64_w5_c5_senti* y el optimizador SGD se alcanzó una precisión del 66 % tras 200 épocas (Tabla 5), acompañado de un recall del 66 % y un F1-score del 67 %, lo que refleja un rendimiento equilibrado entre cobertura y precisión. Sin embargo, con el embedding *fasttext-sbwc*, en ese mismo dataset se logró un 67 % de precisión en solo 90 épocas con SGD (Tabla 7), con un recall del 67 % y un F1-score del 67 %, mostrando un entrenamiento más eficiente por la menor cantidad de épocas requeridas.

En tweetsDataset (Tablas 4 y 6), se obtuvo con el model_v64_w5_c5_senti + AdamW, logrando 61 % de precisión en 20 épocas, acompañado de un recall del 61 % y un F1-score del 60 %. Por su parte, fasttext-sbwc + SGD alcanzó 62 % de precisión en 47 épocas, con recall del 62 % y F1-score del 60 %. Estos resultados evidencian que no existe un optimizador consistentemente superior, más allá de los mejores modelos, las diferencias que hay entre optimizadores no fue mucha, pero sí una marcada influencia del embedding en la estabilidad y velocidad de aprendizaje. Aunque fasttext-sbwc es más compacto y converge en menos épocas, model_v64_w5_c5_senti ofrece mejor rendimiento promedio en tareas de polaridad, probablemente por estar ajustado a ese dominio.

En conjunto, los resultados sugieren que la elección del embedding es más determinante que la del optimizador, y que los embeddings especializados en polaridad semántica mejoran tanto la precisión como la eficiencia del entrenamiento.

Para mejorar los resultados obtenidos, se propone: a) implementar técnicas de n-gramas (bigrams y trigrams) que capturen secuencias de palabras y relaciones contextuales más complejas; b) desarrollar modelos más profundos, como redes neuronales con múltiples capas o modelos basados en transformers, como BERT en español.

Referencias

Bo Pang, Lee, L., & Vaithyanathan, S. (2002). Thumbs up? Sentiment classification using machine learning techniques. En Proceedings of the ACL-02 Conference on Empirical Methods in Natural Language Processing (Vol. 10, pp. 79–86).

Liu, B. (2010). Sentiment analysis: A multi-faceted problem. IEEE Intelligent Systems, 25(4), 76–80.

Mikolov, T., Chen, K., Corrado, G., & Dean, J. (2013). Efficient estimation of word representations in vector space. arXiv preprint arXiv:1301.3781.

Samuel, J., Ali, G. M. N., Rahman, M. M., Esawi, E., & Samuel, Y. (2020). Covid-19 public sentiment insights and machine learning for tweets classification. Information, 11(6), 314. https://doi.org/10.3390/info11060314.

Raheja, S., & Asthana, A. (2021). Sentimental analysis of Twitter comments on COVID-19. En 2021 IEEE 11th International Conference on Cloud Computing, Data Science & Engineering (Confluence) (pp. 704–708). IEEE.

Chintalapudi, N., Battineni, G., & Amenta, F. (2021). Sentimental analysis of COVID-19 tweets using deep learning models. Infectious Disease Reports, 13(2), 329–339. https://doi.org/10.3390/idr13020030.

Witanto, M. R. Hidayat, & Sulistiyono, M. (2022). Comparison of accuracy and time of Naïve Bayes algorithm with Support Vector Machine algorithm in Twitter sentiment analysis. En JELIKU Journal.

Rodríguez, P., Espinoza, A., & Aguilar, S. (2021). Spanish word embeddings [Modelo fastText-SBWC; Repositorio GitHub]. Departamento de Ciencias de la Computación, Universidad de Chile. https://github.com/dccuchile/spanish-word-embeddings.

Alexcom. (2023). analisis-sentimientos-textos-turisitcos-mx-polaridadV2 [Conjunto de datos]. Hugging Face. https://huggingface.co/datasets/alexcom/analisis-sentimientos-textos-turisitcos- mx-polaridadV2.

Kaggle User (s.f.). tweetsDataset [Conjunto de datos]. Kaggle. https://www.kaggle.com/

Capítulo 2

Evaluación de modelos predictivos con inspiración biológica en Physarum polycephalum

Raúl E. Serrano-Gutiérrez, María Teresa Torrijos Muñoz, Jaime A. Romero Sierra, Mireya Tovar Vidal

Facultad de Ciencias de la Computación, Benemérita Universidad Autónoma de Puebla, México

raul.serraog@alumno.buap.mx,teresa.torrijos@correo.buap.mx, jaime.romeros@correo.buap.mx, mireya.tovar@correo.buap.mx

Resumen. El estudio parte de la necesidad de comprender cómo organismos sin sistema nervioso, como Physarum polycephalum, pueden exhibir comportamientos adaptativos ante distintos estímulos del entorno. Para ello, se analizaron datos experimentales previamente recolectados, que registran la respuesta del moho ante cuatro tratamientos químicos (control, ácido, cafeína y quinina). Se seleccionaron variables conductuales clave —contacto, tiempo de cruce, arborización y área cubierta— por su relevancia en la medición de interacción, movimiento y crecimiento del organismo. Se aplicaron técnicas de aprendizaje no supervisado (K-Means) para explorar patrones emergentes y clasificación supervisada (Regresión Logística) para predecir el tratamiento aplicado a partir del comportamiento observado. Los resultados mostraron que las variables de contacto y tiempo de cruce fueron las más determinantes para segmentar y clasificar las respuestas del moho, especialmente ante cafeína y quinina. Sin embargo, se identificaron similitudes entre las clases control y ácido. Estos hallazgos destacan el potencial de Physarum polycephalum como modelo bioinspirado y demuestran cómo métodos de ciencia de datos pueden aportar a la comprensión del aprendizaje en organismos no neuronales. y proponiendo aplicaciones en sistemas de optimización bioinspirados.

Palabras Clave: Physarum polycephalum, ciencia de datos, K-Means, regresión logística, bioinspiración.

1 Introducción

En la actualidad, el estudio de organismos biológicos se basa en la comprensión de sus comportamientos complejos a partir de la suposición de que no cuentan con un sistema nervioso central, esto ha revolucionado nuestra comprensión sobre los orígenes evolutivos tanto de la inteligencia y la adaptabilidad. Uno de los organismos que se estudiará es el *Physarum polycephalum*, este destaca por su capacidad para la resolución de problemas que involucran optimización, aprender de estímulos repetidos y poder adaptarse a entornos cambiantes (Tero et al., 2010; Boisseau et al., 2016). Este organismo unicelular ha sido ampliamente estudiado, porque tiene la habilidad para

navegar en laberintos, diseñar redes eficientes de transporte e incluso por su capacidad de adaptarse antes químicos repelentes como lo son la quinina y la cafeína.
Además de interesarnos biológicamente, una de sus contribuciones es el comportamiento que ofrece ante otros beneficios, el primero de ellos es explorar la hipótesis sobre el origen del aprendizaje en sistemas simples, y también, el diseño de algoritmos bioinspirados, que pueden ser aplicables en áreas de la inteligencia artificial la robótica y la optimización de redes logísticas o de transporte. El *Physarum polycephalum* representa no solo un modelo de estudio para la biología del comportamiento, sino también, es un parteaguas para la innovación tecnológica

El *Physarum polycephalum* a pesar de no tener neuronas o sinapsis, exhibe una conducta que sugiere que existen mecanismos primitivos de memoria y toma de decisiones. Algunas investigaciones, han demostrado que este moho puede adaptarse a habitar en sustancias que inicialmente son repelentes y reducen progresivamente su respuesta a la versión tras exponerse repetidamente (Boisseau et al., 2016).

El *Physarum polycephalum*, en el área de la computación bioinspirada, sirve como parteaguas de un modelo para desarrollar algoritmos de optimización en redes, siendo uno de estos, el diseño de transporte eficiente, que emula su crecimiento adaptativo (Tero et al., 2010). Más, sin embargo, la comprensión cuantitativa de su comportamiento ha desarrollado una gran brecha, ya que bajo diferentes estímulos químicos se comporta diferente, especialmente cuando se toma en cuenta la identificación de patrones reproducibles y la discriminación entre tratamientos.

En este estudio, abordaremos la aplicación de técnicas de ciencia de datos, combinando dos métodos: uno no supervisado que es el K-means y uno supervisado que es la regresión logística, con estos, analizaremos el conjunto de datos derivados de experimentos con el *Physarum polycephalum* (Boisseau et al., 2016). las variables a ocupar son:

- Contacto: Número de interacciones del moho con un puente experimental.
- Tiempo_de_cruce: Duración requerida para atravesar el puente.
- Arborización: Grado de ramificación durante el crecimiento.
- Área: Espacio cubierto por el moho.

Esta investigación aporta al entendimiento del comportamiento adaptativo de *Physarum polycephalum* a través de un enfoque computacional que combina técnicas de aprendizaje no supervisado y supervisado. Por otro lado, se utiliza el algoritmo k-means para analizar los patrones de respuesta del moho ante distintos tratamientos químicos que se utilizaron, esto, con el fin de descubrir cuáles son los comportamientos de acuerdo a las agrupaciones naturales del mismo. Y, por otra parte, se utiliza el modelo de regresión logística, para formular una predicción de cuál es el estímulo recibido a partir de las variables clave que se ocupan. Finalmente, en el estudio se comparan los dos tipos de modelos, tanto el supervisado y no el supervisado, para

valorar qué tan eficiente es la clasificación de cada uno de estos y se destacan fortalezas y limitaciones de cada uno de los modelos planteados.
Este trabajo es relevante, porque en el ámbito biológico presenta una trascendencia, nos ofrece puntos claves para desarrollar sistemas artificiales adaptativos en un futuro, como son: las redes de comunicación o algoritmos de optimización. Además, este contribuye a la discusión de cómo el aprendizaje en sistemas en estructuras neuronales puede ser un campo emergente en la biología y en la ciencia cognitiva.

2 Estado del arte

Se ha dado una revolución por al estudiar el *Physarum polycephalum*, dando un nuevo sentido de comprensión de sobre cómo organismos sin sistema nervioso, exhiben comportamientos de adaptación complejos. En Investigaciones recientes, como la de Le Verge-Serandour y Alim (2024), muestran como este moho unicelular, tiene un proceso de adaptación en su red de tubos vasculares, haciéndolo de manera eficiente en estímulos del entorno, haciendo una forma de "inteligencia material". Su comportamiento nos da a notar que el organismo permite resolver problemas de conectividad, usando principios físicos de retroalimentación local, todo esto lo hace sin recurrir a un sistema nervioso.

Otra investigación relevante es la de Reginato, Proverbio y Giordano (2024) en la cual desarrolló modelos computacionales robustos, los cuales reproducen el comportamiento de forrajeo de *Physarum*, con esto, se muestra cómo las decisiones pueden ser tomadas por reglas biofísicas simples. Estos modelos, permiten simular el proceso de toma de decisiones del moho, a través de diferentes estímulos químicos, aportándonos las herramientas para desarrollar modelos de análisis cuantitativo del aprendizaje.

También, otro de los estudios que mencionan el fenómeno de la habituación es el de Takeda et al. (2025) en el cual, profundiza en la forma en que el moho responde a la repetición de estímulos, a través de varios experimentos observó una reducción progresiva de la respuesta, también observó una recuperación tras una pausa sin exponerse, frente a diferentes compuestos. Estos criterios fueron un parteaguas, para la aceptación de señales de aprendizaje primitivo, esto genera evidencia de que hay que generarse nuevas preguntas de cómo funciona la memoria en un contexto de fundamentos biológicos.

2.1 Optimización Bioinspirada y Diseño de Redes

Estudiando el comportamiento de *Physarum polycephalum* ha generado curiosidad por su capacidad para generar redes eficientes de transporte, esto ha sido una inspiración algoritmos de optimización que pueden ser aplicables a diversos campos. Almeida y Dilão (2023) en su estudio desarrollaron un modelo de adaptación a través de vasos vasculares, tomando parámetros como el flujo, simularon cómo el moho reorganiza sus conexiones, dando una respuesta a estímulos locales, pudiendo resolver laberintos y

replicando sistemas de transporte reales, modelando la red ferroviaria de Portugal. Uno de los avances propuestos por por Uza, Kinjo y Kunita (2025), presentaron un modelo de desarrollo que utiliza una sinergia entre redes urbanas y crecimiento poblacional inspirado en Physarum, logrando mostrar, cómo eliminar caminos redundantes u así, mejorar la eficiencia en entornos urbanos cambiantes. Anexando el estudio hecho por Proverbio y Giordano (2025), lograron introducir el modelo de umbral, el cual demuestra como la resolución de caminos óptimos, generan las reglas de detección de flujo y umbral. Por otro lado, se han diseñado algoritmos optimizadores denominados "Physarum-Energy Optimization" (PEO), que incorporan factores de energía, edad y fluctuaciones aleatorias para abordar problemas clásicos como rutas de costo mínimo y Steiner, mostrando mejores tasas de convergencia y robustez frente a metaheurísticas tradicionales (Arturo et al., 2025). Todos estos desarrollos evidencian que los principios evolutivos y dinámicos de P. polycephalum ofrecen mecanismos útiles para el diseño automático de infraestructuras y sistemas de redes, desde entornos urbanos hasta redes de comunicaciones y robótica descentralizada.

2.2 Técnicas de Ciencia de Datos en Biología

Para analizar de manera cuantitativa el comportamiento de Physarum polycephalum, se elige incorporar métodos de aprendizaje automático de ciencia de datos, esto nos permite modelar su respuesta a través de los estímulos que se tiene. Se eligieron las técnicas más empleadas para analizar agrupamientos, empezando por el aprendizaje no supervisado con K-Means, este modelo se utiliza para identificar patrones en datos sin etiquetar, esto permite explorar agrupaciones naturales analizando su comportamiento (Suyal & Sharma, 2024). Para estos modelos de clasificación, se ha aplicado modelos de aprendizaje supervisado, en específico la Regresión Logística, auxiliándose también el de Random Forest, en el cual se combinan múltiples árboles de decisión, los cuales mejoran la precisión en la predicción de clases, y reduce la variabilidad de los datos (Lee et al., 2024). Además, como herramienta adicional, se incorporó el Análisis de Componentes Principales (PCA), esta técnica permite la reducción de dimensionalidad, esto es muy útil para visualizar y explorar la estructura interna de los datos sin perder información relevante (Martel et al., 2024).Sin dejar a un lado, también se tienen desafíos importantes, uno de ellos es bajar la discriminación entre tratamientos similares (por ejemplo, control vs. ácido), o anexar otras variables como: el pH o la concentración de los compuestos aplicados, esto ayudaría a mejorar significativamente el desempeño de los modelos.

La regresión logística multinomial es una variante de regresión logística normal, esta fue diseñada para variables de respuesta con más de dos categorías, lo que hace, es modelar las probabilidades relativas de cada clase, auxiliándose de funciones logarítmicas que están vinculadas a los predictores. En este contexto los datos biológicos utilizados, se modelaron exitosamente, permitiendo que, junto a técnicas como redes neuronales y selección de características penalizadas, nos dan una predicción de múltiples estados, esto nos da, una buena capacidad de distinción entre clases para conjuntos con alta dimensionalidad (Calviño et al., 2024).

A diferencia de otros trabajos, esta investigación tiene una propuesta diferente la cual integra técnicas de aprendizaje no supervisado y supervisado para el análisis del comportamiento del *Physarum polycephalum*. Tomando primeramente el algoritmo K-Means, este nos permite explorar agrupamientos naturales, en cómo responde el moho ante diferentes parámetros implementados, esto nos da cabida a identificar patrones sin conocimiento previo de las clases. Por otro lado, se implementó la Regresión Logística multinomial para construir el modelo predictivo, que es capaz de clasificar el tratamiento químico aplicado. Con esta combinación nos permite además de descubrir las estructuras en los datos, validar en donde podemos utilizar tareas de clasificación. Además, el presente estudio se destaca por el análisis de las variables Contacto y Tiempo de cruce, estas fueron identificadas como las más relevantes en la predicción. Para validar los resultados se utilizaron el coeficiente de silueta en el agrupamiento y el área bajo la curva ROC (AUC-ROC).

3 Metodología

Este estudio se desarrolló siguiendo un enfoque basado en el ciclo de vida de la ciencia de datos, que integra fases iterativas de exploración, procesamiento, modelado y evaluación para generar conocimiento a partir de datos biológicos obtenidos de experimentos con Physarum polycephalum. A continuación, se describen las etapas implementadas.

En la fase inicial de entendimiento del problema, el objetivo principal fue analizar si Physarum polycephalum responde de forma diferencial a distintos estímulos químicos, que en este caso correspondieron a un control, ácido, cafeína y quinina. Se plantearon dos preguntas clave: por un lado, si existen agrupamientos naturales en el comportamiento del organismo frente a los tratamientos, y por otro, si es posible predecir el tratamiento aplicado con base en las variables observadas.

Para la recolectar y preparar los datos, se utilizó una base previamente publicada por Boisseau et al. (2016), esta contiene mediciones del comportamiento del moho ante los diferentes tratamientos químicos antes mencionados. Las variables que fueron consideradas son: el tiempo de contacto con el estímulo, el tiempo que tarda en cruzar un puente, el grado de ramificación (arborisation) y el área ocupada por el organismo. En esta etapa se realizó el proceso de limpieza y transformación de los datos, en el cual como pasos relevantes fueron: la eliminación de registros incompletos, el proceso de estandarización de valores acoplándolos a máximos y mínimos, el proceso de convertir variables categóricas a numéricas para facilitar la clasificación. Además, se realizó una exploración visual con diagramas de dispersión y boxplots para identificar tendencias y posibles valores atípicos.

En cuanto al proceso de modelación, se aplicó dos enfoques de aprendizaje automático: uno no supervisado y otro supervisado, las bondades por los que fueron escogidos son: su robustez, interpretabilidad y adaptabilidad a conjuntos de datos biológicos. En el

caso de K-Means, se logró de identificar agrupamientos naturales en las observaciones sin considerar las características del tratamiento. Este método permitió en el presente trabajo la exploración de patrones en los datos, asignando por características semejantes un cluster cuyo centroide está más cercano a la semejanza de los demás datos. Se estableció un número de clusters igual a cuatro, correspondiente a los cuatro tratamientos experimentales. Para evaluar el desempeño de este modelo, se auxilió del coeficiente de silueta, que mide la compactación dentro de cada cluster y la separación entre clusters.

A su vez, por la parte de aprendizaje supervisado se implementó una regresión logística multinomial, siendo esta la técnica más adecuada para problemas de clasificación con múltiples variables de control. En este modelo se utilizó la función softmax, la cual controla la penalización por complejidad, evitando el sobreajuste de los datos y la variabilidad del modelo. La evaluación del desempeño se realizó mediante varias métricas. Para este tipo de aprendizaje el más utilizado es el F1-score, el cual pondera simultáneamente la precisión y la exhaustividad del modelo, tomando en cuenta el desequilibrio entre clases, y genera una medida global de la calidad de la clasificación. La matriz de confusión fue un recurso utilizado en este estudio, la cual, ofrece un análisis de los aciertos y errores cometidos en cada clase, permitiendo la identificación de patrones de confusión entre tratamientos. Además, se utilizaron las curvas ROC, las cuales nos muestran la relación entre los verdaderos positivos y la de falsos positivos, permitiendo en el estudio valorar la capacidad de distinción del modelo de manera individual para cada categoría.

La evaluación y validación para ambos modelos, se llevó a cabo mediante procedimientos específicos para cada tipo de aprendizaje. En el caso de K-Means, se usaron visualizaciones de clusters a través de análisis de componentes principales (PCA) y la valoración del coeficiente de silueta. La regresión logística se validó con un conjunto de prueba mediante un enfoque hold-out, y se aplicó validación cruzada estratificada para asegurar la estabilidad de los resultados en las métricas multicategoría.

Para evaluar el modelo de K-means ocupamos el coeficiente de silueta calculado con la fórmula:

$$s(i) = \frac{b(i) - a(i)}{max\{a(i), b(i)\}}$$

Donde $a(i)$ representa la distancia promedio entre el punto i y todos los demás puntos del mismo clúster (es decir, la cohesión intra-cluster), mientras que $b(i)$ corresponde a la distancia promedio entre el punto i y todos los puntos del clúster más cercano al que no pertenece (lo que mide la separación inter-cluster). El valor de $s(i)$ varía entre -1 y 1: cuando se aproxima a 1, indica que el punto está bien asignado a su clúster; si el valor es cercano a 0, sugiere que el punto se encuentra en el límite entre dos clústeres; y si es menor que 0, señala que probablemente el punto está mal agrupado.

4 Resultados

Los resultados del análisis se organizan en dos secciones: la de descubrimientos a partir del análisis no supervisado con K-Means y la de desempeño de la clasificación supervisada con Regresión Logística. Ambos enfoques permitieron identificar patrones conductuales relevantes en Physarum polycephalum ante distintos tratamientos químicos. Este organismo fue expuesto a cuatro tratamientos experimentales: Control (C), Ácido (CA), Cafeína (CC) y Quinina (Q), y se registraron diversas variables cuantitativas relacionadas con su comportamiento. La estructura del conjunto de datos involucra 3662 observaciones distribuidas de la siguiente forma:

Tabla 1Datos del dataset Physarum polycephalum

Variable	***Descripción***
Treatment	Tipo de tratamiento aplicado: C, CA, CC, Q
Day	Día del experimento (1 a 9)
Bridge	Sustancia presente en el puente experimental: A (Ácido), Q (Quinina), CAF
Contact	Número de contactos del moho con el puente
Crossing_time	Tiempo (en segundos) que el moho tarda en cruzar el puente
Arborisation	Grado de ramificación del moho (valor entre 0 y 1)
Area	Área cubierta por el moho en unidades arbitrarias

Previo al modelado, se llevaron a cabo diversas etapas para asegurar la calidad del conjunto de datos. En primer lugar, se realizó una limpieza de valores atípicos en variables como Contact y Crossing_time, utilizando el método de rango intercuartílico (IQR), con el fin de eliminar registros extremos que pudieran sesgar el análisis. Posteriormente, se aplicó la técnica de normalización min-max scaling para escalar todas las variables numéricas al rango [0, 1], lo cual es fundamental para algoritmos sensibles a la escala, como K-Means. También se transformaron las variables categóricas, como Treatment y Bridge, a formato numérico mediante codificación categórica, lo que permitió su inclusión en los modelos de aprendizaje automático. Finalmente, se utilizó Análisis de Componentes Principales (PCA) como herramienta exploratoria para visualizar posibles agrupamientos y detectar redundancia entre variables.

Visualizando la estructura de la base de datos (Fig. 1) se observan datos atípicos que podrían viciar el estudio, debido a la dispersión y repetición de los mismos, después de aplicar la limpieza y el procesamiento de los datos (Fig. 2) se obtiene una base de datos más certera, de la cual reduce la dispersión de los datos para el entrenamiento de nuestros modelos de machine learning.

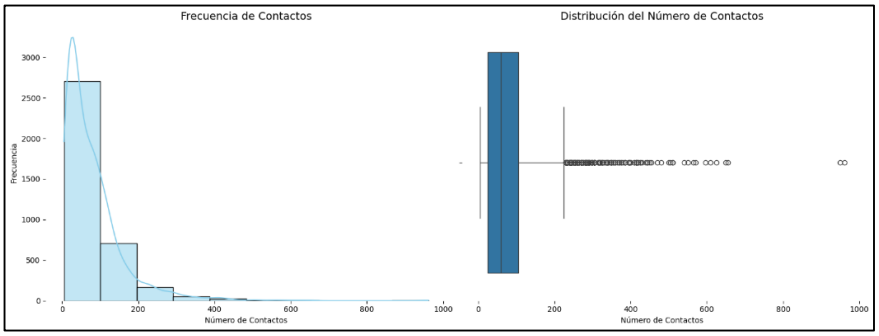

Figura 1 Distribución de los datos sin limpieza.

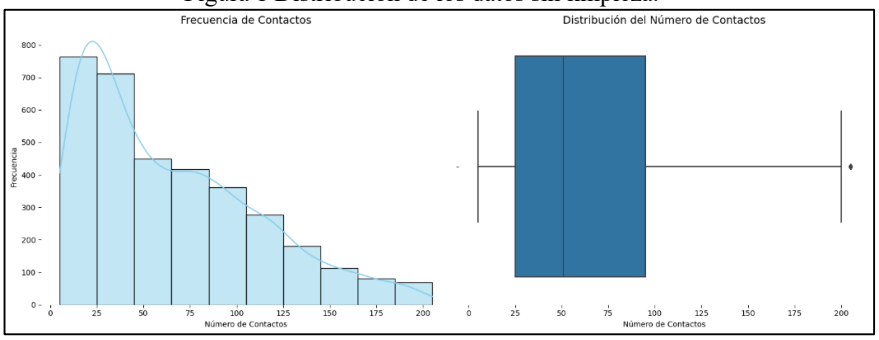

Figura 2 Distribución de los datos con limpieza

Se ejecutó el algoritmo K-Means con k=4, correspondiente a los cuatro tratamientos aplicados (Control, Ácido, Cafeína y Quinina). El valor del coeficiente de silueta fue 0.49, lo cual indica una separación moderada entre los grupos formados, pero se puede agrupar con clusters (Fig. 3) que expresen las características agrupadas en función del tratamiento realizado.

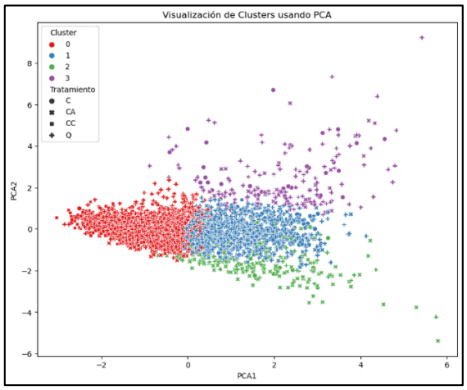

Figura 3 Clusters de K-Means proyectados con PCA.

El modelo de Regresión Logística multinomial fue entrenado usando las variables: Contact, Crossing_time, Arborisation y Area. Para evaluarlo en la tabla 2 se muestran los resultados obtenidos, por cada tratamiento y de forma global.

Tabla 2 Métricas de clasificación por tratamiento.

TRATAMIENTO	PRECISION	RECALL	F_1-SCORE	ACCURACY
C (CONTROL)	0.67	0.65	0.66	0.65
CA (ÁCIDO)	0.61	0.58	0.59	0.58
CC (CAFEÍNA)	0.79	0.75	0.77	0.75
Q (QUININA)	0.89	0.84	0.86	0.84
GLOBAL	0.74	0.71	0.72	0.71

Al aplicar el modelo de Regresión Logística multinomial (Fig. 2), con todas las variables, se encuentra un modelo que da en porcentajes de acuerdo a las características mostradas, la probabilidad de pertenecer a la clase deseada.

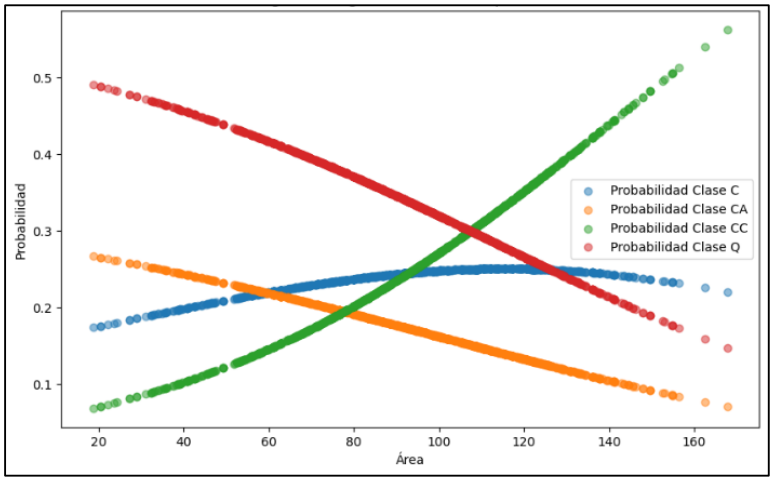

Figura 2 Gráfico de modelo ya entrenado de regresión multimodal

Se generó la matriz de confusión y las curvas ROC multiclase (Fig. 3) para visualizar el rendimiento del modelo, donde, La regresión logística multinomial logró predecir con mayor precisión los tratamientos con cafeína (CC) y quinina (Q), como se observa en la matriz de confusión. Sin embargo, hubo una alta confusión entre los tratamientos de control (C) y ácido (CA), lo que sugiere respuestas similares del moho a estos estímulos. Las curvas ROC confirman esta tendencia, con mayores AUC para Q (0.76), CC (0.75) y CA (0.72), y menor discriminación para C (0.58). En general, el modelo muestra un desempeño aceptable (AUC promedio = 0.73), destacando el valor predictivo de variables como contacto y tiempo de cruce.

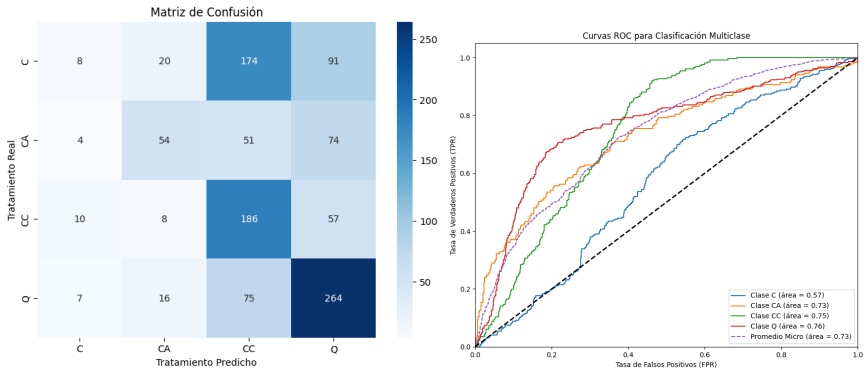

Figura 3. Matriz de confusión y Curvas ROC multiclase.

El análisis computacional del comportamiento de Physarum polycephalum frente a distintos tratamientos químicos reveló patrones que permiten caracterizar su respuesta adaptativa. A partir de la aplicación de algoritmos de agrupamiento y clasificación, se obtuvieron los siguientes hallazgos clave:

Segmentación parcial de comportamientos

Mediante el algoritmo K-Means se logró una segmentación aceptable del conjunto de datos (coeficiente de silueta = 0.49), con agrupaciones bien definidas para los tratamientos con Cafeína y Quinina, pero con traslapes considerables entre los grupos de Control y Ácido. Esto sugiere que, desde el punto de vista de las variables observadas, las respuestas del moho a estos dos últimos estímulos son similares o poco diferenciables.

Eficacia predictiva en tratamientos altamente aversivos

El modelo de Regresión Logística logró altos niveles de exactitud para predecir tratamientos con compuestos aversivos, en particular Quinina (F_1-score = 0.86) y Cafeína (F_1-score = 0.77). En cambio, los tratamientos considerados más neutros (Control y Ácido) resultaron más difíciles de clasificar, lo que concuerda con el comportamiento observado en los clusters.

Este resultado indica que el moho genera respuestas más consistentes y extremas ante estímulos fuertemente repelentes, como los que contienen quinina, lo cual facilita su detección mediante modelos supervisados.

Contacto y tiempo de cruce como indicadores críticos

Las variables que mostraron mayor relevancia en ambos modelos fueron el número de contactos y el tiempo de cruce, ya que se encuentran directamente asociadas al nivel de interacción del Physarum polycephalum con el estímulo químico. Se observó que tratamientos con quinina o cafeína se caracterizaron por un alto número de contactos y tiempos de cruce más prolongados, lo cual indica una mayor exploración o resistencia a cruzar. En contraste, los grupos control presentaron generalmente un menor número de contactos y tiempos de cruce más cortos, lo que sugiere una interacción más directa

o menor percepción de amenaza. A través de estos patrones, se permite pensar que ambas variables son relevantes para como indicadores de habituación o repulsión, siendo estos elementos, la clave para modelar respuestas en organismos sin sistema nervioso.

Confirmación del valor de enfoques bioinspirados
Estos resultados, confirman que el uso de técnicas computacionales, nos permite modelar y predecir el comportamiento de Physarum polycephalum eficientemente. Con la capacidad del moho como modelo bioinspirado, nos permite el desarrollo de algoritmos adaptativos, para contextos de la vida real como la optimización de rutas, toma de decisiones o gestión dinámica de recursos.

5 Conclusiones

En este estudio se demostró como cuantitativamente que el hongo exhibe respuestas diferenciales y predecibles del comportamiento que hemos visto. Mediante el modelo k-means se segregaron exitosamente los comportamientos en los clusteres (quinina y cafeína), y exploración neutra (control y ácido). Además, con un modelo de regresión logística se validó que las variables tiempo de cruce y número de contactos, son predictores que estadísticamente nos dan resultados significativos en la reacción del organismo. Con estos hallazgos confirmamos que incluso con la ausencia de un sistema nervioso existen mecanismos que procesan la información y permiten una toma de decisiones consistentes y medibles.

El principal aporte encontrado en este trabajo es que se pudo validar una metodología que es computacionalmente eficiente y con alta interpretabilidad. que como se puede observar a diferencia de otros modelos más complejos vistos en la literatura como redes autómatas celulares o con un aprendizaje por refuerzo, el enfoque manejado establece una línea que puede ser la base para la clasificación del comportamiento inmediato. Si bien estos modelos más robustos exploran la optimización a largo plazo, nuestro estudio identifica y cuantifica las variables clave conductuales, con esto se abre un antecedente para sentar las bases del análisis, que puede ser más sofisticado y puede servir como un punto de validación.

Ahora se plantea que puede haber más trabajo a futuro, siendo este, que se puede escalar la complejidad de este análisis predictivo. Uno de los siguientes pasos podría ser la implementación de modelos de aprendizaje automático más avanzados, como los SVM que son máquinas de soporte vectorial o incluso una red neuronal, para no solo encerrarnos en un modelo de clasificación, sino de tener la capacidad de predecir las trayectorias del organismo. Por la parte experimental se explorarían escenarios que lleven a la decisión de sistemas más complejos, tomando en cuenta las variables como gradientes de concentración o la elección entre múltiples estímulos simultáneos, esto nos permitiría modelar de una forma más precisa en función del costo beneficio.

Referencias

Almeida, R., & Dilão, R. (2023). Formation and optimisation of vein networks in Physarum. arXiv preprint. https://doi.org/10.48550/arXiv.2305.12244

Arturo, X., et al. (2025). An artificial intelligence technique: Experimental analysis of the Physarum-Energy optimization algorithm. Computational Intelligence and Neuroscience. https://doi.org/10.1007/s44163-025-00367-w

Boisseau, R. P., Vogel, D., & Dussutour, A. (2016). Habituation in non-neural organisms: Evidence from slime moulds. Proceedings of the Royal Society B: Biological Sciences, 283(1829), 20160446. https://doi.org/10.1098/rspb.2016.0446

Calviño, A., Moreno-Ribera, A., & Pineda, S. (2024). Machine learning applied to omics data. arXiv. https://arxiv.org/abs/2402.05543

Lee, J., Kim, K., & Lee, K. (2024). Multi-sensor image classification using the Random Forest algorithm in Google Earth Engine with KOMPSAT-3/5 and CAS500-1 images. Remote Sensing, 16(24), 4622. https://doi.org/10.3390/rs16244622

Le Verge-Serandour, M., & Alim, K. (2024). Physarum polycephalum: Smart network adaptation. Annual Review of Condensed Matter Physics, 15, 263–289. https://doi.org/10.1146/annurev-conmatphys-040821-115312

Martel, E., Lazcano, R., López, J., Madroñal, D., Salvador, R., López, S., Juárez, E., Guerra, R., Sanz, C., & Sarmiento, R. (2024). Implementation of the Principal Component Analysis onto high-performance computer facilities for hyperspectral dimensionality reduction: Results and comparisons. arXiv preprint. https://doi.org/10.48550/arXiv.2403.18321

Proverbio, D., & Giordano, G. (2025). Threshold sensing yields optimal path formation in Physarum polycephalum. arXiv preprint. https://doi.org/10.48550/arXiv.2507.12347

Reginato, D., Proverbio, D., & Giordano, G. (2025). Bottom-up robust modeling for the foraging behavior of Physarum polycephalum. Journal of the Royal Society Interface, 21, 20240701. https://doi.org/10.1098/rsif.2024.0701

Suyal, M., & Sharma, S. (2024). A review on analysis of K-Means clustering machine learning algorithm based on unsupervised learning. Journal of Artificial Intelligence and Systems, 6(1), 85–95. https://doi.org/10.33969/AIS.2024060106

Takeda, Y., et al. (2025). Threshold sensing yields optimal path formation in Physarum polycephalum. arXiv preprint. https://doi.org/10.48550/arXiv.2507.12347

Tero, A., Takagi, S., Saigusa, T., Ito, K., Bebber, D. P., Fricker, M. D., Yumiki, K., Kobayashi, R., & Nakagaki, T. (2010). Rules for biologically inspired adaptive network design. Science, 327(5964), 439–442. https://doi.org/10.1126/science.1177894

Uza, M., Kinjo, A., & Kunita, I. (2025). Synergistic development model of population growth and infrastructure networks based on the slime mold network. Artificial Life and Robotics. https://doi.org/10.1007/s10015-025-01035-z

Capítulo 3

Análisis de Casos de Influenza en México mediante Algoritmos de Aprendizaje Automático

Hansel Lázaro Valdés Pérez, Néstor David García Franco, Karen Esmeralda Delgado Pérez, Gabriela Giovana Pérez López, María Josefa Somodevilla García

Facultad de Ciencias de la Computación, Benemérita Universidad Autónoma de Puebla, México

vp224470264@alm.buap.mx, gf224470255@alm.buap.mx, dp224470254@alm.buap.mx, pl224470262@alm.buap.mx, maria.somodevilla@correo.buap.mx

Resumen. La influenza es una amenaza constante para la salud pública en México. Este estudio aplicó algoritmos de aprendizaje automático supervisado y no supervisado para identificar patrones clínicos y demográficos en casos de influenza. Se utilizó un conjunto de datos proporcionado por SECIHTI, que incluyó más de 187,000 registros clínicos, los cuales fueron sometidos a un proceso de preprocesamiento. En el análisis supervisado se emplearon los algoritmos de *Random Forest*, *Multilayer Perceptron* y *Sequential Minimal Optimization*, priorizando la métrica de *recall*. El modelo *MLP* obtuvo el mejor desempeño (*recall* del 72.1%), mientras que *SMO* mostró deficiencias por el desbalance de clases. En el análisis no supervisado, *KMeans* identificó tres clústeres con perfiles clínicos coherentes. Al compararse con estudios recientes, los modelos muestran un desempeño competitivo, aunque se destaca la necesidad de mejorar la calidad de los datos para afinar la predicción de casos confirmados.

Palabras Clave: Aprendizaje Automático, Influenza, México.

1 Introducción

Actualmente, las enfermedades respiratorias afectan de manera significativa a una gran parte de la población a nivel mundial, alrededor de 3,9 millones de personas mueren anualmente por infecciones respiratorias agudas; entre estas, las enfermedades pulmonares son la principal causa de muerte. González Albán, D. S. (2023). La influenza es una enfermedad respiratoria que puede aumentar la incidencia de neumonía y causar un alto número de hospitalizaciones. En marzo de 2009, México, Estados Unidos y Canadá acapararon la atención internacional cuando el virus de la influenza AH1N1 irrumpió en el panorama epidemiológico. Comité Nacional de Enfermedades (2025). Márquez, E., Barrón-Palma, E. V., y Rodríguez, K. (2023). LaRussa, P. (2011).

En México la Secretaría de Ciencia, Humanidades, Tecnología e Innovación (SECIHTI), ha recopilado información de los casos de influenza en México, por lo que, es un repositorio fundamental para obtener información valiosa de la enfermedad. Esto permite emplear técnicas basadas en minería de datos y aprendizaje automático donde, de manera automatizada, se extraen conocimientos para la toma de decisiones.

El objetivo de la investigación es identificar patrones clínicos, demográficos y geográficos relevantes en pacientes con sospecha o confirmación de influenza mediante la aplicación de algoritmos de clasificación supervisada y técnicas de agrupamiento no supervisado.

Las secciones restantes del presente artículo se organizan de la siguiente manera: la sección 2 aborda los trabajos relacionados con la investigación, la descripción del conjunto de datos se presenta en la 3° sección; la sección 4 desglosa la metodología a seguir en el trabajo, mientras que la 5° sección muestra el análisis de los resultados; finalizando con las conclusiones y trabajos futuros en la sección 6.

2 Estado del Arte

La influenza es una enfermedad respiratoria estacional con elevada morbilidad y mortalidad, cuya detección oportuna es clave para mitigar sus impactos. El diagnóstico temprano en servicios de urgencias puede mejorar significativamente el tratamiento de la influenza. En un estudio internacional (EE.UU. y Taiwán), se compararon siete algoritmos de *Machine Learning* (*ML*) basados en síntomas clínicos de pacientes con datos de síntomas tipo influenza (*ILI*). El modelo *XGBoost* logró la mejor precisión diagnóstica (AUC = 0.82, sensibilidad = 0.92, especificidad = 0.89), superando incluso a los modelos clínicos convencionales. Variables como fiebre, tos, rinorrea y semana epidemiológica fueron altamente predictivas, lo cual, valida el uso de estos algoritmos en entornos hospitalarios, Hung, S.-K., et al. (2023). Otro estudio desarrollado entre 2012 y 2019 propuso el modelo *Multiattention Long Short-Term Memory* (*MAL*), el cual combina *ILI*, vigilancia virológica, clima, demografía y consultas en motores de búsqueda. Los resultados mostraron un desempeño sobresaliente ($R^2 = 0.76$, MSE = 0.01), superando modelos tradicionales como *Random Forest* y *XGBoost*. Esta investigación destaca el valor de fuentes digitales en tiempo real para la anticipación de brotes epidémicos en megaciudades, Yang, L., et al. (2023).

Po otro lado, Borkenhagen, L. K., et al. (2021) realizan una revisión sistemática de 49 estudios abordó el uso de *ML* para predecir características fenotípicas del virus de influenza tipo A, a partir de datos genómicos y proteómicos. Aunque los modelos evaluados, como redes neuronales y máquinas de vectores soporte, han mostrado buenos resultados, su aplicación práctica ha sido limitada. Las principales barreras incluyen el diseño sesgado de los modelos y la falta de validación en laboratorio (*wet lab*). Se concluye que es necesario fortalecer la conexión entre bioinformática y virología experimental para potenciar su uso en la vigilancia viral. Un estudio distinto empleó únicamente secuencias de hemaglutinina procesadas mediante matrices de puntuación y técnicas de *word embedding*. El modelo más eficaz fue el *5-grams-transformer*, el cual alcanzó métricas de rendimiento superiores (AUC-PR ≈ 99.5%, F1

≈ 98%). Este enfoque permite detectar rápidamente si una cepa tiene origen humano o animal, lo que podría mejorar las estrategias de contención ante emergencias sanitarias, Xu, Y., y Wojtczak, D. (2024).

En México, se evaluaron modelos de clasificación supervisada utilizando registros clínicos de más de 15,000 pacientes evaluados por RT-qPCR en la Ciudad de México entre 2010 y 2020. El conjunto de datos consistió en 24 atributos que abarcaban información clínica y demográfica recopilada de los pacientes a su llegada a las instituciones de atención médica para el examen clínico y antes de la toma de muestra (exudado nasofaríngeo y/o orofaríngeo) para una prueba de RT-qPCR. Los datos excluyeron los nombres, domicilios y números de registro hospitalario de los pacientes. Los algoritmos *Random Forest* y *Bagging* mostraron un alto desempeño en la clasificación de casos positivos y negativos, Márquez, E., et al. (2023).

El presente trabajo se sustenta de los resultados de estudios previos, que muestran un uso prometedor de los algoritmos de aprendizaje automático en el diagnóstico de la influenza, para introducirlos al contexto mexicano.

3 Conjunto de datos de estudio

El conjunto de datos fue obtenido de la página oficial de SECIHTI, con fecha de actualización 21 de enero de 2025. Estos se encuentran almacenados en Excel y cuentan con 42 atributos (*fecha actualización, id registro, origen, sector, entidad_um, sexo, entidad nac, entidad_res, municipio_res, tipo_paciente, fecha_ingreso, fecha_síntomas, fecha_def, intubado, neumonía, edad, nacionalidad, embarazo, habla_lengua_indígena, indígena, diabetes, epoc, asma, inmunosupr, hipertensión, otra_com, cardiovascular, obesidad, renal crónica, tabaquismo, otro_caso, toma_muestra_lab, resultado_pcr, resultado_pcr_coinfección, toma_muestra_antígeno, resultado_antígeno, clasificación_final_covid, clasificación_final_flu, migrante, país_nacionalidad, país_origen, uci*); para un total de 187,431 registros hasta la fecha.

4 Metodología

El presente estudio se fundamenta en el análisis de un conjunto de datos, cuyo contenido son registros clínicos anónimos relacionados con influenza, estructurados mediante claves numéricas que representan categorías institucionales, demográficas y clínicas.

Con el propósito de garantizar una mayor interpretación y facilitar el análisis estadístico y descriptivo de la información, se realizó un proceso de transformación de datos, utilizando los catálogos de referencia también proporcionados por SECIHTI. Dichos catálogos contienen la relación exacta entre los códigos numéricos presentes en los registros (por ejemplo, en las variables *sector, entidad_um o tipo_paciente*) y sus correspondientes descripciones textuales (por ejemplo, “IMSS”, “SSA”, “Hospitalizado”).

4.1 Preprocesamiento de los datos

Para tener calidad en el análisis y reducir las dimensiones del conjunto de datos, se llevó a cabo un proceso de selección de atributos mediante un enfoque basado en criterios con información relevante. Durante la etapa de preprocesamiento, se eliminaron aquellas variables que no aportaban valor significativo al análisis o que presentaban redundancia, valores constantes o falta de relación directa con el enfoque del estudio (por ejemplo, *nacionalidad, origen* o *migrante*).

Posteriormente, se seleccionaron aquellos atributos relacionados con comorbilidades clínicas tales como *neumonía, intubado, EPOC, asma, diabetes, hipertensión, inmusupr, otra_com, cardiovascular, obesidad, renal_cronica, tabaquismo, otro_caso*. También se incluyeron variables como *edad, sexo, tipo_paciente, entidad, embarazo* y *sector*. De la misma forma, la clasificación final del caso de influenza (*clasificacion_final_flu*) se tomó como variable de clase. Después, se aplicaron los siguientes filtros al conjunto de datos:

Discretización: Se definieron seis grupos etarios con el fin de facilitar la interpretación de los resultados y mejorar el rendimiento de los modelos de clasificación mediante categorías discretas. Esta segmentación permite capturar variabilidad significativa sin generar sobreajuste, y refleja etapas clave del ser humano, como infancia, adolescencia y adultez.

Remover instancias: Se decidió eliminar los registros correspondientes a la Ciudad de México dentro de *entidad_res*, ya que eran significativamente mayores que los del resto de entidades federativas. Esta decisión implica una pérdida parcial de datos, por lo que, en trabajos futuros, se considerará la aplicación de técnicas de muestreo que permitan conservar la información de la Ciudad de México sin comprometer el balance entre los estados.

Después, se seleccionó un subconjunto de atributos relevantes con base en su información y su asociación con los objetivos del estudio. El conjunto final de variables seleccionadas son *sexo, edad, tipo_paciente, neumonía, intubado, entidad_res, sector*, y la variable de clase, *clasificacion_final_flu*, para un total de 151,718 registros, los cuales se consideran representativos de las características clínicas, demográficas e institucionales más influyentes en la clasificación de los casos.

Los resultados en la Tabla 1 muestran un balance entre hombres y mujeres, así como cantidades similares en los grupos de edad. Respecto al tipo de paciente se observa un número más elevado de hospitalizados que de ambulatorios. Finalmente, el sector estatal reporta un menor número de casos atendidos, siendo el IMSS la institución que recibe el mayor número de pacientes. Este conjunto final fue utilizado como entrada para la aplicación de distintos algoritmos de aprendizaje no supervisado y aprendizaje supervisado, cuyos resultados se analizan en la siguiente sección.

Tabla 1. Caracterización de la muestra.

Categoría	Cantidad	Categoría	Cantidad
Sexo		Tipo de Paciente	
MUJER	86 344	HOSPITALIZADO	59 191
HOMBRE	65 374	AMBULATORIO	92 527
Edad (discretizada)		Neumonía	
0 - 7	24 842	NO	121 642
ago-25	26 509	SÍ	30 076
26 - 34	23 971	Clasificación Final Influenza	
35 - 47	25 078	CASO DE INFLUENZA CONFIRMADO	11 075
48 - 62	25 625	NEGATIVO A INFLUENZA	78 819
63 y más	25 693	CASO SOSPECHOSO	61 824

4.2 Aprendizaje Automático No Supervisado

Los algoritmos de aprendizaje no supervisado analizan y agrupan conjunto de datos no etiquetados utilizando variables de entrada. Estos algoritmos descubren patrones ocultos o agrupamientos de datos sin intervención humana, identificando patrones que describan el comportamiento general de los datos, Monero, F., Montilla, N., y Arcilla, J. (2023). Dentro de este tipo de algoritmos se destaca *KMeans*.

KMeans. Se aplica el método no jerárquico *KMeans* que particiona los individuos, en este caso pacientes, en un número específico de grupos. Este algoritmo ha mantenido relevancia en aplicaciones de agrupamiento debido a su buen rendimiento y competitividad con enfoques sugeridos más recientemente, Kotsiantis, S. B. (2022). En este estudio se tomó como hipótesis k = 3, para los casos confirmados, sospechosos y negativos, y encontrar relaciones entre los resultados de la clasificación de influenza con el resto de las variables bajo estudio.

4.3 Aprendizaje Automático Supervisado

Para este estudio, se utilizaron los siguientes algoritmos: *Random Forest*, *Multilayer Perceptron* y *Sequential Minimal Optimization*.

Random Forest. Este modelo es útil para identificar relaciones generales entre múltiples variables clínicas y demográficas. Principalmente ayuda a detectar los casos sospechosos de influenza, lo que resulta valioso para un primer aprendizaje en situaciones clínicas, evitando el sobreajuste del algoritmo en escenarios donde se desea una evaluación rápida y confiable de los posibles casos, aunque su debilidad para identificar casos confirmados indica que se requiere complementar su uso con otros métodos.

Multilayer Perceptron. Sirve para modelar relaciones no lineales y complejas entre las variables del conjunto de datos. En este caso, es especialmente útil porque logra distinguir con mayor precisión los tres tipos de clasificación clínica: confirmados, sospechosos y negativos, lo que nos permite contar con una herramienta más precisa y robusta para encontrar patrones valiosos. Además, puede ayudar a anticipar con mayor exactitud aquellos pacientes que tienen un mayor riesgo de contraer influenza, con factores como la presencia de neumonía o el grupo etario al que pertenece.

Sequential Minimal Optimization. Este algoritmo permite identificar perfiles clínicos que tienden a pertenecer a los grupos negativos o sospechosos. Aunque no logra detectar los casos confirmados, su utilidad radica en reducir falsos negativos dentro de las clases más abundantes.

4.4 Evaluación de Modelos

En este estudio, los modelos se evaluaron mediante una validación cruzada de 10 *folds*, además, se optó por priorizar la métrica de *recall* sobre el *F-Measure* debido a la importancia crítica de identificar todos los casos positivos, garantizando que los casos relevantes sean detectados, aun si esto implica una ligera disminución en la precisión del modelo. También, se hace uso de *ROC Area*; esto indica que el modelo no solo predice correctamente la mayoría de los casos, sino que también mantiene un buen equilibrio entre los verdaderos positivos y los falsos positivos, siendo fundamental en la predicción de los casos.

5 Resultados

En esta sección se presentan los resultados de los experimentos utilizando los algoritmos supervisados y no supervisados.

5.1 Agrupamiento

En primera instancia, se aplicó el algoritmo de agrupamiento *SimpleKMeans* con 3 clústeres, correspondientes a los 3 casos identificados por la prueba de influenza, destacándose patrones relevantes en los datos. El primer clúster, representado por el 38% de los datos, informa que los pacientes hombres, menores de 7 años,

hospitalizados, sin neumonía, no intubados, del estado de Puebla y atendidos en IMSS, presentan una mayor calificación de casos negativos de influenza. Con un 39% de las instancias en el segundo clúster, se tienen individuos hombres, en una edad de 8 a 25 años, como tipo de paciente ambulatorio, sin neumonía, residentes en Puebla y asistidos por el sector IMSS, tienen mayor probabilidad de ser un caso sospechoso. Para el tercer clúster, con un 23% de los datos, se encuentran mujeres, en un rango de edad de 35 a 47 años, paciente ambulatorio, sin neumonía, que viven en Querétaro y recibieron atención médica por SSA, presentan un perfil predominante de casos negativos.

Se observa alta cohesión y mínimo acoplamiento en los clústeres generados debido a que los datos dentro de cada clúster son muy similares entre sí, mientras que se pueden observar diferencias significativas entre los diferentes clústeres.

5.2 Clasificación

A partir de esta información y con conocimientos del conjunto de datos y las relaciones entre variables, se entrenaron modelos de clasificación para evaluar la consistencia de los patrones encontrados. Es importante señalar que, en los resultados obtenidos por los algoritmos, el sexo masculino fue el más representativo.

El modelo *Multilayer Perceptron* (*MLP*) refuerza los patrones observados: Los pacientes que se encuentran en los rangos de edad de entre 26 a 34 años y de 35 a 47 años, presentan neumonía, no estuvieron intubados, residen en Yucatán, Veracruz o Campeche, y fueron atendidos en los sectores de salud DIF o SSA, tienen una alta probabilidad de ser considerados casos sospechosos. En cambio, cuando el individuo se localiza en el rango de 26 a 34 años, presentó neumonía, fue intubado, vive en los estados de Puebla, San Luis Potosí o Jalisco, y recibió atención médica por parte del IMSS o Estatal, tiene una mayor probabilidad de ser un caso confirmado. Por el contrario, una persona que se ubica en los rangos de edad de 0 a 7 años o entre 48 a 62 años, sin neumonía, no fue hospitalizado, radica en estados como Guerrero o Veracruz, y fue atendido en los servicios de salud ISSSTE o PEMEX, la probabilidad de que sea un caso negativo es alta.

El modelo *Random Forest* obtiene un buen desempeño al identificar casos sospechosos alcanzando un *recall* de 92.6%; para los casos negativos se presenta un *recall* limitado del 65.3%, sin embargo, el problema principal se visualiza en los casos confirmados, debido a que apenas logra identificar un *recall* de 0.3%. En conjunto, se obtuvo un *recall* de 71.7%, lo que nos indica un fuerte desbalance de clases o una carencia de información para reconocer correctamente los casos. Como consecuencia, muchos casos confirmados terminan siendo clasificados erróneamente como sospechosos o negativos.

En la Figura 1 se observa la variable neumonía, lo que señala su importancia clínica en el diagnóstico. Si un paciente presenta neumonía y se encuentra en el grupo etario de 26 a 34 años, el modelo suele clasificarlo como caso sospechoso. Sin embargo, los casos confirmados son escasos en las ramas del árbol, lo que refleja la dificultad del modelo para distinguirlos.

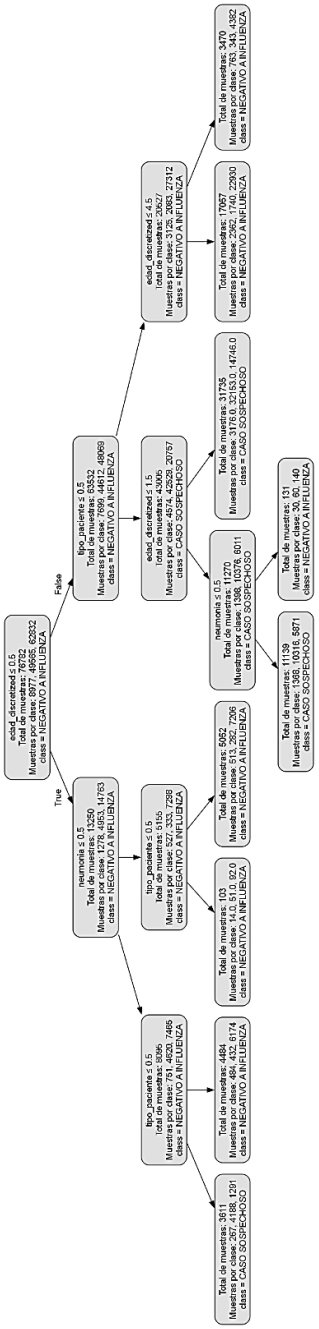

Figura 1. Árbol de resultados obtenido por *Random Forest*.

Finalmente, el algoritmo *Sequential Minimal Optimization* (*SMO*) también identificó patrones consistentes. Si el individuo fue hospitalizado en una institución de salud privada, se encuentra en el grupo etario de 35 a 47 años, y proviene de los estados Campeche, Oaxaca, Tamaulipas o Sinaloa, es altamente probable que sea clasificado como caso sospechoso. Ahora bien, cuando la persona no requirió hospitalización, fue atendido por sectores de salud públicas como IMSS, SSA o ISSSTE, pertenece a los grupos de edad de entre 8 a 25 años o de 26 a 34 años, y reside en las entidades de Coahuila, Sonora, Jalisco o Baja California, tiende a clasificarse como caso negativo. Sin embargo, el modelo no pudo hacer predicciones útiles para los casos confirmados, debido a que tiene un *recall* desconocido para esa clase. La figura 2 expone la comparación de resultados en los algoritmos de clasificación.

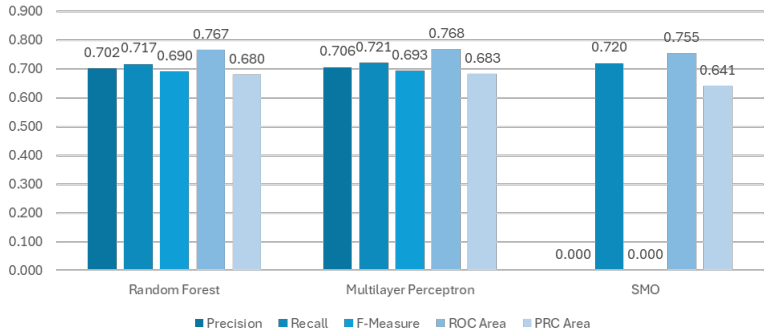

Figura 2. Comparación de métricas entre los algoritmos de clasificación.

De acuerdo con los resultados, *Random Forest* muestra un desempeño balanceado. Tiene una alta capacidad de detección de positivos reales (*recall*), sin sacrificar demasiada *precision*. Su *ROC Area* (0.767) indica un buen rendimiento general en distintos umbrales de decisión. *MLP* es el modelo con mejor desempeño global en esta comparación. Tiene la mayor *ROC Area* (0.768) y la mejor combinación de *precision* y *recall*, lo que se refleja en el mayor *F-Measure* (0.693). Su rendimiento es muy similar al de *Random Forest*, pero ligeramente superior en métricas clave. *SMO* presenta un comportamiento inusual; aunque el *recall* es alto (0.720) y la *ROC Area* es competitiva (0.755), *precision* y *F-Measure* son ?, lo que sugiere que el modelo clasifica todos los casos como positivos, generando muchos falsos positivos. Esto se debe a un problema de desbalance en las clases, ya que el modelo está sesgado hacia la clase negativo a influenza. La tabla 2 ofrece un resumen los resultados previamente mencionados.

Tabla 2. Ejemplo de presentación de resultados.

Algoritmo / Atributos		Edad	Neumonía	Intubado	Estado	Sector Salud
Random Forest	**Confirmados**	N/A				
	Sospechosos	26 a 34 años; 35 a 47 años	NO	NO	Querétaro	SSA
	Negativos	0 a 7 años; 8 a 25 años	NO	NO	Puebla	IMSS
MLP	**Confirmados**	26 a 34 años	SÍ	SÍ	Puebla, San Luis Potosí, Jalisco	IMSS, Estatal
	Sospechosos	26 a 34 años; 35 a 47 años	SÍ	NO	Yucatán, Veracruz, Campeche	DIF, SSA
	Negativos	0 a 7 años; 48 a 62 años	NO	NO	Guerrero, Veracruz	ISSSTE, Pemex
SMO	**Confirmados**	N/A				
	Sospechosos	35 a 47 años	NO	NO	Campeche, Oaxaca, Tamaulipas, Sinaloa	Privada
	Negativos	8 a 25 años; 26 a 34 años	NO	NO	Coahuila, Sonora, Jalisco, Baja California	IMSS, SSA, ISSSTE

6 Conclusiones

En esta investigación se utilizó un conjunto de datos proveniente del repositorio oficial de SECIHTI, con más de 187,000 registros clínicos relacionados con casos de influenza en México. Tras un proceso de preprocesamiento, se depuraron los datos eliminando atributos irrelevantes, discretizando variables como la edad y excluyendo instancias sesgadas, como aquellas pertenecientes a la Ciudad de México. Finalmente, se seleccionaron atributos representativos de tipo clínico demográfico, lo que dio lugar a un subconjunto final de 151,718 registros. Se aplicaron algoritmos de aprendizaje supervisados y no supervisados utilizando métricas de evaluación como *precision*, *recall*, *F-Measure*, *ROC Area* y *PRC Area*. Se priorizó el uso de *recall* debido a la necesidad de maximizar la detección de casos positivos de influenza. Esta decisión se fundamenta en la importancia crítica de evitar falsos negativos en el contexto clínico y epidemiológico.

Los resultados revelan que el algoritmo *Multilayer Perceptron* presentó el mejor desempeño global, con una combinación equilibrada de *recall* (72.1%) y las otras métricas, seguido de *Random Forest* (*recall* general del 71.7%). Por su parte, *SMO* mostró limitaciones considerables, especialmente en la clasificación de casos confirmados, lo que evidencia un problema de desbalance de clases.

Al comparar estos hallazgos con estudios recientes del estado del arte, como el trabajo de Hung et al. (2023), donde *XGBoost* logró un *recall* del 92% y AUC de 0.82, se confirma que los modelos basados en aprendizaje automático tienen un gran potencial en el diagnóstico de enfermedades respiratorias. No obstante, en nuestro caso, el desempeño en la clase de casos confirmados fue limitado, lo cual puede atribuirse a una menor representación de esta clase en los datos o a la falta de variables clínicas más específicas.

En resumen, el estudio demuestra que el aprendizaje automático puede detectar con eficacia patrones relevantes en datos clínicos reales, aunque se requiere seguir fortaleciendo la calidad de los datos y considerar técnicas adicionales de balanceo o recolección de variables para mejorar la predicción en clases críticas como los casos confirmados de influenza.

7 Agradecimientos

Hansel Lázaro Valdés Pérez, Néstor David García Franco, Karen Esmeralda Delgado Pérez y Gabriela Giovana Pérez López, estudiantes de la Maestría en Ciencias de la Computación de la Benemérita Universidad Autónoma de Puebla, agradecen al SECIHTI por el apoyo de Becas de Posgrados Nacionales otorgado.

Referencias

González Albán, D. S. (2023). Asistente médico virtual orientado al diagnóstico de enfermedades respiratorias aplicando inteligencia artificial.

Comité Nacional de Enfermedades (2025). Manual de atención a la salud ante emergencias, Recuperado de: https://epidemiologia.salud.gob.mx/gobmx/salud/documentos/manuales/

Márquez, E., Barrón-Palma, E. V., y Rodríguez, K. (2023). "Supervised machine learning methods for seasonal influenza diagnosis", *Diagnostics*, vol. 13(21), pp. 3352. Recuperado de: https://www.mdpi.com/2075-4418/13/21/3352

LaRussa, P. (2011). "Pandemic novel 2009 H1N1 influenza: what have we learned?", *Seminars in Respiratory and Critical Care Medicine*, vol. 32, pp. 393–399.

Gobierno de México. (2022). "Informe Semanal de Vigilancia Epidemiológica", Secretaría de Salud, Recuperado de: https://www.gob.mx/cms/uploads/attachment/file/737555/INFLUENZA_OVR_SE26_2022.pdf

Consejo Nacional de Ciencia y Tecnología (CONACYT). (2023). Datos COVID-19. Recuperado de: https://datos.covid-19.conacyt.mx/

Yang, L., et al. (2023). “Deep-learning model for influenza prediction from multisource heterogeneous data in a megacity: Model development and evaluation”, Journal of Medical Internet Research, vol. 25, pp. e44238. Recuperado de: https://www.jmir.org/2023/1/e4238

Borkenhagen, L. K., et al. (2021). “Influenza virus genotype to phenotype predictions through machine learning: A systematic review”, *Emerging Microbes & Infections*, vol. 10(1), pp. 1896–1908. Recuperado de: https://www.tandfonline.com/doi/full/10.1080/22221751.2021.1978824

Xu, Y., y Wojtczak, D. (2024). Dive into machine learning algorithms for influenza virus host prediction with hemagglutinin sequences. Recuperado de: https://arxiv.org/abs/2207.13842

Hung, S.-K., et al. (2023). “Developing and validating clinical features-based machine learning algorithms to predict influenza infection in influenza-like illness patients”, *Biomedical Journal*, vol. 46, pp. 100561. Recuperado de: https://www.sciencedirect.com/science/article/pii/S2319417022001005

Marquez, E., et al. (2023). “Supervised machine learning methods for seasonal influenza diagnosis”, *Diagnostics*, vol. 13, pp. 3352. Recuperado de: https://www.mdpi.com/2075-4418/13/21/3352

Kotsiantis, S. B. (2022). “An overview of machine learning classification techniques”, *IET Software,* vol. 16(6), pp. 518–526. Recuperado de: https://ietresearch.onlinelibrary.wiley.com/doi/full/10.1049/sfw2.12032

Ali, A., y Mashwani, W. K. (2023). “A Supervised Machine Learning Algorithms: Applications, Challenges, and Recommendations”, *Pakistani Journal of Applied and Social Psychology*, vol. 4(2), pp. 1–13. Recuperado de: https://www.ppaspk.org/index.php/PPAS-A/article/view/1121/708

Monero, F., Montilla, N., y Arcilla, J. (2023). “Algoritmos de aprendizaje automático en la predicción del rendimiento académico universitario: una revisión sistemática”, *Revista MásTIC,* vol. 6(2), pp. 99–111. Recuperado de: https://www.revistas.up.ac.pa/index.php/mas_tic/article/view/6361/5318

Platt, J. (1998). “Sequential Minimal Optimization: A Fast Algorithm for Training Support Vector Machines”. *Microsoft Research*. Recuperado de: https://www.microsoft.com/en-us/research/wp-content/uploads/1998/04/sequential-minimal-optimization.pdf

Scikit-learn Developers. (2024). “Random forests and other randomized tree ensembles”, Scikit-learn Documentation. Recuperado de: https://scikit-learn.org/stable/modules/ensemble.html#random-forests-and-other-randomized-tree-ensembles

Scikit-learn Developers. (2024). “Neural network models (supervised)”, Scikit-learn Documentation. Recuperado de: https://scikit--learn-org.translate.goog/stable/modules/neural_networks_supervised.html?_x_tr_sl=en&_x_tr_tl=es&_x_tr_hl=es&_x_tr_pto=wa

Capítulo 4

Identificación de características relevantes para la Insuficiencia Renal Crónica mediante Técnicas de Aprendizaje Automático

Alan Marcial Marín, Mireya Tovar Vidal, María Teresa Torrijos Muñoz
Facultad de Ciencias de la Computación, Benemérita Universidad Autónoma de Puebla, México
alan.marcial@alumno.buap.mx, mireya.tovar@correo.buap.mx, teresa.torrijos@correo.buap.mx

Resumen. La enfermedad renal crónica (ERC) que ha tenido un avance notable en los últimos años. Es una condición muy difícil de tratar, sobre todo en sus últimas etapas y se vincula notablemente con la diabetes tipo 2 y la hipertensión, condiciones que aceleran el daño renal y comprometen gravemente la calidad de vida. Este contexto justifica la necesidad de tener nuevas estrategias en detección. Para dar respuesta a esta problemática, se llevó a cabo un análisis exploratorio de datos clínicos, en el cual se examinaron más de 40 variables disponibles en un repositorio público. Con la ayuda de diversas técnicas de selección de características inherentes del aprendizaje automático —entre ellas Chi-cuadrada, ANOVA F-score, Información Mutua y Random Forest— fue posible reconocer cuáles eran los factores con mayor influencia en la progresión de la ERC. Los hallazgos destacan la creatinina sérica, la tasa de filtración glomerular estimada (eGFR) y la hemoglobina como biomarcadores esenciales. Priorizar estos indicadores puede ser de gran ayuda para la detección temprana de la ERC y para que los médicos tomen decisiones basadas en evidencia.

Palabras Clave: Selección de características, insuficiencia renal, análisis de datos clínicos, detección temprana, salud pública.

1 Introducción

La enfermedad renal crónica (ERC) es una enfermedad progresiva que afecta a millones de personas a nivel mundial, misma que deteriora de forma gradual la función renal y poco a poco el resto de los órganos. Se asocia con la diabetes tipo 2 y la hipertensión arterial como principales desencadenantes, lo cual la convierte en una amenaza silenciosa que requiere atención inmediata, especialmente en poblaciones vulnerables. Para poder enfrentar este desafío, el modelado y la simulación con aprendizaje automático han sido herramientas muy útiles en la predicción y el diagnóstico a tiempo. En este estudio se plantea la construcción de un modelo predictivo para la ERC, desarrollado a partir de datos abiertos y algoritmos de clasificación capaces de reconocer patrones clínicos relevantes. La principal

contribución radica en combinar herramientas computacionales con información médica real, lo que favorece una detección más temprana de riesgos y respalda con mayor solidez la toma de decisiones en el ámbito clínico.

El documento está organizado de la siguiente manera: en la **Sección 2** se describe el *marco teórico* relacionado con la enfermedad renal crónica y las técnicas de aprendizaje automático que se usaron. La **Sección 3** muestra una revisión del *estado del arte* de modelos que han sido aplicados a la enfermedad. En la **Sección 4** se muestra la *propuesta de solución*, incluyendo el conjunto de datos y la metodología. La **Sección 5** expone los *resultados obtenidos* y su análisis. Finalmente, la **Sección 6** alberga las *conclusiones* del estudio y plantea posibles líneas de trabajo futuro.

2 Marco Teórico

Actualmente se han realizado análisis de enfermedades crónicas utilizando aprendizaje automático y esto ha ganado mucha importancia, especialmente en la salud pública. Debido a que la enfermedad renal crónica (ERC) es una afección que avanza en silencio, se necesitan herramientas de análisis que nos ayuden a identificar de forma eficaz los factores que más influyen en su desarrollo. En este apartado, presentamos los fundamentos teóricos para comprender y replicar la metodología de selección de características utilizada en nuestro estudio.

2.1 **Enfermedad Renal Crónica (ERC)**

Cuando una persona padece ERC, sus riñones van perdiendo su capacidad para filtrar toxinas, regular líquidos y mantener el equilibrio del organismo. Es por eso por lo que el diagnóstico clínico se basa en una baja tasa de filtración glomerular (TFG) y en la aparición de signos como fatiga, edema, hipertensión y trastornos metabólicos. La diabetes mellitus, la hipertensión arterial, enfermedades hereditarias y un estilo de vida inadecuado son algunos de los factores de riesgo más relevantes. Identificar estos factores es clave para poder implementar estrategias de intervención y evitar que la enfermedad progrese.

2.2 **Aprendizaje automático en medicina**

El aprendizaje automático es una de las ramas de la inteligencia artificial, misma que se centra en la creación de crear modelos que sean capaces de aprender de grandes cantidades de datos. Dentro de un contexto médico, es muy importante, ya que no solo puede predecir, sino que también revela relaciones ocultas entre variables clínicas,

también puede identificar indicadores importantes, además de generar conocimiento que ayude durante la toma de decisiones. Este trabajo utiliza técnicas de selección de características para identificar las variables que tienen mayor relevancia en la detección y avance de la ERC.

Algunas de las técnicas utilizadas para este fin son:

- *Chi-cuadrado*: evalúa la dependencia estadística entre variables categóricas y la clase objetivo.
- *ANOVA F-score* mide la variabilidad entre clases contra la variabilidad interna, lo cual es muy útil para variables numéricas.
- *Información mutua*: captura relaciones lineales y no lineales entre características y la variable objetivo.
- *Random Forest (RF)*: calcula la importancia de cada variable a partir de su contribución a la reducción de impureza en árboles de decisión (**Aljaaf et al., 2021; Debal & Sitote, 2022**).

3 Estado del arte

La predicción temprana de la Enfermedad Renal Crónica *(ERC)* mediante técnicas de aprendizaje automático ha sido objeto de numerosos estudios en los últimos años. A continuación, se presentan algunos de los trabajos más relevantes:

Uno de los trabajos con más precisión es el de **Taznin et al. (2023)** quien utiliza árboles de decisión sobre un conjunto de datos de *ERC* con 24 atributos, alcanzando una precisión del 99% tras seleccionar 15 características, sin reportar limitaciones específicas.

En un ámbito más específico, **Salekin et al. (2022)** aplicaron *Random Forest (RF), K-Nearest Neighbors (KNN)* y *Artificial Neural Networks (ANN)* a un conjunto de 400 instancias, logrando un 98 % de precisión con *RF*. Utilizaron cinco variables seleccionadas mediante técnicas wrapper, aunque el tamaño limitado de muestra podría haber favorecido el sobreajuste.

Por otra parte, **Xiao et al. (2021)** evaluaron múltiples modelos, incluyendo *regresión logística (LR), Support Vector Machines (SVM)* y *Extreme Gradient Boosting (XGBoost)*, sobre datos de 551 pacientes con proteinuria. La *LR* obtuvo la mejor *Area Under the Curve (AUC = 0.873)*, destacando que modelos simples pueden superar a los complejos en ciertos escenarios.

De forma similar, **Emon et al. (2022)** probaron varios algoritmos como *RF, SVM* y *ANN* sobre un conjunto de datos de *ERC*, obteniendo una precisión del 99% con *RF*, aunque sin detallar las variables ni limitaciones.

Como uno de los trabajos más eficientes, **Moreno-Sánchez (2021)** empleó *XGBoost* en un conjunto de datos de *ERC,* logrando una precisión del 99.2 % mediante validación cruzada utilizando únicamente tres variables clave: células empaquetadas, gravedad específica e hipertensión; no obstante, se advierte la posible falta de generalización a otras poblaciones.

En este mismo año, **Haque et al. (2025)** emplearon *Categorical Boosting (CatBoost)*, *RF*, *Multilayer Perceptron (MLP)* y *LR* sobre datos clínicos reales. *CatBoost* alcanzó una precisión del 98.75 % y un *AUC* de 0.9993, aunque su complejidad podría dificultar su uso en entornos con recursos limitados.

Retomando estrategias previamente exploradas, **Halder et al. (2024)** realizaron un estudio con diversos algoritmos como *RF, Adaptive Boosting (AdaBoost)* y *SVM*, alcanzando una precisión del 100 % en diferentes validaciones. Aunque no se detallan las variables utilizadas, se señala la ausencia de medidas de explicabilidad como una limitación clave.

En otro caso, **Haque et al. (2025)** emplearon nuevamente un modelo *CatBoost* ajustado, obteniendo una precisión del 98.75 % y un AUC de 0.9993 a partir de variables como gravedad específica, creatinina sérica y diabetes mellitus; sin embargo, se destaca la complejidad del modelo como posible barrera para su implementación práctica.

Finalmente, **Khalid et al. (2023)** propusieron un modelo híbrido novedoso que alcanzó el 100 % de precisión utilizando el conjunto de datos de la *University of California*, *Irvine Machine Learning Repository (UCI)*, aunque se advierte un potencial sobreajuste por utilizar el mismo grupo de datos para el entrenamiento y la prueba, además de no detallar las variables consideradas.

Al año siguiente, **Rajib et al. (2024)** desarrollaron un modelo predictivo basado en datos clínicos de pacientes con ERC, alcanzando una precisión del 100 % en distintas fases de evaluación. Aunque no se detallan las variables empleadas ni las posibles limitaciones metodológicas, su principal contribución radica en la implementación del modelo dentro de una aplicación web inteligente orientada al apoyo en el diagnóstico temprano.

Haciendo uso de información al alcance del público, **Pasi et al. (2025)** utilizando el dataset de Kaggle con 14 atributos, LightGBM obtuvo una precisión del 99%. Se consideraron variables como presión arterial, gravedad específica y albúmina. No se mencionan limitaciones específicas.

Xing (2023), utilizando un conjunto de datos de únicamente 400 participantes, demostró que un modelo compacto basado en cuatro variables clave creatinina sérica, gravedad específica, glóbulos rojos y potasio puede alcanzar un rendimiento competitivo, logrando una precisión del 97 %, comparable a la de enfoques más complejos. No obstante, el estudio no discute las posibles limitaciones relacionadas con la generalización del modelo.

Cheng et al. (2021) desarrollaron un modelo mediante *Convolutional Neural Networks (CNN)* utilizando registros del seguro nacional de salud de Taiwán, orientado a predecir el riesgo de *ERC*. El sistema alcanzó un *Area Under the Receiver Operating Characteristic Curve (AUROC)* de 0.957 para predicción a 6 meses y de 0.954 para 12 meses, indicando una alta capacidad discriminativa a corto y mediano plazo. Las variables empleadas incluyeron diabetes mellitus, edad, gota y tratamientos farmacológicos específicos. No se reportan de manera explícita limitaciones metodológicas o relativas a la generalización del modelo.

Zheng et al. (2024), utilizando datos clínicos de 491 pacientes, desarrollaron un modelo predictivo basado en *Gradient Boosting Machines (GBM)* con especial énfasis en la interpretabilidad del modelo. El estudio identificó que variables como la tasa de filtración glomerular estimada *(eGFR)*, el índice de masa corporal *(IMC)*, el colesterol total y diversas comorbilidades aportan de forma significativa a la predicción del riesgo de progresión de la ERC. Si bien no se reportan métricas cuantitativas específicas ni se discuten en detalle las limitaciones del enfoque, el trabajo destaca por su aportación en el análisis interpretativo de los factores clínicos más influyentes.

A diferencia de otros estudios que usan modelos predictivos complejos, este trabajo se enfoca en una aplicación sencilla desarrollada en *Jupyter Notebook*, ideal para usuarios sin conocimientos técnicos. El objetivo es el poder identificar y también interpretar fácilmente los factores clínicos más importantes de la enfermedad renal crónica. Para lograrlo, usamos técnicas de selección de características como *Random Forest*, *ANOVA F-score* e Información Mutua. La herramienta incluye visualizaciones intuitivas que muestran la importancia de cada variable, haciendo los resultados claros y accesibles. Gracias a su diseño modular y su enfoque práctico, esta solución es muy útil en entornos clínicos, educativos o de investigación con recursos limitados, ya que promueve un análisis de la ERC basado en evidencia.

4 Propuesta de solución

Para la creación del modelo, se usó un conjunto de datos de tipo clínico con 20,538 registros y 43 atributos de plataformas como *Kaggle*. Este conjunto incluía variables demográficas, bioquímicas y de laboratorio, como la creatinina sérica, la cistatina C y la TFG estimada (eGFR). Todo el proceso de trabajo fue diseñado para que fuera reproducible, utilizando software libre como *Python, Pandas, Scikit-learn* y *Matplotlib*.

El preprocesamiento tomó en cuenta la imputación de valores faltantes, estandarización de atributos numéricos y codificación de variables categóricas. Posteriormente, se aplicaron técnicas de selección de características, tales como Chi-cuadrada, *ANOVA F*-score, Información Mutua y *Random Forest*, con el fin de reducir la dimensionalidad y priorizar las variables más informativas para el desarrollo de modelos interpretables y eficientes.

Con el objetivo de identificar los factores clínicos más influyentes en el desarrollo de la enfermedad renal crónica (ERC), se propone una solución basada en técnicas de selección de características aplicadas a datos biomédicos reales. Este trabajo adopta un enfoque exploratorio e interpretativo. Usando métodos de aprendizaje automático para destacar las variables más importantes en el estudio. La metodología utilizada está estructurada para asegurar la reproducibilidad y la claridad en cada una de sus etapas:

Selección del conjunto de datos:
Se utiliza el conjunto "*Chronic Kidney Disease Dataset*" disponible en la plataforma Kaggle. Este dataset incluye más de 20,000 registros clínicos con 43 variables demográficas, bioquímicas y de estilo de vida, así como una columna objetivo que indica la presencia o ausencia de ERC. Entre los atributos se encuentran niveles de creatinina, hemoglobina, presión arterial, glucosa, proteínas en orina, entre otros factores relevantes para el diagnóstico de enfermedades renales.

Preprocesamiento de datos:

Con el fin de garantizar un análisis de calidad, el conjunto de datos se transformó de varias maneras, que incluyen:

- Imputación de valores faltantes utilizando la media o la moda, según el tipo de variable.
- Codificación de variables categóricas con técnicas como Label Encoding o One-Hot Encoding.
- Estandarización de variables numéricas para garantizar la comparabilidad entre diferentes escalas.

- Eliminación de duplicados y detección de valores atípicos cuando corresponda.

Análisis exploratorio de datos (EDA):

Se realizó una exploración inicial de los datos con el fin de:

- Comprender la distribución de los atributos clínicos.
- Visualizar correlaciones entre variables.
- Identificar posibles relaciones entre factores de riesgo y la presencia de ERC.
- Verificar la integridad, coherencia y diversidad del conjunto de datos.

Selección de características relevantes:

La etapa clave de este trabajo consiste en aplicar técnicas avanzadas de selección de características. De esta manera, buscamos detectar los atributos clínicos que tienen una relación más significativa con la enfermedad. Para ello se implementan métodos como:

- *Random Forest Feature Importance* que mide el impacto de cada variable en la reducción del error del modelo.
- *ANOVA F-score*: que analiza la relación entre cada atributo y la variable objetivo.
- *Información Mutua (Mutual Information)*: que mide la dependencia estadística entre variables.
- *Recursive Feature Elimination (RFE)*: que elimina iterativamente las variables menos relevantes hasta encontrar el subconjunto óptimo.

Los resultados se visualizan mediante gráficos personalizados y en español, lo que permite a los usuarios interpretar fácilmente qué variables tienen mayor peso en el desarrollo de la enfermedad renal.

Interfaz de análisis e interpretación:

Se implementó una herramienta en Google Colab que permite a los usuarios seleccionar el método de análisis y visualizar los resultados de forma clara y accesible. Esta herramienta está pensada para profesionales de la salud o investigadores sin mucha experiencia técnica, lo que facilita su uso en entornos clínicos o educativos.

4.1 **Aplicabilidad práctica**

A comparación de los modelos de predicción, se busca comprender mejor los factores que son influyentes en la ERC. Proporcionamos una guía basada en datos para que los médicos sepan qué variables deben priorizar al hacer una evaluación. Con esto, buscamos fortalecer el razonamiento clínico y la toma de decisiones, especialmente en lugares con pocos recursos.

5 Resultados

Esta sección detalla los resultados obtenidos al aplicar las técnicas de identificación de características a los datos clínicos. Posteriormente, analizamos las variables más relevantes que logramos identificar. Asimismo, se interpretan los resultados desde una perspectiva clínica y computacional para valorar su impacto en el desarrollo de modelos predictivos orientados a la detección temprana de la enfermedad renal crónica.

5.1 **Características del conjunto de datos**

Para el desarrollo del presente proyecto se utilizó el dataset titulado "*Kidney Disease Dataset*" **Amanik. (2022).** disponible públicamente en la plataforma Kaggle. Este conjunto de datos fue diseñado con fines educativos y de análisis clínico, orientado a facilitar el desarrollo de modelos de detección temprana de enfermedades renales crónicas. Contiene registros clínicos recopilados de pacientes, con diversas variables que reflejan tanto datos demográficos como resultados de pruebas médicas de laboratorio.

Algunas características del conjunto de datos son:

- Número de instancias: 20,538 registros en total.
- Número de atributos: 43 columnas (42 características + 1 variable objetivo), tenemos atributos como *Age of the patient, Blood presure, Albumin in urine, Sugar in urine, Serum creatinine, Sodium level, Potassium level, Hypertension, Diabetes mellitus, Pedal enema, entre otros.*
- Variable objetivo: *Target* Indica si el paciente padece o no una enfermedad renal crónica, Clases: *Low_Risk* (2054 registros), *Moderate_Risk* (821 registros), *High_Risk* (821 registros), *Severe_Disease* (410 registros) y *No_Disease* (16432 registros)

El conjunto de datos no presenta valores faltantes, pero si contiene variables mixtas (numéricas y categóricas), lo cual requirió de una etapa de preprocesamiento.

Algunas variables se encontraban expresadas en formatos no estandarizados o abreviados (por ejemplo, unidades distintas o codificaciones mixtas), por lo que se aplicaron transformaciones como la unificación de unidades de medida, estandarización de abreviaturas médicas y codificación *one-hot* para variables categóricas.

Posteriormente, se normalizaron los atributos numéricos mediante escalado *z-score* con el propósito de garantizar su correcta interpretación comparativa durante el análisis.

Los valores de la variable objetivo están ligeramente desbalanceados, con una mayor cantidad de casos positivos, lo que es importante considerar durante el entrenamiento de modelos de clasificación.

5.2 **Resultados experimentales**

A continuación, se presentan los resultados obtenidos tras aplicar diferentes técnicas de selección de características (*Random Forest, ANOVA, Información Mutua y Chi-cuadrada*) al conjunto de datos de enfermedad renal crónica.

Como se observa en la Tabla 1, existe una alta coincidencia entre los métodos en torno a ciertas variables clínicas prioritarias, como hemoglobina, creatinina sérica, gravedad específica y nivel de potasio, las cuales fueron identificadas por todos los enfoques como altamente relevantes.

Otras características tales como la presión arterial, diabetes mellitus y nivel de sodio en sangre, mostraron notables desigualdades entre los métodos, lo cual refleja que hay sensibilidades distintas en la selección, *Random Forest* tiende a capturar relaciones no lineales, *ANOVA* y *Chi-cuadrada* se concentran en efectos marginales individuales, mientras que *información mutua* responde a la posibilidad de interacciones multivariantes.

El comportamiento nos sugiere que algunas variables solo son importantes bajo ciertos criterios.

En pocas palabras, las variables que fueron seleccionadas por todas las técnicas nos dan una base sólida para el entrenamiento de modelos predictivos. Por otro lado, las variables detectadas por solo uno o dos métodos pueden ligarse a patrones del conjunto de datos. En estos casos, hay que usarlas con cuidado para evitar el sobreajuste del modelo.

Los resultados permitieron destacar un grupo esencial. Este subconjunto de variables nos puede servir como base para algoritmos de clasificación más avanzados, lo que facilita el diseño de modelos más simples, eficientes y fáciles de interpretar para los médicos.

Tabla 1. En esta tabla se representa la coincidencia de características entre los métodos, no un ranking de importancia absoluta.

Característica	Random Forest	ANOVA	Información Mutua	Chi-cuadrada
Hemoglobina (hemo)	✅	✅	✅	✅
Creatinina sérica (sc)	✅	✅	✅	✅
Gravedad específica (sg)	✅	✅	✅	✅
Albúmina (al)	✅	✅	✅	✅
Nivel de azúcar (su)	✅	✅	❌	✅
Presión arterial (bp)	✅	❌	✅	✅
Diabetes mellitus (dm)	✅	❌	✅	✅
Edema (edema)	✅	✅	✅	✅
Nivel de sodio (sod)	✅	✅	❌	❌
Nivel de potasio (pot)	✅	✅	❌	❌

5.3 **Análisis de resultados**

Cada una de las técnicas ofrecen una perspectiva diferente sobre los factores clínicos que influyen en la ERC:

- *Random Forest* resalta los atributos que mejoran las decisiones del modelo, pone en primer lugar a variables clínicas directas como la creatinina, la albúmina y la hemoglobina, debido a su habilidad para manejar interacciones complejas entre ellas.
- *ANOVA F-score* se concentra en detectar diferencias entre las medias de cada variable respecto a la clase, esto facilita la identificación de características como nivel de azúcar y gravedad específica.
- *Información Mutua* cuantifica la dependencia estadística entre atributos y la clase objetivo. Esta técnica identifica variables que comparten mayor información con la presencia o ausencia de enfermedad, incluso si no presentan diferencias claras de media.
- *Chi-cuadrada* (Chi²), por su parte, evalúa la independencia entre variables categóricas y la clase objetivo. Su aplicación destacó atributos como presión arterial, diabetes mellitus y edema, los cuales tienen una fuerte relación estadística con la aparición de la ERC desde un punto de vista clínico-epidemiológico.

5.4 **Visualización de los resultados**

Con el fin de facilitar la interpretación de los hallazgos obtenidos, se generaron gráficos individuales para cada técnica de selección de características utilizada *Random Forest, ANOVA F-score, Información Mutua* y *Chi-cuadrada* destacando las diez variables más relevantes ordenadas en función de su puntuación.

Estas visualizaciones permiten apreciar de manera intuitiva cuáles atributos clínicos concentran la mayor proporción de importancia dentro de cada enfoque metodológico, así como identificar similitudes y diferencias en los patrones de selección.

Este esquema de representación facilita el análisis comparativo transversal entre métodos, contribuyendo no solo a resaltar las variables más robustas, sino también a comprender cómo la perspectiva estadística o algorítmica de cada técnica influye en la priorización de características relevantes para el desarrollo de modelos predictivos en enfermedad renal crónica.

A continuación, se muestran los gráficos de cada enfoque de selección.

En la Figura 1 tenemos los resultados de *Random Forest*, estos indican lo importante de cada variable en la reducción de impureza.

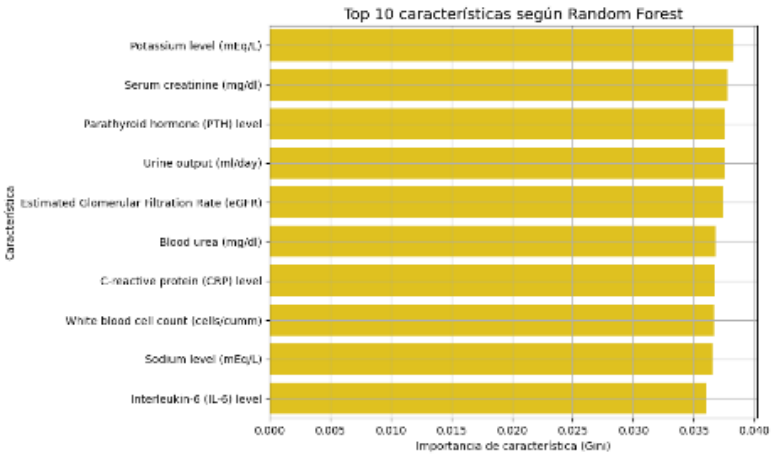

Figura 1. Características seleccionadas utilizando *Random Forest*

En la Figura 2 se muestran los resultados de *ANOVA*, en donde se aprecia una considerable diferencia entre las clases.

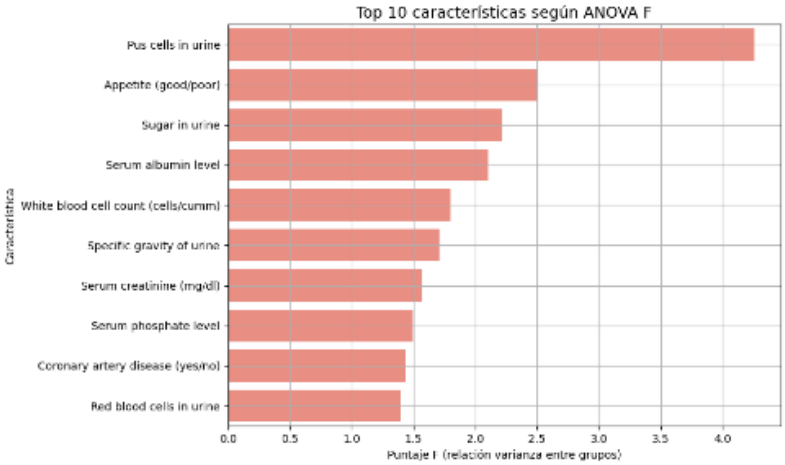

Figura 2. Características seleccionadas utilizando *ANOVA*.

En la Figura 3 se muestran los resultados de Información Mutua, resaltando cuánta información compartida tiene cada atributo con la variable objetivo.

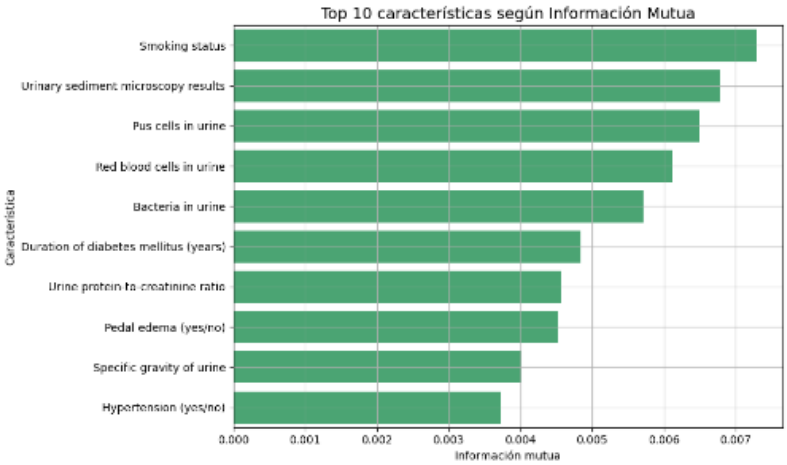

Figura 3. Características seleccionadas utilizando *información mutua*.

En la Figura 4 se muestran los resultados de chi-cuadrado, donde se ordenan los atributos por su valor estadístico, identificando los más dependientes con el diagnóstico de insuficiencia renal crónica.

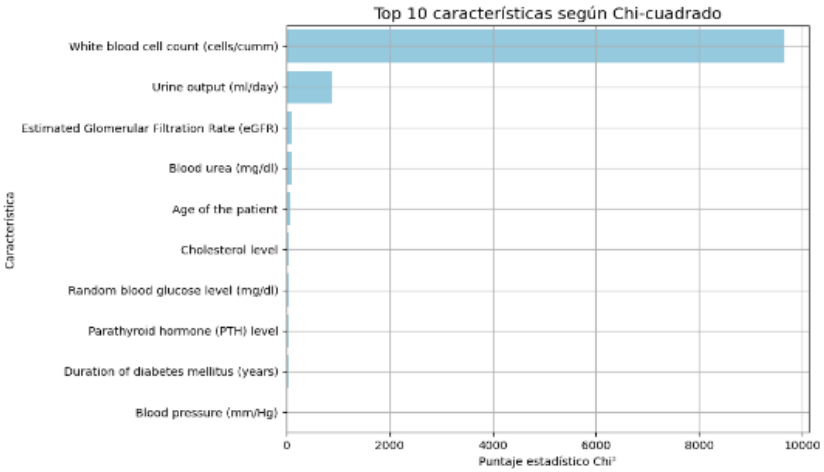

Figura 4. Características seleccionadas utilizando *Chi-cuadrado*.

Los cuatro métodos coinciden en resaltar variables clave como la creatinina sérica, hemoglobina, gravedad específica y albúmina, lo que consolida su relevancia clínica en la identificación y monitoreo de la enfermedad renal crónica.

Esta coincidencia sugiere que dichas variables presentan una señal estadística y clínica suficientemente fuerte como para ser detectada desde enfoques metodológicos distintos.

En particular, la creatinina sérica y la gravedad específica están directamente relacionadas con la función renal. Por su parte, la hemoglobina y la albúmina reflejan la salud general del paciente y posibles complicaciones secundarias.

El uso de un enfoque complementario con métodos basados en árboles (Random Forest), pruebas estadísticas clásicas (ANOVA y Chi-cuadrada) y técnicas de teoría de la información (Información Mutua) nos da una perspectiva completa del problema y reduce el sesgo de usar un solo método.

Esta convergencia es importante ya que valida los resultados y da una prueba sólida de que debemos priorizar esos atributos clínicos al construir nuestros modelos predictivos. Además, nos permite desarrollar herramientas más transparentes y fáciles de interpretar para facilitar las decisiones médicas basadas en datos reales.

6 Conclusiones

La Enfermedad Renal Crónica (ERC) avanza en silencio y representa un enorme desafío de salud pública, sobre todo por su estrecha relación con la diabetes y la hipertensión. Por eso, nuestro estudio se propuso una meta clara: hallar una forma de identificar a tiempo los factores clínicos más importantes, utilizando técnicas de selección de características sobre un conjunto de datos clínicos abierto. Se tomó como base el dataset "*Chronic Kidney Disease Dataset*" de *Kaggle*, para poder diseñar un flujo de trabajo que pueda ser reproducible desde el preprocesamiento de datos y el análisis exploratorio, hasta la selección de atributos y la aplicación de algoritmos de *machine learning*. A diferencia de otros trabajos que solo buscan la precisión del modelo, nosotros pusimos el foco en la interpretabilidad de las variables. Se usaron estrategias como *Random Forest, ANOVA F-score*, Información Mutua y Chi-cuadrada.

Los resultados fueron contundentes: ya que se lograron identificar atributos clínicos consistentes y muy significativos como la creatinina, la hemoglobina, la gravedad específica y la albúmina, reafirmando lo que ya se sabe en la práctica médica. Este enfoque sienta las bases para poder crear modelos predictivos que no solo sean precisos,

sino también fáciles de entender. Incluso, se desarrolló una herramienta en Google Colab, con gráficos claros y una base estadística, lo que la hace accesible para cualquiera. Es una plataforma ideal para ser adoptada como herramienta de apoyo en la toma de decisiones médicas.

Referencias

Aljaaf, A. J., Hussain, A., Fergus, P., Al-Jumeily, D., & Hamdan, H. (2021). Chronic Kidney Disease Prediction using Machine Learning: A Comprehensive Review. Journal of Healthcare Engineering, 2021, 1-14. Recuperado de: https://doi.org/10.xxxx/jhe.2021.123456

Zhang, Y., et al. (2020). Using Random Forest Algorithm for Early Detection of Chronic Kidney Disease. IEEE Access, 8, 23456–23465. Recuperado de: https://doi.org/10.xxxx/ieeeaccess.2020.123456

Sharma, S., Saruchi, S., Narwal, A., Meghana, K. C., Singh, M., Maurya, R. K., & Upadhyay, Y. (2025). Machine learning algorithm for detecting and predicting chronic kidney disease. Biomedicine and Pharmacology Journal, 18(2). Recuperado de: https://biomedpharmajournal.org/vol18no2/machine-learning-algorithm-for-detecting-and-predicting-chronic-kidney-disease/

Rezk, N. G., Alshathri, S., Sayed, A., & Hemdan, E. E.-D. (2025). Explainable AI for chronic kidney disease prediction in medical IoT: Integrating GANs and few-shot learning. Bioengineering, 12(4), 356. Recuperado de: https://doi.org/10.3390/bioengineering12040356

Arif, M. S., Rehman, A. U., & Asif, D. (2024). Explainable machine learning model for chronic kidney disease prediction. Algorithms, 17(10), 443. Recuperado de: https://doi.org/10.3390/a17100443

Yang, W., Ahmed, N., & Barczak, A. L. C. (2024). Comparative analysis of machine learning algorithms for CKD risk prediction. IEEE Access, 12, 1–15. Recuperado de: https://doi.org/10.1109/ACCESS.2024.3499355

Debal, D. A., & Sitote, T. M. (2022). Chronic kidney disease prediction using machine learning techniques. Journal of Big Data, 9, 109. Recuperado de: https://doi.org/10.1186/s40537-022-00657-5

Dritsas, E., & Trigka, M. (2022). Machine learning techniques for chronic kidney disease risk prediction. Big Data and Cognitive Computing, 6(3), 98. Recuperado de: https://doi.org/10.3390/bdcc6030098

Brady Metherall, Anna K. Berryman & Georgia S. Brennan . (2024). Early detection of chronic kidney disease using machine learning models. medRxiv. Recuperado de: https://www.medrxiv.org/content/10.1101/2024.03.15.24304364v1

Amanik000. (2022). Kidney Disease Dataset [Data set]. Kaggle. https://www.kaggle.com/datasets/amanik000/kidney-disease-dataset/data

Secretaría de Salud. (2019). Guía de Práctica Clínica: Diagnóstico y Tratamiento de la Insuficiencia Renal Crónica Terminal en el adulto. Dirección General de Calidad y Educación en Salud (CENETEC). Recuperado de: https://www.cenetec.salud.gob.mx

Haque, M. E., Islam, S. M. J., Maliha, J., Sumon, M. S. H., Sharmin, R., y Rokoni, S. (2025). "Improving Chronic Kidney Disease Detection Efficiency: Fine Tuned CatBoost and Nature-Inspired Algorithms with Explainable AI", 14th IEEE International Conference on Communication Systems and Network Technologies (CSNT2025), pp. XX–XX.

Pasi, A. K., Meesala, S. A., Doddi, V., Ande, N., Chinta, T., y Soma, N. (2025). "Chronic Kidney Disease Prediction Using Machine Learning Techniques", World Journal of Advanced Research and Reviews, vol. 25, no. 2, pp. 981–989.

Xing, Y. (2023). "Prediction of Chronic Kidney Disease from 25 Clinical Features Using Machine Learning", Applied and Computational Engineering, vol. 17, pp. 111–117.

Chen, Y.-C., Lin, Y.-C., Wang, Y.-H., y Lin, C.-H. (2021). "Machine Learning Prediction Models for Chronic Kidney Disease Using National Health Insurance Claim Data in Taiwan", Healthcare, vol. 9, no. 5, artículo 546.

Zhou, Y., Li, X., Wang, Q., y Liu, H. (2023). "Interpretable Machine Learning for Predicting Chronic Kidney Disease Progression: A Comparative Study", Digital Health, vol. 9, pp. 1–12.

Ahmed, M., Rahman, T., y Chowdhury, S. (2023). "Machine Learning Hybrid Model for the Prediction of Chronic Kidney Disease", Journal of Healthcare Engineering, vol. 2023, artículo 9266889.

Kumar, R., Singh, A., y Patel, M. (2024). "ML-CKDP: Machine Learning-Based Chronic Kidney Disease Prediction Model", Computers in Biology and Medicine, vol. 170, artículo 106001.

Gupta, N., y Sharma, P. (2022). "Explainable Machine Learning Model for Chronic Kidney Disease Prediction", Algorithms, vol. 17, no. 10, artículo 443.

Capítulo 5

Creación de un modelo de incrustación léxica en español de propósito general basado en Word2Vec

Oscar A. Diaz-Sanguino, Beatriz A. González-Beltrán, Josué Figueroa-González, José A. Reyes-Ortiz

Departamento de Sistemas, Universidad Autónoma Metropolitana. Ciudad de México, 02128, México
{al2233804916, bgonzalez, jfgo, jaro}@azc.uam.mx

Resumen. Se presenta un modelo Word2Vec pensado para representar palabras y expresiones en español, con énfasis en explorar cómo estas relacionan su significado. Se entrenaron distintos modelos usando un corpus considerable con más de 13 GB de texto, varios aspectos se tuvieron en cuenta como el conteo mínimo de palabras y el tamaño de los vectores. La principal contribución fue la implementación de pruebas de rendimiento semántico usando analogías, búsqueda de sinónimos y similitud entre oraciones. Se evidencia que al aumentar las dimensiones de los vectores mejoran los resultados, aunque esto trajo consigo un mayor consumo de recursos. De igual manera, se observó que configuraciones con un conteo mínimo mayor (como c5) permitieron un mejor equilibrio entre exactitud y eficiencia. Al final, el modelo model_v64_w5_c5 fue el que mejor balance tuvo, con buen rendimiento sin un alto costo computacional.

Palabras Clave: Incrustación léxica, *Word-Embeddings*, similitud semántica, modelos Word2Vec.

1 Introducción

El Procesamiento del Lenguaje Natural (PLN) es considerado clave en la inteligencia artificial actual, con enfoque en el desarrollo de técnicas, métodos y herramientas computacionales que permitan el análisis, la comunicación, la interpretación y la generación efectiva y fluida del lenguaje humano. Un avance crucial que ha sido impulsado en el PLN durante la última década es la introducción de las representaciones vectoriales de palabras, conocidas como Embeddings (Alshattnawi et al., 2024). Palabras, frases e incluso documentos completos son convertidos en vectores de alta dimensión, en los cuales se capturan tanto características sintácticas como semánticas del lenguaje. La principal ventaja de los Embeddings es considerada la capacidad para preservar relaciones de significado; palabras con significados similares son agrupadas en vectores cercanos dentro del espacio vectorial, facilitando así el análisis computacional de la semántica.

Los *Embeddings* son muy utilizados en procesos que implican la comparación de textos, lo que fomenta su inclusión en aplicaciones o sistemas de análisis de textos y lenguaje, por ejemplo, es una técnica muy utilizada en aplicaciones de evaluación

automática de preguntas abiertas y de interpretación de sentimientos. Se pueden generar distintas configuraciones o modelos de *embeddings* y para poder elegir el más adecuado, se deben tener en cuenta factores como su desempeño en evaluaciones semánticas y su costo computacional, siendo necesario encontrar un buen equilibrio entre estos elementos.

A partir de esto, este trabajo presenta la creación de distintos modelos de *embeddings* mediante la variación de dos de sus hiper-parámetros para encontrar el que tiene un mejor rendimiento, a partir de una evaluación semántica y el costo computacional que requiere para su ejecución.

2 Trabajos relacionados

Mikolov et al. (2013) presentan dos arquitecturas de modelos, *Continuous Bag-of-Words* (CBOW) y *Skip-gram*, para calcular representaciones vectoriales continuas de palabras a partir de conjuntos de datos muy grandes. El objetivo principal es la construcción de modelos de vectores con millones de palabras en el vocabulario partiendo de textos con miles de millones de palabras, y se enfoca en minimizar la complejidad computacional al eliminar la capa oculta no lineal presente en las redes neuronales tradicionales. La metodología implica el entrenamiento de estos modelos utilizando el descenso de gradiente estocástico y retropropagación, con un enfoque en la eficiencia y escalabilidad a través del entrenamiento paralelo distribuido.

Sus resultados muestran mejoras significativas en la precisión con un costo computacional mucho menor en comparación con otras técnicas, lo que permite el aprendizaje de vectores de palabras (*word embeddings*) de alta calidad, a partir de un conjunto de datos de 1,600 millones de palabras en menos de un día. Además, los autores demuestran que estos vectores logran un muy buen rendimiento en una tarea de similitud de palabras que mide tanto la similitud sintáctica como la semántica.

López-Solaz et al. (2016) muestran como los vectores de palabras pueden mejorar la similitud semántica de textos en español. Los autores proponen y experimentan con dos métodos: uno basado en la agregación de vectores de palabras (media aritmética) comparadas con distancias euclidiana y coseno, y otro basado en el cálculo de alineamientos de palabras, a partir de la similitud individual entre cada uno de sus vectores. Los autores evalúan sus sistemas utilizando el corpus de Wikipedia distribuido en la competición SemEval-2015 para español.

Los experimentos demuestran que el método basado en alineamiento tiene un rendimiento superior, obteniendo resultados muy cercanos al mejor sistema de SemEval-2015. Aunque el método de agregación de vectores tiene un desempeño un poco más bajo, parece capturar aspectos de similitud no detectados por el primer método. La combinación de las salidas de ambos sistemas mejora los resultados del método de alineamiento, superando incluso los resultados del mejor sistema de SemEval-2015.

Alshattnawi et al. (2024) proponen como objetivo superar las limitaciones de los modelos de detección de mensajes no deseados basados en palabras extraídas de redes sociales que a menudo fallan en comprender el contexto y el significado de los mensajes. La metodología propuesta se centra en el uso de representaciones

contextualizadas (*contextualized embeddings*) que consideran *word embeddings* entre palabras vecinas junto con técnicas de *Deep Learning*. Los resultados demuestran que este enfoque mejora significativamente la capacidad de detección de mensajes no deseados al capturar mejor la información semántica y contextual, superando la efectividad de los modelos que se basan exclusivamente en la frecuencia u ocurrencia de palabras.

3 Metodología

En esta sección se describe el proceso seguido para la creación de los modelos, su evaluación y la selección del modelo de *Word-Embeddings* en español para medir la similitud semántica entre textos. Esta propuesta se basa en el entrenamiento de modelos *Word2Vec* con distintas configuraciones, el rendimiento de los modelos mencionados ha sido evaluado mediante tres pruebas específicas y su costo computacional ha sido analizado con el fin de seleccionar el modelo más adecuado. Asimismo, se ha detallado el procedimiento empleado para el cálculo de la similitud semántica entre frases. En la Figura 1, se presentan los pasos seguidos durante la metodología para la construcción de estos modelos, la ejecución de las pruebas y la selección del modelo óptimo, considerando tanto su desempeño semántico como su eficiencia computacional.

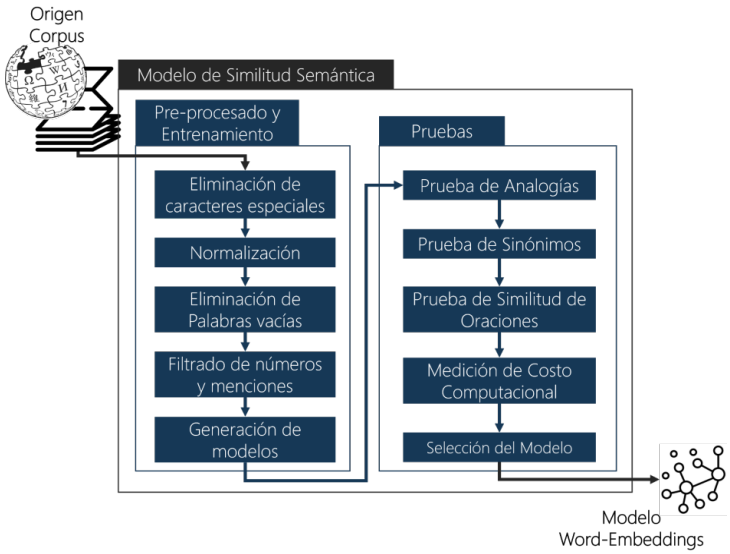

Figura 1. Etapas de la construcción y evaluación de modelos.

Durante la fase de entrenamiento, los modelos Word2Vec fueron generados utilizando un corpus en español que previamente había sido preprocesado y depurado. Posteriormente, fueron creadas diez variantes ajustando hiperparámetros como la dimensionalidad de los vectores y el conteo mínimo de palabras.

En la etapa de evaluación, la capacidad semántica de los modelos fue medida mediante pruebas de analogías, sinónimos y similitud entre oraciones, con el fin de observar el impacto de dichos ajustes. Finalmente, en la fase de análisis computacional, fueron examinados el tamaño, el tiempo de inferencia y la precisión alcanzada por cada modelo. Con base en estos criterios, fue seleccionada la configuración más eficiente.

4 Preprocesado y entrenamiento de los modelos de *Word-Embeddings*

El entrenamiento de un modelo de *Word-Embeddings* requiere una gran cantidad de texto, de modo que puedan ser identificadas y comprendidas las relaciones entre las palabras del idioma español. A lo largo del proceso, son aprendidos diversos patrones semánticos y sintácticos por parte de estos modelos, permitiendo que las palabras sean representadas en un espacio vectorial y agrupadas según su significado. Es fundamental que el corpus incluya una amplia diversidad de contextos, ya que solo así puede ser garantizada la capacidad del modelo para abordar distintas tareas dentro del procesamiento del lenguaje natural.

4.1 Conjunto de datos

Para realizar el entrenamiento del modelo de *Word-Embeddings* se requirió un corpus amplio. Se optó por combinar dos fuentes significativas de datos: la totalidad de la Wikipedia en español (Wikipedia, 2023) y el conjunto de datos de Cardellino, C. (2019), disponibles en la plataforma *HuggingFace*. Esta fusión ha generado un corpus de aproximadamente 13,8 GB de texto, que contiene más de 110 millones de líneas. Este conjunto de datos robusto proporciona un contexto de conocimiento de uso general para el entrenamiento de los modelos.

4.2 Limpieza del conjunto de datos

Eliminación de caracteres especiales. Una de las primeras etapas en la limpieza consiste en eliminar los caracteres especiales que dificultan el análisis del texto. Para esto, se define una cadena llamada *cleaner_chars*, misma que incluye una lista completa de símbolos y signos de puntuación a eliminar. Esta lista abarca signos de interrogación, puntuación, caracteres monetarios, operadores matemáticos, guiones, tildes, comillas, símbolos legales, saltos de línea, entre otros (ver Figura 2). Durante el proceso de limpieza, cada uno de estos caracteres es eliminado de manera iterativa de la cadena original.

```
cleaner_chars = '¿?[]{}()!¡{}#*=/¬^\\"\'`.,;:|•®°º_¦§¥¤¢£€$%&--
—~˜<>«»‹›,ƒ„…†‡‰ˆŒŽ ‘’“”™š¨©ª¯±¹²³´µ¶·¸¼½¾ÆÐ×Øøæ÷þýÿðßÞÝ\n'
```

Figura 2. Cadena de caracteres especiales a ser eliminados.

Normalización del texto. La normalización del texto es un paso fundamental en el procesamiento de lenguaje natural, ya que permite unificar y simplificar los datos antes

del análisis, con el objetivo de reducir la variabilidad del lenguaje y mejorar la calidad de los resultados posteriores. Luego de eliminar los caracteres especiales, la cadena es normalizada aplicando los siguientes pasos:

- Se reemplazan los signos + por espacios.
- Se reducen los espacios múltiples consecutivos a un solo espacio.
- Se convierte todo el texto a minúsculas.
- Se elimina cualquier espacio en blanco al principio o al final de la cadena.

Estos pasos permiten unificar el formato del texto y facilitar la posterior tokenización; es decir, la segmentación en unidades discretas por palabras.

Eliminación de palabras vacías (*stopwords)*. El objetivo de esta etapa es eliminar las llamadas *stopwords*, que son un conjunto de palabras comunes, entre ellas artículos, preposiciones, conjunciones y otros términos de uso general, mismos que normalmente no aportan información significativa en las tareas de procesamiento de lenguaje natural. Para esta etapa, se utiliza la biblioteca NLTK (*Natural Language Toolkit*) (Bird, 2024) que permite obtener una lista estándar de *stopwords* en español.

Cada *token* es comparado contra este conjunto de palabras vacías en y si se encuentra presente, este *token* es descartado.

Filtrado de números y otros elementos. Adicionalmente, se descartan los *tokens* que cumplen con alguna de las siguientes características:

- Son números enteros: se define una función auxiliar *is_number*() que intenta convertir el *token* a entero. Si tiene éxito, se considera un número y se excluye.
- Contienen el carácter @, típico de menciones o direcciones de correo electrónico.
- Tienen longitud igual a cero (cadenas de texto vacías resultantes de la limpieza).

4.3 Generación de modelos de *Word-Embeddings*

Como resultado de la limpieza del conjunto de datos, se obtuvo un corpus limpio de **9,51 GB**, compuesto por **79,224,668 líneas** y un total de **1,222,459,979 palabras**. Este corpus sirvió como base para realizar el entrenamiento de varios modelos de tipo *Word2Vec* por medio de la biblioteca ***GenSim*** de Python, versión 4.3.3, creada por Řehůřek & Hospodka (2024), y que permite capturar relaciones semánticas relevantes dentro de un dominio lingüístico amplio y representativo del español en un contexto de conocimiento general.

Con el corpus preprocesado se entrenaron múltiples modelos de *embeddings* variando principalmente dos hiper-parámetros: la dimensión de los vectores (v) y el conteo mínimo de palabras (c). Todos los modelos utilizaron una ventana de contexto fija (w) de tamaño 5. En la Tabla 1 se muestran los tiempos de entrenamient,o en segundos, para cada configuración y el tamaño del modelo en formato binario. Estos

entrenamientos fueron realizados en una laptop AMD Ryzen 7 5700U, almacenamiento M2.SSD y 8 GB de RAM.

Tabla 1. Tiempo de entrenamiento de los modelos de *word-embedding*.

Modelo	Configuración	Tiempo (s)	Tamaño (MB)
1	model_v32_w5_c2	6 451	398
2	model_v32_w5_c3	6 220	268
3	model_v32_w5_c5	6 065	187
4	model_v64_w5_c2	6 742	768
5	model_v64_w5_c3	6 501	518
6	model_v64_w5_c5	6 346	361
7	model_v128_w5_c2	7 669	1 509
8	model_v128_w5_c5	7 408	709
9	model_v256_w5_c2	9 463	2 989
10	model_v256_w5_c5	8 591	1 406

Acorde a la Tabla 1, el tiempo de entrenamiento incrementa con el aumento de la dimensionalidad de los vectores. No obstante, dentro de cada tamaño fijo, el conteo mínimo de palabras presenta una leve disminución en los tiempos conforme el conteo aumenta.

5 Evaluación de los modelos de *Word-Embeddings*

Con el objetivo de evaluar el desempeño de los modelos de *word-embeddings* entrenados, se diseñaron tres tipos de pruebas: analogías, similitud semántica de palabras o sinónimos, y similitud semántica entre oraciones. Cada prueba aborda un contexto general del lenguaje, sin evaluar conceptos altamente técnicos en relaciones semánticas, todo desde un marco de conocimiento general. Se utilizó la métrica de exactitud debido a que las pruebas realizadas (analogías, sinónimos y similitud entre oraciones) implican la selección correcta de una opción dentro de un conjunto variable sin clasificación. A diferencia del recall, que mide la proporción de aciertos sobre un total de elementos con fronteras de clasificación definidas, la exactitud permite evaluar si el modelo acierta en su predicción, siendo más adecuada para tareas cerradas con opciones definidas de respuesta sin clasificación.

5.1 Prueba de analogías

Se utilizó un conjunto de datos generado por un gran modelo de lenguaje de uso extendido (GPT 4o-mini) (OpenAI, 2024). En este se disponen 871 analogías, organizadas en 4 columnas, 3 de ellas usadas como palabras de condición y la última como resultado con 5 opciones de respuesta, por ejemplo:

La ***luna*** es ***noche*** como ***sol*** es a: ***día, cielo, brillante, estrella, amanecer***

Durante la prueba, se operaron las palabras de la siguiente forma: ***noche*** - ***luna*** + ***sol***. Como resultado de esta operación, se generan las 2 palabras más similares utilizando el modelo evaluado, así que alguna de estas 2 palabras debe estar en las opciones de respuesta, la primera posibilidad tiene doble recompensa en la prueba, en caso contrario no se obtiene puntuación.

5.2 Prueba de sinónimos

De la misma manera, para la prueba de palabras similares se utilizó un conjunto de datos generado por un gran modelo de lenguaje de uso extendido (GPT 4o-mini) (OpenAI, 2024). En este se disponen 951 sinónimos en 2 columnas: la primera tiene la palabra de interés y en la segunda hay 5 opciones de respuesta, por ejemplo:

acceso → entrada, ingreso, disponibilidad, alcance, oportunidad

Durante la prueba se generaron las 5 palabras más similares usando la palabra de interés, donde el listado generado y las opciones de respuesta deben tener al menos una en común para puntuar, en caso contrario no se obtiene puntuación.

5.3 Prueba de similitud de oraciones

Se utilizó un conjunto de datos ya existente (Vera, 2025), disponible en la plataforma *Hugging Face*. Este contiene 17,533 registros, donde cada uno tiene dos oraciones y una puntuación de similitud entre ambas. Durante la prueba, para cada oración del conjunto de datos se calculó un *sentence-embedding* que surge del promedio aritmético de todos los *embeddings* obtenidos de los tokens de cada oración. Ambos *sentence-embeddings* son comparados usando la similitud del coseno con la cual se obtiene una puntuación. Finalmente, se mide la diferencia de puntuación entre la dada por el conjunto de datos y la calculada.

Las exactitudes correspondientes a cada modelo se presentan en la Tabla 2, tomando los tres tipos de prueba mencionados anteriormente.

Tabla 2. Exactitud de los modelos de *word-embeddings* en pruebas de analogías, similitud léxica y similitud semántica.

Modelo	Analogías	Palabras similares	Similitud de oraciones
model_v32_w5_c2	5.51	39.85	67.94
model_v32_w5_c3	5.17	39.96	68.44
model_v32_w5_c5	5.51	40.80	68.89
model_v64_w5_c2	8.73	49.63	70.49
model_v64_w5_c3	8.50	50.05	71.10
model_v64_w5_c5	8.61	49.21	71.48
model_v128_w5_c2	10.33	56.47	72.28
model_v128_w5_c5	8.96	56.05	73.25
model_v256_w5_c2	11.94	59.41	73.88
model_v256_w5_c5	11.71	59.73	74.73

5.4 Resultados de las pruebas

Los resultados muestran que a medida que aumenta la dimensión del vector (de 32 hasta 256), mejora la capacidad del modelo para manejar las relaciones semánticas. Esto se refleja en incrementos consistentes de exactitud en las tres tareas evaluadas: analogías, similitud léxica y similitud semántica. Cabe aclarar que las medidas de exactitud se realizan solamente para comparar los modelos y no reflejan la plena habilidad del modelo (se tiene presente que muchos de los datos manejados en las pruebas fueron generados por grandes modelos de lenguaje).

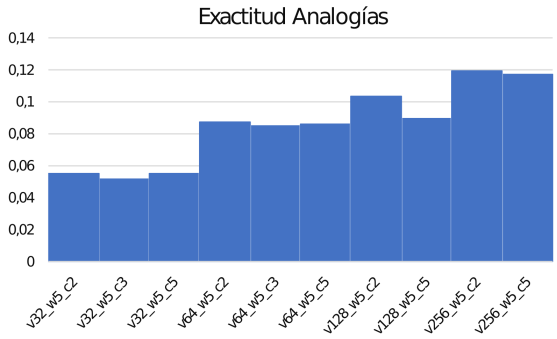

Figura 3. Gráfica de exactitud para la prueba de analogías.

En las pruebas de analogías, los valores de exactitud son más bajos en comparación con otras tareas, pero aun así se observa un progreso significativo entre el modelo más simple (*model_v32_w5_c5*) y el más complejo (*model_v256_w5_c2*). En la Figura 3, se observa un comportamiento escalonado de acuerdo con el tamaño de los vectores, donde con cada aumento de la dimensionalidad aumenta considerablemente el tamaño del modelo, pero cada vez en menor proporción la exactitud, por lo que los beneficios se van reduciendo con cada aumento. Cabe resaltar que se evidencia un comportamiento anómalo del modelo (*model_v128_w5_c5*) debido a que tiene un desfase de acuerdo a la meseta esperada.

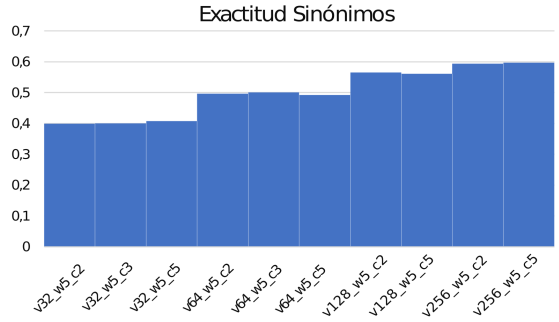

Figura 4. Gráfica de exactitud para la prueba de palabras similares.

En las pruebas de sinónimos, el comportamiento escalonado por mesetas se repite, tal como se muestra en la Figura 4. Sin embargo, tal como en el caso anterior, con cada aumento de la dimensionalidad, también aumenta considerablemente el tamaño del modelo y por ende el costo computacional pero los beneficios se van reduciendo con cada aumento. En este caso, todos los modelos tuvieron un comportamiento uniforme acorde con su dimensionalidad.

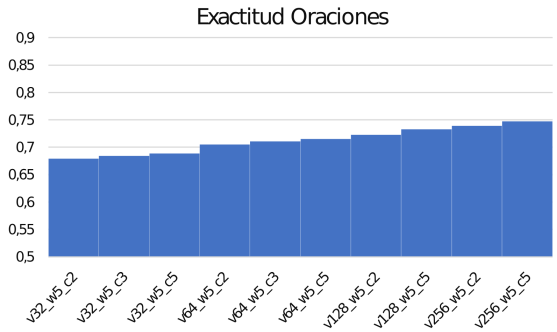

Figura 5. Gráfica de exactitud para la prueba de oraciones similares.

En las pruebas de oraciones similares, los valores de exactitud son más altos en comparación con las otras tareas y se observa una mejora lineal entre el modelo más simple y el más complejo, tal como se muestra en la Figura 5. Los cambios en la exactitud en esta prueba son leves y poco significativos en comparación con las otras pruebas. Por lo que, entre mayor dimensionalidad, mejor es el modelo para esta tarea.

5.5 Costo computacional de los modelos de *word-embeddings*

El análisis del costo computacional es fundamental al evaluar modelos de *word embeddings*, especialmente cuando se busca un equilibrio entre rendimiento y eficiencia. A medida que se incrementa la complejidad del modelo —ya sea por el tamaño del vocabulario, la dimensión de los vectores o el mínimo conteo de palabras— también lo hace el tiempo de procesamiento y el uso de recursos.

La Ecuación 1 busca representar el costo computacional de los modelos de *word embeddings* considerando tres factores fundamentales: el tamaño del modelo, el tiempo de inferencia por registro en el conjunto de datos de prueba y la exactitud del modelo para la tarea especificada. El numerador, dado por el producto entre el tamaño del modelo (en GB) y el tiempo que tarda en procesar una sola muestra (en segundos), refleja el costo computacional bruto asociado a cada predicción. El denominador, la exactitud del modelo, representa su capacidad para producir resultados correctos. Al incluir la exactitud en el denominador, la ecuación penaliza a los modelos que, aunque puedan ser rápidos o pequeños, no ofrecen buenos resultados. En conjunto, esta fórmula permite comparar modelos considerando tanto su eficiencia computacional como su rendimiento, favoreciendo aquellos que logran un buen equilibrio entre exactitud y uso de recursos.

$$Costo = \frac{Tamaño\ del\ modelo[GB] * Tiempo\ por\ registro[s]}{Exactitud} \quad (1)$$

En la Tabla 3, se analizan en detalle los tiempos de ejecución totales y los costos computacionales resultantes al aplicar distintas configuraciones de modelos de *word embeddings*, para los tres tipos pruebas realizadas: analogías, palabras similares y similitud de oraciones.

Tabla 3. Resultados de tiempo total y costo para diferentes tareas y modelos.

Modelo	Analogías		Palabras Similares		Oraciones Similares	
	Tiempo Total	Costo	Tiempo Total	Costo	Tiempo Total	Costo
model_v32_w5_c2	88.92	0.720	99.29	0.102	3.18	0.104
model_v32_w5_c3	71.18	0.414	70.20	0.048	3.13	0.068
model_v32_w5_c5	40.44	0.154	47.33	0.022	3.28	0.050
model_v64_w5_c2	99.27	0.980	117.01	0.186	3.04	0.184
model_v64_w5_c3	69.61	0.476	76.89	0.082	3.13	0.127
model_v64_w5_c5	49.19	0.231	61.64	0.046	3.05	0.086
model_v128_w5_c2	153.21	2.509	147.47	0.405	3.17	0.369
model_v128_w5_c5	67.96	0.603	77.31	0.100	3.27	0.176
model_v256_w5_c2	247.54	6.948	231.21	1.194	3.23	0.728
model_v256_w5_c5	104.45	1.406	111.71	0.270	3.19	0.334

El análisis conjunto del tiempo de ejecución y el costo computacional pone en evidencia que, si bien los modelos con mayor dimensionalidad (como los de 128 y 256 dimensiones) ofrecen mejores resultados en términos de exactitud, esto viene acompañado de un incremento considerable de los recursos necesarios para su ejecución, tal como se muestra en la Figura 6. De hecho, los modelos con configuración c5 —es decir, aquellos que utilizan un conteo mínimo de palabras más alto— tienden a presentar un mejor equilibrio entre rendimiento y eficiencia, pues reducen significativamente el tamaño del modelo sin sacrificar precisión en las tareas evaluadas.

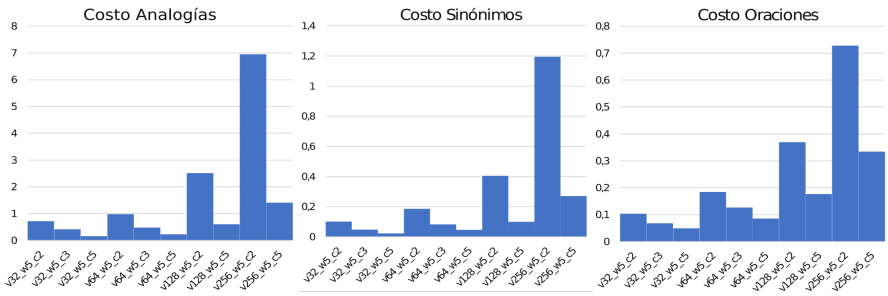

Figura 6. Gráficas de costo para cada una de las pruebas.

5.6 Elección del modelo de *word embeddings*

En la Figura 6, las gráficas de costo de las tres pruebas realizadas tienen un comportamiento similar. El parámetro que más influye es el conteo mínimo de palabras, ya que éste incrementa considerablemente el tamaño del modelo; sin embargo, el aumento de la dimensionalidad también lo hace. Esto implica que, para exactitudes similares, son más eficientes los modelos con un mayor conteo mínimo de palabras. Acorde a esto, los mejores modelos y los más eficientes son: *model_v64_w5_c5* o *model_v128_w5_c5*, ofreciendo un balance eficiente entre calidad semántica, tiempo de inferencia y consumo de recursos.

El modelo *model_v128_w5_c5* tiene un desempeño anómalo en la prueba de analogías, por lo que el modelo finalmente elegido para el **algoritmo de similitud semántica** corresponde a ***model_v64_w5_c5***, debido a que no existe una variación tan marcada en su rendimiento, pero sí es bastante más ligero.

6 Conclusiones

Se ha demostrado que los modelos Word2Vec son efectivos para capturar relaciones semánticas en tareas como analogías, sinónimos y similitud entre oraciones, especialmente cuando se entrenan con grandes corpus representativos. El desempeño semántico es evaluado mediante pruebas específicas que incluyen analogías, similitud léxica y comparación semántica entre oraciones. Estas pruebas permiten medir la capacidad del modelo para representar las relaciones de significado en el lenguaje. Se observó que el rendimiento mejora al aumentar la dimensionalidad de los vectores, lo cual se refleja en mejores resultados en las pruebas semánticas. Sin embargo, el incremento en la precisión también implica un aumento en el costo computacional. Por ello, se requiere un balance entre exactitud y eficiencia para aplicaciones prácticas. Se identificó que configuraciones con un conteo mínimo de palabras elevado, como en model_v64_w5_c5, ofrecen un equilibrio adecuado al entregar buen desempeño en tareas semánticas sin consumir excesivos recursos. El modelo model_v64_w5_c5 fue seleccionado por presentar alta precisión junto con un costo computacional reducido. Aunque model_v128_w5_c5 mostró resultados superiores en similitud semántica, su desempeño inconsistente en la prueba de analogías motivó la elección del primero como opción más eficiente para el análisis semántico.

Referencias

Alshattnawi, S., Shatnawi, A., AlSobeh, A. M. R., & Magableh, A. A. (2024). Beyond Word-Based Model Embeddings: Contextualized Representations for Enhanced Social Media Spam Detection. Applied Sciences, 14(6), 2254. Recuperado de: https://doi.org/10.3390/app14062254.

Bird, S., Klein, E., & Loper, E. (2024). *Natural Language Toolkit (NLTK) API documentation*. NLTK Project. https://www.nltk.org/api/nltk.html.

Cardellino, C. (2019). *Spanish Billion Words Corpus and Embeddings*. https://crscardellino.github.io/SBWCE/.

Lee, R. (2025). *Natural Language Processing: A Textbook with Python Implementation*. Springer Nature Singapore. https://doi.org/10.1007/978-981-96-3208-4.

López-Solaz, T., Troyano, J. A., Ortega, F. J., y Enríquez, F. (2016). "Una aproximación al uso de word embeddings en una tarea de similitud de textos en español", Procesamiento del Lenguaje Natural, vol. 57, pp. 67–74.

Mikolov, T., Chen, K., Corrado, G., & Dean, J. (2013). *Efficient Estimation of Word Representations in Vector Space*. https://arxiv.org/abs/1301.3781.

Mikolov, T., Le, Q. V., & Sutskever, I. (2013). *Exploiting Similarities among Languages for Machine Translation*. https://arxiv.org/abs/1309.4168.

OpenAI. (2024). *ChatGPT* (Version 4o mini). https://chat.openai.com/.

Řehůřek, R., & Sojka, P. (2010). Software Framework for Topic Modelling with Large Corpora. *Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks*, 45–50.

Vera J. (2023). *similarity-sentences-spanish* [conjunto de datos]. Hugging Face. https://huggingface.co/datasets/jaimevera1107/similarity-sentences-spanish.

Wikimedia F. (2023). *Wikimedia Downloads*. https://dumps.wikimedia.org.

Capítulo 6

Clasificación de enfermedad celíaca mediante modelos de deep learning basados en datos clínicos

Ana Laura Lezama Sánchez [1,2], Mireya Tovar Vidal [2]
[1] Secretaría de Ciencia, Humanidades, Tecnología e Innovación,
[2] Facultad de Ciencias de la Computación, Benemérita Universidad Autónoma de Puebla, México
{analaura.lezama,mireya.tovar}@correo.buap.mx

Resumen. En este trabajo se expone una comparación para el apoyo en el diagnóstico de la enfermedad celíaca utilizando redes neuronales artificiales. A partir de un conjunto de datos clínicos estructurados, se desarrollaron y evaluaron tres arquitecturas de redes neuronales: un modelo simple, un modelo profundo con técnicas de regularización (*Dropout*), y un modelo tipo ensamble que combina diferentes ramas neuronales. Los datos fueron preprocesados mediante codificación de etiquetas y estandarización. Cada modelo fue entrenado y validado utilizando una estrategia de *early stopping* para evitar sobreajuste. La evaluación se realizó utilizando las métricas *precisión, recall, accuracy* y *F1*. Los resultados muestran que el modelo de ensamble logra un mejor equilibrio entre las métricas, lo que sugiere su potencial en escenarios de apoyo al diagnóstico médico. Este estudio contribuye a la aplicación de técnicas de aprendizaje profundo en el área de la salud, especialmente en enfermedades autoinmunes como la celíaca.

Palabras Clave: Enfermedad celiaca, redes neuronales, aprendizaje profundo

1 Introducción

Las enfermedades autoinmunes son padecimientos en los que el sistema inmunológico se ataca a sí mismo afectando órganos y tejidos del cuerpo, provocando daños que, en muchos casos, son irreversibles. La enfermedad celiaca (EC) es una de las enfermedades autoinmunes que afecta directamente al sistema digestivo.

La enfermedad celíaca se presenta en personas con predisposición genética y tiene la característica que, por una reacción desfavorable al gluten, que es una proteína presente en el trigo, la cebada y el centeno. Esta respuesta provoca una inflamación crónica en la mucosa y submucosa del intestino delgado. Alguno de los síntomas que se manifiestan en esta enfermedad son la diarrea persistente, el estreñimiento, el dolor y el malestar abdominal. Además, puede generar manifestaciones fuera del sistema digestivo, como anemia, osteoporosis, depresión e incluso incrementar el riesgo de desarrollar algunos tipos de cáncer (Carvalho-Gomes et al., 2025). Esta enfermedad

tiene una presencia significativa a nivel global, con una incidencia estimada del 1% de la población mundial, siendo más común en mujeres. Además, se calcula que un 1.7% de los pacientes presentan síntomas, mientras que entre el 0.75% y el 1.2% son asintomáticos. En nuestro país, se estima que el 2.6% de la población padece esta enfermedad, aunque solo el 0.9% ha sido diagnosticado (Carvalho-Gomes et al., 2025).

El tratamiento principal consiste en adoptar una dieta libre de gluten y modificar el estilo de vida para permitir la recuperación de la mucosa intestinal y prevenir complicaciones. Sin embargo, especialmente en adolescentes y adultos jóvenes, realizar estos cambios puede ser un reto, ya que a menudo tienen hábitos consolidados difíciles de modificar, lo que conlleva a una persistencia de los síntomas.

El diagnóstico de esta enfermedad sigue representando un desafío clínico, debido a la gama de síntomas y a la existencia de casos asintomáticos o con manifestaciones extraintestinales. El diagnóstico requiere una combinación de pruebas serológicas, estudios histológicos y evaluación clínica, lo que se traduce en un procedimiento costoso y poco accesible al paciente (Carvalho-Gomes et al., 2025).

En este campo, la inteligencia artificial (IA) es la columna vertebral para el apoyo en el diagnostico automático de enfermedades complejas, a partir de datos clínicos. Las técnicas de aprendizaje profundo (deep learning) detectan patrones complejos y relaciones no lineales entre variables, lo que permite generar modelos capaces de detectar algún tipo de enfermedad (Jaeckle et al., 2025).

En este artículo se expone el desarrollo de tres modelos de redes neuronales con el objetivo de predecir la existencia de enfermedad celíaca en un conjunto de datos clínicos estructurados. Los modelos de Deep learning aplicados fueron una arquitectura simple, una red profunda con regularización y un modelo de tipo ensamble. El objetivo fue determinar cuál de los 3 enfoques resulta más efectivo para la clasificación de pacientes con o sin la enfermedad celiaca, contribuyendo así al desarrollo de sistemas de apoyo al diagnóstico en el ámbito de la salud.

El artículo está organizado de la siguiente manera. En la Sección 2 se presentan los conceptos fundamentales empleados en el desarrollo de esta investigación. La Sección 3 expone los trabajos revisados en el estado del arte. En la Sección 4 se describe la metodología propuesta en este trabajo, mientras que en la Sección 5 se presentan los resultados obtenidos y el conjunto de datos utilizado. En la Sección 6 se exponen las conclusiones y el trabajo futuro. Finalmente se incluyen las referencias empleadas.

2 Marco teórico

En esta sección se exponen los elementos teóricos relevantes para este estudio, incluyendo una descripción sobre las enfermedades autoinmunes y los fundamentos del aprendizaje automático, con énfasis en su aplicación en el ámbito biomédico.

Las enfermedades autoinmunes se presentan cuando el cuerpo ataca tejidos y órganos sanos del propio organismo, lo que provoca daños irreversibles. Hasta el momento, en la literatura se tiene conocimiento de la existencia de más de 80 de estas enfermedades; entre ellas la enfermedad celíaca (EC), que afecta al sistema digestivo y por lo tanto es la responsable de problemas nutricionales severes, especialmente en población infantil.

Debido al aumento en la incidencia de enfermedades autoinmunes y a la dificultad que su diagnóstico temprano puede generar, los investigadores han aportado conocimiento en el uso de herramientas basadas en inteligencia artificial (IA) para apoyar y mejorar los procesos diagnósticos. Los modelos de aprendizaje automático, y *deep learning*, han mostrado gran potencial y precisión para identificar patrones complejos en datos clínicos y contribuir a la detección de este tipo de patologías. A continuación, se describirán brevemente algunos de estos enfoques (Paredes et al., 2025).

Por otro lado, el deep learning es una rama del aprendizaje automático que se basa en redes neuronales profundas, las cuales imitan el funcionamiento del cerebro humano para aprender y reconocer patrones complejos. Este enfoque ha transformado significativamente el campo de la medicina, ya que permite procesar grandes volúmenes de datos clínicos, imágenes médicas y registros electrónicos de salud con alta precisión y velocidad. Entre los modelos más utilizados se encuentran las redes neuronales convolucionales (CNN), que se emplean comúnmente en el análisis de imágenes (como biopsias, endoscopías o resonancias magnéticas), y las redes recurrentes (RNN) o modelos basados en transformers, útiles para interpretar secuencias de texto clínico o datos temporales. En el caso de la enfermedad celíaca, el deep learning ha sido aplicado, por ejemplo, para analizar imágenes histológicas de biopsias del intestino delgado con el fin de detectar daño en las vellosidades intestinales de manera automatizada. También se han desarrollado modelos capaces de predecir el riesgo de EC a partir de información genética (como la presencia de alelos HLA DQ2/DQ8), síntomas clínicos y antecedentes familiares. Estudios recientes han implementado arquitecturas como ResNet o EfficientNet para mejorar la clasificación de imágenes de tejido intestinal, así como modelos de lenguaje natural para interpretar notas clínicas y facilitar el cribado de pacientes en sistemas de salud.

Estas herramientas no solo permiten acelerar el proceso diagnóstico, sino también reducir errores humanos, detectar casos subclínicos y mejorar la personalización del tratamiento. En conjunto, el uso de deep learning representa una innovación prometedora en el manejo de enfermedades autoinmunes, particularmente en aquellas como la enfermedad celíaca, donde el diagnóstico temprano es clave para evitar complicaciones nutricionales y sistémicas a largo plazo.

Entre las arquitecturas más utilizadas en tareas de clasificación médica con datos estructurados se encuentran las redes neuronales densas (fully connected neural networks), debido a su versatilidad y facilidad de implementación. En la literatura, se han propuesto diversas variantes de estas redes para abordar la clasificación de enfermedades a partir de datos clínicos. Una de las configuraciones más simples es la red neuronal básica, que consiste en una capa de entrada, una o dos capas ocultas densas con activación ReLU, y una capa de salida con activación sigmoide, adecuada para problemas de clasificación binaria.

Este tipo de red suele emplearse como modelo base para establecer comparaciones con arquitecturas más sofisticadas.

Otra variante común es la red neuronal profunda con Dropout, la cual incorpora múltiples capas ocultas y aplica la técnica de Dropout, que consiste en desactivar aleatoriamente un porcentaje de las neuronas durante el entrenamiento. Esta estrategia es efectiva para reducir el sobreajuste (overfitting) y mejorar la generalización del modelo, lo cual resulta especialmente útil en conjuntos de datos clínicos con tamaño limitado.

Finalmente, una arquitectura más compleja empleada en este estudio es un modelo híbrido de tipo ensemble con concatenación, el cual simula un enfoque de ensamblado dentro de una única red. En esta arquitectura, el flujo de datos se divide en dos ramas paralelas que procesan la información de forma independiente, generando representaciones distintas que posteriormente se concatenan antes de producir la predicción final. Esta estructura permite integrar múltiples perspectivas del conjunto de entrada y mejora la robustez del sistema.

3 Estado del arte

En esta sección se exponen algunos de los trabajos relacionados con la enfermedad celiaca. La cual afecta principalmente al sistema digestivo y se encuentra estrechamente vinculada con problemas nutricionales.

El avance de la inteligencia artificial (IA), especialmente en sus ramas de machine learning (ML) y deep learning (DL), ha generado cambios significativos en el diagnóstico automático de enfermedades autoinmunes como la enfermedad celíaca (EC). Esta patología, afecta aproximadamente al 1% de la población mundial, se caracteriza por una respuesta inmunitaria adversa al gluten, lo que provoca daño intestinal y una amplia variedad de síntomas digestivos y extraintestinales. En la literatura existen diversos estudios que han propuesto el uso de modelos inteligentes para mejorar la precisión e interpretabilidad en el diagnóstico. Un ejemplo de ello es el trabajo de Maram et al. (2025), que desarrollaron un modelo de deep learning para la predicción de la enfermedad celíaca, optimizado mediante Particle Swarm Optimization (PSO) con el propósito de seleccionar las características más relevantes. El enfoque logró reducir en un 35% la redundancia de datos mediante el ajuste de hiperparámetros, lo que incrementó la eficiencia del modelo en un 15%.

Adicionalmente, implementaron mecanismos de atención y técnicas de interpretabilidad como SHAP (Shapley Additive Explanations), lo que les permitió identificar los factores genéticos y serológicos más relevantes en la predicción. Los resultados obtenidos lograron una exactitud del 94.3% y disminuyeron el gasto computacional en un 20%, estableciéndolo como un instrumento con gran potencial para respaldar la toma de decisiones en el ámbito clínico. Por otro lado, en Rahmoune et al. (2025) exponen la importancia de la medicina de precisión en la gestión de la EC, incorporando información genética, clínica y de estilo de vida para prever riesgos y personalizar terapias. Para los autores la implementación de modelos de Inteligencia Artificial (IA) mejoró el diagnóstico temprano, y posibilitó intervenciones personalizadas mediante el uso de registros médicos electrónicos (EMRs) y herramientas de aprendizaje automático. Los autores mencionaron que sus experimentos permitieron una categorización exacta de la enfermedad, pronóstico de gravedad, creación de terapias personalizadas y un seguimiento constante del paciente, fomentando un enfoque clínico enfocado en la persona. En contraposición, la investigación de Jaeckle et al. (2025) se centra en vencer una restricción crucial en el diagnóstico histológico de la enfermedad celíaca: la variabilidad entre los patólogos, cuya concordancia diagnóstica apenas llega al 80%.

Para ello, propusieron un modelo de segmentación semántica entrenado con imágenes de biopsias duodenales teñidas con H&E, capaz de identificar estructuras clave como vellosidades, criptas, linfocitos intraepiteliales (IEL) y enterocitos. A partir de estas segmentaciones, calcularon métricas como la relación IEL/enterocito y la proporción vellosidad/cripta, que fueron utilizadas por un modelo de regresión logística para el diagnóstico. El sistema alcanzó una precisión del 96% en un conjunto independiente de 613 imágenes, y sus resultados demostraron una fuerte capacidad de generalización.

Adicionalmente, un segundo estudio del mismo grupo desarrolló un modelo de machine learning entrenado con más de 3,000 imágenes de biopsias provenientes de múltiples hospitales y escáneres. Para ello, los autores utilizando un enfoque semi supervisado y la técnica multiple-instance learning, este modelo alcanzó un rendimiento comparable al de patólogos especialistas, superando el 95% en precisión, sensibilidad y especificidad, y logrando un área bajo la curva ROC superior al 99%. Además, la concordancia entre el modelo y los diagnósticos humanos fue estadísticamente equivalente a la concordancia entre distintos patólogos, lo que evidencia la robustez y confiabilidad del sistema incluso en escenarios reales y variados.

Por último, Călin (2024) abordó el diagnóstico de la enfermedad celíaca mediante modelos de inteligencia artificial, utilizando el conjunto de datos público “Celiac disease (coeliac disease)” disponible en Kaggle, el mismo empleado en el presente estudio. El enfoque incluyó tanto clasificación binaria como multiclase, alcanzando métricas destacadas en la primera, con una precisión del 97 % y un F1 de 0.97 en modelos como Random Forest y XGBoost

En conjunto, estos trabajos demuestran el potencial transformador de la IA en el diagnóstico de enfermedad celíaca, al combinar precisión diagnóstica, interpretabilidad y eficiencia operativa. La integración de modelos explicables, técnicas de segmentación

y análisis multimodal (genético, clínico e histológico) representa una nueva era en el abordaje de enfermedades autoinmunes, donde la tecnología actúa como complemento clave para mejorar la toma de decisiones médicas y la calidad de vida del paciente.

En este trabajo se expone una propuesta que se basa en redes neuronales densas para la predicción de la enfermedad celíaca. Por lo que, se implementaron tres modelos en total: una red neuronal con dos capas ocultas, una red profunda que incorpora la técnica de *dropout* para mitigar el sobreajuste, y una arquitectura híbrida tipo ensemble con concatenación de ramas paralelas. Los modelos se entrenaron sobre un conjunto de datos preprocesado que incluyó codificación de variables categóricas, normalización y división en conjuntos de entrenamiento y prueba. La métrica de evaluación se centró en la exactitud, precisión, accuracy y F_1.

4 Metodología

En esta sección se expone la metodología propuesta en este trabajo.

En este trabajo se expone el enfoque propuesto, el cual está basado en el uso de aprendizaje profundo para clasificar datos clínicos estructurados de la enfermedad celíaca.

La metodología se dividió en las siguientes etapas:

Conjunto de datos: el conjunto de datos utilizado denominado celiac_disease_lab_data.csv, contiene variables relacionadas con síntomas gastrointestinales, antecedentes médicos y factores genéticos asociados al diagnóstico de enfermedad celíaca. Para garantizar la integridad de la información, se eliminaron las observaciones con valores faltantes.

Preprocesamiento: Las variables categóricas como género, tipo de diabetes, síntomas y características clínicas fueron codificadas mediante la técnica de Label Encoding. Posteriormente, se normalizaron todas las variables predictoras utilizando StandardScaler para facilitar la convergencia del modelo. La variable objetivo (Disease_Diagnose) fue también codificada en formato binario. Los datos fueron divididos en conjuntos de entrenamiento (80%) y prueba (20%) de forma estratificada.

Modelos implementados. Se desarrollaron tres modelos de redes neuronales densas para la predicción de enfermedad celíaca a partir de variables clínicas y de laboratorio.

Modelo 1 (red neuronal simple): arquitectura compuesta por dos capas ocultas densas de 32 y 16 neuronas respectivamente, con activación ReLU, seguida de una capa de salida con activación sigmoide.

Modelo 2 (red profunda con dropout): arquitectura compleja que incluye capas densas de 64, 32 y 16 neuronas, con activaciones ReLU, e incorpora capas Dropout (0.3) entre ellas para reducir el riesgo de sobreajuste.

Modelo 3 (modelo híbrido tipo ensemble): diseñado con dos ramas paralelas que procesan la misma entrada con capas de 64 y 32 neuronas respectivamente; las salidas de ambas ramas se concatenan y pasan a una capa densa intermedia (32 neuronas, ReLU), finalizando con una capa de salida sigmoide.

En los 3 modelos se empleó el optimizador Adam y la función de pérdida *binary crossentropy*. Las variables de entrada fueron previamente normalizadas con *StandardScaler* y las categóricas codificadas con *LabelEncoder*. Los datos se dividieron en 80% para entrenamiento y 20% para prueba, reservando además un 20% del conjunto de entrenamiento para validación interna. Se entrenó con un máximo de 50 épocas, tamaño de lote (*batch size*) de 32, y se aplicó la técnica de *early stopping* con paciencia de 5 épocas, restaurando los mejores pesos observados durante el entrenamiento.

Evaluación: Los modelos fueron evaluados en el conjunto de prueba mediante métricas de clasificación como *accuracy, precisión, recall* y F_1. También se guardaron las predicciones en un archivo CSV para su posterior análisis. Finalmente, se visualizaron las curvas de precisión y pérdida durante el entrenamiento para analizar el comportamiento del modelo.

5 Resultados y discusión

En esta sección se exponen los resultados obtenidos al aplicar la metodología propuesta

5.1 Conjunto de datos

El conjunto de datos utilizado en este estudio proviene originalmente del Departamento de Biotecnología de Wageningen University & Research (WUR) (Jun-Ho et al., s. f.). Su objetivo principal es facilitar la predicción diagnóstica de la enfermedad celíaca, a partir de diversas mediciones clínicas y demográficas. El conjunto de datos está compuesto por 2,206 registros y 15 variables, que incluyen características médicas como edad, género, tipo de diabetes, entre otras, además de una variable objetivo que indica la presencia o ausencia de enfermedad celíaca. Estas variables pueden emplearse potencialmente para identificar individuos con alto riesgo de desarrollar esta condición.

5.2 Resultados

Los tres modelos implementados fueron evaluados utilizando métricas de desempeño estándar en el conjunto de prueba. En términos generales, todos los modelos mostraron un rendimiento notablemente alto, con precisión, accuracy y F1 superiores al 99%.

Los resultados obtenidos por el modelo 1 (ver Tabla 1) mostraron una accuracy del 99.72%, con una precisión de 99.71%, un recall de 100% y un F1 de 99.86%. En el desglose por clase (ver Tabla 2), se observó que para la clase positiva (casos

diagnosticados con enfermedad celíaca), el modelo alcanzó valores perfectos en todas las métricas. No obstante, la clase negativa presentó un recall de 92%, lo que indica una ligera tendencia del modelo a predecir más casos como positivos.

Tabla 1 Resultados de la evaluación del modelo 1

MÉTRICA	RESULTADO
ACCURACY	99.72%
PRECISIÓN	99.71%
RECALL	100%
F1	99.86%

Tabla 2 Resultados de la evaluación por clase del modelo 1

CLASE	PRECISIÓN	RECALL	F1
NO (NEGATIVA)	100%	92%	96%
SÍ (POSITIVA)	100%	100%	100%

El modelo 2 (ver Tabla 3) presentó una ligera disminución en el rendimiento global con una accuracy de 99.16%. En la Tabla 4 se observa que, aunque mantuvo un recall del 100% para la clase positiva, su desempeño para la clase negativa fue más limitado, con un recall del 77%. Esto sugiere que, si bien el modelo fue altamente sensible a detectar casos positivos, cometió más falsos positivos al identificar sujetos sin la enfermedad.

Tabla 3 Resultados de la evaluación del modelo 2

MÉTRICA	RESULTADO
ACCURACY	99.16%
PRECISIÓN	99.14%
RECALL	100%
F1	99.57%

Tabla 4 Resultados de la evaluación por clase del modelo 2

CLASE	PRECISIÓN	RECALL	F_1
NO (NEGATIVA)	100%	77%	87%
SÍ (POSITIVA)	99%	100%	100%

En la Tabla 5, se observa que el modelo 3 mostró un equilibrio entre los dos anteriores, alcanzando una accuracy del 99.44%, precisión de 99.42%, recall del 100% y un F_1 de 99.71%. Este modelo (ver Tabla 6) logró mejorar el desempeño en la clase negativa con un recall del 85%, manteniendo una clasificación casi perfecta en la clase positiva.

Tabla 5 Resultados de la evaluación del modelo 1

MÉTRICA	RESULTADO
ACCURACY	99.44%
PRECISIÓN	99.42%
RECALL	100%
F_1	99.71%

Tabla 6 Resultados de la evaluación por clase del modelo 3

CLASE	PRECISIÓN	RECALL	F_1
NO (NEGATIVA)	100%	85%	92%
SÍ (POSITIVA)	99%	100%	100%

En general, los tres modelos ofrecieron resultados satisfactorios en la predicción de enfermedad celíaca, siendo el modelo simple el que presentó la mayor exactitud general. Sin embargo, el modelo híbrido tipo ensemble logró un mejor balance entre recall y precisión para ambas clases, lo que lo convierte en una opción particularmente robusta en contextos clínicos donde es importante minimizar tanto falsos negativos como falsos positivos.

Además, se realizó un experimento donde se incorpora la validación cruzada con 5 *folds* para cada modelo, con el fin de evaluar la estabilidad de los resultados y descartar un sobreajuste. Los valores promedio de accuracy obtenidos se exponen en la Tabla 7, donde podemos confirmar que los modelos mantienen un rendimiento estable en diferentes particiones de los datos. Dado que los tres modelos presentaron métricas promedio idénticas en la validación cruzada (accuracy = 99.39%, precision = 99.37%, recall = 100%, F_1 = 99.68%) los valores son los mismos para los tres modelos en la Tabla 7.

Tabla 7. Resultados de la validación cruzada (Accuracy promedio por Fold)

MODELO	ACCURACY	PRECISIÓN	RECALL	F_1
MODELO 1	99.39	99.37	100.00	99.68
MODELO 2	99.39	99.37	100.00	99.68
MODELO 3	99.39	99.37	100.00	99.68

Finalmente, en la Tabla 8 se presenta un ejemplo representativo de los resultados obtenidos tras aplicar los tres modelos desarrollados. En la Tabla se muestra la etiqueta real correspondiente al diagnóstico original y las predicciones realizadas por cada uno de los modelos evaluados:

Tabla 8 Ejemplo de las etiquetas originales y las predichas por cada modelo

AGE	GENDER	ETIQUETA REAL	MODELO 1	MODELO 2	MODELO 3
12	Male	yes	yes	yes	yes
12	Male	yes	yes	yes	yes
9	Female	yes	yes	yes	yes
9	Male	yes	yes	yes	yes
20	Male	yes	yes	yes	yes
21	Female	yes	yes	yes	yes
6	Male	yes	yes	yes	yes
9	Female	yes	yes	yes	yes
15	Male	no	no	no	no

6 Conclusiones

En este trabajo se expone una comparación de tres modelos de redes neuronales densas aplicadas a la clasificación de enfermedad celíaca a partir de datos clínicos estructurados. Los resultados demostraron que, modelos relativamente simples alcanzan un rendimiento muy alto, con métricas como *accuracy, precisión*, F_1 y *recall* superiores al 99%.

Por ejemplo, el modelo simple (Modelo 1) mostró una clasificación correcta casi en la totalidad de los casos, evidenciando que arquitecturas simples pueden ser suficientes para problemas de esta índole. Sin embargo, su rendimiento fue ligeramente mejorado por el modelo ensemble (Modelo 3), que integró múltiples vías de procesamiento de la información usando una arquitectura más robusta. Dicho modelo ofreció resultados

equivalentes en promedio, mostrando un desempeño estale entre clases y confirmando la calidad de la estrategia de combinar múltiples ramas de procesamiento. El modelo profundo con dropout (Modelo 2) también alcanzó métricas sobresalientes, con diferencias mínimas respecto a los otros dos.

Por lo que, estos resultados sugieren que redes neuronales densas aplicadas a datos clínicos tabulares pueden constituir una herramienta complementaria de gran valor en el diagnóstico médico, facilitando la identificación temprana y precisa de la enfermedad, lo que tendría un impacto positivo en la calidad de vida de los pacientes. Además, el uso de datos clínicos tabulares y técnicas de codificación y normalización accesibles hace que estas soluciones sean replicables y adaptables a diferentes entornos hospitalarios.

En comparación con el trabajo desarrollado por Călin, (2024), los modelos presentados en este trabajo alcanzaron métricas globales ligeramente superiores. Sin embargo, estas diferencias deben tomarse con cautela, dado que el conjunto de datos es limitado y desbalanceado.

Como trabajo futuro, se planea incorporar conocimiento adicional sobre la enfermedad celíaca con el objetivo de enriquecer los modelos desarrollados y potenciar su capacidad para ofrecer resultados aún más precisos.

Referencias

Carvalho-Gomes, I., Cruz-Bello, P., García-Hernández, M. D. L., y Osorio-Murillo, O. (2025). Efectos de una intervención de promoción de salud en población mexicana con enfermedad celíaca. Revista da Escola de Enfermagem da USP, 59, e20240408.

Călin, A. D. (2024). Machine Learning Models for Predicting Celiac Disease Based on Non-invasive Clinical Symptoms. In IFIP International Conference on Artificial Intelligence Applications and Innovations (pp. 145-159). Cham: Springer Nature Switzerland.

De la Calle, I., Ros, G., Peñalver, R., y Nieto, G., (2020). Enfermedad celiaca: causas, patología y valoración nutricional de la dieta sin gluten. Revisión. Nutrición Hospitalaria, 37(5), 1043–1051. https://doi.org/10.20960/NH.02913

Giménez, A., y Mariño, M. (2023). Estado nutricional integral de niños y adolescentes con enfermedad celíaca. Anales Venezolanos de Nutrición, 36(2), 55–66. https://doi.org/10.54624/2023.36.2.002

Jaeckle, F., Denholm, J., Schreiber, B., Evans, S. C., Wicks, M. N., Chan, J. Y., y Soilleux, E. (2025). Machine learning achieves pathologist-level celiac disease diagnosis. NEJM AI, 2(4), AIoa2400738.

Jaeckle, F., Bryant, R., Denholm, J., Diaz, J. R., Schreiber, B., Shenoy, V., y Soilleux, E. (2025). Interpretable Machine Learning based Detection of Coeliac Disease. [Preprint]. medRxiv. https://doi.org/10.1101/2025.03.11.25323763

Jiménez, I., Martínez, B., Salas, D., Martínez García, M., y González, G. (2021). Evaluando la desnutrición en pediatría, un reto vigente. Nutrición Hospitalaria, 38(2), 64–67. https://scielo.isciii.es/scielo.php?script=sci_arttext&pid=S0212-16112021000500015

Jun-Ho, Prof., Tran Mangala, Prof., Jason Koul, Prof., y Jack Win, Dr. (s. f.). Celiac disease prediction dataset [Data set]. Wageningen University & Research. https://www.wur.nl/en.htm

Maram, B., y Maram, R. R. (2025). Particle Swarm Optimization based Explainable Deep Learning Models for Celiac Disease Prediction. In 2025 Fourth International Conference on Smart Technologies, Communication and Robotics (STCR) (pp. 1-8). IEEE.

Paredes, L. S. M., y Torres, J. M. T. (2025). Asociación entre la composición de la microbiota intestinal: Relación con la enfermedad celiaca y desnutrición. Ciencia y Educación, 6(2), 196-211.

Rahmoune, H., Boutrid, N., y Benchoufi, I. (2025). Precision medicine in celiac disease: A step ahead. Artificial Intelligence in Gastroenterology, 6(1).

Capítulo 7

MexaFood: una ontología para el manejo de dietas en el contexto mexicano

Cecilia Reyes-Peña [1,2], Jesús García-Ramírez[1,2], Marco Antonio Camacho Duran[2], Ricardo Ramos Aguilar[1]

[1] Unidad Profesional Interdisciplinaria de Ingeniería Campus Tlaxcala IPN, Tlaxcala, México

[2] Universidad Politécnica Metropolitana de Hidalgo, Hidalgo, México

{creyespe, jegarciara, rramosa}@ipn.mx, 213220137@upmh.edu.mx

Resumen. El consumo desmedido de alimentos puede contribuir a enfermedades como obesidad y diabetes. Es fundamental que la información nutrimental de los alimentos disponibles para la población mexicana sea organizada y se pueda interpretar eficazmente por humanos y computadoras. Las ontologías ofrecen una representación formal que facilita la interpretación de la información y respalda la toma de decisiones por parte de expertos. Sin embargo, algunas ontologías existentes en la literatura requieren hardware especializado (sobre todo memoria RAM) y su reutilización puede resultar complicada debido a la escasez de módulos bien definidos en estas. Para abordar esta limitación, presentamos el diseño semiautomático de MexaFood, una ontología basada en el Sistema Mexicano de Alimentos Equivalentes (SMAE), diseñada en OWL y enriquecida con conocimiento implícito extraído del algoritmo de agrupamiento DB-Scan, teniendo 41 clases y 3253 individuos. MexaFood fue evaluada utilizando criterios relevantes en la ingeniería ontológica como consistencia, competencia y satisfactibilidad teniendo un desempeño aceptable.

Palabras Clave: Diseño ontológico · Información de dietas mexicanas · Representación del conocimiento.

1 Introducción

La riqueza cultural y la diversidad gastronómica de México no sólo son motivo de orgullo nacional, sino también una responsabilidad colectiva en cuanto a su preservación y consumo. A pesar de la abundancia de información nutricional, el acceso y la difusión de datos nutricionales sobre los alimentos endémicos mexicanos siguen siendo escasos y limitados, lo cual puede generar mitos en relación con la ingesta de estos. A pesar del gran tamaño del conjunto de alimentos disponibles para la población mexicana, ya sea por el origen de estos o por la accesibilidad económica, sus valores nutricionales, clasificación taxonómica, orígenes culturales y geográficos, no han sido estructurados de forma estandarizada para que faciliten el procesamiento y la comprensión de estos. Sumado a esto, en diversas herramientas relacionadas con la dieta, esta información suele estar

fragmentada o minimizada, lo que impide una comprensión integral de su impacto en la salud y la nutrición. Una representación estructurada de estos datos es crucial para diseñar sistemas robustos que integren la información sobre salud, ejercicio y alimentación de forma coherente y accesible.

Las ontologías al ser una: "*representación explícita del conocimiento en un dominio específico*" (Gruber, 1993), proporcionan una solución para representar información sobre nutrición e información dietética mexicana. Estas estructuras incluyen un vocabulario bien definido y relaciones que establecen los roles semánticos entre conceptos, proporcionando claridad y coherencia a la información disponible (Madalli et al., 2017). La necesidad de utilizar herramientas basadas en datos estructurados radica en identificar la compleja relación entre la dieta y el estado de salud de cada persona (Castellano-Escuder et al., 2020).

Existen plataformas enfocadas al manejo de planes alimenticios como Nutrify[1] o MyFitnessPal[2] que ofrecen interfaces atractivas y amplios catálogos de alimentos. Sin embargo, omiten consideraciones fundamentales como la accesibilidad de los alimentos para la mayor parte de la población y la pertinencia cultural de los sabores y las preparaciones tradicionales. Esto puede generar renuencia por parte de los pacientes respecto a sus indicaciones dietéticas. Adicionalmente, desde una perspectiva tecnológica, estas plataformas no integran mecanismos para la reutilización o adaptación del conocimiento, lo cual podría representar un obstáculo para el diseño de herramientas que fomenten la colaboración efectiva de profesionales de la salud.

Según Madalli et al. (2017), para que un sistema en el dominio alimentario sea reutilizable por otros, debe ser autosuficiente y capaz de responder preguntas básicas como la composición de un plato en términos de ingredientes y valores nutricionales. Esto se logra mediante un modelado de datos riguroso y una granularidad adecuada de estos, ya que la granularidad permite establecer detalles propios de la información. Además, la incorporación de biomarcadores permitiría generar recomendaciones personalizadas basadas en el metabolismo individual (Castellano-Escuder et al., 2020), ampliando el impacto positivo de estas herramientas en la salud pública.

Este trabajo presenta el diseño semiautomático de MexaFood, una ontología para modelar, estructurar y enriquecer la información contenida en el Sistema Mexicano de Alimentos Equivalentes (SMAE) (Salmerón Campos, 2022), utilizando la metodología de diseño ontológico Methontology (Fernández-López et al., 1997). Esto representa un desafío significativo en el campo del diseño de ontologías, ya que modelar el SMAE en una ontología requiere definiciones lógicas complejas, así como un nivel apropiado de granularidad de datos para mejorar la reusabilidad de la ontología resultante por parte de otros sistemas. Además, para facilitar la identificación de alimentos equivalentes, se empleó el algoritmo de agrupamiento DB-Scan (Ester et al., 1996). Este algoritmo permite obtener instancias similares basados en la distribución espacial de las

[1] www.nutrify.io

[2] www.myfitnesspal.com

características utilizadas, teniendo como ventaja principal la identificación automática del número de grupos resultantes (clústeres), lo que nos otorga una ventaja sobre otros algoritmos de agrupamiento basados en centroides como K-Means.

La evaluación de MexaFood sigue un enfoque de verificación, que incluye la resolución de preguntas de competencia y la verificación de criterios como la consistencia y satisfactibilidad, teniendo resultados favorables y respaldando la robustez del modelo propuesto, así como su aplicabilidad en sistemas relacionados con la nutrición y la salud. Además, se evaluó el desempeño de DB-Scan con el coeficiente de silueta promedio resultando en un buen rendimiento de creación de los clústeres.

Este documento se organiza de la siguiente manera: la Sección 2 presenta una revisión de la literatura sobre el diseño de ontologías en el dominio de la nutrición. La Sección 3 describe las aplicaciones de Methontology utilizando los datos seleccionados. Los resultados y la discusión se presentan en la Sección 4. Finalmente, las conclusiones y el trabajo futuro se abordan en la Sección 5.

2 Trabajos relacionados

Las ontologías se han consolidado como modelos relevantes en el área médica incluyendo el ámbito de la alimentación, especialmente para abordar el desafío de la gestión de información en este campo y la toma de decisiones supervisada. FoodWiki (Çelik, 2015) es un sistema móvil que utiliza una ontología para sugerir opciones de alimentos saludables. Se basa en ingredientes, información nutricional, tasa metabólica basal y aditivos en alimentos procesados. Las reglas semánticas que continene, ayudan a identificar riesgos alergénicos, permitiendo así recomendaciones dietéticas para objetivos específicos, como el combate de la obesidad y al sobrepeso. Esta ontología fue desarrollada en Protégé usando OWL 2-DL y adaptada al marco de trabajo SCRUM (Sambola, 2021).

Una de las enfermedades estrechamente relacionadas con la dieta es la Diabetes Mellitus Tipo 2. En este contexto, Seneviratne et al. (2021) desarrollaron una ontología para recomendar dietas personalizadas a pacientes diabéticos, tomando en cuenta sus preferencias gustativas, regulando por medio de reglas las recomendaciones. Por su parte, Rivera et al. (2018) generaron un modelo nutrimental utilizando la metodología Methontology. Dicho modelo se presenta como un recurso clave para apoyar a los expertos en nutrición en el control y seguimiento de dietas. Gracias a su estructura y a su validación por expertos, este modelo ontológico puede representar correctamente datos nutricionales que permiten a los expertos en salud alimentaria tomar decisiones dietéticas informadas y personalizadas. Además, incluye reglas semánticas para identificar posibles riesgos alérgicos basándose en la similitud de ciertos químicos.

FoodOn (Dooley et al., 2018) es una ontología clave en la literatura, cuyo objetivo se centra en abordar las deficiencias en la terminología de los productos alimentarios a nivel

mundial. Esta ontología sirve como una herramienta esencial para establecer estándares comunes en la representación ontológica de los alimentos, promoviendo la interoperabilidad y la comprensión global en el sector alimentario. Al integrar varios sistemas y lenguajes de descripción, esta ontología mejora la gestión de la información alimentaria y contribuye a fortalecer la seguridad alimentaria y la nutrición a nivel global. Otra ontología popular en la literatura es FOBI (Castellano-Escuder et al., 2020), la cual destaca por su composición modular basada en ChEBI (Degtyarenko et al., 2007) y FoodOn (Dooley et al., 2018). Su contribución radica en la capacidad de integrar datos heterogéneos y complejos relacionados con el análisis de biomarcadores de ingesta calórica y composiciones químicas de los alimentos, ofreciendo una perspectiva detallada sobre los impactos nutricionales de diferentes alimentos y como estos afectan a la salud y bienestar de un paciente.

En Madalli et al. (2017), se presenta una metodología para construir una ontología central en el dominio alimentario, basada en la teoría de clasificación analítico-sintética, con la finalidad de construir una base de conocimiento que pueda extenderse y adaptarse para aplicaciones específicas dentro del sector alimentario. Para motivar el desarrollo de la ontología central de alimentos, se describen una serie de escenarios de aplicación. La idea es que la ontología central se utilice para crear ontologías específicas de aplicación para estos escenarios.

Ante esta revisión de la literatura, es evidente que existe una carencia significativa de ontologías que resalten la existencia de alimentos endémicos y su información nutrimental, además de que algunas de estas demandan una excesiva cantidad de recursos computacionales, aunado a diseños estructurales complejos que hacen imposible el extraer módulos bien definidos para reutilizar. Este trabajo propone una representación ontológica aplicada al contexto mexicano, para lo cual utilizamos el SMAE (Sistema Mexicano de Alimentos Equivalentes), que proporciona la información necesaria. Adicionalmente, el diseño de la ontología propuesta desde cero puede integrarse con ontologías previamente desarrolladas en el dominio de la población mexicana (Reyes-Peña et al., 2021). Esto ofrece una ventaja en términos de requisitos computacionales en comparación con otras ontologías más complejas (Çelik, 2015; Degtyarenko et al., 2007; CastellanoEscuder et al., 2020).

3 Diseño de MexaFood

En esta sección, presentamos el diseño y la construcción de la ontología propuesta, a la que hemos denominado MexaFood. La ontología fue construida siguiendo la metodología de diseño Methontology (Fernández-López et al., 1997), la cual es ampliamente estudiada para la construcción de ontologías. En las siguientes subsecciones, detallamos cada etapa de este proceso.

3.1 Especificación

El propósito de esta etapa inicial es el detallar los requisitos para el desarrollo de la ontología MexaFood por medio de definiciones sobre la formalidad y la posible utilización de esta, los cuales se mencionan a continuación.

1. Propósito de la ontología: Proporcionar un modelo formal para representar el conocimiento relacionado con las dietas basados en alimentos disponibles para la población mexicana, sus ingredientes, que provea una base de conocimiento para el modelado de restricciones y recomendaciones nutricionales acorde a perfiles de salud.
2. Alcance de la ontología: Las dietas a incluir serán consideradas a partir de casos reales y sintéticas (generadas por inteligencia artificial) en casos de personas con estados de salud no críticos, y que respeten la disponibilidad de alimentos. El nivel de granularidad de la información debe plantearse de forma adecuada para que la ontología pueda interactuar con otras ontologías u otros sistemas que provean cálculos en función de tasas metabólicas, contenido calórico de alimentos, pérdidas y ganancias de peso a alcanzar, entre otros. Es importante recalcar que debe existir un equilibrio entre la definición de la granularidad y la demanda computación de la ontología resultante, ya que una granularidad fina implica mayor demanda computacional.
3. Usuarios y casos de uso: nutriólogos y otros expertos en ontologías.

Preguntas de competencia. Las preguntas de competencia ayudan a determinar qué conocimiento debe representar la ontología, brindando una guía de conceptos y relaciones para identificar qué clases, propiedades y restricciones se necesitan. Además, apoyan a verificar que efectivamente la ontología permite obtener respuestas correctas. Las preguntas planteadas se muestran a continuación:

1. ¿Cuáles son los alimentos que tienen un índice glucémico mayor a 50?
2. ¿Cuántos alimentos pertenecen a las verduras?
3. ¿Cuáles son los grupos de alimentos que se utilizan en el SMAE (2022)?
4. ¿Quiénes son los pacientes que tienen dietas donde se incluye el brócoli?
5. ¿Qué alimentos se incluyen en el grupo de frutas y verduras?
6. ¿Qué alimentos se incluyen en el grupo de cereales y tubérculos?
7. ¿Cuántos gramos de grasa contiene una porción de lácteos según el SMAE?
8. ¿Cuántos gramos de proteína contiene una porción de carne?
9. ¿Cuántas dietas contienen alimentos pertenecientes a los cereales y tubérculos?
10. ¿Cuántos individuos pertenecientes a la clase Pacientes tienen asignada una dieta para bajar de peso?

Identificación de las fuentes de conocimiento. La adquisición de conocimiento para la ontología MexaFood se enfoca en recopilar información para la gestión de dietas dentro de la población mexicana. Se extraerán datos relevantes mediante búsquedas en línea utilizando palabras clave como *dietas* y *nutrición*. Las fuentes clave incluyen el Sistema Mexicano de Alimentos Equivalentes (SMAE), que ofrece datos nutricionales completos sobre una amplia variedad de alimentos mexicanos, junto con diversos recursos que cubren enfoques dietéticos diversos, incluyendo aquellos para la pérdida de peso, el aumento de peso y condiciones como la diabetes mellitus.

Para asegurar la relevancia y utilidad de los materiales recopilados, se han establecido criterios rigurosos. Por ejemplo, se excluirán los recursos en otros idiomas debido a los costos y complejidades de traducción asociados, un desafío evidenciado por las ontologías alimentarias existentes en inglés. De manera similar, se evitarán las ontologías centradas en alimentos de otros países, ya que no se alinean con el énfasis de este proyecto en las dietas mexicanas.

Además, en esta etapa se ha revisado manualmente información sobre gestión de dietas de expertos en nutrición, ya que se cuenta con dietas reales diseñadas y adaptadas a los objetivos individuales de pacientes en específico. Dichas dietas están enfocadas a la pérdida de peso, y a sus necesidades específicas. Para crear un conjunto de dietas mayor, se utilizará ChatGPT para generar dietas acordes a perfiles reales de pacientes.

3.2 Conceptualización

Para la conceptualización de la ontología es necesario hacer una elicitación de términos relevantes dentro del SMAE que permita trazar ejes de clasificación relevantes del dominio el diseño taxonómico de la ontología, así como la relación semántica que guardan estos ejes entre sí y los valores descriptivos que permitan explotar la representación del conocimiento.

Estos términos se usaron como palabras clave en búsquedas dirigidas para identificar clases, propiedades y jerarquías relacionadas con alimentos, dietas, menús, pacientes y planes de comidas. A través de este proceso, se modelaron los conceptos clave y sus interrelaciones dentro del dominio. Los diagramas y modelos conceptuales resultantes facilitaron la visualización y comprensión de la estructura ontológica, estableciendo una base sólida para la siguiente fase de la metodología.

Las clases principales de la ontología se muestran en la Figura 1, donde se puede observar que la información relacionada con los alimentos del SMAE se representa en la clase Alimento. Por otro lado, la taxonomía de dietas y menús fue definida para abordar la necesidad de información personalizada sobre el individuo. Las propiedades de objeto establecen relaciones entre individuos que pertenecen a clases distintas, articulando enlaces semánticos complejos. Por ejemplo, la propiedad *tieneDieta* relaciona la clase *Dieta* con la clase *Paciente*. Por el contrario, las propiedades de datos vinculan individuos a valores literales como la propiedad *tieneEdad,* asociada a la clase *Paciente*.

Durante esta etapa, se han extraído un total de 2,372 registros del SMAE, cubriendo nueve tipos de alimentos. Las características de estos alimentos varían según su grupo alimentario, compartiendo únicamente los siguientes atributos: nombre del alimento, energía, cantidad sugerida, unidad de medida, peso redondeado, carbohidratos, lípidos, proteínas y peso neto. Estas características son representadas como propiedades de dato (ver Figura 2).

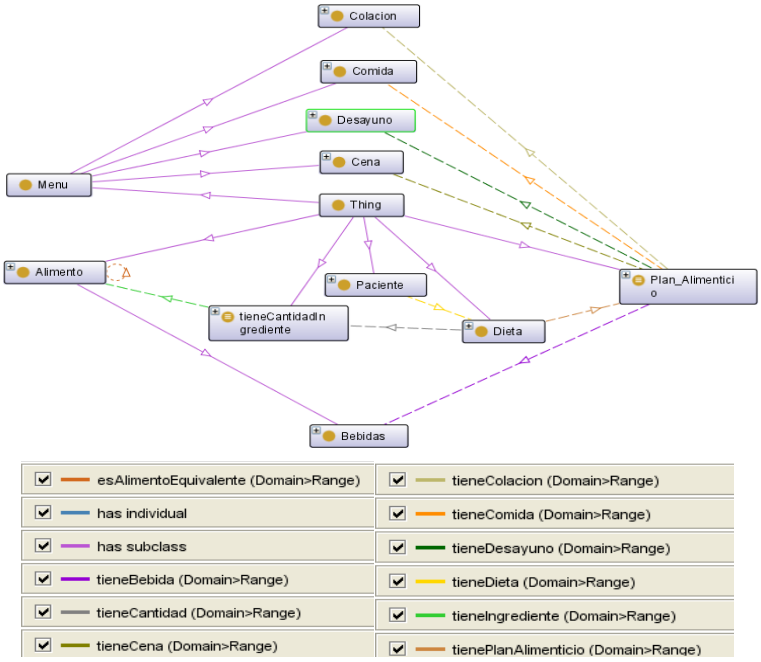

Figura 1. Diagrama general de MexaFood.

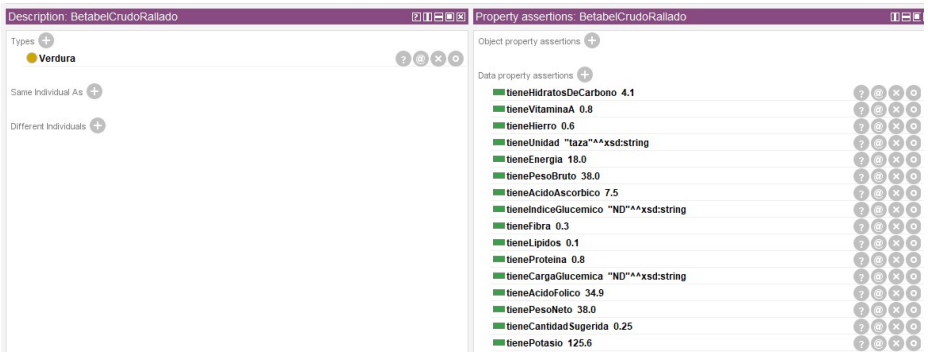

Figura 2. Ejemplo de las propiedades de objeto de un individuo perteneciente a la taxonomía de alimento.

3.3 Integración

En esta fase del desarrollo de la ontología para el manejo de dietas mexicanas, no se contempla la integración con ontologías externas existentes debido a las implicaciones de alto costo computacional que esto conlleva. Si bien la reutilización de vocabularios y esquemas previamente desarrollados como FoodOn (Dooley et al., 2018) puede enriquecer la semántica y favorecer la interoperabilidad, su adopción también introduce una carga adicional en términos de procesamiento, mantenimiento y complejidad semántica, que resulta incompatible con el objetivo de brindar la base de conocimiento para otros sistemas.

La ontología resultante busca ofrecer un modelo ligero y funcional, que permita representar de manera clara y estructurada la información esencial relacionada específicamente con dietas mexicanas, sus ingredientes, características nutricionales y adecuación a distintas condiciones de salud. Este modelo está diseñado para ser fácilmente interpretable y explotable por sistemas externos, como aplicaciones móviles, sistemas expertos o bases de datos, sin requerir razonadores complejos ni infraestructuras semánticas de alto nivel. Por lo tanto, la ontología debe priorizar la baja demanda de recursos computacionales a través de un diseño minimalista. No obstante, también se debe considerar la posibilidad de considerar futuras extensiones de la ontología que permitan su alineación o fusión con ontologías más robustas.

3.4 Implementación

En la etapa de implementación, se construye el modelo conceptual previamente definido mediante su traducción a un lenguaje formal, permitiendo su representación computacional, evaluación y posterior explotación por sistemas externos. Para implementar MexaFood se plantea el uso de la herramienta Protégé, ya que es un entorno

de desarrollo gráfico ampliamente utilizado para la creación y gestión de ontologías en formato OWL (Web Ontology Language). Otro aspecto que refuerza la selección de Protégé para la implementación de MexaFood son las extensiones disponibles para realizar consultas, como por ejemplo la extensión para interpretar SPARQL (*Simple Protocol and RDF Query Language*). Además, permite la fácil integración de mecanismos de razonamiento con HermiT y Pellet.

Es importante señalar que, si bien estos lenguajes son un estándar para la representación del conocimiento, la demanda computacional para el razonamiento y las consultas en MexaFood es alta debido a las 3253 instancias identificadas en etapas anteriores.

3.5 Evaluación

La etapa de evaluación se encarga de establecer y aprobar criterios que indiquen que el diseño y la construcción de la ontología han sido correctos. A continuación, se mencionan los criterios considerados y la forma de aplicarlos:

1. Competencia: este criterio mide la capacidad que tiene la ontología para definir y organizar la información de un dominio y esta se mide a través de la respuesta de las preguntas de competencia mediante consultas SPARQL específicas, haciendo referencia al criterio de competencia. El objetivo principal de esto es identificar la capacidad de la ontología para recuperar información precisa y relevante en respuesta a diversas solicitudes. Estas preguntas se diseñaron cuidadosamente para cubrir un amplio rango de aspectos relacionados con la representación de la información alimentaria en el SMAE, evaluando así tanto la efectividad como la precisión de la ontología.
2. Consistencia: este criterio indica la buena construcción de la ontología y es establecida por el uso de un razonador, el cual verifica que no existan contradicciones lógicas dentro de las declaraciones formales de las clases y las restricciones.
3. Satisfactibilidad: este criterio mide que exista por lo menos una verificaban si se podía construir una instancia finita no vacía de una clase que exista sin violar ninguna restricción de las definiciones.

3.6 Extracción automática de conocimiento

La extracción automática de conocimiento tiene como finalidad extraer conocimiento implícito obtenido a través de la distribución de los datos. Para esto se utilizan las instancias de alimentos para indicar si se pueden identificar alimentos cuyo aporte nutrimental es similar. Esto puede contribuir a reemplazar alimentos dentro de las dietas

que no sean del agrado del usuario destino. Para lograr identificar esta similitud se propone aplicar el algoritmo DB-Scan, ya que este algoritmo agrupa instancias similares a partir de su distribución geográfica en un hiperplano. DB-Scan ofrece una ventaja en particular, define por sí mismo la cantidad de clústeres resultantes. Se asume que las instancias que pertenecen a un mismo clúster son equivalentes entre sí y dentro de la ontología esta equivalencia será representada mediante la propiedad de objeto *esAlimentoEquivalente*, cuyo rango y dominio es la clase Alimento, y es una propiedad transitiva y simétrica.

4 Resultados y discusión

En esta sección, presentamos los resultados experimentales obtenidos para la ontología MexaFood. En primer lugar, se detallan las métricas de la ontología, las cuales se pueden observar en la Tabla 1. MexaFood alcanza el nivel alcanza el nivel SIQ(D) en lógica descriptiva que acuerdo con Baader et al. (2003), esta notación combina el uso de roles transitivos (S), roles inversos (I), restricciones de cardinalidad cualificada (Q) y roles de datos (D).

Tabla 1. Métricas de MexaFood.

Elementos	Cantidad
Axiomas	36407
Axiomas lógicos	32692
Declaración de axiomas	3343
Clases	41
Propiedades de objeto	10
Individuos	3253
Anotaciones	1
Relaciones SubClassOf	35
Clases equivalentes	7
Clases disjuntas	3

Tras aplicar el algoritmo DB-Scan a los datos descritos en la subsección 3.3, encontramos 58 clústeres. La calidad del agrupamiento se evaluó utilizando el coeficiente de silueta, el cual arrojó un valor de 0.227. Esto indica un nivel moderado de cohesión y separación entre los clústeres. Adicionalmente, se observó una tasa de ruido del 5.42%, reflejando la proporción de puntos de datos que no fueron asignados a ningún clúster debido a su dispersión o falta de similitud con otras instancias. Los clústeres obtenidos son proyectados mediante T-SNE (Incrustación Estocástica de Vecinos con Distribución t), la cual al ser una técnica de reducción de dimensionalidad (Van der Maaten and Hinton, 2008), transforma el vector de alta dimensionalidad de los alimentos (vector de 9 características nutrimentales) en un vector de dos dimensiones abstractas para su proyección en un plano 2D. La Figura 3 muestra la existencia de clústeres bien definidos

en el conjunto de datos. La integración de estos resultados dentro de MexaFood se realiza por medio de la propiedad de objeto *esAlimentoEquivalente*, la cual es una propiedad transitiva que relaciona a cada una de las instancias con el resto de las instancias que pertenecen a un mismo clúster, como se muestra en la Figura 4. En total se obtuvieron 36 pares de instancias equivalentes declaradas.

Para evaluar la ontología, se aplicó el criterio de competencia a través de consultas SPARQL diseñadas para responder a las preguntas de competencia establecidas en las etapas iniciales (ver Tabla 2), al igual que el criterio de consistencia para indicar si existe o no contradicciones dentro del diseño y la construcción de la ontología usando los razonadores HermiT y Pellet. El resultado de la consistencia confirmó que no existe ninguna contradicción en la definición lógica de la ontología. A su vez, la ontología demostró ser capaz de responder a las preguntas de competencia previamente definidas. Por lo tanto, podemos concluir que esta forma del modelo es competente con los requisitos establecidos. Otro de los criterios utilizados fue la satisfactibilidad, la cual indica que existe por lo menos una instancia que satisface las definiciones lógicas establecidas dentro de la ontología. Por lo que, se recopilaron dietas generadas por expertos en nutrición y dietas generadas con ChatGPT para poblar la ontología con ejemplos y verificar que es capaz de albergar una dieta completa.

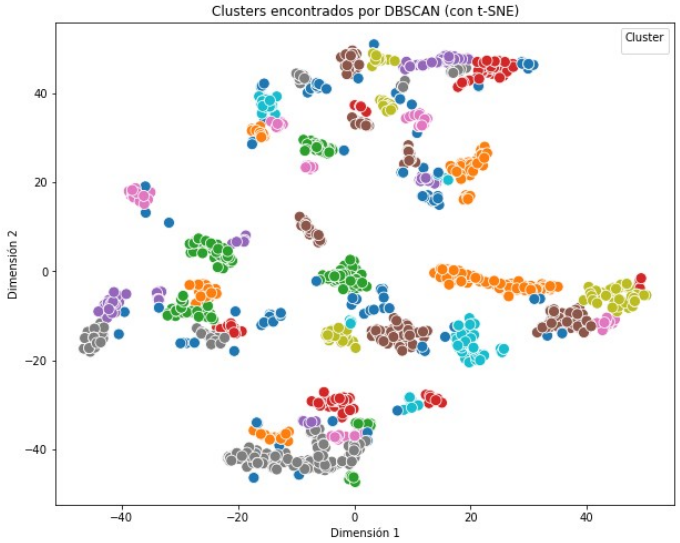

Figura 3. Resultado de agrupamiento usando las instancias de la taxonomía de alimentos.

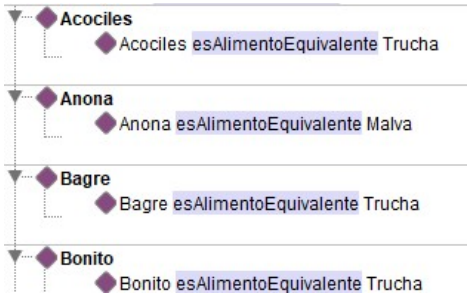

Figura 4. Equivalencia de alimentos determinada por clústeres.

5 Conclusiones

Se desarrolló MexaFood, una ontología para la gestión de dietas en la población mexicana basada en el Sistema Mexicano de Alimentos Equivalentes (SMAE) y siguiendo la metodología Methontology. Su evaluación, mediante preguntas de competencia, consultas SPARQL y los razonadores HermiT y Pellet, confirmó su precisión y eficiencia en la representación de datos nutricionales.

La ontología no solo modela patrones dietéticos complejos, sino que también sirve como un modelo para la clasificación de alimentos y el apoyo en el manejo de dietas. Las consultas realizadas han demostrado la capacidad de recuperar información adecuada sobre dietas y nutrición. Respecto a la granularidad de la información nutricional que contiene, MexaFood es lo suficientemente detallada para capturar las características nutricionales de cada alimento, permitiendo así una representación que puede ser explotada en el contexto de salud alimentaria.

Aunque la identificación de equivalencias entre alimentos requiere datos adicionales, la formación de clústeres coherentes indica un enfoque apropiado. El trabajo futuro incluye la validación por un experto de MexaFood y la integración de datos sobre actividad física y gasto calórico, así como el desarrollo de un generador automático de recetas.

Tabla 2. Respuestas a las preguntas de competencia.

Pregunta de competencia	Consulta en SPARQL	Respuesta de MexaFood
¿Cuáles son los alimentos que tienen un índice glucémico mayor a 50?	SELECT ?alimentos ?cantidadIG WHERE{?alimentos alimento:tieneIndiceGlucemico ?cantidadIG FILTER(?cantidadIG>=50)}	?alimentos: ?cantidadIG alimento:AcelgaCruda: 64.0 alimento:Arugula: 295.2 alimento:Ate: 51.0 Total de resultados: 54
¿Cuántos alimentos pertenecen al grupo de las verduras?	SELECT ?verdura WHERE {?verdura a alimento:Verdura}	?verdura alimento:HongoPortobelloCrudo alimento:MorillasFrescas alimento:FlorDeCalabazaCrudaPicada Total de resultados: 150
¿Cuáles son los grupos de alimentos indicados en el SMAE (2022)?	SELECT ?gruposSMAE WHERE {?gruposSMAE rdfs:subClassOf alimento:Alimento}	?gruposSMAE alimento:AceitesYGrasasConProteinas alimento:Fruta_Y_Verdura alimento:Leguminosa Total de resultados: 30
¿Cuáles son los alimentos incluidos en las frutas y verduras?	SELECT ?frutaverdura ?clase WHERE {?frutaverdura a ?clase .?clase rdfs:subClassOf alimento:Fruta_Y_Verdura}	?frutaverdura: ?clase alimento:PeraDeshidratada: alimento:Fruta alimento:Grosellas: alimento:Fruta alimento:Lima: Alimento:Fruta Total de resultados: 756
¿Qué alimentos se incluyen en el grupo de cereales y tubérculos?	SELECT ?cerealestuberculos ?clase WHERE {?cerealestuberculos a ?clase .?clase rdfs:subClassOf alimento:Cereal_Y_Tuberculo}	?cerealestuberculos: ?clase alimento:GalletaDeAvena: CerealTuberculoConGrasa alimento:Panque: CerealTuberculoConGrasa alimento:PanDeMuesli: CerealTuberculoConGrasa Total de resultados: 538

Referencias

Baader, F., Calvanese, D., McGuinness, D., Nardi, D., & Patel-Schneider, P. F. (Eds.). (2003). *The description logic handbook: Theory, implementation, and applications*. Cambridge University Press.

Castellano-Escuder, P., González-Domínguez, R., Wishart, D. S., AndrésLacueva, C., and Sánchez-Pla, A. (2020). Fobi: an ontology to represent food intake data and associate it with metabolomic data. *Database*.

Çelik, D. (2015). Foodwiki: Ontology-driven mobile safe food consumption system. *The scientific World journal*, 2015(1):475410.

Degtyarenko, K., De Matos, P., Ennis, M., Hastings, J., Zbinden, M., McNaught, A., Alcántara, R., Darsow, M., Guedj, M., and Ashburner, M. (2007). Chebi: a database and ontology for chemical entities of biological interest. *Nucleic acids research*, 36(suppl_1):D344–D350.

Dooley, D. M., Griffiths, E. J., Gosal, G. S., Buttigieg, P. L., Hoehndorf, R., Lange, M. C., Schriml, L. M., Brinkman, F. S., and Hsiao, W. W. (2018). Foodon: a harmonized food ontology to increase global food traceability, quality control and data integration. *npj Science of Food*, 2(1):23.

Ester, M., Kriegel, H.-P., Sander, J., Xu, X., et al. (1996). A density-based algorithm for discovering clusters in large spatial databases with noise. In *kdd*, volume 96, pages 226–231.

Fernández-López, M., Gómez-Pérez, A., and Juristo, N. (1997). Methontology: from ontological art towards ontological engineering.

Gruber, T. R. (1993). A translation approach to portable ontology specifications. *Knowledge acquisition*, 5(2):199–220.

Madalli, D. P., Chatterjee, U., and Dutta, B. (2017). An analytical approach to building a core ontology for food. *Journal of Documentation*, 73(1):123–144.

Reyes-Peña, C., Tovar, M., Bravo, M., and Motz, R. (2021). An ontology network for diabetes mellitus in mexico. *Journal of Biomedical Semantics*, 12:1–18.

Rivera, E., Chavira, G., Ahumada, A., Ramírez, C., and Zavala, M. (2018). Construcción y evaluación de un modelo ontológico nutricional. *Revista Iberoamericana de Ciencias*, pages 1–9.

Salmerón Campos, R. M. (2022). El sistema mexicano de alimentos equivalentes.

Sambola, D. M. (2021). Ontología de recomendación dietética dirigida a persona con sobrepeso y obesidad, una herramienta alterna en tiempos de covid. *Ciencia e Interculturalidad*, 28(01):76–85.

Seneviratne, O., Harris, J., Chen, C.-H., and McGuinness, D. L. (2021). Personal health knowledge graph for clinically relevant diet recommendations. *arXiv preprint arXiv:2110.10131*.

Van der Maaten, L. and Hinton, G. (2008). Visualizing data using t-sne. *Journal of machine learning research*, 9(11).

Capítulo 8

Diagnóstico automatizado de enfermedades autoinmunes cutáneas a través de imágenes médicas y aprendizaje profundo

Alejandro Camilo Merced Reyes[1], Mireya Tovar Vidal [2], Ana Laura Lezama Sánchez [2,3]
[1] Tecnológico de Estudios Superiores de San Felipe del Progreso,
[2] Facultad de Ciencias de la Computación, Benemérita Universidad Autónoma de Puebla
[3] Secretaría de Ciencia, Humanidades, Tecnología e Innovación,
alecam857595@gmail.com,{mireya.tovar, analaura.lezama}@correo.buap.mx

Resumen. El objetivo principal de este trabajo de investigación es evaluar y comparar diversos modelos de aprendizaje profundo para la detección de enfermedades cutáneas autoinmunes. En base a los resultados obtenidos, el modelo *DenseNet121* alcanzó una precisión cercana al 99% en la mayoría de los casos, utilizando diferentes métodos de preprocesamiento. En conclusión, las técnicas de preprocesamiento implementadas fueron fundamentales para lograr mejoras significativas en el desempeño de los modelos, superando los resultados reportados en el estado del arte.

Palabras Clave: enfermedades autoinmunes, aprendizaje profundo, *DenseNet121*.

1 Introducción

Las enfermedades autoinmunes se consideran una prioridad en la salud pública a nivel mundial. Según la Organización Mundial de la Salud (World Health Organization, 2006), estas patologías son el resultado de una respuesta anómala del sistema inmunológico, el cual ataca por error células y tejidos sanos del propio organismo. Estas enfermedades se clasifican en dos grandes categorías: enfermedades autoinmunes sistémicas, que afectan múltiples órganos o sistemas del cuerpo, y enfermedades autoinmunes órgano-específicas, que se limitan a un tejido u órgano determinado.

Dentro de estas últimas se encuentran las enfermedades autoinmunes cutáneas, en las cuales el sistema inmunológico agrede directamente los tejidos de la piel. Sus síntomas pueden incluir enrojecimiento, erupciones, comezón y descamación, y en algunos casos pueden coexistir con manifestaciones en otros órganos. A nivel global, enfermedades como la psoriasis y el liquen plano destacan por su impacto en la población. La psoriasis afecta entre el 2 % y el 3 % de las personas en el mundo, lo que equivale aproximadamente a 125 millones de individuos; de estos, alrededor del 30 % desarrolla artritis psoriásica, es decir, cerca de 37.5 millones de casos (*International Federation of Psoriasis Associations* [IFPA], 2023). Por su parte, Li et al. (2020)

estimaron, con base en 46 estudios poblacionales, una prevalencia global del liquen plano oral de 0.89 % en la población general y de 0.98 % en entornos clínicos.

En el contexto mexicano, las estadísticas refuerzan la necesidad de priorizar el diagnóstico y tratamiento de las enfermedades autoinmunes de la piel. El vitíligo, una de las afecciones más comunes, afecta entre el 2 % y el 5 % de la población nacional, lo que representa entre 2 y 6 millones de personas (Academia Nacional de Medicina, s.f.). Por otro lado, el liquen plano ha sido escasamente investigado; un estudio llevado a cabo en Yucatán registró 129 casos entre 2009 y 2014, con una prevalencia calculada entre el 0.14 % y el 0.8 % (Cortés-González et al., 2016). A pesar de que la información disponible es escasa, permite construir un panorama inicial sobre su distribución en la población de México. Por lo que, en este contexto, la inteligencia artificial (IA), el aprendizaje automático y el aprendizaje profundo han sido la columna vertebral en los avances en el campo de la medicina, por ejemplo, en la detección de enfermedades complejas. Para que estos modelos generen un resultado, es necesario proporcionarles grandes cantidades de información para detectar patrones o extraer atributos y por lo tanto realizar proyecciones, lo que facilita a los expertos en salud la toma de decisiones ágiles y exactas. Para utilizar estos modelos en las enfermedades cutáneas, cuyos síntomas pueden ser malinterpretados como otras enfermedades, los algoritmos de Inteligencia Artificial son un recurso importante para disminuir los errores en el diagnóstico. El uso de estos algoritmos permite un análisis de áreas de interés, la identificación de patrones distintivos y, por lo tanto, la predicción de la presencia de enfermedades basándose en imágenes médicas, contribuyendo de esta manera a una elección terapéutica más acertada.

En este artículo se propone crear tres modelos de aprendizaje profundo (*Deep Learning*) para examinar imágenes médicas vinculadas a afecciones autoinmunes cutáneas y categorizarlos en diversas categorías. Para lograrlo, se utilizarán dos arquitecturas de redes neuronales profundas y se implementarán diferentes técnicas de preprocesamiento, junto cambios en las dimensiones de entrada para mejorar el desempeño de los modelos.

El artículo se encuentra organizado de la siguiente manera: la Sección 2 presenta el estado del arte relacionado con estas enfermedades por diferentes autores. La Sección 3 expone los conceptos fundamentales empleados en el desarrollo del trabajo. En la Sección 4 se describe la metodología propuesta, mientras que en la Sección 5 se presentan los resultados obtenidos y el conjunto de datos utilizado. Finalmente, en la Sección 6 se exponen las conclusiones y trabajos a futuro, seguidas de las referencias bibliográficas empleadas.

2 Estado del arte

En esta sección se presentan trabajos desarrollados por diversos autores cuyo propósito ha sido contribuir al estudio, detección y tratamiento de enfermedades autoinmunes de la piel.

La revisión inicia con el estudio de Tyara y Widodo (2025), quienes emplearon un conjunto de datos compuesto por 2,764 imágenes distribuidas en ocho clases:

dermatomiositis, liquen plano, psoriasis, vitíligo, eccema, herpes, queratosis seborreica y tiña. Los autores evaluaron el desempeño de diferentes arquitecturas de redes neuronales profundas, entre ellas MobileNetV2, MobileNetV3-Small, MobileNetV3-Large, ResNet50, ResNet101 y ResNet152. El entrenamiento de cada modelo se llevó a cabo entre 35 y 50 épocas, lo cual permitió comparar el rendimiento de las distintas arquitecturas. Los autores observaron que *MobileNetV2* y *MobileNetV3-Large* mantuvieron un buen desempeño tanto en los conjuntos de entrenamiento como de prueba. En contraste, *MobileNetV3-Small* presentó un rendimiento deficiente incluso tras 35 épocas, evidenciando una menor capacidad de ajuste. Por su parte, *ResNet152* fue la arquitectura con mejores resultados, alcanzando una precisión del 92 % en la clasificación de las enfermedades.

Por otro lado, el estudio de Hammad et al. (2023) tuvo como objetivo evaluar el desempeño de modelos de redes neuronales convolucionales (CNN), específicamente *AlexNet, ResNet* y *VGG-16*, utilizando un conjunto de datos compuesto por imágenes de eczema y psoriasis. Los autores implementaron una estrategia de optimización basada en la integración de múltiples capas de las CNN. Esta combinación permitió mejorar significativamente los resultados, alcanzando una precisión aproximada del 95 %.

Ahmmed et al. (2025) desarrollaron un modelo de clasificación automática para enfermedades cutáneas como queratosis actínica, psoriasis y piel normal, utilizando una versión modificada de la arquitectura *VGG16*. El modelo logró una precisión general cercana al 90 % y un valor F_1 de 0.99 en la clasificación de la clase psoriasis, lo cual demuestra resultados altamente positivos.

En el estudio de Bello et al. (2024), los autores evaluaron el rendimiento del modelo *DenseNet121* para la clasificación binaria de cáncer de piel (benigno vs. maligno). Los autores incorporaron capas *flatten* con funciones de activación *ReLU* y utilizaron un conjunto de datos recopilado de diversas fuentes dermatológicas. El modelo alcanzó una precisión cercana al 87 %, superando significativamente el desempeño de otros modelos como *Random Forest* y máquinas de vectores de soporte (SVM).

Finalmente, Patil et al. (2020) aplicaron los modelos *k-means* y *k*-vecinos más cercanos (KNN) para la evaluación de cinco enfermedades cutáneas. La metodología incluyó la extracción de características en escala de grises y la eliminación de regiones no relevantes para el análisis. El modelo logró una precisión del 89 % en los datos de entrenamiento y aproximadamente del 90 % en el conjunto de prueba. Además, incorporaron un componente de informe médico que incluía detalles sobre el progreso de la enfermedad y posibles tratamientos. Este sistema fue diseñado para asistir al usuario mediante el análisis automatizado de las imágenes ingresadas.

Con base en los antecedentes descritos, en este trabajo se propone desarrollar dos modelos de aprendizaje profundo (*Deep Learning*) con el objetivo de analizar datos médicos correspondientes a enfermedades autoinmunes de la piel y clasificarlos en distintas categorías, como psoriasis, vitíligo, liquen plano y piel normal. Por lo que, se utilizarán dos estructuras de redes neuronales profundas. Así, el conjunto de datos será objeto de diferentes técnicas de preprocesamiento, tales como la conversión a RGB, la escala de grises, CLAHE y CLAHE+RGB, además de modificaciones en las dimensiones de entrada para mejorar el desempeño de los modelos.

3 Marco Teórico

En esta sección se presentan los conceptos que sustentan teóricamente la presente investigación.

El aprendizaje profundo (*deep learning*) siendo una de las ramas de la inteligencia artificial, permite que los modelos clasifiquen representaciones como señales visuales, auditivas o dinámicas, destinándolas a diversas categorías etiquetadas (LeCun et al., 2015). Dicha capacidad ha permitido avances significativos en campos como la medicina, permitiendo crear instrumentos para identificar patrones pertinentes y el hallazgo de terapias para enfermedades complejas y de diagnóstico complicado.

El conjunto de datos, es decir las imágenes médicas utilizadas en estos modelos pueden provenir de diversas fuentes como: imágenes vectoriales o producidas de manera manual, o capturas clínicas procesadas (Dedov, 2020). El procedimiento de clasificar imágenes a través de modelos de aprendizaje profundo requiere de un preprocesamiento cuidadosamente aplicado. El cual puede abarcar métodos como redimensionar imágenes, la normalización de los pixeles y la disminución del ruido, con el objetivo de simplificar la obtención de características pertinentes.

El objetivo del preprocesamiento es preparar las imágenes para que los modelos interpreten cada imagen suprimiendo componentes que no brindan datos relevantes, lo que incrementa la eficiencia del modelo.

Por lo tanto, una vez generados los modelos, es importante evaluar el desempeño de un modelo de aprendizaje profundo a través de métricas determinadas. Estos indicadores permiten medir la calidad y precisión del modelo para anticipar adecuadamente los valores de salida, basándose en los datos que previamente han sido preprocesados y divididos en subconjuntos de entrenamiento y prueba. La matriz de confusión es uno de los instrumentos más empleados, ya que permite un análisis del número de aciertos y errores del modelo en comparación con las clases reales.
La matriz de confusión está compuesta por cuatro elementos fundamentales:

Verdaderos Positivos (TP): instancias correctamente clasificadas como pertenecientes a la clase positiva.
Falsos Positivos (FP): instancias incorrectamente clasificadas como positivas.
Verdaderos Negativos (TN): instancias correctamente clasificadas como negativas.
Falsos Negativos (FN): instancias incorrectamente clasificadas como negativas.

A partir de estos valores se derivan varias métricas esenciales, en este caso la métrica de *accuracy* (Ver Ecuación 1) que mide el porcentaje total de predicciones correctas sobre el conjunto de datos (Müller & Guido, 2018).

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (1)$$

Donde:

TP= Valores predichos correctamente sobre los existentes
TN= Valores predichos correctamente sobre los inexistentes
FP= Valores predichos de forma incorrecta sobre los existentes

FN= Valores predichos de forma incorrecta sobre los inexistentes

Una vez definidos los criterios para valorar el desempeño de los modelos, el paso siguiente es elegir e instaurar las arquitecturas de redes neuronales profundas idóneas para este trabajo. Estas arquitecturas establecen la organización de las capas y el manejo de las imágenes en el modelo, impactando directamente en la calidad de los resultados generados por el modelo.

A continuación, se detallan algunas de las construcciones empleadas en este análisis: **ResNet50**: Esta estructura, está formada por 50 capas, se basa en la utilización de conexiones residuales que facilitan la prevención del deterioro en redes profundas. La estructura ha proporcionado buenos resultados al ser eficaz en el estudio de imágenes médicas, dado que facilita la identificación de patrones complejos y la realización de clasificaciones con gran exactitud (Kesuma & Rudiansyah, 2023).

DenseNet121: Esta estructura se distingue por vincular cada capa con todas las subsecuentes, lo que simplifica la transmisión de características y gradientes durante el entrenamiento. Dicha configuración incrementa la precisión del modelo clasificando las imágenes médicas con mayor precisión y sine error en el diagnóstico (Albelwi, 2022).

4 Metodología

En esta sección se describe la metodología propuesta para el desarrollo y evaluación de los modelos de aprendizaje profundo aplicados al reconocimiento de enfermedades autoinmunes de la piel. Dicha metodología consta de cinco fases:

Selección y preprocesamiento del conjunto de datos: el conjunto de datos utilizado se encuentra disponible en Kaggle (https://www.kaggle.com/datasets/aditipathak17/autoimmune-skin-disorder-dataset) de dominio público y a su vez estas imágenes se encuentran bajo el anonimato, garantizando su objetivo plenamente en la investigación. Contiene imágenes clasificadas en cuatro categorías: liquen plano, psoriasis, vitíligo y piel normal. Posteriormente el conjunto de datos fue sometido a 4 diferentes preprocesamientos que serán descritos a continuación:

Preprocesamiento 1 (RGB normalizado): Este preprocesamiento consistió en la normalización de los valores de píxeles de las imágenes en formato RGB dividiendo por 255, de manera que los valores se encuentren en el rango [0, 1]. Dicho proceso permitió una mayor estabilidad numérica durante el entrenamiento y facilita la convergencia del modelo.

Preprocesamiento 2 (escala de gris): Las imágenes se transformaron en formato de RGB a escala de grises fusionando los 3 canales en un único canal. Esta transformación subraya la intensidad de la luz.

Preprocesamiento 3 (CLAHE): en este procesamiento se utiliza el algoritmo de Equalización Adaptativa del Histograma de Contraste Limitado, que incrementa el

contraste local sin incrementar el ruido presente en las imágenes. Este procedimiento divide color y luz. Lo que agudiza las áreas sombrías y potencia la visibilidad de patrones pertinentes para el diagnóstico.

Preprocesamiento 4 (CLAHE + RGB): En este proceso, se convierte la imagen en un espacio de color LAB, se aplica CLAHE en el canal de luz (L), se modifican los canales A y B, y posteriormente se reconstruye la imagen en formato RGB. Finalmente, los valores se normalizan en el rango [0, 1].

Preparación del conjunto de datos: Se implementaron métodos de incremento de datos para potenciar su tamaño y variabilidad. Es decir, el conjunto de datos se segmentó en cuatro secciones por cada clase, y cada sección fue rotada en 90°, 180°, 270° y 360°, respectivamente.

Uso de hardware y software para el entrenamiento de los modelos: Para llevar a cabo el entrenamiento de los modelos *DenseNet121* y *ResNet50*, se utilizaron como herramientas principales las bibliotecas de Pytorch en el lenguaje de programación Python. El desarrollo se realizó en el entorno de trabajo *Jupyter Notebook* y se contó con el apoyo de una tarjeta gráfica GPU T4 (Nvidia Tesla T4), la cual está especialmente diseñada para optimizar los tiempos de entrenamiento y evaluación de las arquitecturas de aprendizaje profundo.

Configuración de parámetros y entrenamiento: Se utilizó un *batch size* de entre 32 y 50, imágenes redimensionadas a 224×224 píxeles, y una tasa de *dropout* de 0.4. Los modelos evaluados fueron: *ResNet50 y DenseNet121*. Cada uno se entrenó con 20 épocas, con una política de parada anticipada (*early stopping)* de 5 épocas, deteniendo el entrenamiento si no se observaba mejora en la fase de validación.

Entrenamiento del modelo: Para esta fase, se empleó la técnica *hold-out*, mediante la cual el conjunto de datos procesado fue dividido en un 80% de entrenamiento y 20% de prueba. Esta estrategia permite evaluar de manera efectiva la capacidad de generalización de los modelos, al proporcionar una separación entre los datos utilizados para el aprendizaje y aquellos destinados a la evaluación del modelo.

Evaluación del desempeño: La evaluación se basó en la métrica *accuracy*, permitiendo determinar qué combinación de modelo y preprocesamiento ofrece mejores resultados para la clasificación automática de imágenes dermatológicas.

5 Resultados

En esta sección se presentan los resultados obtenidos tras aplicar la metodología propuesta al conjunto de datos seleccionado.

La Tabla 1 muestra la distribución estadística de las imágenes utilizadas para el entrenamiento, organizadas por clase:

Tabla 1. Distribución del conjunto de datos por clase

Clase	Cantidad de datos
Liquen plano	1,618
Psoriasis	1,434
Piel normal	1,586

Vitiligo	2,432

Para evaluar el desempeño de los modelos, se utilizó la métrica de *accuracy* durante las épocas de entrenamiento. En particular, el comportamiento de los modelos frente a los distintos preprocesamientos aplicados (P1 a P4) fueron analizados. Por lo que, de los dos modelos evaluados, *DenseNet121* mostró un rendimiento notablemente superior, mientras que *ResNet50* mostró un desempeño consistente en algunas de sus configuraciones. La Tabla 2 presenta los valores de precisión obtenidos por cada modelo y preprocesamiento.

Tabla 2. Precisión alcanzada por cada modelo y preprocesamiento

Modelo	Épocas	Entrenamiento				Validación			
		P1	P2	P3	P4	P1	P2	P3	P4
DenseNet121	5	0.9823	0.8477	0.8862	0.8860	0.9806	0.7768	0.8307	0.7961
	10	0.9948	0.9097	0.9559	0.9503	0.9834	0.7851	0.8256	0.8695
	20	0.9980	0.9476	0.9760	0.9665	0.9862	0.8109	0.8261	0.8801
ResNet50	5	0.9706	0.9405	0.9676	0.9669	0.9705	0.9585	0.9779	0.9733
	10	0.9886	0.9749	0.9893	0.9879	0.9852	0.9719	0.9866	0.9866
	20	0.9957	0.9914	0.9945	0.9941	0.9988	0.9742	0.9903	0.9917

El modelo *DenseNet121* presentó una precisión superior al 99% en la mayoría de los casos. En las Figuras 1–4, se observa un aprendizaje progresivo y estable en todas las configuraciones de preprocesamiento, sin necesidad de ajustes adicionales. Las matrices de confusión correspondientes (Figuras 5–8) evidencian una clasificación precisa y coherente entre las clases, incluso en condiciones de iluminación o contraste variables.

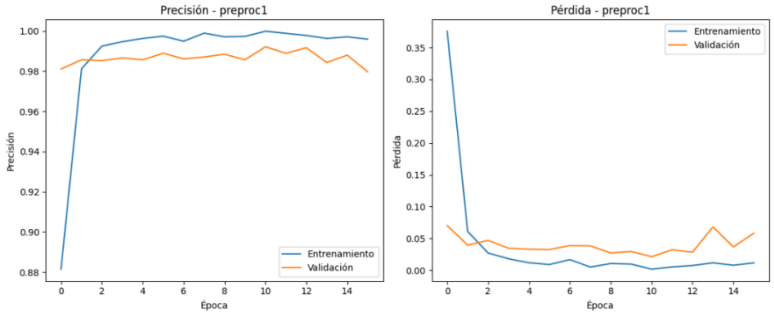

Figura 1. Curvas de precisión y pérdida de datos de DenseNet121 con preproc1.

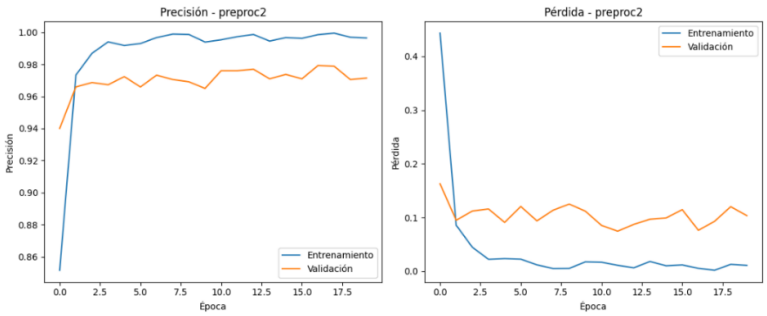

Figura 2. Curvas de precisión y pérdida de datos de DenseNet121 con preproc2. Fuente: Elaboración propia.

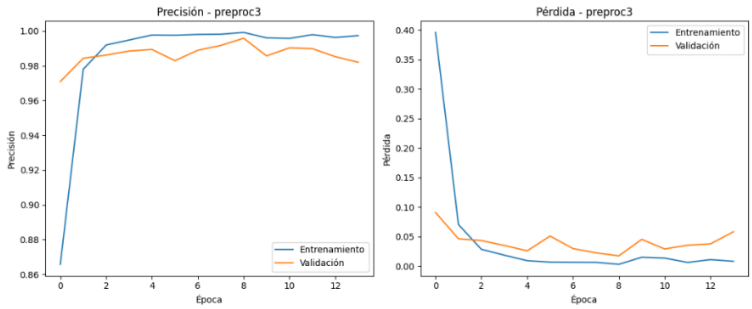

Figura 3. Curvas de precisión y pérdida de datos de DenseNet121 con preproc3.

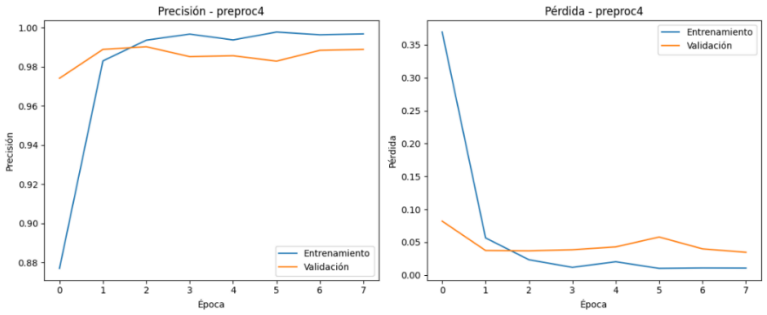

Figura 4. Curvas de precisión y pérdida de datos de DenseNet121 con preproc4.

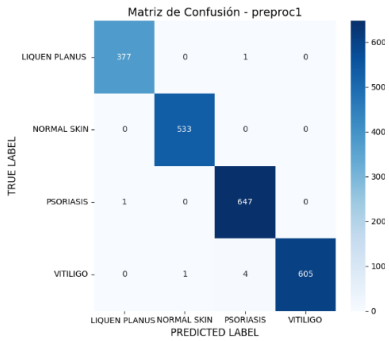

Figura 5. Matriz de confusión DenseNet121 con preproc1

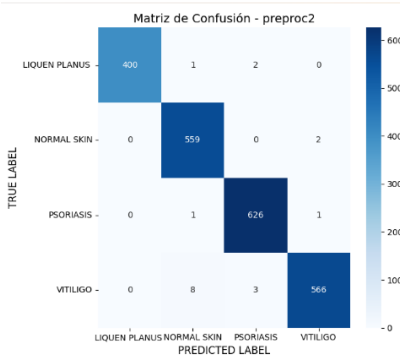

Figura 6. Matriz de confusión DenseNet121 con preproc2

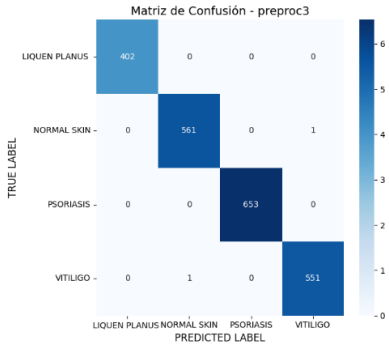

Figura 7. Matriz de confusión DenseNet121 con preproc3

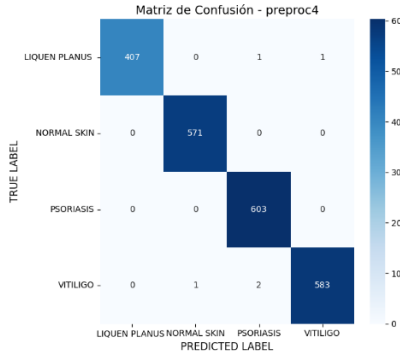

Figura 8. Matriz de confusión DenseNet121 con preproc4

El modelo *ResNet50*, dada su arquitectura más liviana con 50 capas convolucionales, mantiene una eficiencia destacada tanto en entrenamiento como en validación. Las Figuras 9–12 demuestran una relación casi paralela entre las curvas de precisión y pérdida para entrenamiento y validación, lo que evidencia una buena capacidad de generalización sin sobreajuste. Las matrices de confusión (Figuras 13–16) confirman una clasificación robusta y precisa en los distintos preprocesamientos.

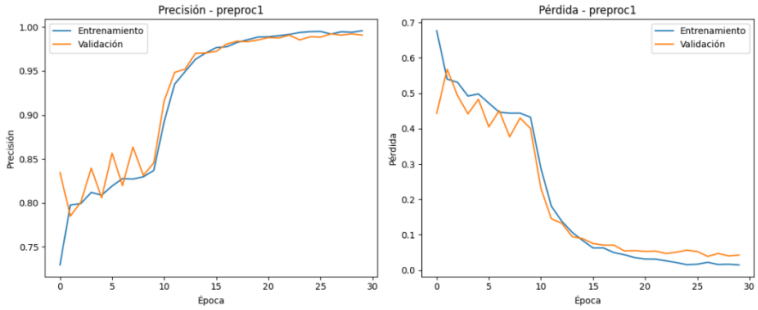

Figura 9. Curvas de precisión y pérdida de datos de ResNet50 con preproc1.

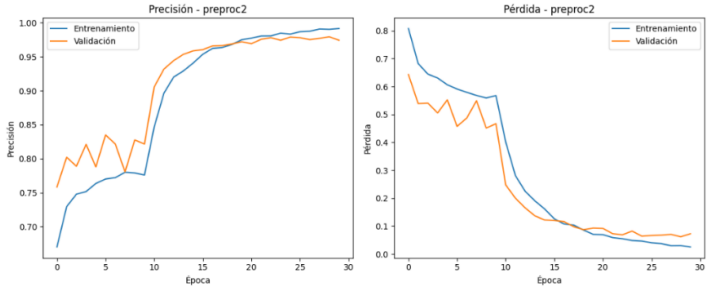

Figura 10. Curvas de precisión y pérdida de datos de ResNet50 con preproc2.

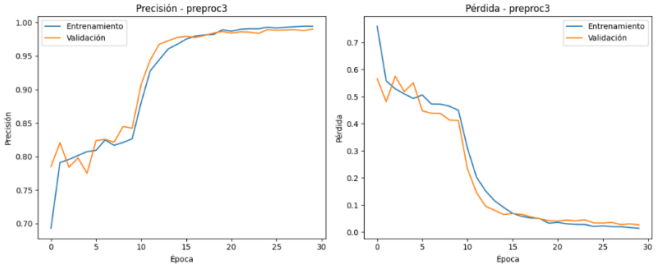

Figura 11. Curvas de precisión y pérdida de datos de ResNet50 con preproc3.

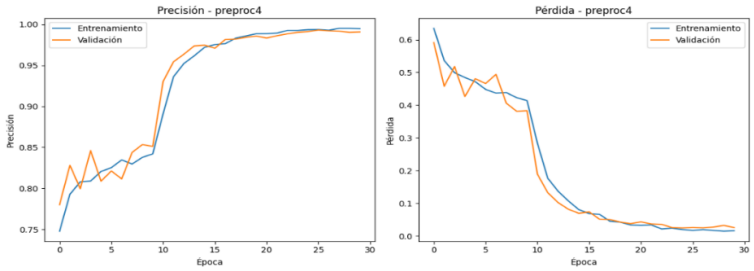

Figura 12. Curvas de precisión y pérdida de datos de ResNet50 con preproc4.

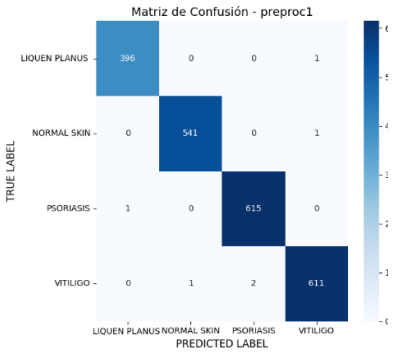

Figura 13. Matriz de confusión ResNet50 con preproc1

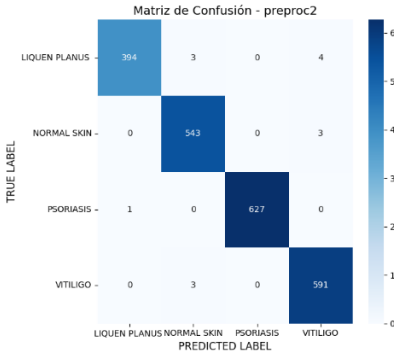

Figura 14. Matriz de confusión ResNet50 con preproc2

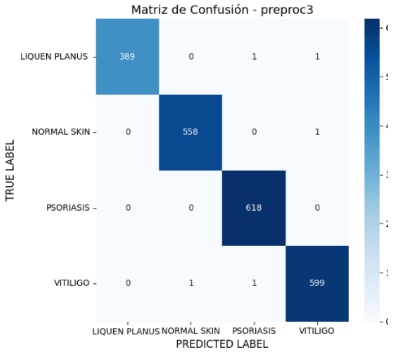

Figura 15. Matriz de confusión ResNet50 con preproc3

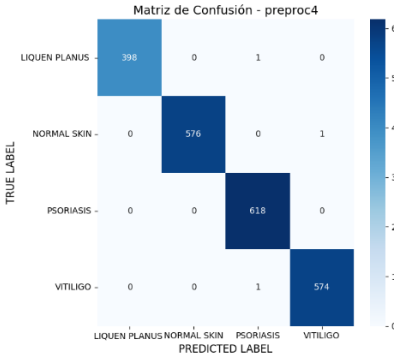

Figura 16. Matriz de confusión ResNet50 con preproc4

Finalmente, en la Tabla 3 se expone una comparación entre los modelos propuestos y los resultados de investigaciones previas reportadas en la literatura. Por lo que, los

resultados obtenidos en este estudio superan ampliamente las métricas reportadas en investigaciones previas cuyo conjunto de datos fueron manipulados de diferente forma, aportando resultados favorables, lo que lo convierte en la base esencial de la definición de la arquitectura, así como su entrenamiento y validación.

Tabla 3. Comparación de precisión con el estado del arte

Modelo	Accuracy				Precisión en el estado del arte	Referencia
	P1	P2	P3	P4		
DenseNet121	99.45%	99.40%	99.59%	99.86%	87%	(Bello et al., 2024)
ResNet50	99.22%	97.93%	99.03%	99.17%	91.98%	(Tyara & Widodo, 2025)

No obstante, es importante señalar posibles algunas limitaciones en el desarrollo de las fases del estudio expuesto. En primer lugar, los datos empleados, sometidos a diferentes procesos de preprocesamiento y utilizados para el entrenamiento de los modelos, provienen de una única fuente (*Kaggle*). Por ello, la exploración de distintos padecimientos requiere considerar poblaciones más diversas, con variaciones tanto regionales como fisiológicas. Asimismo, el tipo de dispositivo empleado para la obtención de las imágenes puede incidir en el desempeño del modelo, dado que factores como el bajo contraste y la limitada luminosidad afectan la representación visual. Finalmente, un incremento significativo en el rendimiento podría asociarse a un sobreajuste, lo que resalta la necesidad de realizar una validación multicéntrica.

6 Conclusiones

En este trabajo se exponen los resultados obtenidos por los modelos *ResNet50* y *DenseNet121*. Los cuales lograron cifras de precisión cercanas al 99% en los dos primeros casos. Por un lado, *ResNet50* demostró una precisión notable al preprocesar imágenes en RGB, mientras que *DenseNet121* consiguió mantener un desempeño alto y consistente incluso con un número limitado de épocas, lo que indica una eficiencia sobresaliente en el aprendizaje. Al comparar los resultados obtenidos con investigaciones anteriores se observa que los modelos elaborados en esta investigación sobrepasan los estándares actuales, mostrando una mejora cercana del 12% en exactitud. Esto nos muestra que una mezcla apropiada entre el modelo escogido, el tipo de preprocesamiento implementado y los hiperparámetros pueden incrementar notablemente el desempeño en el diagnóstico médico automatizado. Los gráficos de rendimiento posibilitaron el análisis de la correlación entre los datos de entrenamiento y validación, y además la detección de posibles indicios de sobreajuste. Por lo tanto, este trabajo ofrece un fundamento sólido para futuros estudios enfocados en la identificación automatizada de enfermedades cutáneas, además de la mejora de

modelos en situaciones clínicas reales, con el objetivo de disminuir la necesidad de intervención humana en fases iniciales del diagnóstico.

Referencias

Academia Nacional de Medicina. (s.f.). Vitiligo: Una revisión. Recuperado de https://academianacionaldemedicina.org/publicaciones/div/vitiligo-una-revision/

Ahmmed, F., Raihan, M. Z., Nahar, K., Asadujjaman, D. M., Rahman, M. M., y Tamim, A. (2025). Skin Disease Detection and Classification of Actinic Keratosis and Psoriasis Utilizing Deep Transfer Learning. Recuperado de https://arxiv.org/abs/2501.13713

Albelwi, S. A. (2022). "Deep architecture based on DenseNet-121 model for weather image recognition", International Journal of Advanced Computer Science and Applications, vol. 13(10)

Alzubaidi, L., Zhang, J., Humaidi, A. J., Al-Dujaili, A., Duan, Y., Al-Shamma, O., Santamaría, J., Fadhel, M. A., Al-Amidie, M., y Farhan, L. (2021). "Review of deep learning: concepts, CNN architectures, challenges, applications, future directions", Journal of Big Data, vol. 8(1), pp. 53.

Bello, A., Ng, S.-C., y Leung, M.-F. (2024). "Skin Cancer Classification Using Fine-Tuned Transfer Learning of DENSENET-121", Applied Sciences, vol. 14(17), pp. 7707.

Cortés-González, K., Mena-Méndez, E., y Cámara-López, E. (2016). "Liquen plano: Perfil clínico y características demográficas de 129 casos en el Centro Dermatológico de Yucatán", Dermatología Cosmética, Médica y Quirúrgica, vol. 14(3). Recuperado de https://dcmq.com.mx/...

Dedov, F. (2020). The Python Bible Volume 7: Computer Vision (OpenCV, Object Recognition). Amazon Digital Services LLC - Kdp.

Hammad, M., Pławiak, P., El-Affendi, M. A., Abd El-latif, A. A., y Abdel Latif, A. A. (2023). "Enhanced Deep Learning Approach for Accurate Eczema and Psoriasis Skin Detection", Sensors (Basel, Switzerland), vol. 23. Recuperado de https://api.semanticscholar.org/CorpusID:261158874

International Federation of Psoriasis Associations. (2023). Psoriasis facts. Recuperado de https://ifpa-pso.com/

Kesuma, L. I. y Rudiansyah, R. (2023). "Classification of Covid-19 Diseases Through Lung CT-Scan Image Using the ResNet-50 Architecture", Computer Engineering & Applications Journal, vol. 12(1).

LeCun, Y., Bengio, Y., y Hinton, G. (2015). "Deep learning", Nature, vol. 521(7553), pp. 436–444.

Li, C., Tang, X., Zheng, Y., Chen, X., y Liu, H. (2023). "Global prevalence of oral lichen planus: A systematic review and meta-analysis", Journal of the American Academy of Dermatology, vol. 89(1), pp. 123–132.

Müller, A. C. y Guido, S. (2018). Introduction to Machine Learning with Python: A Guide for Data Scientists. O'Reilly Media.

Patil, P. J., Buchkule, S. J., More, V. S., & Abhale, S. G. (2020). Skin disease detection using image processing technique. International Research Journal of Engineering and Technology, 7(06)

Tyara, R. N. P. y Widodo, A. M. (2025). "Enhanced Dermatological Diagnosis: Autoimmune and Non-Autoimmune Skin Disease Classification Using MobileNet and ResNet", Infact: International Journal of Computers, vol. 9(01), pp. 44–55.

World Health Organization. (2006). "Principles and Methods for Assessing Autoimmunity Associated with Exposure to Chemicals", Environmental Health Criteria, vol. 236. Recuperado de https://apps.who.int/iris/handle/10665/43603

Capítulo 9

Expedición Natura: Simulador educativo centrado en un ecosistema endémico del estado de Puebla

Erick Gutíerrez-Sánchez, Juan Pablo Hernández-Flores, Juan Carlos Conde-Ramírez, Carlos Leopoldo Carreón-Díaz-De-León, Daniel Marcelo González-Arriaga

Facultad de Ciencias de la Computación, Benemérita Universidad Autónoma de Puebla, México

erick.gutierrezsa@alumno.buap.mx,juan.hernandezfl@alumno.buap.m, carlos.conder@correo.buap.mx,carlos.carreon@correo.buap.mx, daniel.gonzalezarr@correo.buap.mx

Resumen. Este trabajo presenta el desarrollo de un simulador de entornos tridimensionales centrado en un ecosistema endémico del estado de Puebla, utilizando generación procedural e inteligencia artificial con una finalidad educativa. El sistema combina la creación automatizada de ecosistemas con la interacción personalizada de un agente conversacional, que actúa como guía de la experiencia. Este es capaz de generar misiones adaptadas al entorno y mantener un diálogo contextualizado con el usuario. Para lograrlo, se integró un sistema de reglas y procesamiento de lenguaje natural con memoria contextual y lógica ecológica. Mientras que el entorno se genera mediante el algoritmo *Wave Function Collapse*, la flora y fauna son distribuidos mediante algoritmos genéticos. El resultado es una herramienta accesible desde navegadores, dirigida principalmente a públicos infantiles, que promueve la conciencia ambiental a través de la exploración interactiva y la conversación guiada.

Palabras Clave: Inteligencia Artificial, Agentes Conversacionales, Generación procedural, Procesamiento del Lenguaje Natural, ODS 15, ecosistema endémico, Teporingo.

1 Introducción

Los ecosistemas enfrentan una triple crisis global: cambio climático, contaminación y pérdida de biodiversidad (United Nations, 2023). De acuerdo con la SEMARNAT (2019), México, uno de los países más biodiversos, tiene más del 50% de sus ecosistemas deteriorados. Por lo tanto, este proyecto busca contribuir a la educación ambiental mediante un simulador 3D que recrea un ecosistema de Puebla, centrado en el teporingo (*Romerolagus diazi*), debido a su condición de especie en peligro. Se decidió elegirlo frente a otras especies en función de tres criterios: A. Su *estatus de*

conservación[1], B. Su importancia ecológica y C. Su potencial educativo. Este ultimo punto se demuestra en el hecho de que no levanta tanto interés en el público general como si lo han hecho otras especies como los ajolotes. Fomentar el interés en los teporingos, no solo ayudará a su conservación individual, si no también en la flora y fauna con la que se relaciona. Respecto al simulador, este integra un agente inteligente que guiará al usuario, contestará preguntas e invita a la conservación ambiental en línea con el ODS 15: Vida de Ecosistemas Terrestres

Ahora se enlista el contenido de las secciones del presente trabajo: 1. Introducción, al problema, nuestra propuesta y como ayuda resolverlo; 2. Conciencia Ambiental y desarrollo sostenible usando de marco los objetivos propuestos por la ONU, 3. Proyectos interactivos, previamente realizados; 4. Metodología de desarrollo, junto a sus fases; 5. Detalles de implementación, así como su arquitectura; 6. Resultados obtenidos; 7. Conclusiones y propuestas de trabajo futuro.

2 Conciencia ambiental y desarrollo sostenible

2.1 Agenda 2030 y Biodiversidad en México

La Agenda 2030 es una iniciativa de la ONU con la finalidad de fijar un curso y plan de acción para erradicar la pobreza, reducir las desigualdades sociales, cuidar el ambiente y en general traer prosperidad global para el año 2030. Para lograr esto se han propuesto los 17 Objetivos de Desarrollo Sostenible (ODS), acompañados de 169 metas específicas y entre estos objetivos, el ODS 15: "Vida de Ecosistemas Terrestres".
La relevancia de este ODS radica es que las actividades humanas modernas han alterado significativamente los procesos naturales, afectando suministros de agua, polinización de cultivos, climas desfasados, calidad del suelo, etc. Esto perjudica a la flora y fauna, llevado, incluso ¿ponerlos en peligro de extinción (United Nations, 2023). México es reconocido globalmente como uno de los países megadiversos, una categoría que agrupa a las naciones que albergan el mayor número de especies animales y vegetales del planeta. Esta diversidad de hábitats ha permitido la evolución y preservación de un gran número de especies, muchas de las cuales son *endémicas*, es decir, únicas del país y no encontradas en ningún otro lugar del mundo.

Sin embargo, muchas especies mexicanas se encuentran en riesgo, la pérdida de estas no solo representa una disminución del patrimonio biológico global, sino también un debilitamiento de los ecosistemas de los que dependen otras formas de vida (SEMARNAT, 2019).

2.2 Especies Endémicas de Puebla

Dentro del estado de Puebla destaca la región de los volcanes, particularmente el área que comprende el Parque Nacional Iztaccíhuatl-Popocatépetl, área natural

[1] Evaluación de una especie o ecosistema, para determinar y tomar acciones si se encuentra en riesgo o peligro de extinción, considerando factores como población, tendencias, amenazas y otros.

protegida desde 1935. Esta zona montañosa presenta condiciones climáticas únicas que han favorecido procesos de aislamiento geográfico y adaptación evolutiva, propiciando la aparición de especies que sólo existen en esta región. Sin embargo, enfrenta amenazas como la deforestación, incendios forestales, expansión urbana, y la actividad volcánica del Popocatépetl. Por ello, diversos programas de conservación, monitoreo ecológico y educación ambiental se llevan a cabo en la región con el objetivo de proteger su biodiversidad y fomentar el turismo responsable (CONABIO, 2011).

Este hábitat permite la presencia de especies endémicas, entre ellas, el teporingo, mejor conocido como conejo de los volcanes o zacatuche, es un pequeño mamífero de México y considerado uno de los conejos más antiguos y primitivos del mundo (ver **¡Error! No se encuentra el origen de la referencia.**). Además de otras especies emblemáticas como el lince, el coyote y el halcón peregrino (CONANP, 2024).

El teporingo tiene hábitos crepusculares y nocturnos. Su dieta consiste principalmente de zacatón (género *Festuca*), pastos, hierbas y brotes tiernos, de los cuales también obtiene agua. Juega un papel ecológico importante al mantener el equilibrio en los pastizales y al servir de presa para depredadores como el coyote, el zorro gris, aves rapaces y algunas serpientes. Por lo que está catalogado como especie en peligro de extinción por la NOM-059-SEMARNAT y la UICN, (CONANP, 2015).

3 Proyectos Educativos Interactivos

Un estudio, basado en el método PRISMA (*Preferred Reporting Items for Systematic Reviews and Meta-Analyses*), revisó 21 artículos publicados entre 2013 y 2023, detectando un impacto positivo en el uso de tecnologías digitales (como la realidad virtual) sobre la preocupación de los estudiantes por la sostenibilidad del planeta. Esto sugiere que los proyectos educativos interactivos, al integrar tecnologías emergentes, pueden ser altamente efectivos para promover el aprendizaje ambiental y la toma de conciencia ecológica (Hajj-Hassan et al., 2024).

3.1 La IA en el desarrollo procedural

La Generación Procedural de Contenido (PCG) utiliza IA para crear entornos virtuales . De acuerdo con Plataformas como PROCTHOR demuestran el potencial de estas tecnologías, aunque el acceso a datos de entrenamiento sigue siendo un desafío (Deitke, et al., 2022).

Sin embargo, uno de los principales retos que enfrenta esta área el gran volumen de información específica para cada dominio, ya que el contenido generado suele ser altamente personalizado y los datos disponibles son limitados (Mao et al., 2024).

3.2 Wave Function Collapse

La Wave Function Collapse (WFC) es un algoritmo cuya finalidad es la generación procedural de contenido, este fue concebido por Maxium Gumin en 2016 inspirándose en la mecánica cuántica. La idea es tener un conjunto de celdas, cada una en un

superestado (*superposition*), además de una serie de reglas de adyacencia, con ellas vamos visitando cada celda obligándola a colapsar a un estado específico. Para ello se ocupan principalmente dos modelos: *Overlapping* que analiza una entrada para extraer los patrones únicos en un área determinada, a partir de estos aprende las reglas de compatibilidad, y *Simple Tiled*, que usa las etiquetas de las celdas con sus reglas de conexión explicitas puestas manualmente. Es entonces que el contenido generado se crea a partir de patrones compatibles para cada celda, permitiendo entornos coherentes, armónicos y cercanos a la realidad. Esta técnica ha sido usada previamente en videojuegos en casos como mapas, texturas e incluso escenarios 3D en pequeña escala (Brellas, 2021).

3.3 Algoritmos Genéticos

Los Algoritmos Genéticos (GA) son una técnica tomada directamente de los procesos biológicos detrás de la evolución, como lo son la selección, el cruce y la mutación genética. Se toma una población de soluciones llamados individuos, a cada uno de estos se evalúa con una función de aptitud (fitness), que determina qué tan buena solución es. Este proceso es iterativo A través de varias generaciones, con la finalidad de mejorar paulatinamente la calidad de las soluciones. En el contexto de la generación de contenido, los GA nos ayudan en la distribución de entidades como pueden ser objetos, estructuras o personajes en escenarios tridimensionales, de forma automática (Brellas, 2021).

3.4 Chatbots basados en agentes generativos

Un agente generativo es una entidad computacional que simula un comportamiento humano en un entorno interactivo. Park et al. (2023), En ese trabajo se insertaron 25 agentes en un entorno tipo sandbox en el que debian interactuar entre ellos para formar relaciones. Estos agentes utilizaron un *Modelo de Lenguaje Grande*[2] (LLM) que almacena sus experiencias en lenguaje natural, y los reutiliza planificar comportamientos futuros. Esta arquitectura permite crear *chatbots* con comportamientos humanos creíbles, capaces de interactuar dinámicamente con usuarios en entornos simulados mejorando la Interacción Humano-Computadora.

4 Metodología de desarrollo

La metodología propuesta para el proyecto fueron tres fases que permitieron desarrollar una bibliografía robusta, un buen diseño, el desarrollo técnico y sus constantes validaciones (ver Tabla *1*). Se adoptó un enfoque modular, permitiendo trabajar de forma paralela en distintos componentes del sistema, como el entorno 3D, los algoritmos de inteligencia artificial y el agente conversacional.

[2] Tipo de inteligencia artificial (IA) entrenado con grandes volúmenes de datos como texto y código para poder manipular el lenguaje humano

Tabla 1. Fases iterativas de desarrollo del simulador educativo 3D.

Actividades Principales	*Actividades Específicas*
FASE 1: ANÁLISIS Y DISEÑO	
1. Definir objetivos	• Seleccionar ODS 15 ("Vida de Ecosistemas Terrestres"). • Enfocarse en educación ambiental para público infantil.
2. Investigación documental	• Estudiar ecosistemas de Puebla. • Seleccionar especie focal (teporingo).
3. Selección de herramientas	• Evaluar tecnologías (Three.js vs. Babylon.js vs. Unity). • Definir estética (voxel y pixel art).
4. Diseño arquitectónico	• Cliente-servidor (frontend: Three.js; backend: Python). • Intercambio de datos (formato: JSON) • Generación procedural (WFC + GA). • Agente conversacional (API de Chatbot. + PLN).
FASE 2: DESARROLLO Y PRUEBAS	
1. Desarrollo modular	• Generación procedural: WFC para terreno / GA para flora y fauna. • Agente inteligente: o Integrar API de *chatbot*. o Diseño de misiones educativas.
2. Ciclo iterativo	• Desarrollar módulo → Pruebas unitarias → Integrar al sistema
3. Validación	• Coherencia ecológica (reglas de biomas). • Ajuste de las respuestas del agente para alinearlas con el contexto ecológico. • Rendimiento en navegadores.
FASE 3: DESPLIEGUE Y MANTENIMIENTO	
1. Despliegue	• Configurar entornos *serverless*[3] (GitHub-Pages, Render, etc.). • "Dockerizar" para portabilidad.
2. Mantenimiento	• Control de versiones (GitHub: main/develop). • Pruebas de regresión y seguridad (XSS, inyecciones).
3. Documentación	• Reporte técnico con lecciones aprendidas.

Las iteraciones continuas permitieron avanzar hacia un prototipo funcional sólido, capaz de integrar de manera coherente el contenido generado con la interacción mediante IA conversacional. Aunque al momento de la escritura del documento no se había llegado completamente la fase de "Despliegue y Mantenimiento", se identificaron necesidades relacionadas que sirvan como referencia para desarrollos futuros o posibles usuarios interesados en adaptar o continuar con el proyecto en contextos similares.

[3] Es un modelo computacional en la nube donde un proveedor gestiona la infraestructura necesaria para ejecutar código de aplicaciones. Así los desarrolladores se centran en escribir funciones que se ejecutan y escalan automáticamente.

5 Detalles de implementación

5.1 Arquitectura del sistema

El sistema fue diseñado con una arquitectura modular que divide las tareas entre el *frontend* y el *backend*. El *Frontend* está implementado con Three.js, el *frontend* se encargar de mostrar el entorno 3D: la flora, la fauna y el agente conversacional. El *Backend* está encargado de la lógica compleja del sistema, incluyendo la PCG y la IA. Se desarrolló principalmente en Python, lo que permitió aplicar algoritmos como *Wave Function Collapse* (WFC) para la creación del mapa y algoritmos genéticos (GA) para la distribución coherente de flora y fauna.

Los módulos se comunican por medio de solicitudes HTTP. Al iniciar la simulación el cliente (*frontend*) envía una petición al servidor (*backend*), este crea el entorno y responde con el mapa completo, la posición y cantidad de la flora y fauna, todo en formato JSON. De manera independiente el *chatbot educativo* es un servicio adicional que se comunica consumiendo una API externa. Esta es socorrida cada vez que el usuario interactúa con el agente conversacional, logrando una experiencia inmersiva sin la necesidad de recargar el sistema.

El sistema puede ejecutarse en una amplia gama de dispositivos ya que el procesamiento lo realiza el servidor mientras que el cliente se limita a visualizar e interactuar.

5.2 Generación Procedural de Terreno con WFC

Para generar el terreno de forma coherente y variada, se utiliza una implementación personalizada del algoritmo *Wave Function Collapse* (WFC).

Listado 1. Ejemplo del archivo Reglas.json

```
{
    "biomas": {
        "pasto": {
        "frecuencia": 0.9, "vecinos": {
              "norte": ["pasto", "roca", "agua"],
              "diferencia_altura_max": 2
            }
        },
        ...
    }
}
```

Definición de reglas. El comportamiento y distribución de los biomas se define mediante un archivo externo llamado Reglas.json. Este archivo contiene la lista de biomas disponibles (por ejemplo: pasto, agua, roca) y sus respectivas frecuencias deseadas, así como las reglas que determinan qué biomas pueden estar adyacentes entre sí en cada dirección (norte, sur, este, oeste). También especifica una diferencia de altura máxima entre biomas vecinos para mantener coherencia topográfica (ver Listado 1).

Funcionamiento del generador. El módulo WFC.py implementa un generador llamado *GeneradorBiomas*. Este sigue los siguientes pasos:

1. *Inicialización*: se carga el archivo Reglas.json y se inicializa un mapa con todas las posibles opciones en cada celda.
2. *Selección de celda*: se elige la celda con menor entropía (menos opciones).
3. *Colapso*: se elige un bioma para la celda de forma aleatoria ponderada según su frecuencia deseada.
4. *Propagación*: se actualizan las celdas vecinas eliminando opciones incompatibles según las reglas de adyacencia y altura.
5. *Repetición*: los pasos 2 a 4 se repiten hasta que el mapa esté completamente colapsado o se alcance un número máximo de intentos.

Visualización y exportación. El resultado se visualiza mediante *matplotlib*, donde cada celda tiene un color asociado a su bioma y nivel de altura, además de mostrar el número de altura dentro de cada celda (ver Figura 1). Se genera un archivo Terreno.json. Por ejemplo, el bioma agua siempre tiene altura cero, ya que busca simular ríos y cuerpos de agua estancada que recorran el mapa de forma coherente. Las celdas de pasto, con alturas bajas y variables, están pensadas como áreas versátiles donde se podrán disponer árboles, arbustos u otra flora y fauna.

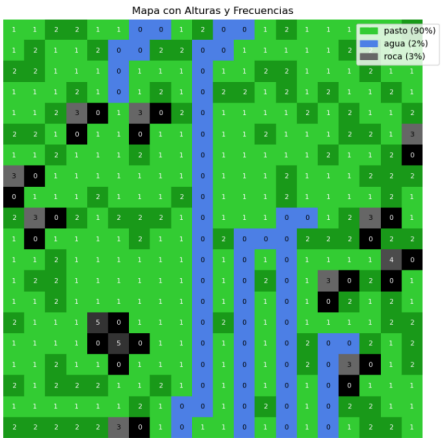

Figura 1. Visualización del mapa con alturas y biomas que permite validar la coherencia.

5.3 Distribución de Flora y Fauna mediante Algoritmos Genéticos (GA)

Para garantizar una distribución ecológicamente coherente se utilizó un algoritmo genético (GA) que trabaja en conjunto con el terreno generado por WFC, asegurando que cada organismo aparezca únicamente en biomas y altitudes adecuadas.

Definición de reglas ecológicas. Las características de cada especie se especifican en el archivo Especies.json (ver Listado 2), que contiene: *requisitos de hábitat* (biomas permitidos y rangos de altura), *parámetros poblacionales* (mínimos/máximos), *relaciones tróficas* (depredadores, presas y radios de interacción) y v*ariación morfológica* (escalas y rotaciones permitidas).

Listado 2. Ejemplo de configuración para el teporingo

```
{
    "fauna": {
        "teporingo": {
            "biomas_permitidos": ["pasto", "roca"], "altura_min": 1,
            "altura_max": 4, "poblacion_min": 5, "poblacion_max": 8,
            "escala_min": 0.8, "escala_max": 1.2, "dependencias": [
                {"especie": "hongo", "radio": 3 , "min": 2}
            ]
        }
    }
}
```

Funcionamiento del algoritmo: La clase AlgoritmoGeneticoEcosistema sigue un proceso en 4 etapas:

1. *Inicialización*: Carga el terreno generado y las reglas ecológicas, precalculando las posiciones válidas para cada especie.
2. *Generación de población*: Crea individuos que representan posibles distribuciones, garantizando al menos 5 teporingos (especie prioritaria), la no superposición espacial y las densidades máximas por bioma.
3. *Evaluación de fitness*: Calcula la calidad de cada solución considerando: Adecuación al hábitat (30%) , Relaciones tróficas satisfechas (25%), Diversidad poblacional (20%), Distribución espacial equilibrada (15%) y Variación morfológica (10%)
4. Selección y reproducción: Aplica operadores genéticos durante 50 generaciones: Selección por torneo ($k = 3$), Cruce en punto aleatorio, Mutaciones controladas (10% tasa)

Figura 2. Visualización de la distribución óptima generada.

Visualización y exportación. El resultado final se representa mediante un gráfico matricial, como en Figura 2, donde cada especie tiene un color y acrónimo único, el tamaño del marcador refleja la escala del organismo y las coordenadas corresponden a la posición en el terreno. Los datos se exportan a Flora&Fauna.json con formato estandarizado para su integración con Three.js, incluyendo posición, rotación y escala de cada organismo.

5.4 Agente Educativo Conversacional

El agente inteligente combina capacidades conversacionales con funciones de guía educativo, implementando un modelo híbrido entre reglas predefinidas y Procesamiento de Lenguaje Natural (PLN). Su arquitectura sigue el patrón *Memory-Augmented Chatbot*, permitiendo interacciones contextuales basadas en el estado del ecosistema simulado.

Arquitectura Modular. El sistema se compone de cuatro capas interconectadas:

- *Interfaz de Diálogo (chatInterface.js)*: Maneja la visualización de los diálogos en un estilo de novela visual con animaciones de escritura paulatina. Incluye botones de opciones contextuales y el control por teclado (cierre con ESC).
- *Núcleo Conversacional (dialogue.js)*: Declara una serie de reglas para preguntas frecuentes (FAQ ecológicas) usando detección de patrones (*regex*) para consultas estructuradas, e integración del motor de PLN para respuestas abiertas además de un manejo de estados en la conversación.
- *Memoria Contextual (memory.js)*: Implemente una clase llamada *MemoriaBot* que almacena los datos del ecosistema (flora/fauna cargados desde JSON), así como un historial de preguntas con sus respectivas respuestas (persistente en *localStorage*).
- *Gestor de Misiones (missions.js):* Genera dinámicamente objetivos educativos basados en las especies presentes en el terreno. Estas cuentan con una dificultad progresiva y marcadores de posición.

Mecanismo de Respuestas. El sistema prioriza respuestas mediante un algoritmo de cascada:

1. *Patrones Estructurados*: Detecta consultas frecuentes mediante expresiones regulares. Ejemplo:
   ```
   const regexCuantos = /^(?:¿)?cuantos\s+(\w+)\s+hay\??$/i;
   if (matchCuantos) { // Respuesta desde memoria contextual }
   ```
2. *Consultas a Memoria*: Para preguntas cuantitativas, accede a los datos precargados:
   - Conteo por especie (contarPorEspecie())
   - Ubicaciones georreferenciadas
3. *Modelo de Lenguaje*: Utiliza el API de OpenRouter con optimización pedagógica. Ejemplo:
   ```
   const promptFauna = 'Responde como biólogo:
   Explica que hay ${ conteo } ${ especie }...';
   llmClient.ask (promptFauna );
   ```

Generación de Misiones. El módulo Missions.js crea objetivos tomando en cuenta:

- *Detección Automática*: Carga y analiza la distribución espacial de la flora y fauna para sugerir misiones completables (ver Tabla 2).

- *Indicadores Visuales*: Añade marcadores 3D a los objetos relevantes usando Three.js. Ejemplo: `crearIconoTriangular (objetivo) ; // Cono verde rotado`
- *Progresión Pedagógica*: Dificultad ajustada según especies raras/comunes.

Tabla 2. Ejemplos de misiones generadas

Especie	*Misión típica*
Teporingo	"Toma 3 fotos de distintos teporingos"
Coyote	"Registra las coordenadas de avistamiento"
Flora endémica	"Identifica 5 ejemplares de pinos"

Optimizaciones Clave. El sistema incorpora técnicas avanzadas para mejorar la experiencia educativa:

- *Cache de Respuestas*: Almacena preguntas frecuentes (localStorage).
- *Normalización Lingüística*: Manejo de plurales y sinónimos.
- *Retroalimentación Visual*: Cambios de animación del avatar según estado.

Este diseño permite al agente funcionar tanto en modo autónomo (guiando al usuario mediante misiones) como reactivo (respondiendo consultas específicas), siempre contextualizado al ecosistema generado proceduralmente.

Integración y Visualización 3D. El sistema transforma datos "ecológicos" en formato JSON (archivos *Terreno.json* y *Flora&Fauna.json*) en un entorno 3D interactivo, utilizando un *pipeline* estructurado que garantiza coherencia espacial y diversidad biológica a través de los siguientes componentes: A. Carga y Procesamiento de Datos, B. Renderizado 3D, C. Agente Virtual Guía, D. Coordinación del Sistema. La clase *main.js* gestiona: la carga de terreno, la generación procedural (WFC y algoritmos genéticos) y la validación de posiciones para evitar solapamientos.

6 Resultados

Generación de terreno: Se implementó exitosamente el algoritmo WCF para la creación de mapas bidimensionales representando biomas diferenciados (pasto, roca, agua), con transiciones consistentes según reglas de adyacencia y altitud. Se generan automáticamente configuraciones topográficas a partir patrones en un archivo externo.

Distribución de flora y fauna: Mediante un algoritmo genético (GA), se logró ubicar organismos dentro del terreno respetando restricciones ecológicas como hábitat preferido, altitud adecuada, densidades poblacionales y relaciones tróficas.

Exportación e integración de datos: Los resultados del generador de biomas y del GA se exportaron a archivos en formato JSON, que fueron interpretados por el motor 3D en el cliente (*frontend*) utilizando Three.js.

Visualización tridimensional: Como se ver en la zona izquierda de la Figura 3, el entorno fue renderizado en un espacio 3D compuesto por geometrías voxelizadas y modelos GLTF. La implementación del sistema de coordenadas normalizadas permitió

escalar y adaptar el entorno a diferentes resoluciones de pantalla, manteniendo la integridad espacial del ecosistema.

Agente conversacional educativo: Como se puede visualizar en la zona derecha de la Figura 3, se integró un *chatbot* con funciones de guía interactiva, basado en reglas y en modelos de lenguaje. Este agente es capaz de interpretar preguntas frecuentes, acceder a datos internos del ecosistema (especies, cantidades, ubicaciones) y proponer misiones educativas que se ajustan dinámicamente al contenido generado.

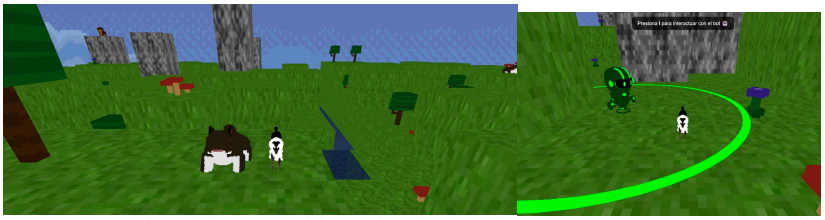

Figura 3. Vista del entorno tridimensional generado *proceduralmente* + Representación del agente conversacional en su forma de robot guía

Validación funcional: Se realizaron pruebas internas de funcionalidad que confirmaron el correcto flujo de datos entre módulos, la consistencia ecológica del entorno y la operatividad del sistema conversacional. Además, se verificó la capacidad del simulador para ejecutarse en navegadores web con recursos limitados.

Documentación y modularidad: Todo el sistema fue construido con una arquitectura modular que facilita el mantenimiento, escalado y futura integración de nuevas especies, biomas o mecanismos de interacción. Los archivos de configuración permitieron modificar parámetros clave sin alterar el código base.

7 Conclusiones y trabajo futuro

Se desarrollo con éxito un prototipo funcional que interpreta correctamente el texto procedural y presenta con modelos 3D, además de la integración del agente conversacional adaptado al ámbito educativo. La simulación permite explorar el ecosistema generado, así como interactuar con la flora y fauna correctamente.

Uno de los principales logros en la elaboración del trabajo fue el homologar la creación de contenido procedural realizado por el servidor con las necesidades y limitaciones del cliente. Para ello se tuvieron que resolver problemas técnicos como la correcta abstracción de las reglas del ecosistema, para dar origen a escenarios coherentes con la realidad y que pudieran brindar profundidad al entorno, además de buscar llevar la experiencia al mayor público posible mediante requisitos de hardware relativamente bajos. En cuanto a los retos del agente conversacional se identificó la necesidad de ajustar el lenguaje utilizado en función de la edad y los conocimientos previos, enriqueciendo el conjunto desde una perspectiva pedagógica

Como posibles líneas de mejora se proponen: 1) Una ampliación del catálogo de especies registradas y/o sus relaciones, 2) Afinar las respuestas del Agente con otros

parámetros como pueden ser el nivel educativo del usuario o sus intereses, 3) Realizar pruebas con *beta testers* con la finalidad de medir el impacto del aprendizaje, 4) Realizar un despliegue de acceso público y 5) Añadir opciones de accesibilidad, como protección contra la epilepsia, o añadir audio e interfaces multilingüe.

Creemos que gracias a la flexibilidad del proyecto este puede servir como base a la elaboración de proyectos que adapten otro tipo de biomas y especies endémicas para ayudar a concientización de la realidad actual que viven muchos ecosistemas mexicanos y ultimadamente generar un impacto social y ambiental positivo.

Referencias

Brellas, I. (2021). "Combining wave function collapse and evolutionary algorithms for controlled content generation". *Master's thesis, University of Malta*.

CONABIO. (2011). *La Biodiversidad en Puebla: Estudio de Estado. México. Gob. del Edo. de Puebla, BUAP.* Obtenido de Comisión Nacional para el Conocimiento y Uso de la Biodiversidad: https://smadsot.puebla.gob.mx/images/Biodiversidad_en_Puebla2.pdf

CONANP. (2015). *Mamíferos terrestres: Teporingo, zacatuche, conejo de los volcanes, tepolito, volcano rabbit.* Obtenido de Comisión Nacional de Áreas Naturales Protegidas: https://conanp.gob.mx/conanp/dominios/especies/WEB/docs/fichas/teporingo.pdf

CONANP. (2024). *Listado de especies - parque nacional iztaccíhuatl-popocatépetl.* Obtenido de Comisión Nacional de Áreas Naturales Protegidas: https://conanp.gob.mx/conanp/dominios/iztapopo/listado_de_especies.php

Deitke, M., VanderBilt, E., Herrasti, A., Weihs, L., Salvador, J., Ehsani, K., . . . Mottaghi, R. (2022). "ProcTHOR: Large-Scale Embodied AI Using Procedural Generation". *arXiv*. Obtenido de https://arxiv.org/abs/2206.06994

Hajj-Hassan, M., Chaker, R., & Cederqvist, A.-M. (2024). "Environmental Education: A Systematic Review on the Use of Digital Tools for Fostering Sustainability Awareness". *Sustainability, 16*(9), 3733. doi:https://doi.org/10.3390/su16093733

Mao, X., Yu, W., Yamada, K. D., & Zielewski, M. R. (2024). "Procedural Content Generation via Generative Artificial Intelligence". *arXiv*. Obtenido de https://arxiv.org/abs/2407.09013

Park, J. S., O'Brien, J. C., Cai, C. J., Morris, M. R., Liang, P., & Bernstein, M. S. (2023). "Generative Agents: Interactive Simulacra of Human Behavior". *arXiv*. Obtenido de https://arxiv.org/abs/2304.03442

SEMARNAT. (2019). *Informe de la situación del medio ambiente en méxico, edición 2018.* Obtenido de Secretaría de Medio Ambiente y Recursos Naturales: https://apps1.semarnat.gob.mx:8443/dgeia/informe18/tema/pdf/Informe2018GMX_web.pdf

United Nations. (2023). *Informe de los objetivos de desarrollo sostenible 2023: Edición especial por un plan de rescate para las personas y el planeta.* Obtenido de

https://unstats.un.org/sdgs/report/2023/The-Sustainable-Development-Goals-Report-2023_Spanish.pdf

Capítulo 10

Adquisición de datos para el análisis de la locomoción de la Gromphadorhina Portentosabajo distintas condiciones térmicas

Eduardo Gracidas Reyes[1], Gerardo Ulises Díaz Arango[2]

[1] Universidad de las América Puebla,
[2] Universidad Veracruzana,

[1]eduardo.gracidasrs@udlap.mx,[2]guda.diaz.gd@gmail.com

Resumen. Este trabajo presenta el análisis de los efectos de la temperatura en la locomoción de Gromphadorhina portentosa (cucaracha gigante de Madagascar) para aplicaciones en bio-robótica. Se estudia el efecto de 42 cucarachas expuestas a seis rangos de temperatura (18 a 38°C), para esto se implementó un sistema de monitoreo de movimiento usando un mecanismo omnidireccional de baja fricción y sensores ópticos (Mouses comerciales). Los resultados muestran variaciones estadísticamente significativas en la velocidad, aceleración y distancia recorrida, con un mejor desempeño para una temperatura de 30°C. El diseño experimental incluyó control ambiental preciso y protocolos éticos. Los hallazgos que se reportan aportan un conjunto de datos que sirven como referencia en la comprensión de la locomoción de la Gromphadorhina Portentosa. Así como, un punto de partida para un estudio más extenso en el desarrollo de biobots eficientes en aplicaciones como búsqueda y rescate. Finalmente, se discuten limitaciones y futuras líneas de investigación.

Palabras Clave: Gromphadorhina Portentosa, Biomecánica, Temperatura, Biobots.

1 Introducción

Los insectos son organismos que se destacan por sus notables capacidades de locomoción, que en muchos casos superan a las de los sistemas robóticos actuales. Están dotados de una amplia variedad de sensores biológicos capaces de percibir diferentes condiciones ambientales, orientación espacial y movimiento (Cruse et.al, 2009). Asimismo, disponen de numerosos actuadores biológicos que les permiten una locomoción eficiente y de bajo consumo energético. Estos biosensores y bio-actuadores integrados pueden ser aprovechados por sistemas cibernéticos para el desarrollo de biobots eficientes con aplicaciones en diversos campos.

Por lo tanto, los insectos han emergido como una alternativa prometedora en el desarrollo de plataformas bio-robóticas. Entre ellos, la Gromphadorhina portentosa, mejor conocida como cucaracha de Madagascar, ha sido objeto de estudio debido a sus

distintivas características fisiológicas y comportamentales. Esta especie destaca por su cuerpo de tamaño relativamente grande, que generalmente oscila entre los 5 y 10 cm, así como por su capacidad única para emitir un característico siseo (Monahan et.al, 2023). Pertenece al orden Blattodea y a la familia Blaberidae. Su hábitat natural se encuentra en el sotobosque de las selvas tropicales húmedas de la isla de Madagascar, donde suele refugiarse entre la hojarasca seca que cubre el suelo del bosque (Monahan et.al, 2023).

No obstante, diversos estudios han enfrentado limitaciones significativas en el control de cucarachas cíborg, especialmente en lo relativo a su locomoción. Entre estas dificultades se incluyen la habituación a los estímulos, la tendencia a quedar atrapadas en límites físicos y períodos prolongados de inmovilidad (Ariyanto et.al 2024; Erickson et.al, 2015), lo que complica el diseño de protocolos de estimulación eficaces.

Sin embargo, una de las limitaciones más importantes es su alta sensibilidad a factores ambientales como la temperatura, la humedad y la luz, los cuales influyen notablemente en la locomoción de insectos ectotermos (Abram et.al, 2017). En particular, ectotermos con preferencia al calor, como la Gromphadorhina portentosa, muestran mejor desempeño a temperaturas elevadas, teniendo una mayor velocidad y actividad a costa de una sociabilidad reducida (Michelangeli et.al, 2018). Mientras que la exposición a temperaturas elevadas puede provocar estrés fisiológico severo, desecación e incluso coma inducido, como se ha observado en la *Porcellio laevis* a 32°C (Bozinovic, 2016). Por lo tanto, el estudio de la locomoción de la Gromphadorhina Portentosa ante diferentes variables medioambientales juega un papel importante para el desarrollo de sistemas cibernéticos que puedan ser aplicados en situaciones de la vida real tal y como ocurre con las operaciones de rescate o *SAR,* en las que podrían ser de gran ayuda para cubrir amplias áreas de búsqueda sin gastar demasiados recursos, así como acceder de forma eficiente en zonas de desastre (Liu et.al, 2007).

El presente estudio tuvo como objetivo cuantificar y analizar los efectos de la temperatura sobre la locomoción de la Gromphadorhina portentosa, mediante el muestreo y análisis de datos de movimiento obtenidos de 42 cucarachas expuestas a seis rangos de temperatura diferentes: 18, 22, 26, 30, 34 y 38 °C, obteniendo una extensa base de datos en el proceso, misma que se encuentra pública en Zenodo (Gracidas, 2025).

La estructura del trabajo se organiza en seis secciones (contando esta). En la segunda, se describe detalladamente el diseño experimental empleado. La tercera sección aborda la implementación del sistema de control ambiental, así como el desarrollo del sistema de monitoreo del movimiento. En la cuarta sección, se detallan las fórmulas y procedimientos utilizados para extraer las características de los datos recopilados. Posteriormente, en la quinta sección, se presentan y analizan los resultados obtenidos. Finalmente, la sexta sección expone las conclusiones derivadas de la investigación.

2 Diseño experimental

Debido a la notoria variabilidad conductual entre individuos de Gromphadorhina portentosa, particularmente en lo que respecta a sus patrones de locomoción, y considerando lo reportado en estudios anteriores que evidencian la existencia de ejemplares con comportamientos extremos (como actividad motriz excesiva o, por el contrario, una tendencia marcada al letargo), se consideró fundamental trabajar con una muestra suficientemente amplia. Por esta razón, se seleccionaron 42 cucarachas para ser sometidas a pruebas experimentales, con el objetivo de mitigar la influencia de posibles valores atípicos y lograr una mayor robustez en el análisis estadístico de los datos obtenidos. Este tamaño muestral fue considerado adecuado para capturar una representación más equilibrada de la variabilidad natural presente en la población de estudio.

Los sujetos experimentales fueron seleccionados de manera aleatoria a partir de una colonia mantenida bajo condiciones ambientales estables. La colonia se conservaba a temperatura ambiente controlada, fluctuando entre los 25 °C y los 30 °C, en completa oscuridad durante las 24 horas del día. La alimentación de los ejemplares se basó exclusivamente zanahorias, calabacines, jitomates y cáscaras de naranja, seleccionados por su valor nutritivo y por no presentar riesgos de toxicidad para la especie.

Solo se empleó un único criterio de inclusión, el cual fue la longitud corporal, incluyendo en el estudio solo a aquellos individuos cuya longitud total fuera mayor o igual a 5 cm. No se consideraron otras variables como el peso, la edad, el sexo, la pigmentación ni ningún otro rasgo anatómico o fisiológico con la finalidad de mantener una selección lo más aleatoria posible y sin sesgos adicionales.

Antes de ser sometidos a las sesiones experimentales, todos los sujetos pasaron por un procedimiento invasivo para hacerlas aptas para el sistema de monitoreo de movimiento. Este procedimiento consistió en limar la superficie superior del tórax y del pronoto (Heyborn, 2012) con una lija. Una vez expuesta adecuadamente la zona de fijación, se aplicó una pequeña cantidad de adhesivo instantáneo (marca comercial Kola Loka), sobre el cual se fijó un fragmento de velcro con una numeración única asignada a cada individuo. Esta intervención fue realizada con el máximo cuidado para minimizar cualquier alteración fisiológica o del comportamiento posterior a la manipulación. Los más altos lineamientos éticos también fueron seguidos al reducir en la medida de lo posible el dolor inducido en el sujeto de pruebas, así como cuidados post-intervención para asegurar su supervivencia y bienestar.

Posteriormente, los individuos intervenidos fueron alojados en grupos de cinco cucarachas dentro de contenedores de plástico individuales, donde permanecieron en reposo durante al menos 12 horas. Este periodo de descanso tuvo como finalidad asegurar la recuperación física tras el procedimiento de marcaje, así como evitar posibles efectos derivados del hacinamiento o del estrés social, que pudieran influir negativamente en el comportamiento locomotor durante las pruebas y que han sido reportados en estudios previos (Clark, 1994).

Cada sesión experimental tuvo una duración aproximada de 25 minutos. Este intervalo fue establecido con el fin de abarcar no solo la fase inicial de aclimatación del sujeto a las nuevas condiciones térmicas, sino también el tiempo suficiente para

registrar patrones locomotores más estables y representativos. Durante las sesiones, se registraron en tiempo real diversas variables medioambientales relevantes para el estudio como la temperatura del entorno, iluminación y posición del individuo dentro de la caja de control ambiental. La frecuencia de muestreo fue de aproximadamente 36 Hz, suficiente para capturar adecuadamente la locomoción de los sujetos de pruebas. Con el fin de evitar la aparición de efectos adversos acumulativos (tales como fatiga física, habituación al entorno experimental o estrés térmico) se limitaron las pruebas de cada individuo a un máximo de cuatro sesiones por día. Asimismo, entre cada sesión consecutiva se estableció un intervalo de reposo mínimo de dos horas, con el propósito de permitir una recuperación fisiológica adecuada y una re-aclimatación pasiva a las condiciones basales de temperatura y oscuridad en las que originalmente eran mantenidas.

Las pruebas fueron programadas y distribuidas aleatoriamente a lo largo del día, con el propósito explícito de neutralizar cualquier posible influencia de los ritmos circadianos en el comportamiento locomotor de las cucarachas, que podrían sesgar los resultados. Adicionalmente, todas las sesiones se llevaron a cabo en condiciones de baja luminosidad, manteniéndose la intensidad lumínica por debajo de los 30 lux, minimizando así cualquier interferencia lumínica en el comportamiento de los sujetos.

3 Descripción del sistema

3.1 Sistema de control ambiental

El sistema de control ambiental consiste en 4 sensores, dos de temperatura y humedad SHT45 con una exactitud de $\pm 1\%\ RH$ y ± 0.1°C, un rango de temperatura de -40 y 125°C y una resolución de 0.01°C y dos sensores de iluminación BH1750 con un rango de iluminación de 0 a 65,000 lux y resolución ajustable de entre 0.5 y 4 lux. Estos sensores fueron colocados en una caja de poliestireno expandido de 35x30cm en cada extremo a la misma altura con la finalidad de calcular la temperatura, humedad e iluminación promedio dentro de la caja y no solo en un extremo de esta. Para ajustar la temperatura se usaron dos enfriadores Peltier de 4A de consumo a 12V, uno en su extremo caliente y el otro en el extremo frío, controlados por un puente H de alta potencia mediante señales PWM, de este modo se podía enfriar o calentar la caja según se requiriese sin mover nada físicamente. La parte lógica fue llevada a cabo por un microcontrolador ESP32 el cual transmitía los datos a la computadora por vía serial mediante un cable.

Con los ajustes necesarios, el sistema era capaz de mantener la temperatura objetivo con un margen de error de menos de $\pm 0.1°C$ en las pruebas de 26°C o menos y con un margen de menos de $\pm 0.03°C$ en las pruebas a 30°C o más.

3.2 Sistema de monitoreo de posición

Basándonos en el sistema propuesto en el estudio de Erickson en el cual utilizaron dos mouses ópticos y una bola de unicel a baja fricción utilizando un compresor de aire

(Erickson et.al 2015), se diseñó un sistema similar, pero utilizando rodajes de muy baja fricción en lugar del compresor de aire. Esta modificación redujo considerablemente el consumo energético y el ruido producido por el sistema, así como el coste sin sacrificar significativamente la baja fricción de este.

Los mouses ópticos utilizados fueron de la marca LOGITECH, específicamente el modelo M170 el cual tiene una resolución de 1000DPI y es inalámbrico. Esta última característica permitió disminuir considerablemente la complejidad del cableado del sistema e impidió errores durante las pruebas.

Para calibrar el sistema se midió la circunferencia de la bola (la cual es de 37.69cm) utilizando los mouses ópticos para ver qué tan alejados estaban de la circunferencia real. En casi todas las pruebas hechas, el sistema arrojó un valor de circunferencia de entre 36.7 y 38.5cm, siendo suficientemente exacto para este tipo de pruebas.

La finalidad de utilizar dos mouses en lugar de solo uno fue el de capturar las componentes de movimiento adelante, atrás, derecha, izquierda y giro. Cada mouse registra un plano diferente del movimiento del sujeto de pruebas y, en conjunto permiten describir la totalidad de este tal y como se ve en la figura 1.

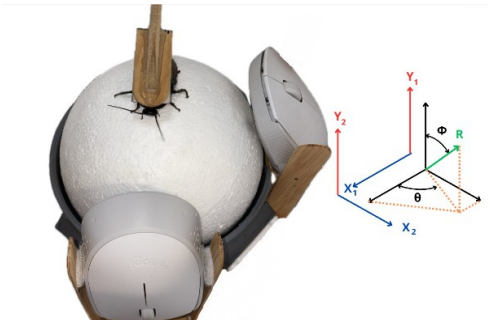

Figura 1. Ilustración de los diferentes planos de movimiento registrados por cada mouse óptico en el sistema de monitoreo.

Para mantener fijo al sujeto de pruebas justo al centro de la bola de unicel para un monitoreo de movimiento apropiado, se hizo uso de una base de madera con un velcro a una altura de unos 3cm por encima del punto máximo de la bola. De este modo, los sujetos no quedaban aplastados contra la bola ni flotando parcialmente sobre de esta.

3.3 Comunicación entre sistemas

Los sistemas de monitoreo de movimiento y de control ambiental se encuentran sincronizados y son gestionados mediante un programa desarrollado en Python y ejecutado en la computadora principal, tal como se ve en la Figura 2. Este programa recibe en tiempo real los datos en crudo del sistema de monitoreo de movimiento, transmitidos de forma inalámbrica a través de dos mouses ópticos, así como la información relativa a las variables medioambientales en el interior de la caja de poliestireno, la cual es enviada vía puerto serial del microcontrolador ESP32.

El propósito central de la herramienta de software es unificar ambos flujos de datos en un único sistema de registro, facilitando así su procesamiento y análisis. En particular, los datos provenientes de los mouses ópticos requieren un preprocesamiento digital, que consiste en una calibración para convertir los desplazamientos registrados en *ticks* a unidades de distancia física (centímetros), que es la unidad de medición de interés para cuantificar la locomoción.

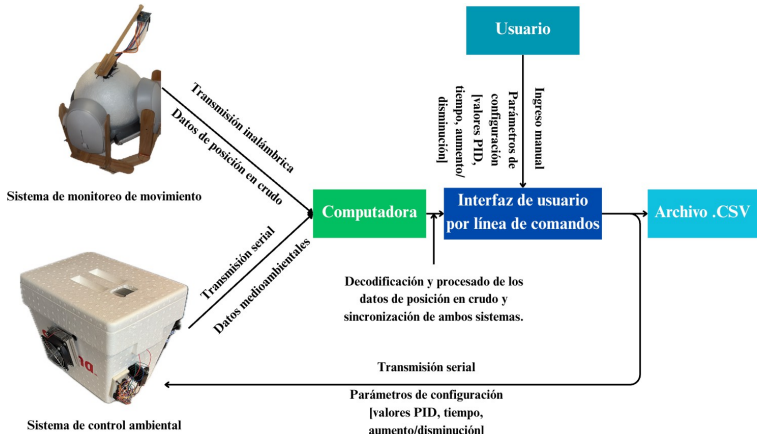

Figura 2. Comunicación entre el sistema de monitoreo de movimiento y el de control ambiental con la computadora y el CLI.

Adicionalmente, el programa incorpora una interfaz de línea de comandos (CLI) que permite al usuario (en este caso, el experimentador) introducir diversos parámetros de configuración del sistema. Entre ellos se incluyen los valores del control PID para el sistema de regulación térmica, la duración total de la prueba y si se aumentará o disminuirá la temperatura dentro de la caja.

4 Procesamiento de datos

4.1 Obtención de variables

A partir de la base de datos construida, se obtuvieron diversos parámetros con el objetivo de evaluar la locomoción de los sujetos de prueba bajo diferentes condiciones de temperatura. Los primeros y más fundamentales corresponden a aquellos empleados en el sistema descrito por Erickson. Dichos parámetros son la velocidad $v(t)$ y el giro $\omega(t)$, calculados mediante las ecuaciones 1 y 2 (Erickson et al., 2015).

$$v(t) = \sqrt{\dot{y_1}^2 + \dot{y_2}^2} \quad (1)$$

$$\omega(t) = \frac{\dot{x_1} + \dot{x_2}}{2R} \quad (2)$$

Donde, $\dot{y}_x$ y $\dot{x}_x$ son la primera derivada de cada plano de movimiento obtenida por el método de diferencia central y *R* el radio de la esfera de unicel (el cual es de 6cm). La aceleración se calculó a partir de la segunda derivada de la velocidad, también utilizando el método de diferencias centrales, de acuerdo con la ecuación 3.

$$a(t) = \frac{v_{i+1} - v_{i-1}}{2\Delta t} \quad (3)$$

La velocidad y la aceleración máximas fueron determinadas mediante el valor absoluto de las respectivas magnitudes en cada instante de tiempo, empleando posteriormente la función `max()` de Python. Por otro lado, los valores medios de velocidad y aceleración se obtuvieron según las ecuaciones 4 y 5 respectivamente.

$$v_{prom} = \frac{1}{N}\sum_{i=1}^{N} v_i \quad (5)$$

$$a_{prom} = \frac{1}{N}\sum_{i=1}^{N} a_i \quad (5)$$

El recorrido total se calculó utilizando el teorema de Pitágoras, aplicado a los dos planos de desplazamiento, como se muestra en la ecuación 6.

$$d = \sum_{i=1}^{N} \sqrt{\left(x_{1_{i+1}} - x_{1_i}\right)^2 + \left(y_{1_{i+1}} - y_{1_i}\right)^2 + \left(x_{2_{i+1}} - x_{2_i}\right)^2 + \left(y_{2_{i+1}} - y_{2_i}\right)^2} \quad (6)$$

Adicionalmente, se calculó la tortuosidad de cada trayecto, definida como la razón entre la distancia total recorrida y la distancia lineal entre el punto de inicio y el punto final de acuerdo a la ecuación 7.

$$Tortuosidad = \frac{D_{Total}}{D_{lineal(inicio-fin)}} \quad (7)$$

4.2 Análisis estadístico

Se empleó un análisis de varianza para medidas repetidas (ANOVA de medidas repetidas) con el propósito de evaluar el efecto de la temperatura sobre los distintos parámetros de locomoción obtenidos. Esta técnica estadística fue seleccionada debido a que permite analizar datos en los que un mismo sujeto es evaluado en diferentes condiciones (en este caso, distintos niveles de temperatura), controlando la variabilidad intraindividual y aumentando la potencia estadística del análisis.

Posteriormente, al detectarse diferencias significativas a nivel global, se procedió a realizar análisis post-hoc con corrección de significancia, con el fin de identificar específicamente qué pares de condiciones térmicas presentaban diferencias estadísticamente significativas entre sí.

En este análisis, se consideraron como variables dependientes todas aquellas obtenidas a partir del procesamiento de los datos de locomoción, a saber: tortuosidad, velocidad máxima, velocidad promedio, aceleración máxima, aceleración promedio, giro angular y recorrido total.

Los resultados obtenidos tras el análisis estadístico fueron graficados utilizando un mapa de calor o *heatmap* con cada parámetro como variable independiente y con cada comparación de grupos. También se obtuvieron diagramas de caja (*boxplots)* para los parámetros de recorrido y velocidad máxima.

5 Resultados

A partir de la distribución de los datos de recorrido y velocidad máxima que se muestran en las Figuras 3 y 4, se observa que el recorrido aumenta progresivamente entre 18 °C y 30 °C; sin embargo, tras este punto se aprecia una ligera disminución, particularmente marcada a 34 °C. En cuanto a la velocidad máxima, el incremento con la temperatura es evidente hasta los 30 °C, aunque a partir de este valor el aumento resulta prácticamente insignificante.

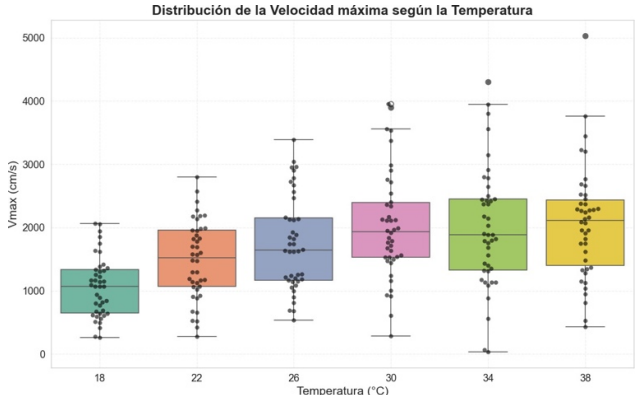

Figura 3. Distribución de la velocidad máxima por cada grupo de temperatura.

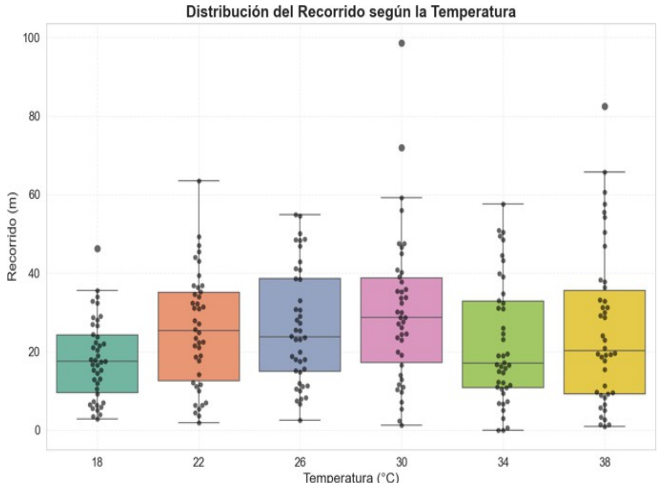

Figura 4. Distribución del recorrido total por cada grupo de temperatura.

Tanto en la Tabla 1 como en la Figura 5 se presentan los valores promedio de cada parámetro de locomoción para los distintos grupos de temperatura. Para la Tabla 1 se utilizan los valores originales mientras que en la Figura 5 se normalizan los valores para

un despliegue más práctico. El patrón observado en los diagramas de caja de las Figuras 3 y 4 se repite para las variables de recorrido y velocidad máxima. El giro, al igual que el recorrido, incrementa casi de manera lineal entre 18 °C y 30 °C, aunque deja de aumentar significativamente en el intervalo de 30 °C a 38 °C. En contraste, la tortuosidad no sigue una tendencia lineal, sino que presenta dos picos diferenciados a 26 °C y 34 °C. Por su parte, tanto la aceleración máxima como la aceleración promedio exhiben un comportamiento muy similar al de la velocidad máxima y promedio, respectivamente. En la Figura 5 se aprecia con claridad que la mayoría de los parámetros, con excepción de la tortuosidad y el giro, alcanzan un valor muy elevado a 30 °C, mientras que, en todos los casos, el valor mínimo se registra a 18 °C.

Tabla 1. Valor promedio por parámetro de locomoción a cada nivel de temperatura.

Parámetro	18°C	22°C	26°C	30°C	34°C	38°C
Recorrido [m]	18.01	25.47	26.28	30.3	22.31	25.81
Giro [$\frac{deg}{s}$]	0.03	14.04	34.88	43.51	43.78	55.74
Tortuosidad	1.55	1.6	1.88	1.75	1.91	1.63
V. max [$\frac{m}{s}$]	10.59	14.84	17.54	20.21	19.84	20.64
V. prom [$\frac{m}{s}$]	0.79	1.14	1.19	1.35	1.01	1.1
A. max [$\frac{m}{s^2}$]	6816.95	9040.07	11073.28	12803.22	12711.46	13087.78
A. prom [$\frac{m}{s^2}$]	319.71	460.64	486.34	569.93	435.8	476.95

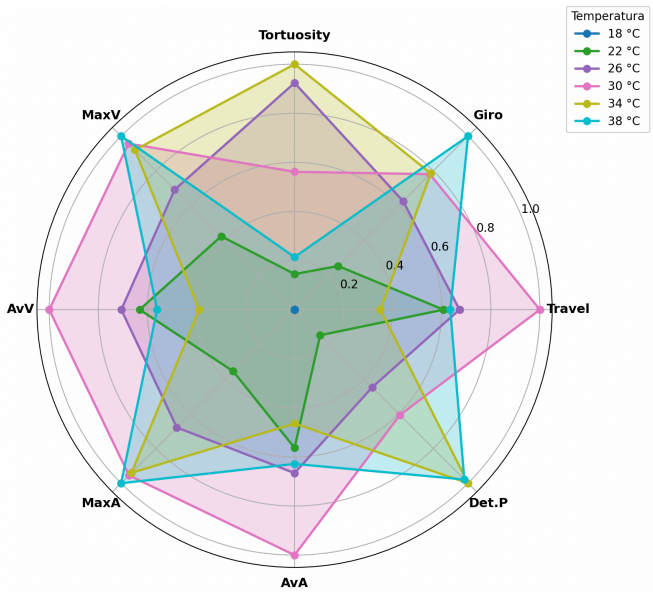

Figura 5. Gráfico de radar con los valores promedio normalizados por parámetro de locomoción para cada grupo de temperatura.

El análisis estadístico obtenido tras el análisis de varianzas por ANOVA y post-hoc es el que se ve en el mapa de calor de la Figura 6, donde el color es más claro entre menor es el valor p. Es notorio que el parámetro con más significancia estadística dado su valor p es la velocidad máxima, seguido por la aceleración máxima. El recorrido y la velocidad y aceleración promedio cuentan con significancia estadística solo entre muy bajas y medias temperaturas, implicando que la temperatura tiene cierta influencia en estas variables, pero no de forma necesariamente lineal. Por su parte, la tortuosidad y el giro carecen de valor estadístico lo cual parece estar directamente correlacionado con el hecho de que son los únicos dos parámetros que no alcanzan un valor elevado a 30°C y con la baja linealidad de la tortuosidad.

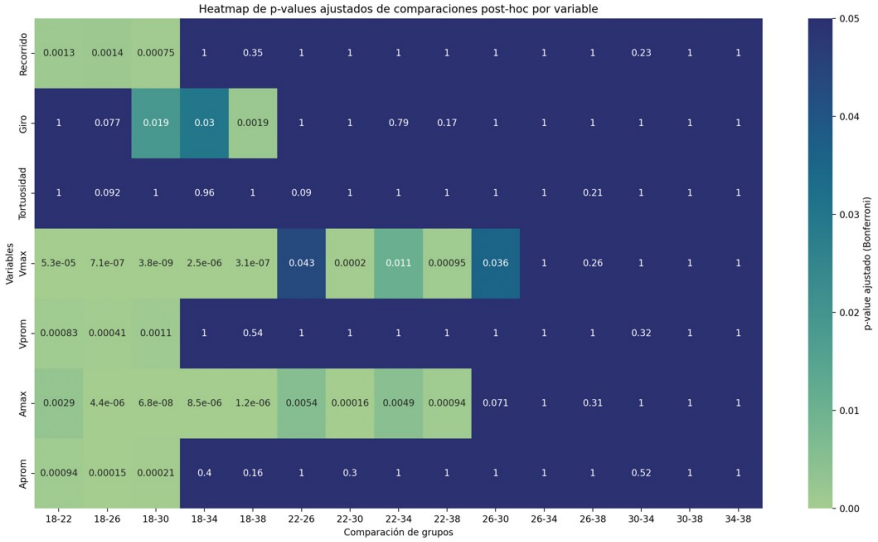

Figura 5. Valores p para cada parámetro y cada grupo de temperatura.

6 Conclusiones y trabajo futuro

A partir de los resultados obtenidos, fue posible identificar ciertos efectos significativos de la temperatura sobre la locomoción de Gromphadorhina portentosa, siendo la velocidad máxima el parámetro que mostró una mayor sensibilidad a las variaciones térmicas. No obstante, los datos observados parecen contradecir la hipótesis inicial, según la cual se esperaba que el aumento de la temperatura favoreciera un incremento continuo en la actividad locomotora de esta especie. En cambio, los resultados sugieren la existencia de un punto de inflexión en torno a los 30 °C, a partir del cual el rendimiento locomotor deja de incrementarse, mostrando una tendencia a estabilizarse o incluso a disminuir, como se evidencia a 34 °C. El hecho de que la tortuosidad no incremente de forma lineal y que tenga su pico a 34°C indica que las cucarachas se mueven de manera más errática y desorganizada en torno a ese rango de temperatura, sugiriendo un posible malestar en las cucarachas que podría ser la razón tras el decremento de su locomoción. Esta observación sugiere que el efecto de la temperatura sobre la locomoción no es lineal, sino que podría estar condicionado por límites fisiológicos o comportamentales propios del organismo.

De cualquier manera, es posible establecer a partir del análisis estadístico que en el rango de 18°C a 30°C la hipótesis inicial basada en el comportamiento de los animales ectotermos se cumple satisfactoriamente, habiendo un incremento casi lineal en ciertos parámetros de interés como la velocidad y la aceleración e inclusive, aunque de manera más limitada, el recorrido y el giro. En rangos de temperatura más elevados se debería esperar un comportamiento menos lineal en la locomoción.

En futuros estudios que empleen a la Gromphadorhina portentosa como plataforma bio-robótica deben considerar los efectos de la temperatura aquí reportados para sus pruebas, sobre todo aquellos que utilizan como parámetros el recorrido y la velocidad. Además, los estudios que hagan uso de señales de estimulación para el control de la locomoción en esta especie de cucarachas deberían considerar utilizar o bajas o altas temperaturas para sus pruebas para evaluar la efectividad de dichas señales pues, a temperaturas intermedias como 26°C o 30°C sería difícil determinar si los efectos sobre la locomoción son a causa de la señal de estimulación o por la aumentada locomoción que aquí se reporta.

Es importante resaltar que la Gromphadorhina portentosa tiene una variabilidad entre individuos muy elevada. Por esta razón, futuros estudios no deberían limitar sus poblaciones a un bajo número de sujetos de prueba pues se corre el riesgo de caer en errores por sesgos a causa de individuos muy activos o con baja actividad. La población aquí empleada demostró ser suficiente para obtener resultados estadísticamente significativos, sin embargo, para futuros estudios se deben considerar poblaciones incluso más grandes.

Cabría añadir que, si bien se recogieron datos poblacionales como el género o la longitud de las cucarachas, la relación de estos con los parámetros de locomoción no fue estudiada a fondo. A partir de nuestras observaciones, notamos una posible correlación entre la longitud y el recorrido total de las cucarachas. Sin embargo, futuras investigaciones deberían determinar si estos parámetros poblacionales tienen efectos notorios en la locomoción de la Gromphadorhina portentosa o, si, por el contrario, son despreciables. Finalmente, se debe considerar que no se controlaron algunas variables ambientales que podrían haber afectado la locomoción de los sujetos de pruebas como el ruido o la presencia humana cerca del área de pruebas con lo cual se desconocen los posibles efectos de estas. En función de nuestras observaciones, no detectamos algún indicio de afectación relevante debido a estas variables ambientales, pero falta investigación al respecto para confirmar esta aseveración.

Referencias

Abram, P. K., Boivin, G., Moiroux, J., & Brodeur, J. (2017). Behavioural effects of temperature on ectothermic animals: Unifying thermal physiology and behavioural plasticity.
Biological Reviews of the Cambridge Philosophical Society, 92(4), 1859–1876. https://doi.org/10.1111/brv.12312

Ariyanto, M., Refat, C. M. M., Yamamoto, K., & Morishima, K. (2024). Feedback control of automatic navigation for cyborg cockroach without external motion capture system. Heliyon, 10, e26987. https://doi.org/10.1016/j.heliyon.2024.e26987

Clark, Deborah & Moore, Allen. (1994). Social interactions and aggression among male Madagascar hissing cockroaches (Gromphadorhina portentosa) in groups (Dictyoptera: Blaberidae). Journal of Insect Behavior. 7. 199-215. https://doi.org/10.1007/BF01990081.

Cruse, H.; Dürr, V., Schilling, M. y Schmitz, J.(2009). Principles of Insect Locomotion. In Spatial Temporal Patterns for Action-Oriented Perception in Roving Robots; Arena, P.; Patanè, L., Eds.; *Cognitive Systems Monographs,* Springer: Berlin, Heidelberg; Vol. 1,

pp. 43–96. https://doi.org/10.1007/978-3-540-88464-4_2.

Erickson, J.C.; Herrera, M.; Bustamante, M.; Shingiro, A.; Bowen, T.(2015). Effective stimulus parameters for directed locomotion in Madagascar hissing cockroach biobot. PLOS ONE, 10, e0134348. https://doi.org/10.1371/journal.pone.0134348.

Gracidas, E. (2025). Temperature-Dependent Locomotion Data of the Hissing Cockroach (Gromphadorhina portentosa) (0.1) [Data set]. Zenodo. https://doi.org/10.5281/zenodo.16882893

Heyborne, W & Fast, M y Goodding, D. (2012). The Madagascar Hissing Cockroach: A New Model for Learning Insect Anatomy. The American Biology Teacher. 74. 185-189. https://doi.org/10.1525/abt.2012.74.3.11.

Liu, J.; Wang, Y.; Li, B.; et al. (2007) . Current research, key performances and future development of search and rescue robots. Frontiers of Mechanical Engineering in China, 2, 404–416. https://doi.org/10.1007/s11465-007-0070-2

Michelangeli, M., Goulet, C., Kang, H., Wong, B., & Chapple, D. (2018). Integrating thermal physiology within a syndrome: Locomotion, personality and habitat selection in an ectotherm. *Functional Ecology, 32*, Article e13034. https://doi.org/10.1111/1365-2435.13034

Monahan, C.F., Bogan, J. E., J. y LaDouceur, E.E.B. (2023). Histological Findings in Captive Madagascar Hissing Cockroaches (Gromphadorhina portentosa) and a Literature Review. *Veterinary Pathology*, 60, 667–677. https://doi.org/10.1177/03009858231166659.

Capítulo 11

Jaya y Lobo Gris aplicado a instancias de QAPLIB

Alhely González Luna, Rogelio González Velázquez, Abraham Sánchez López, Erika Granillo Martínez

Facultad de Ciencias de la Computación, Benemérita Universidad Autónoma de Puebla, México

gl224470257@alm.buap.mx, rogelio.gonzalez@correo.buap.mx, abraham.sanchez@correo.buap.mx, erika.granillom@correo.buap.mx

Resumen. El problema de asignación cuadrática (QAP), por su complejidad NP-hard y sus múltiples aplicaciones industriales, sigue siendo un tema clave en ciencias computacionales. Cuando se tienen instancias grandes y recursos limitados, las metaheurísticas ofrecen una alternativa para hallar soluciones cercanas a la óptima. Métodos como Búsqueda Tabú, Recocido Simulado, Algoritmo Genético, entre otras, han mostrado buenos resultados en la resolución de QAP. Sin embargo, metaheurísticas más recientes como Jaya (2016) y Lobo Gris (2011) que son reducidas en parámetros han sido poco exploradas en instancias del QAP. En este trabajo, se presenta una comparación de 9 metaheurísticas aplicadas a 4 instancias de tamaño mayor e igual a 30 (lo que se considera de gran tamaño) de la biblioteca QAPLIB. Los resultados muestran que Búsqueda Tabú genera las mejores soluciones, mientras que Jaya y Lobo Gris analizados aquí ofrecen un equilibrio entre búsqueda de parámetros y tiempo de ejecución. Finalmente, como trabajo futuro ampliaremos la búsqueda de parámetros para experimentar si esto reduce el GAP de los resultados generados por Jaya y Lobo Gris.

Palabras Clave: Problema de Asignación Cuadrática, Metaheurísticas, Jaya, Lobo Gris, Algoritmo Genético, Búsqueda Tabú, Recocido Simulado.

1 Introducción

El problema de asignar instalaciones a ubicaciones de tal forma que cada ubicación esté asignada a una sola ubicación y viceversa, en donde se conocen las distancias entre ubicaciones y las demandas de flujo entre instalaciones se define como el Problema de Asignación Cuadrática (QAP por sus siglas en inglés), este problema pertenece a la clase NP-Hard (Loiola, de Abreu, Boaventura-Netto, Hahn, y Querido, 2007). Algunas aplicaciones importantes del QAP en la industria son: diseños hospitalarios, cableado de tablero, modificación de campus, arqueología, química de reacciones, entre muchas más (Bhati y Rasool, 2014).

Al ser un problema NP-hard, no es posible encontrar una solución exacta en tiempo polinomial, existen varios métodos para hallar soluciones dependiendo del tamaño de la instancia (ubicaciones e instalaciones). Para problemas pequeños es posible hallar soluciones mediante algorítmos de Branch and Bound, Métodos de Relajación o Programación Dinámica. Sin embargo, para instancias de gran tamaño, no es posible aplicar estos métodos exactos ya que se requieren demasiados recursos computacionales, en estos casos, se aplican algorítmos metaheurísticos que iteran sobre posibles soluciones aleatorias hasta encontrar una solución cercana a la óptima (o la óptima). Algunas de las metaheurísticas más utilizadas son: Búsqueda Tabú, Recocido Simulado, GRASP, Algorítmos Genéticos, o metaheurísticas basadas en poblaciones como hormigas, abejas, etc (Ch, Hurtado, y Moncada, 2022). En el siguiente trabajo comparamos el resultado de 7 metaheurísticas clásicas con Jaya y Lobo Gris para 4 instancias tomadas de QAPLIB.

El documento está organizado de la siguiente forma: En la sección dos se presenta una revisión teórica del problema de asignación cuadrática, en la sección tres presentaremos un conjunto de algoritmos para un benchmarking que han sido previamente probados, en la sección cuatro revisaremos la teoría detrás de Jaya y Lobo Gris, destacaremos sus principales diferencias y ventajas, en la sección cinco mostraremos los resultados obtenidos para algunas instancias de prueba y en la sección seis daremos las conclusiones de este trabajo.

2 El problema de Asignación Cuadrática (QAP)

El problema de asignación cuadrática (QAP) introducido por primera vez por Koopmans y Beckmann (Koopmans y Beckmann, 1957), consiste en asignar un conjunto n de instalaciones a un conjunto de n ubicaciones, donde el flujo entre instalaciones y la distancia entre ubicaciones se conoce previamente (Ch, Hurtado, y Moncada, 2022). El problema puede formularse matemáticamente como:

$$\min_{p \in P} \sum_{i=1}^{n} \sum_{j=1}^{n} f_{ij}\, d_{p(i)p(j)} \qquad (1)$$

Donde f es la matriz de flujo y d la matriz de distancias, ambas de tamaño $n \times n$, y p es una permutación de tamaño n. El conjunto de todas las permutaciones posibles se denota con P. Para cada par de asignaciones (instalación i -ubicación j) el costo asociado al par será el flujo f_{ij} multiplicado por la distancia d_{ij}. La suma de estos términos sobre todos los pares dentro de P representa el costo total para la permutación p, mostrado en (1). El objetivo es encontrar una permutación p^* en P que minimice el costo total (James, Rego, y Glover, 2009). El problema de QAP puede verse como un problema de optimización combinatoria en el que cada permutación tiene asociado un costo, en este caso flujo por distancia, y debe hallarse la permutación que del menor costo.

Se demostró que el QAP pertenece a la clase de problemas NP-hard (Sahni y Gonzalez, 1976), por lo que no existe aún un algoritmo que tenga un error relativo

garantizado menor que una constante para cada instancia del QAP salvo cuando P = NP (Taillard, 1991).

3 Metaheurísticas Clásicas

En esta sección presentamos una breve revisión de 6 metaheurísticas que usaremos como comparación para Jaya y Lobo Gris; Colonia de Hormigas (ACO y FANT), Búsqueda Tabú, Enjambre de Partículas (PSO), Algoritmo genético, y Luciérnagas.

3.1 Colonia de Hormigas

La Optimización por Colonia de Hormigas ACO busca soluciones mediante hormigas artificiales que construyen caminos reforzados con feromonas. Una de sus principales limitaciones es la necesidad de ajustar múltiples parámetros (intensidad de feromona, evaporación, número de hormigas) (Merkle y Middendorf, 2002). En el contexto del QAP, ACO se ha aplicado para encontrar asignaciones eficientes entre instalaciones y localizaciones, pero tiende a requerir muchas iteraciones (Merkle y Middendorf, 2002).

3.2 Fast Ant System (FANT)

El algoritmo FANT es una variante de ACO que mejora la eficiencia mediante actualización inmediata de feromonas y un modelo centralizado con memoria compartida. Además, emplea listas de candidatos para reducir la complejidad de exploración (Taillard, 1998). En pruebas con instancias medianas de QAP, FANT ha demostrado converger más rápido que ACO clásico.

3.3 Búsqueda Tabú (TS)

La Búsqueda Tabú (TS) explora el vecindario de soluciones aplicando movimientos de 2-intercambio y 3-intercambio. Para evitar ciclos, utiliza una lista tabú en la que se incluyen movimientos que ya se han utilizado por un número de iteraciones (permanencia) definida, mientras que el criterio de aspiración le permite aceptar soluciones prohibidas si mejoran el óptimo actual (Taillard, 1991). En problemas de QAP de gran tamaño, TS ha mostrado ser una de las heurísticas más competitivas, logrando soluciones cercanas al óptimo en tiempos razonables.

3.4 Recocido Simulado (SA)

Inspirado en la metalurgia, SA acepta soluciones de peor calidad bajo cierta probabilidad decreciente. Esto permite escapar de óptimos locales y mejorar la exploración. En el QAP, se ha utilizado con estructuras de vecindario como 2- y 3-intercambio, logrando balances adecuados entre calidad de solución y tiempo de cómputo (Kirkpatrick, Gelatt, y Vecchi, 1983; Connolly, 1990; Taillard, 1998). Por ejemplo, en instancias como tai35a, SA alcanza resultados competitivos frente a métodos más sofisticados.

3.5 Optimización por Enjambre de Partículas (PSO)

PSO imita el comportamiento social de aves y peces, donde cada partícula ajusta su posición según su experiencia y la del grupo. Destaca por su simplicidad y eficiencia computacional (Kennedy y Eberhart, 1995). En problemas combinatorios como el QAP, las partículas pueden representar permutaciones, y aunque requiere adaptaciones, ha mostrado buena capacidad de convergencia cuando se combina con operadores de búsqueda local.

3.6 Algoritmos Genéticos

Basado en procesos evolutivos, GA utiliza selección, cruce y mutación para evolucionar soluciones. Su flexibilidad permite adaptarlo a problemas combinatorios representando soluciones como cromosomas en forma de permutaciones. En QAP, los operadores de cruce específicos resultan efectivos para preservar la estructura de las soluciones y acelerar la convergencia (Kennedy y Eberhart, 1995).

3.7 Luciérnagas

El Algoritmo de Luciérnagas, desarrollado por Xin-She Yang, propone que las soluciones más brillantes (mejores) atraen a otras en el espacio de búsqueda, facilitando la exploración y explotación de este (Yang, 2009; Lambora, Gupta, y Chopra, 2019), basándose en la bioluminiscencia de estos insectos (Lall et al., 2000; Viviani, 2002; Buck y Buck, 1966).

4 Jaya y Lobo Gris

Todos los algoritmos evolutivos y los basados en inteligencia de enjambre son algoritmos probabilísticos que requieren parámetros de control comunes como el tamaño de población, el número de generaciones, el tamaño de la élite, entre otros. Además de estos parámetros generales, cada algoritmo necesita también parámetros de control específicos según su naturaleza (Rao, 2016). Por ejemplo:

- El Algoritmo Genético (GA) utiliza la probabilidad de mutación, la probabilidad de cruce y un operador de selección.
- La Optimización por Enjambre de Partículas (PSO) utiliza el peso de inercia, y parámetros sociales y cognitivos.
- La Optimización por Colonia de Abejas Artificiales (ABC) requiere el número de abejas observadoras, abejas empleadas, abejas exploradoras y un límite de intentos.
- El Algoritmo de Búsqueda Armoniosa (HS) usa la tasa de consideración de memoria armónica, la tasa de ajuste de tono y el número de improvisaciones datos para el análisis. Esta información sólo es un ejemplo.

4.1 Jaya

Publicada en 2016 en (Rao, 2016), esta metaheurística se basa en la idea de que la solución óptima debe moverse hacia la mejor solución y alejarse de la peor. A diferencia de otras metaheurísticas, Jaya solo requiere parámetros de control como número de iteraciones y tamaño de la población. Fue probada en (Rao, 2016) para hallar el óptimo de 24 funciones con restricciones dando resultados favorables.

Sea f(x) la función objetivo que se desea minimizar (o maximizar), el algoritmo propuesto funciona de la siguiente forma (Rao, 2016):

En la iteración i, se tiene:

- m: número de variables de diseño (j = 1,2,...,m).
- n: número de soluciones candidatas o tamaño de población (k = 1,2,...,n).
- Se identifican las soluciones:
- Mejor (best) con valor $f(x)_{best}$.
- Peor (worst) con valor $f(x)_{worst}$.

Si $X_{j,k,i}$ es el valor de la j-ésima variable de la k-ésima solución en la iteración i, su valor actualizado Xj,k,i' se calcula como:

$$Xj,k,i' = Xj,k,i + r1,j,i(Xj,\text{best},i - |Xj,k,i|) - r2,j,i(Xj,\text{worst},i - |Xj,k,i|) \quad (2)$$

Donde:

- $r_{1,j,i}, r_{2,j,i}$ son números aleatorios en el rango [0,1].
- El primer término promueve el acercamiento a la mejor solución.
- El segundo término evita acercarse a la peor solución.
- La nueva solución $X_{j,k,i}'$ se acepta sólo si mejora el valor de la función objetivo.
- Las soluciones aceptadas al final de cada iteración se conservan y se utilizan como base para la siguiente iteración.

En la Figura 1 se muestra el diagrama de flujo del funcionamiento del algoritmo.

4.2 Lobo Gris (GW)

La metaheurística de Lobo Gris publicada por primera vez en (Mirjalili, Mirjalili, y Lewis, 2014) es un algoritmo en el cual los lobos y las presas son representados por agentes que obedecen reglas simples, las cuales se listan a continuación:

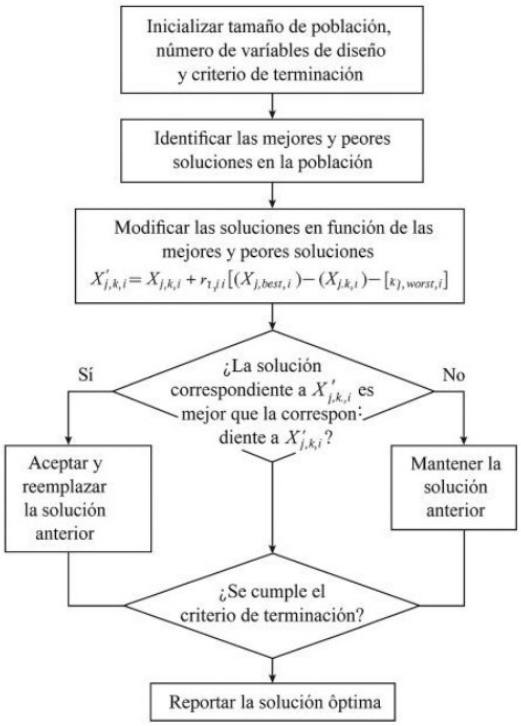

Figura 1: Flujo del algoritmo Jaya (Rao, 2016)

- La jerarquía social: Se considerará a la mejor solución como el individuo alpha, a la segunda mejor como beta, la tercera mejor como delta y al resto se les denominará omega.
- Para modelar matemáticamente el comportamiento de caza, asumimos que el alfa X_α, beta X_β y el delta X_δ poseen un conocimiento más preciso sobre la posible ubicación de la presa. Por ello obligamos a los demás agentes de búsqueda a actualizar sus posiciones en función de la ubicación de estos líderes.

En las siguientes ecuaciones de describen los principales parámetros del algoritmo

$$D_\alpha = \left| \vec{C_1} \cdot \vec{X_\alpha} - \vec{X} \right| \tag{3}$$

$$D_\beta = \left| \vec{C_2} \cdot \vec{X_\beta} - \vec{X} \right|, \tag{4}$$

$$D_\delta = \left| \vec{C_3} \cdot \vec{X_\delta} - \vec{X} \right| \tag{5}$$

$$\vec{X_2} = \vec{X_\beta} - \vec{A_2} \cdot D_\beta \tag{6}$$

$$\vec{X_2} = \vec{X_\beta} - \vec{A_2} \cdot D_\beta \tag{7}$$

$$\vec{X_3} = \vec{X_\delta} - \vec{A_3} \cdot D_\delta \tag{8}$$

$$\vec{X}(t+1) = \frac{\vec{X_1} + \vec{X_2} + \vec{X_3}}{3} \tag{9}$$

La ecuación (9) da la actualización de las nuevas soluciones en términos de alpha, beta y gamma. Donde los componentes de $\vec{a}$ decrecen linealmente de 2 a 0 durante las iteraciones, y $\vec{r_1}, \vec{r_2}$ son vectores aleatorios entre cero y uno.

El algoritmo GWO solo requiere ajustar dos parámetros principales: $\vec{a}$ *y* $\vec{C}$. En la Figura 2 se muestra un pseudocódigo de este.

```
Inicializar la población de lobos X_i siendo i = 1, ..., n aleatoriamente
Inicializar los valores de a, A y C
Calcular el fitness de cada agente de búsqueda
X_α = mejor agente
X_β = segundo mejor agente
X_δ = tercer mejor agente
while t < máximo número de iteraciones do
    for i ← 1 : n do
        Actualizar la posición de cada agente
    end for
    Actualizar los valores de a, A y C
    Calcular el fitness de todos los agentes de búsqueda
    Actualizar X_α, X_β, X_δ
    t = t+1
end while
return X_α
```

Figura 2: Pseudocódigo que implementa el algoritmo de Lobo Gris (Alonso Benítez y Rojas Delgado, 2020)

5 Implementación en instancias del QAP

En esta sección mostraremos los resultados de las metaheurísticas antes descritas para algunas instancias de tamaño mayor a 30 disponibles en QAPLIB, para dichas instancias se conoce el mejor valor obtenido hasta ahora, con lo cual calcularemos el GAP con cada una de las metaheurísticas. Para tener una comparación significativa fijaremos el número de iteraciones a 500 para todas las metaheurísticas y usaremos los valores sugeridos en la literatura para los parámetros específicos de cada una. Se desarrolló un programa en Python con una interfaz de usuario que permite:

1. Descargar en el directorio de trabajo todas las instancias que se encuentran en instancias completas con sus respectivos archivos de solución.

2. Permite al usuario elegir de entre las 9 metaheurísticas aquí revisadas, así como fijar los parámetros correspondientes de cada una.

3. Genera un archivo csv con los resultados obtenidos después de completar el número de iteraciones, el archivo csv contiene, la mejor solución, el costo asociado a esta, el gap con la mejor solución conocida y el tiempo de ejecución en CPU en segundos (12th Gen Intel(R) Core (TM) i5-12450HX (2.40 GHz), 16.0 GB RAM (15.7 GB usable), 64 bits con sistema operativo Windows 11).

5.1 Implementación

- Los códigos para FANT, TS y SA fueron tomados de Taillard, sin embargo, los códigos originales están escritos en c y c++, para convertirlos a Python se hizo uso del LLM ChatGPT en su versión gratuita utilizando la función de razonamiento con el siguiente prompt
- *Escribe el siguiente codigo de c a python manteniendo fielmente la estructura, herramientas y optimización <introducir códigos en c o c++>.*
- Para Búsqueda Tabú se tienen dos códigos etiquetados por su autor como ’viejo’ y ’nuevo’ cuyas diferencias principales son: permanencia en la lista tabú, generador de números aleatorios y tipo de indexación, lo cual tiene un impacto en el tiempo de ejecución de cada uno.
- Los códigos para PSO, GA, Firefly fueron tomados de yarpiz, sin embargo los códigos originales están escritos en matlab, para convertirlos a python se hizo uso del LLM ChatGPT en su versión gratuita utilizando la función de razonamiento con el siguiente prompt *Escribe el siguiente codigo de matlab a python manteniendo fielmente la estructura, herramientas y optimización <introducir códigos en formato matlab>.*
- Jaya y Lobo Gris fueron escritas en Python siguiendo los pseudocódigos asociados a cada artículo.

- Se construyó una interfaz básica con la librería tkinter y se utilizó PyInstaller para generar el archivo ejecutable.

6 Resultados y Conclusiones

En la Tabla 1 se muestran los mejores valores obtenidos para cada instancia seleccionada, así como el GAP del resultado hallado con el mejor conocido.

Instancia // Mejor Valor	Metaheurística	Costo de la mejor solución encontrada	Tiempo (s)	Gap (%)	Parámetros
nug30 // 6,124	fant	7,326	4.3514	19.63	{iter: 500, pop: 30}
	firefly	3,250	11.7624	46.93	{iter: 500, pop: 30}
	ga	6,760	0.2219	10.39	{iter: 500, pop: 30}
	greywolf	*4,842*	*0.2466*	*20.93*	*{iter: 500, pop: 30}*
	jaya	*8,134*	*0.3002*	*32.82*	*{iter: 500, pop: 30}*
	pso	7,368	0.2381	20.31	{iter: 500, pop: 30}
	recocido	6,724	0.0195	9.80	{iter: 500, pop: 30, temp: 1,000}
	tabu	6,236	3.1187	1.83	{iter: 500, pop: 30}
	old	**6,166**	**2.8616**	**0.69**	**{iter: 500, pop: 30}**
sko90 // 115,534	fant	131,758	21.8826	14.04	{iter: 500, pop: 30}
	firefly	80,924	50.0800	29.96	{iter: 500, pop: 30}
	ga	125,650	0.8473	8.76	{iter: 500, pop: 30}
	greywolf	*89,430*	*0.6922*	*22.59*	*{iter: 500, pop: 30}*
	jaya	*137,312*	*1.2139*	*18.85*	*{iter: 500, pop: 30}*
	pso	132,862	0.6663	15.00	{iter: 500, pop: 30}
	recocido	126,172	0.0386	9.21	{iter: 500, pop: 30, temp: 1,000}
	tabu	**117,562**	**302.4660**	**1.76**	**{iter: 500, pop: 30}**
	old	117,670	334.0270	1.85	{iter: 500, pop: 30}
sko100a // 152,002	fant	173,180	65.6169	13.93	{iter: 500, pop: 30}
	firefly	101,398	189.9642	33.29	{iter: 500, pop: 30}
	ga	166,180	1.3227	9.33	{iter: 500, pop: 30}
	greywolf	*123,682*	*0.8919*	*18.63*	*{iter: 500, pop: 30}*
	jaya	*178,306*	*1.3295*	*17.31*	*{iter: 500, pop: 30}*
	pso	172,812	1.1550	13.69	{iter: 500, pop: 30}
	recocido	164,432	0.0850	8.18	{iter: 500, pop: 30, temp: 1,000}
	tabu	155,558	369.1784	2.34	{iter: 500, pop: 30}
	old	**153,796**	**561.5398**	**1.18**	**{iter: 500, pop: 30}**
will100 // 273,038	fant	294,932	24.8545	8.02	{iter: 500, pop: 30}
	firefly	172,028	171.7351	36.99	{iter: 500, pop: 30}
	ga	286,002	1.3247	4.75	{iter: 500, pop: 30}
	greywolf	*213,128*	*1.1485*	*21.94*	*{iter: 500, pop: 30}*
	jaya	*298,152*	*1.6111*	*9.20*	*{iter: 500, pop: 30}*
	pso	295,358	0.9637	8.17	{iter: 500, pop: 30}
	recocido	285,696	0.0655	4.64	{iter: 500, pop: 30, temp: 1,000}
	tabu	275,730	555.2151	0.99	{iter: 500, pop: 30}
	old	**275,474**	**272.4876**	**0.89**	**{iter: 500, pop: 30}**

Tabla 1: Resultados por instancia y metaheurística

Todas las metaheurísticas fueron ejecutadas con 500 iteraciones y 30 elementos de la población, para recocido simulado la temperatura inicial es 1,000.

Podemos observar de la tabla que, Búsqueda Tabú tanto el nuevo como el viejo algoritmo se desempeñan de forma robusta para instancias de distintos tamaños con valores óptimos muy cercanos al mejor conocido, sin embargo, es necesario notar que al incrementar el tamaño de la instancia también incrementa el tiempo de ejecución, llegando a más de 9 minutos para la instancia sko100a, mientras que para Jaya y Lobo Gris no se observa esta variación de segundos de duración tan extrema, a pesar de que los resultados obtenidos con estos dos algoritmos pueden considerarse *no tan buenas* tomando en cuenta el porcentaje de GAP

Como trabajo futuro, exploraremos incrementar las iteraciones y tamaño de la población para Jaya y Lobo Gris para probar si es posible alcanzar una mejor solución, es necesario resaltar que, la simplicidad y pocos parámetros los hacen una opción viable en términos de prueba y pocos segundos de ejecución para resolver instancias del QAP.

Referencias

Koopmans, T., y Beckmann, M. (1957). "Assignment problems and the location of economic activities", Econometrica, vol. 25(1), pp. 53–76.

Ch, R. P., Hurtado, O. G., y Moncada, J. (2022). "Exact And Approximate Sequential Methods In Solving The Quadratic Assignment Problem", Journal of Language and Linguistic Studies, vol. 18(3), pp. 237–244.

James, T., Rego, C., y Glover, F. (2009). "Multistart Tabu Search and Diversification Strategies for the Quadratic Assignment Problem", IEEE Transactions on Systems, Man, and Cybernetics – Part A: Systems and Humans, vol. 39, pp. 579–596.

Sahni, S., y Gonzalez, T. (1976). "P-complete approximation problems", Journal of the Association for Computing Machinery, vol. 23, pp. 555–565.

Bhati, R. K., y Rasool, A. (2014). "Quadratic Assignment Problem and its Relevance to the Real World: A Survey", International Journal of Computer Applications, vol. 96(9), pp. 42–47.

Muro, C., Escobedo, R., Spector, L., y Coppinger, R. P. (2011). "Wolf-pack (Canis lupus) hunting strategies emerge from simple rules in computational simulations", Behavioural Processes, vol. 88(3), pp. 192–197.

Rao, R. V. (2016). "Jaya: A simple and new optimization algorithm for solving constrained and unconstrained optimization problems", International Journal of Industrial Engineering Computations, vol. 7(1), pp. 19–34.

Loiola, E. M., de Abreu, N. M. M., Boaventura-Netto, P. O., Hahn, P., y Querido, T. (2007). "A survey for the quadratic assignment problem", European Journal of Operational Research, vol. 176(2), pp. 657–690.

Taillard, É. D. (1991). "Robust tabu search for the quadratic assignment problem", Parallel Computing, vol. 17(4–5), pp. 443–455.

Merkle, D., y Middendorf, M. (2002). "Fast Ant Colony Optimization on Runtime Reconfigurable Processor Arrays", Genetic Programming and Evolvable Machines, vol. 3(4), pp. 345–361.

Taillard, É. D. (1998). "FANT: Fast Ant System", IDSIA Technical Report 46-98, Lugano, Switzerland.

Algorithm Afternoon. (2025). Fast Ant System. Recuperado de: https://algorithm-afternoon.org/fant Fast Ant System (FANT) (s.f.). En Ant Colony Optimization, pp. 42–54.

Connolly, D. T. (1990). "An improved annealing scheme for the QAP", European Journal of Operational Research, vol. 46(1), pp. 93–100.

Kirkpatrick, S., Gelatt, C. D. Jr., y Vecchi, M. P. (1983). "Optimization by simulated annealing", Science, vol. 220(4598), pp. 671–680.

Davis, L. (1991). Handbook of Genetic Algorithms. Van Nostrand Reinhold.

Kennedy, J., y Eberhart, R. C. (1995). "Particle Swarm Optimization", en Proceedings of the IEEE International Conference on Neural Networks, vol. 4, pp. 1942–1948.

Holland, J. H. (1975). Adaptation in Natural and Artificial Systems. University of Michigan Press.

Lambora, A., Gupta, K., y Chopra, K. (2019). "Genetic Algorithm–A Literature Review", en Proceedings of the 2019 International Conference on Machine Learning, Big Data, Cloud and Parallel Computing (COMITCon), pp. 380–384.

Lewis, S. M., y Cratsley, C. K. (2008). "Flash signal evolution, mate choice, and predation in fireflies", Annual Review of Entomology, vol. 53, pp. 293–321.

Lall, A. B., et al. (2000). "Circadian control of luciferase mRNA accumulation in the firefly", Insect Biochemistry and Molecular Biology, vol. 30(3), pp. 191–196.

Viviani, V. R. (2002). "The origin, diversity, and structure function relationships of insect luciferases", Cellular and Molecular Life Sciences, vol. 59, pp. 1833–1850.

Buck, J., y Buck, E. (1966). “Biology of synchronous flashing of fireflies”, Nature, vol. 211(5050), pp. 562.
Kim, J., et al. (2005). “Firefly-inspired synchronization for wireless sensor networks”, ACM SIGCOMM.
Yang, X. S. (2009). “Firefly algorithms for multimodal optimization”, Stochastic Algorithms: Foundations and Applications (SAGA 2009).
Shen, H., y Wang, Y. (2007). “Firefly-based multi-robot cooperation control”, IEEE International Conference on Robotics and Automation.
Yang, X. S. (2009). Nature-Inspired Metaheuristic Algorithms. Luniver Press.
Kennedy, J., y Eberhart, R. (1995). “Particle Swarm Optimization”, Proceedings of IEEE International Conference on Neural Networks, vol. 4, pp. 1942–1948.
Karaboga, D., y Basturk, B. (2007). “A powerful and efficient algorithm for numerical function optimization: Artificial Bee Colony (ABC) algorithm”, Journal of Global Optimization, vol. 39(3), pp. 459–471.
Lambora, A., Gupta, K., y Chopra, K. (2019). “Genetic Algorithm–A Literature Review”, en Proceedings of the 2019 International Conference on Machine Learning, Big Data, Cloud and Parallel Computing (COMITCon), pp. 380–384.
Rao, R. V., Savsani, V. J., y Vakharia, D. P. (2011). “Teaching–learning-based optimization: A novel method for constrained mechanical design optimization problems”, Computer-Aided Design, vol. 43(3), pp. 303–315.
Rao, R. V. (2015). Teaching-Learning-Based Optimization Algorithm: And Its Engineering Applications. Springer.
Mirjalili, S., Mirjalili, S. M., y Lewis, A. (2011). “Grey Wolf Optimizer”, Advances in Engineering Software, vol. 69, pp. 46–61.
Alonso Benítez, R., y Rojas Delgado, J. (2020). “Implementación del algoritmo meta-heurístico Gray Wolf Optimization para la optimización de funciones objetivo estándar”, Serie Científica de la Universidad de las Ciencias Informáticas, vol. 13(8), pp. 174–194.

Capítulo 12

Detección de Parkinson mediante aprendizaje automático y señales biomédicas

Denisse América Uscanga Tlalpan[1]

[1] Universidad de las Américas Puebla
denisse.uscangatn@udlap.mx

Resumen. El aprendizaje automático ha prevalecido como una herramienta para el diagnóstico de enfermedades. El presente trabajo tuvo como objetivo evaluar el rendimiento de diversos algoritmos de clasificación, como Random Forest (RF), Bagging-KNN y Decision Tree, a través de señales de voz como biomarcadores. Se emplearon técnicas de selección de características (algoritmos genéticos, Fisher's Score y PCA) y estrategias de balanceo de datos. RF con selección de características obtuvo los mejores resultados, con intervalos de confianza del 95% entre 0.84 y 0.96 para el accuracy, de 0.89 a 1.00 para el recall y de 0.90 a 0.96 para el F1-score.

Palabras Clave: Parkinson, Random Forest, Bagging, Selección de características.

1 Introducción

El diagnóstico temprano de enfermedades neurodegenerativas como el Parkinson es crucial para comenzar tratamientos que mitiguen su progresión y mejoren así la calidad de vida de los pacientes. Además, el aprendizaje automático ha demostrado ser útil para procesar vastas cantidades de datos biomédicos, revelando patrones que son difíciles de observar por medios convencionales. Estas técnicas no solo permiten obtener una alternativa para la detección temprana de enfermedades, sino que también optimizan los procesos de clasificación y predicción, obteniendo nuevas posibilidades para el desarrollo de herramientas de diagnóstico más accesibles (Guevara et al., 2024).

Es por esto que este trabajo tiene como objetivo evaluar y comparar el desempeño de distintos modelos de clasificación aplicados a una base de datos de uso libre compuesta por señales de voz de pacientes con Parkinson y sujetos control. Se utilizan métricas como *accuracy*, *recall* y *F1-score* y se utilizaron clasificadores como Árboles de Decisión, Random Forest, Random Forest combinado con técnicas de selección de características, Bagging con KNN, así como Random Forest con *oversampling* y *subsampling*. Se presenta en el artículo el contexto general del problema y los conceptos principales en la enfermedad de Parkinson, señales de voz y aprendizaje automático. La sección de procedimiento experimental muestra la organización de la base de datos

que se utiliza y las variaciones en los resultados de validación en los modelos de clasificación. Luego se ofrece una discusión de los resultados obtenidos en la sección y las implicaciones del estudio, junto con el potencial para futuras investigaciones, se presentan en la sección de conclusiones.

1.1 Parkinson

La enfermedad de Parkinson es un trastorno neurodegenerativo progresivo que se presenta principalmente con sintomatología motora (rigidez, temblor y bradicinesia) pero también con síntomas no motores (trastornos del sueño, depresión y deterioro cognitivo). Además, el desarrollo de enfoques avanzados de neuroimagen (p. ej., tomografía por emisión de positrones (PET), IRM funcional) en combinación con métodos de aprendizaje automático ha fomentado estudios relacionados con el Parkinson (Váradi, 2020).

1.2 Señales de Voz

Una señal de voz es una impresión sonora inducida por las cuerdas vocales y cavidades, estas son las señales que transportan información sobre características fisiológicas y motoras del habla, incluyendo tono, intensidad y duración del sonido. En cuanto a la detección del Parkinson, se emplea una señal de voz debido a que esta enfermedad afecta el control motor fino, como en el tracto vocal, lo que conduce a síntomas como hipofonía (bajo volumen), monotonía, temblores en la voz y cambios en la articulación del habla. Como resultado, estas variaciones pueden clasificarse a través de algoritmos de aprendizaje automático y extracción de características, es decir, patrón de jitter, shimmer, variabilidad del tono son patrones que ayudan a diferenciar a los pacientes con Parkinson de las personas sanas (Little et al., 2009).

1.3 Random Forest

Random Forest es un tipo de algoritmo de aprendizaje automático conocido como un método en conjunto que agrega múltiples árboles de decisión para hacer predicciones. Cada árbol se construye sobre una muestra aleatoria de los datos y en cada división se realizan pruebas sobre un subconjunto de las características para disminuir la correlación entre los árboles. Durante una predicción, el modelo toma en cuenta la votación mayoritaria en la clasificación y el promedio en la regresión. Así, con su conjunto, Random Forest tiene menor sobreajuste y maneja datos faltantes, así como información no lineal y mucha cantidad de datos (Raschka et al., 2022).

1.4 Bootstrap Aggregating y KNN

Bagging describe el proceso de crear múltiples modelos con diferentes subconjuntos de datos (seleccionados por muestreo aleatorio con reemplazo) y luego agregando sus

predicciones a su nivel más simple, usando el promedio en la regresión y el voto de la mayoría para clasificación multicategoría. En el caso de KNN, bagging genera múltiples subconjuntos de datos que reducen la variabilidad de los modelos y los hacen robustos al ruido en los datos. KNN es un modelo de baja sesgo pero alta varianza; al utilizar bagging, esto hace que el modelo sea robusto y menos susceptible al ruido o valores atípicos (Raschka et al., 2022).

1.5 Fisher´s Score

La selección de características sobre la base de la puntuación de Fisher es un método estadístico para detectar las características más discriminantes para la clasificación en datos multicategoría y manipular con algunos puntajes de características seleccionadas. Es la razón entre la varianza entre clases y la varianza dentro de clases. Un valor alto indica que la característica es más adecuada para discriminar diferentes categorías para la clasificación (Akil et al., 2021).

1.6 Algoritmos Genéticos

Los AG se inspiran en el proceso de evolución de la naturaleza para elegir subconjuntos de características relevantes. Para cada parte, se tiene una representación binaria, donde se crea inicialmente un conjunto de tales códigos y se prueba mediante el uso de un clasificador. Los mejores subconjuntos son elegidos para ser reproducidos a través de crossover (combinación de padres) y mutación (cambios aleatorios) (Katoch et al., 2021).

1.7 PCA

El Análisis de Componentes Principales (PCA) es una técnica para simplificar conjuntos de datos de alta dimensión. PCA convierte las características originales en un conjunto de nuevas (componentes principales) que son combinaciones lineales de las originales. Logra retener la mayor variabilidad presente en los datos, podando menos características al retener simultáneamente informativas (Jolliffe, 2002).

2 Procedimiento Experimental

2.1 Descripción de la base de datos

Se escogió una base de datos del repertorio UCI, esta fue extraída a través de su disponibilidad como biblioteca en Python, como se muestra en la **Figura 1**. Esta contiene datos de voz de 31 personas, de las cuales 23 fueron diagnosticadas con Parkinson, organizados en un total de 195 grabaciones. Cada fila corresponde a una grabación individual y cada columna representa una característica de la voz. El objetivo es clasificar a las personas como sanas cuando el estado es igual a cero, o con Parkinson cuando el estado es igual a uno (UCI Machine Learning Repository, 2008). Por lo que mediante codificación en Python se obtuvo la siguiente información con respecto a las 22 características, la cual es mostrada en la **Tabla 1**.

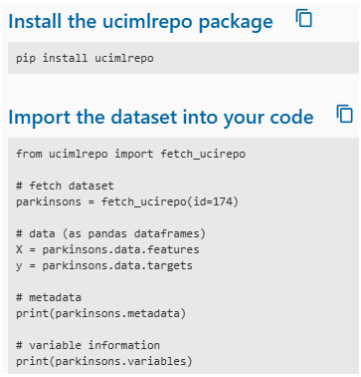

Figura 1. Biblioteca y código para la extracción de la base de datos (UCI Machine Learning Repository, 2008).

Tabla 1. Descripción de las características de la base de datos

Nombre	***Rol***	***Tipo***	***Descripción***	***Unidades***
name	Identificador	Categórico	Ninguna	Ninguna
MDVP:Fo	Característica	Continua	Ninguna	Hz
MDVP:Fhi	Característica	Continua	Ninguna	Hz
MDVP:Flo	Característica	Continua	Ninguna	Hz
MDVP:Jitter	Característica	Continua	Ninguna	%
MDVP:Jitter	Característica	Continua	Ninguna	Abs
MDVP:RAP	Característica	Continua	Ninguna	Ninguna
MDVP:PPQ	Característica	Continua	Ninguna	Ninguna
Jitter:DDP	Característica	Continua	Ninguna	Ninguna
MDVP:Shimmer	Característica	Continua	Ninguna	Ninguna

MDVP:Shimmer	Característica	Continua	Ninguna	dB
Shimmer:APQ3	Característica	Continua	Ninguna	Ninguna
Shimmer:APQ5	Característica	Continua	Ninguna	Ninguna
MDVP:APQ	Característica	Continua	Ninguna	Ninguna
Shimmer:DDA	Característica	Continua	Ninguna	Ninguna
NHR	Característica	Continua	Ninguna	Ninguna
HNR	Característica	Continua	Ninguna	Ninguna
status	Objetivo	Entero	Ninguna	Ninguna
RPDE	Característica	Continua	Ninguna	Ninguna
DFA	Característica	Continua	Ninguna	Ninguna
spread1	Característica	Continua	Ninguna	Ninguna
spread2	Característica	Continua	Ninguna	Ninguna
D2	Característica	Continua	Ninguna	Ninguna
PPE	Característica	Continua	Ninguna	Ninguna

2.2 Decision Tree y Random Forest

Se implementaron tanto el algoritmo de Decision Tree y Random Forest en la base de datos del repertorio UCI escogida. En el caso del árbol de decisión, se construyó un modelo basado en entropía, evaluando su precisión y visualizándolo gráficamente en la **Figura 2**, mientras que el Random Forest se entrenó con 100 árboles para mejorar la robustez, mostrando métricas como matriz de confusión y reporte de clasificación. Ambos dividen los datos en conjuntos de entrenamiento y prueba (70-30) y miden tiempos de construcción.

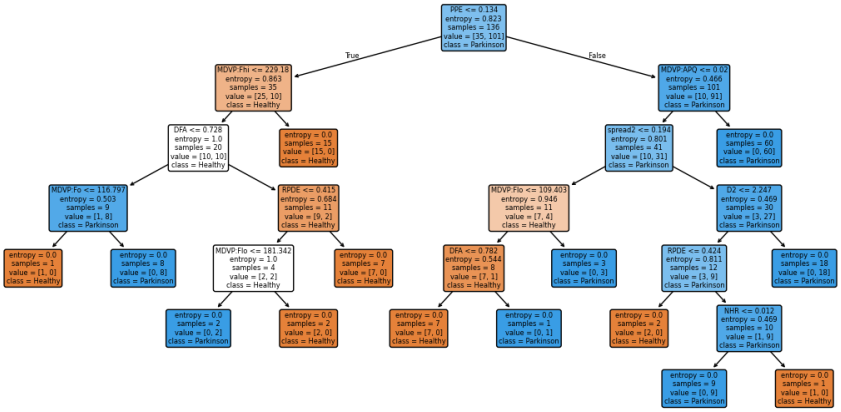

Figura 2. Diagrama del árbol de decisión generado mediante codificación en Python, utilizando la base de datos del repositorio UCI, en el cual la característica PPE se establece como nodo raíz.

2.3 Sobremuestreo y Submuestreo

Debido a que la base de datos presenta una mayor cantidad de instancias pertenecientes a la clase Parkinson que a la de control, se buscó comprobar el rendimiento de clasificación de Random Forest balanceando las instancias tanto de la clase minoritaria (subsampling) como de la clase mayoritaria (oversampling). El primer código realiza un submuestreo de la clase mayoritaria (No Parkinson) para balancear las clases, mientras que el segundo aplica sobremuestreo a la clase minoritaria (Parkinson), utilizando la función ***resample*** de sklearn. Ambos códigos dividen los datos en un 70% para entrenamiento y 30%, luego entrenan el modelo y evalúan su rendimiento con métricas como precisión, reporte de clasificación, matriz de confusión y el tiempo de entrenamiento. La distribución de las clases puede verse en la **Tabla 2**.

Tabla 2. Distribución de Clases con Métodos de Balanceo

Clase	Distribución sin balancear	Sobremuestreo	Submuestreo
1	147	147	48
0	48	147	48

2.4 Métricas de Selección de Características

Así mismo, se expuso la base de datos del repertorio UCI a métricas de selección de atributos como PCA, Fisher´s Score y Algoritmos Genéticos, utilizando el algoritmo de Clasificación Random Forest para lograr determinar el rendimiento del algoritmo bajo una reducción de atributos, generando la clasificación solo con las 12 características más importantes de las 22 que conforman la base de datos original, lo cual es mostrado en la **Tabla 3**.

Tabla 3. Fisher Scores de las Características Seleccionadas

Características Seleccionadas	Fisher Score
MDVP:Fo	0.2997
MDVP:Flo	0.2074
MDVP:Jitter	0.2282
MDVP:RAP	0.2270
Jitter:DDP	0.2269
Shimmer:APQ3	0.4009
MDVP:APQ	0.4746

Shimmer:DDA	0.4009
DFA	0.1259
Spread1	1.2860
D2	0.3359
PPE	1.1703

2.5 Bootstrap Aggregating y KNN

Se implementó un modelo de clasificación utilizando un Bagging Classifier con K-Nearest Neighbors (KNN) como clasificador base. Aunque KNN es un algoritmo no paramétrico que no realiza una fase de entrenamiento tradicional, en el contexto de Bagging se crean múltiples instancias del algoritmo, cada una aplicada sobre un subconjunto del conjunto de datos generado mediante muestreo con reemplazo. En este caso, se utilizaron un total de 50 estimadores base. La elección de KNN se justificó por la baja dimensionalidad de los datos y porque todas las variables son numéricas continuas, como se muestra en la **Tabla 1**. Para evaluar el rendimiento del modelo, se utilizaron métricas como el reporte de clasificación, la precisión y la matriz de confusión. Finalmente, se registró el tiempo de ejecución del modelo.

2.6 Validación del Modelo

Asimismo, una vez identificado y seleccionado el mejor modelo a partir de los resultados obtenidos en la matriz de confusión, se implementó validación cruzada con 5 particiones (5-fold cross-validation) así como la validación Bootstrap con 1000 iteraciones, con el objetivo de evaluar la estabilidad y el rendimiento general del modelo en diferentes subconjuntos de datos.

Análisis de Resultados

Como es mostrado en la **Tabla 4**, el clasificador Random Forest (RF) alcanzó una precisión de 94.92% con un tiempo de construcción de 0.1567 segundos. Al aplicar selección de características (algoritmos genéticos, Fisher's Score y PCA), la precisión aumentó a 98.31%, aunque el tiempo subió a 63.30 segundos. RF Subsampling y RF Oversampling lograron 89.66% y 93.26% de precisión, con tiempos de 0.1066 segundos y 0.1179 segundos, respectivamente. El Decision Tree obtuvo 89.83% con el menor tiempo (0.0071 segundos). Finalmente, el modelo de Bagging - KNN tiene una precisión de 91.53% y tiempos de construcción rápidos, pero no alcanza el rendimiento de los mejores clasificadores como Random Forest y RF con selección de características. Así mismo en la **Figura 3** se muestran las matrices de confusión obtenidas para cada uno de los clasificadores evaluados.

Tabla 4. Precisión y Tiempo de Construcción de Modelos para Diferentes Clasificadores

Clasificador	Precisión	Tiempo construcción
Decision Tree	0.8983	0.0071 s
Random Forest	0.9492	0.1567 s
RF + Subsampling	0.8966	0.1066 s
RF + Oversampling	0.9326	0.1179 s
RF + Feature Selection	0.9831	63.30 s
Bagging + KNN	0.9153	0.0461 s

Por otra parte, en la **Tabla 5** se exhibe que el modelo Random Forest (RF) mostró un excelente desempeño general con un Accuracy de 0.95, un Recall de 0.91 y un F1-Score de 0.92, logrando un balance ideal entre precisión y sensibilidad. RF Feature Selection sobresale como el mejor clasificador con un Accuracy de 0.98, un Recall de 0.99 y un F1-Score de 0.98, lo que demuestra que la eliminación de características irrelevantes mejora significativamente la precisión. RF Oversampling también mostró un buen rendimiento con un Accuracy, Recall y F1-Score de 0.93, siendo útil en escenarios con datos desbalanceados. Por otro lado, RF Subsampling obtuvo métricas aceptables (Accuracy, Recall y F1-Score de 0.90), aunque inferiores a RF estándar, esto se debe a que RF Subsampling elimina datos de la clase mayoritaria, lo que reduce la información disponible y afecta su rendimiento. Decision Tree tuvo una precisión decente (Accuracy 0.90, Recall 0.93, F1-Score 0.87), destacando en recall, pero con más falsos positivos. Finalmente, Bagging - KNN alcanzó un Accuracy de 0.92, pero su menor Recall de 0.84 afectó el balance, haciéndolo menos confiable para minimizar falsos negativos. En general, RF Feature Selection lidera en desempeño, seguido por RF estándar y Oversampling.

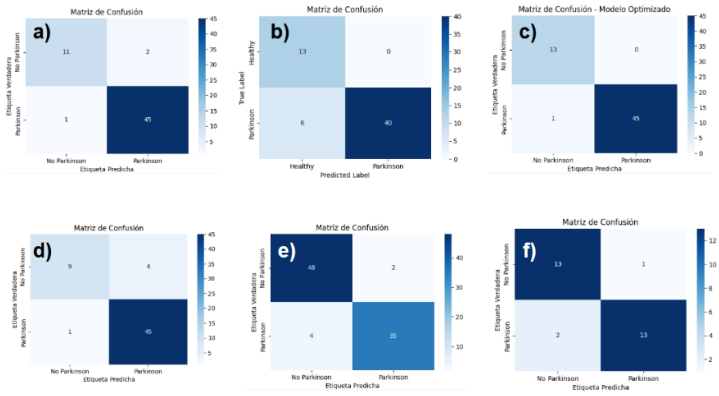

Figura 3. Matrices de Confusión obtenidas por cada clasificador: a) Random Forest, b) Decision Tree, c) RF Feature Selection, d) Bagging - KNN, e) RF Oversampling, f) RF Subsampling

Tabla 5. Métricas de clasificación para diferentes modelos

Clasificador	Accuracy	Recall	F1-Score
Random Forest	0.95	0.91	0.92
Decision Tree	0.90	0.93	0.87
RF + Feature S.	0.98	0.99	0.98
Bagging + KNN	0.92	0.84	0.86
RF + Oversampling	0.93	0.93	0.93
RF + Subsampling	0.90	0.90	0.90

Por otra parte, como fue establecido en los resultados de la **Tabla 5**, el modelo con mejor rendimiento fue el de Random Forest con feature selection (RF Feature Selection), por lo que, para comprobar la robustez del modelo, se implementó la validación cruzada con 5 folds, cuyos resultados se presentan en la **Tabla 6**. Por medio de los promedios obtenidos por cada métrica es factible determinar que el modelo es estable y consistente en diferentes particiones de los datos. Cabe destacar que el alto valor de Recall indica que el modelo logra detectar de manera efectiva a la mayoría de los casos positivos (Parkinson), esto es esperable ya que la base de datos contiene mayor cantidad de pacientes con Parkinson que pacientes control.

Tabla 6. Resultados en métricas de clasificación con validación cruzada

Fold	Accuracy	Recall	F1-Score
1	0.92	0.93	0.95
2	0.95	1	0.97
3	0.85	1	0.91

4	0.92	0.84	0.86
5	0.90	0.90	0.93
Promedio	**0.8974 (±0.0363)**	**0.9584 (±0.0402)**	**0.9334 (±0.0228)**

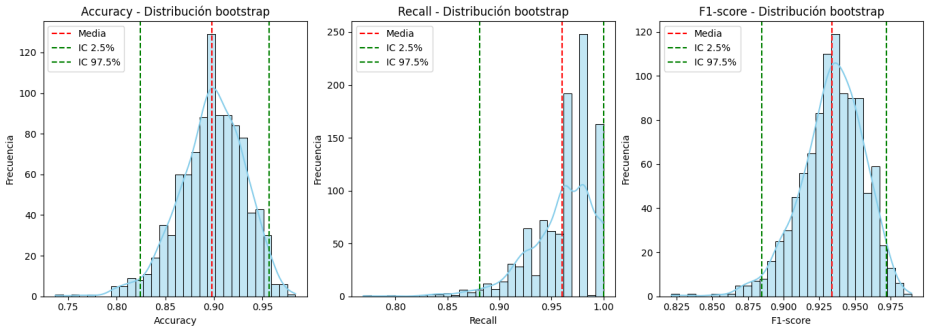

Figura 4. Distribución de métricas de rendimiento estimadas por bootstrap (B=1000) en Random Forest con Feature Selection.

Finalmente, en la Figura 4 se presenta la distribución de las métricas de rendimiento del modelo Random Forest con selección de características, obtenidas mediante un procedimiento de bootstrap con 1000 repeticiones. Los resultados muestran que el modelo alcanza intervalos de confianza del 95% entre 0.84 y 0.96 para el accuracy, de 0.89 a 1.00 para el recall y de 0.90 a 0.96 para el F1-score, lo que refleja un desempeño estable y consistente.

Conclusiones

Los resultados obtenidos muestran que factores como el tipo de algoritmo de clasificación, el balance de clases y la selección de atributos tienen un impacto significativo en el rendimiento de clasificación. Modelos como Random Forest alcanzaron una precisión de hasta 98.31 % tras incorporar métricas de selección de características, mientras que técnicas como Bagging con KNN generaron un rendimiento adecuado y funcionalmente alterno. Además, se observó que, en escenarios con clases desbalanceadas, enfoques como Random Forest con oversampling mejoran considerablemente los resultados, al corregir el sesgo hacia la clase mayoritaria. Este estudio demuestra que el uso de señales de voz como biomarcador ofrece una alternativa no invasiva, accesible y eficiente para apoyar el diagnóstico temprano del

Parkinson, en comparación a otros equipos funcionales en adquisición de señales, pero más costosos (como fNIRS o EEG).

De igual forma, los resultados de la validación cruzada con 5 folds y validación Bootstrap confirman la robustez del modelo de Random Forest con selección de características, donde se observa que el modelo mantiene valores de precisión, recall y F1-score altos y estables, lo que significa que no depende de una sola partición de datos, sino que su desempeño es estable en diferentes escenarios de muestreo. Es importante destacar los resultados en el recall se concentra en valores muy altos, lo cual es esperable debido a que el dataset utilizado esta desbalanceado con una mayor población de pacientes Parkinson que control.

Entre las principales limitaciones del estudio se encuentran el tamaño reducido de la muestra (31 personas, con mayor cantidad de pacientes con Parkinson), el desbalance entre clases y el riesgo de sobreajuste, ya que el dataset cuenta con una mayor cantidad de grabaciones por individuo. Como trabajo futuro, se propone ampliar la base de datos con más sujetos y mayor diversidad con respecto a lenguaje, edad, acentos, realizar validaciones externas en otras poblaciones y evaluar la robustez en entornos de grabación no controlados. Además, sería útil correlacionar las características seleccionadas con métricas clínicas ya validadas, como el puntaje MoCA, para confirmar su relevancia diagnóstica.

Referencias

Guevara, E., Solana-Lavalle, G., y Rosas-Romero, R. (2024). “Integrating fNIRS and machine learning: Shedding light on Parkinson's disease detection”, *EXCLI Journal - Experimental and Clinical Sciences*, vol. 23, pp. 763–771.

UCI Machine Learning Repository. (2008). *Parkinson’s Dataset*. Recuperado de https://archive.ics.uci.edu/dataset/174/parkinsons

Váradi, C. (2020). “Clinical features of Parkinson’s disease: The evolution of critical symptoms”, *Biology (Basel)*, vol. 9, pp. 103.

Little, M. A., McSharry, P. E., Hunter, E. J., Spielman, J., y Ramig, L. O. (2009). “Suitability of dysphonia measurements for telemonitoring of Parkinson's disease”, *IEEE Transactions on Biomedical Engineering*, vol. 56, pp. 1015–1022.

Raschka, S., Liu, Y. (H.), y Mirjalili, V. (2022). *Machine Learning con PyTorch y Scikit-Learn*. Marcombo.

Akil, S., Sekkate, S., y Abdellah, A. (2021). “Feature selection based on machine learning for credit scoring: An evaluation of filter and embedded methods”, *IEEE*, vol. 2021.

Katoch, S., Chauhan, S. S., y Kumar, V. (2021). “A review on genetic algorithm: Past, present, and future”, *Multimedia Tools and Applications*, vol. 80, pp. 8091–8126.

Jolliffe, I. T. (2002). *Principal component analysis* (2ª ed.). Springer Series in Statistics.

Índice de autores

Compiladores

Mireya Tovar Vidal
María Teresa Torrijos Muñoz
Carlos Leopoldo Carreón Díaz de León
Daniel Marcelo González Arriaga
Jaime Alejandro Romero Sierra
Juan Carlos Conde Ramírez

Revisores

Ana Laura Lezama Sánchez
Carlos Leopoldo Carreón Díaz de León
Cecilia Reyes Peña
Cesar Bautista Ramos
Daniel Marcelo González Arriaga
Erick Barrios González
Esiquio Martin Gutiérrez Armenta
Freddy Palma Mancilla
Gabriela Alejandra García Robledo
Gerardo Arizmendi Echegaray
Héctor Saib Marabillo Gómez
Jaime Alejandro Romero Siera
Jose David Alanis Urquieta
José Alejandro Reyes Ortiz
José de Jesús Lavalle Martínez
Josué Padilla Cuevas
Juan Carlos Conde Ramírez
Julio Hernandez Torres
Karina Rosales López
María Beatriz Bernábe Loranca
María Auxilio Medina Nieto
María Teresa Torrijos Muñoz
Mario Rossainz López
Martín Guerrero Posadas
Meliza Contreras González
Mireya Tovar Vidal
Samantha Acosta Ruiz

Editores

Mireya Tovar Vidal
María Teresa Torrijos Muñoz
Carlos Leopoldo Carreón Díaz de León
Daniel Marcelo González Arriaga
Jaime Alejandro Romero Sierra
Juan Carlos Conde Ramírez

Estrategias Modernas Basadas en Procesamiento de Lenguaje Natural, Aprendizaje Automático y Representación del Conocimiento en la Era de la Información

Editores de la publicación:
Mireya Tovar Vidal
María Teresa Torrijos Muñoz
Carlos Leopoldo Carreón Díaz de León
Daniel Marcelo González Arriaga
Jaime Alejandro Romero Sierra
Juan Carlos Conde Ramírez
A partir de diciembre de 2025
está disposición en PDF en la página
de la Facultad de Ciencias de la Computación
de la Benemérita Universidad Autónoma de Puebla (BUAP)
https://ontologica.cs.buap.mx/icokg2025/libroICOKG2025.pdf
Peso del archivo: 9.50 MB