Instituto Tecnológico de Orizaba
División de Estudio de Posgrado e Investigación
Maestría en Sistemas Computacionales

TESIS

Desarrollo de un módulo de software para la detección temprana de la enfermedad de Parkinson mediante técnicas de Deep Learning

PRESENTADO POR:

I.T.I. Jaime Hernández Ramírez M21011175

PARA OBTENER EL GRADO DE

Maestro en Sistemas Computacionales

DIRECTOR DE TESIS

Dr. Giner Alor Hernández

CODIRECTOR DE TESIS

MSC. Nancy Aracely Cruz Ramos

Mayo 2023

Contenido

CAPÍTULO 1. ANTECEDENTES....11
1.1 Marco teórico....11
1.1.1 Inteligencia artificial....11
1.1.1.1 Aprendizaje profundo....12
1.1.1.2 Técnicas de aprendizaje profundo....12
1.1.1.3 Algoritmos de Aprendizaje Profundo....13
1.1.1.4 Redes Neuronales Artificiales....14
1.1.1.5 Redes Neuronales Convolucionales....15
1.1.2 Enfermedades neurodegenerativas....15
1.1.2.1 Enfermedad de Parkinson....16
1.1.3 Software....16
1.1.4 Lenguajes de programación....17
1.1.4.1 Python....17
1.1.5 Bibliotecas de Aprendizaje Profundo....18
1.1.5.1 Scikit-Learn....18
1.2 Situación tecnológica, económica y operativa de la empresa....18
1.3 Planteamiento del problema....19
1.4 Objetivo general y objetivos específicos....20
1.4.1 Objetivo general....20
1.4.2 Objetivos específicos....20
1.5 Justificación....20
CAPÍTULO 2. ESTADO DE LA PRÁCTICA....22
2.1 Trabajos relacionados....22
2.2 Análisis comparativo....31
2.3 Solución propuesta....36
2.3.1 Tecnologías del entorno de desarrollo y despliegue....36
2.3.1.1 Visual Studio Code....36
2.3.1.2 Docker....37
2.3.2 Tecnologías de la capa de presentación e integración....37
2.3.2.1 TypeScript....37
2.3.2.2 Angular....37
2.3.2.3 IONIC....38
2.3.2.4 PHP....38
2.3.3 Tecnologías de la capa de servicios....38

2.3.3.1 Python 39
2.3.3.2 Flask 39
2.3.3.3 Scikit-Learn 39
2.3.4 Tecnologías de la capa de persistencia 39
2.3.4.1 MySQL 40
2.3.5 Justificación de la solución 40
CAPÍTULO 3. APLICACIÓN DE LA METODOLOGÍA 42
3.1 Análisis 42
3.1.1 Requerimientos 43
3.1.2 Casos de uso 44
3.1.3 Características de la aplicación 50
3.1.4 Conjuntos de datos de ejercicios de trazabilidad para la detección de la Enfermedad de Parkinson 51
3.1.5 Selección de Algoritmos de Clasificación 52
3.2 Diseño 55
3.2.1 Arquitectura del módulo 55
3.2.2 Modelo de persistencia 57
3.2.2.1 Diccionario de datos 58
3.2.3 Maquetación del módulo 63
3.2.3.1 Componentes personalizados 63
3.2.3.2 Inicio de sesión 66
3.2.3.3 Panel estadístico 67
3.2.3.4 Visualizar ejercicio 69
3.2.3.5 Panel del perfil 70
3.2.3.6 Catálogo de pruebas 71
3.2.3.7 Aplicación de la prueba 71
3.2.3.8 Resultados de la prueba 72
3.3 Desarrollo 73
3.3.1 Implementación de los algoritmos de clasificación 73
3.3.2 Implementación del módulo 75
CAPÍTULO 4. RESULTADOS 82
4.1 Casos de estudio: Identificación temprana de síntomas de la Enfermedad de Parkinson 82
4.1.1 Caso de estudio 1: Paciente con diagnostico positivo 82
4.1.2 Caso de estudio 2: Paciente con diagnostico negativo 82
4.1.3 Caso de estudio 3: Paciente con diagnostico negativo 83
4.1.4 Datos iniciales 83

4.1.5 Clasificación automática de datos 84
4.1.6 Resultados 85
CAPÍTULO 5. CONCLUSIONES 89
5.1 Recomendaciones 89
PRODUCTOS ACADÉMICOS 90
DERECHOS DE AUTOR 91
BIBLIOGRAFÍA 92

Índice de figuras

Figura 3.1 Metodología de desarrollo....42
Figura 3.2 Diagrama de casos de uso del contexto Ejercicios....44
Figura 3.3 Diagrama de casos de uso del contexto General....45
Figura 3.4 Diagrama de casos de uso del contexto Pacientes....46
Figura 3.5 Diagrama de casos de uso del contexto Pruebas....47
Figura 3.6 Diagrama de casos de uso del contexto Usuarios....49
Figura 3.7 Arquitectura del módulo web para la detección temprana de la EP....56
Figura 3.8 Esquema relacional de base de datos....58
Figura 3.9 Mockup - Componente - Encabezado....63
Figura 3.10 Mockup - Componente - Menú lateral....64
Figura 3.11 Mockup - Componente - Tarjeta de concepto....64
Figura 3.12 Mockup - Componente - Tarjeta estadística....64
Figura 3.13 Mockup - Componente - Tarjeta de gráfico....65
Figura 3.14 Mockup - Componente - Tarjeta de gráfico compacto....65
Figura 3.15 Mockup - Componente - Tarjeta de datos....66
Figura 3.16 Mockup - Pantalla de inicio de sesión....66
Figura 3.17 Mockup - Pantalla de panel estadístico (Principal)....68
Figura 3.18 Mockup - Pantalla de panel estadístico (Historial del paciente)....68
Figura 3.19 Mockup - Pantalla de panel estadístico (Detalles de la prueba)....69
Figura 3.20 Mockup - Pantalla visualizar ejercicio....69
Figura 3.21 Mockup - Pantalla panel del perfil....70
Figura 3.22 Mockup - Pantalla catálogo de pruebas....71
Figura 3.23 Mockup - Pantalla de aplicación de la prueba....72
Figura 3.24 Mockup - Pantalla de resultados....72
Figura 3.25 Flujo operacional para la clasificación de ejercicios....75
Figura 3.26 Pantalla de inicio de sesión....76
Figura 3.27 Pantalla de panel principal....76
Figura 3.28 Pantalla de historial del paciente....77
Figura 3.29 Pantalla de detalles de la prueba....78
Figura 3.30 Pantalla de detalles del ejercicio....78
Figura 3.31 Pantalla de catálogo de pruebas....79
Figura 3.32 Pantalla de aplicación de la prueba....80
Figura 3.33 Pantalla de conclusión de la prueba....80
Figura 3.34 Pantalla de panel del perfil....81

Figura 4.1 Muestras de trazabilidad del caso de estudio 1 - Positivo 83
Figura 4.2 Muestras de trazabilidad del caso de estudio 2 – Negativo.....84
Figura 4.3 Muestras de trazabilidad del caso de estudio 3 - Negativo.....84
Figura 4.4 Módulo web - Panel del expediente del paciente.....85
Figura 4.5 Módulo web - Catálogo de pruebas 86
Figura 4.6 Módulo web - Aplicación de la prueba – Lienzo.....87
Figura 4.7 Módulo web - Aplicación de la prueba – Realizado 87
Figura 4.8 Módulo web - Resultados de la prueba.....88

Índice de tablas

Tabla 2.1. Análisis comparativo, estado de la práctica 32
Tabla 3.1 Conjuntos de datos de ejercicios de trazabilidad para la clasificación de la enfermedad de Parkinson 52
Tabla 3.2 Resultados comparativos de los algoritmos de clasificación 52
Tabla 3.3 Resultados de la curva ROC.....53
Tabla 3.4 Diccionario de entidad "PARKINSON.TIPOS.EJERCICIOS".....59
Tabla 3.5 Diccionario de entidad "PARKINSON.TIPOS.PRUEBAS" 59
Tabla 3.6 Diccionario de entidad "PARKINSON.CAT.EJERCICIOS" 60
Tabla 3.7 Diccionario de entidad "PARKINSON.PROC.PRUEBAS" 60
Tabla 3.8 Diccionario de entidad "PARKINSON.PROC.DETALLES_PRUEBAS" 61
Tabla 3.9 Diccionario de entidad "USUARIOS.REL.PACIENTES_CUIDADORES" 62
Tabla 3.10 Diccionario de entidad "USUARIOS.REL.PACIENTES_MEDICOS" 62
Tabla 4.1 Valores de referencia para la identificación de Parkinson 85

Índice de listados

Listado 3.1 Código de implementación para el proceso de entrenamiento el modelo de clasificación Random Forest.....73
Listado 3.2 Código de implementación para endpoint POST /predict.....74

Resumen

La enfermedad de Parkinson se categoriza como una enfermedad neurodegenerativa que se manifiesta esencialmente en adultos de edad mayor, y difícilmente logra detectarse en personas menores a los 50 años, por ello una de sus características más graves, es que, no cuenta con una cura desarrollada y requiere un tratamiento continuo mediante la aplicación de medicamentos especializados que alivien los malestares presentados, por lo tanto, a través de diversos trabajos relacionados, investigadores identifican patrones y atributos clave que contribuyen en la detección y seguimiento de la enfermedad, sin embargo, se carece de herramientas tecnológicas que impulsen este proceso.

Acorde con lo anterior, el presente trabajo de tesis propone el desarrollo de un módulo que contribuya eficazmente en los procesos de análisis y detección temprana de la enfermedad de Parkinson mediante técnicas de Aprendizaje Profundo que estudien algunos de los comportamientos presentados por este padecimiento.

Por consiguiente, los beneficios representan contar con una serie de actividades de trazabilidad digital que permiten al paciente registrar el comportamiento y dificultad resultante en el proceso, de esta manera, obtener información de relevancia y proveer un seguimiento de análisis frente a conjuntos de datos que determinarán de forma temprana un posible caso positivo del padecimiento, y así, permitir al personal médico proveer atención de manera eficaz y mejorando la calidad de vida del paciente.

Abstract

Parkinson's disease is categorized as a neurodegenerative disease that essentially manifests in older adults and is rarely detected in people under the age of 50. One of its most serious characteristics is that it does not have a developed cure and requires continuous treatment through the application of specialized medications that alleviate the symptoms presented. Therefore, through various related works, researchers identify key patterns and attributes that contribute to the detection and monitoring of the disease. However, there is a lack of technological tools that drive this process.

In accordance with the above, the present thesis proposes the development of a module that contributes effectively to the processes of analysis and early detection of Parkinson's disease through Deep Learning techniques that study some of the behaviors presented by this condition.

Therefore, the benefits represent having a series of digital traceability activities that allow the patient to record the behavior and resulting difficulty in the process, in this way, obtaining relevant information and providing a follow-up analysis against data sets that will determine on a possible positive case of the disease early on. This allows medical personnel to provide effective care, improving the patient's quality of life.

Introducción

Se reconoce la enfermedad de Parkinson como el segundo padecimiento neurodegenerativo más frecuente y de constante crecimiento por adultos mayores de la población, de acuerdo con la Organización Mundial de la Salud, la enfermedad de Parkinson, se cataloga como una enfermedad que afecta directamente al sistema nervioso del individuo que lo padece, uno de los puntos característicos graves de esta enfermedad, es que no se cuenta actualmente con una cura y se abstiene a generar tratamientos que únicamente reduzcan el malestar presentado durante un periodo prolongado.

Por otro lado, a pesar de las investigaciones clínicas exitosas que dedican esfuerzo en el análisis y recolección de información clínica, la falta de herramientas tecnológicas impulsa la necesidad de desarrollar procesos de recolección, análisis y clasificación de datos en apoyo al personal médico y pacientes.

Conforme a lo anterior, el presente trabajo de tesis propone el desarrollo de un módulo de software para la detección temprana de la enfermedad de Parkinson mediante técnicas de Aprendizaje Profundo, el cual, busca recolectar y analizar algunos de los comportamientos presentados por este padecimiento a través de una aplicación móvil.

Este trabajo conforma cinco capítulos fundamentales: en el capítulo uno, Antecedentes, se aborda el marco teórico, planteamiento del problema, objetivos y justificación del proyecto; en el capítulo dos, Estado de la práctica se presentan los trabajos relacionados en la investigación y su análisis comparativo, así como también, la propuesta de solución; en el capítulo tres, Aplicación de la metodología, se concentra el desarrollo del proyecto en sus tres fases, análisis, diseño y desarrollo; en el capítulo cuatro, Resultados, se demuestran los casos de estudio implementados para la las pruebas de funcionamiento y desempeño del módulo, finalmente, en el capítulo cinco, se expresan las conclusiones, recomendaciones y trabajos a futuro identificados durante el desarrollo del proyecto.

Capítulo 1. Antecedentes

Este capítulo abarca una variedad de conceptos de gran relevancia. Además, se aborda la problemática a solucionar, el objetivo general, los objetivos específicos y la justificación del presente trabajo.

1.1 Marco teórico

A continuación, se describen términos relacionados con el tema de investigación.

1.1.1 Inteligencia artificial

La Inteligencia Artificial (IA) se conoce como la rama de la ciencia encargada del desarrollo del pensamiento artificial para dar solución a problemas complejos desde el punto de vista de la ingeniería, visto de otro modo, el concepto de IA se define como la habilidad para aprender de forma efectiva, reaccionar de forma adaptativa, realizar decisiones apropiadas, comunicar y entender ideas a través del lenguaje de forma sofisticada. Y aunque generalmente la IA se asocia en el área de la informática, tiene relación con muchos otros campos de estudio, tales como las matemáticas, la psicología, la cognición y la biología, por mencionar algunos [1]–[3].

El término de Inteligencia Artificial tiene origen en el año de 1956 por John McCarthy del Instituto Tecnológico de Massachusetts, quien estableció las bases de la IA como un campo independiente dentro de la informática [4], posteriormente en la actualidad, su éxito, es causa de que las tecnologías de IA se difunden activamente en la vida cotidiana, a través de tecnologías tales como redes neuronales, controles difusos, árboles de decisión y sistemas basados en reglas que permiten una óptima interacción dinámica entre usuarios y dispositivos inteligentes.

Desde la perspectiva del conocimiento, la IA asume la existencia de representaciones mentales que sirven para modelar el conocimiento a través del procesamiento secuencial de representaciones simbólicas, teniendo como objetivo, desarrollar métodos y sistemas para resolución de problemas complejos donde la actividad intelectual humana es el principal factor de resolución, como, por ejemplo, sistemas de

planeación y predicción, el procesamiento del lenguaje y del habla y procesos de reconocimiento visual. Así mismo, un segundo objetivo prioritario de la IA, es desarrollar modelos que simulen a los organismos vivos, en esencia, al cerebro humano, mejorando con ello el entendimiento sobre su proceso de pensamiento [1], [2], [5].

1.1.1.1 Aprendizaje profundo

En el estudio realizado por Goodfellow et al. [6], describen al Aprendizaje Profundo (AP) o Deep Learning como un enfoque de la Inteligencia Artificial (IA) que logra un alto nivel de potencia y flexibilidad al aprender representaciones del mundo a través de la extracción y concatenación jerárquica de conceptos más simples. Por otro lado, en [7], se presenta el AP como un conjunto de técnicas y procedimientos algorítmicos basados en el Aprendizaje Automático (*Machine Learning*) que permiten que la IA aprenda de manera similar a un ser humano. Sin embargo, desde otro punto de vista, Kim [8] resume la definición del AP como el uso de redes neuronales profundas, considerando que esta técnica es el enfoque más preciso del AP.

Si bien se considera el AP como una técnica de resolución de problemas que provee gran soporte a sistemas de aprendizaje complejos, la interpretación de este concepto se precisa mejor como una familia de algoritmos cuyo propósito es simular el proceso del pensamiento humano para la comprensión de conjuntos de información determinados, esto se logra mediante diversas técnicas de aprendizaje que analizan conjuntos de datos previos como modelos de entrenamiento.

1.1.1.2 Técnicas de aprendizaje profundo

Como se menciona anteriormente, el AP abarca amplio un conjunto de técnicas y arquitecturas que se caracterizan por utilizar múltiples capas de procesamiento de información no lineal de manera jerárquica. Dependiendo del propósito y uso específico, como síntesis y generación o reconocimiento y clasificación, se pueden categorizar tres clases principales [9]:

- **Redes profundas para el aprendizaje no supervisado o generativo**
  El propósito principal de esta categoría radica en el análisis de patrones, la síntesis de datos en observación, o bien, la agrupación de información, sin necesidad de etiquetar los grupos con base en patrones u objetivos.
- **Redes profundas para el aprendizaje supervisado**
  Mediante estos modelos se analizan patrones ya conocidos y categorizados de información, con el fin de realizar una clasificación directa y de mayor precisión.
- **Redes profundas híbridas**
  Este grupo de técnicas provee una solución combinada de las dos categorías previas, con el fin de tener la capacidad de discriminar información menos relevante mediante el aprendizaje no supervisado y contener un grupo de datos de mayor precisión para finalmente ejecutar un proceso supervisado.

1.1.1.3 Algoritmos de Aprendizaje Profundo

También conocidos como algoritmos de AP. Estos algoritmos han revolucionado el campo de la inteligencia artificial al permitir el procesamiento de datos complejos y la extracción de características de manera automática [10].

Se comprende que, el aprendizaje profundo se basa en redes neuronales artificiales profundas, que son estructuras compuestas por múltiples capas de neuronas interconectadas. Los algoritmos definen la forma en que estas capas permiten que el modelo aprenda representaciones jerárquicas de los datos, lo que a su vez posibilita la comprensión de patrones y la toma de decisiones sofisticadas [10].

Una de las ventajas clave de los algoritmos de AP es su capacidad para manejar datos no estructurados y de alta dimensionalidad, como imágenes, texto y audio. Llevando a avances significativos en áreas como el reconocimiento de imágenes, el procesamiento del lenguaje natural y la traducción automática [10].

Entre los algoritmos de AP más utilizados se encuentran las redes neuronales artificales (RNA) y redes neuronales convolucionales (CNN), las cuales, destacan por

rendimiento en tareas de visión por computadora, como la clasificación de imágenes y la detección de objetos. [10].

1.1.1.4 Redes Neuronales Artificiales

Las Redes Neuronales Artificiales (RNA) se definen como un sistema de mapeo no lineal, cuya estructura se basa en principios observados en los sistemas nerviosos de humanos y animales, donde se establece un grupo de procesadores ligados por conexiones, donde cada procesador cuenta con un conjunto de unidades de procesamiento denominadas neuronas [11].

Con esto, se infiere que la idea principal de una RNA, es simular el comportamiento de una red neuronal biológica, de manera que cada neurona se emula mediante algoritmos matemáticos, permitiendo un punto de entrada que recibe la llegada de señales con distintos registros de información, para posteriormente, emitir una señal de salida, dicha señal de salida dependerá siempre de una función de evaluación, como se muestra en la figura 1.1.

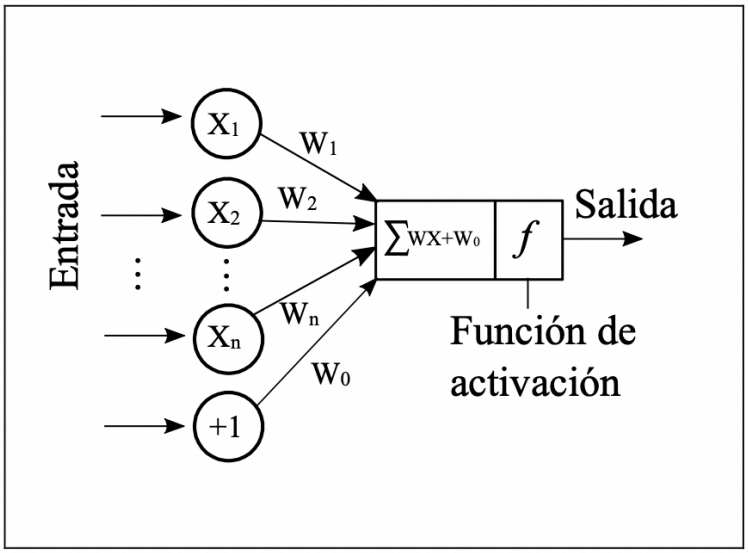


Figura 1.1 Modelo de Red Neuronal Artificial

Una de las ventajas destacables de las RNA frente a otros sistemas de procesamiento son un proceso que provee robustez, por lo que aun cuando se presenten incidencias en algunos elementos, el procesamiento de la información continuará trabajando para

ofrecer los resultados en el resto del conjunto de datos, así mismo se presentan como una solución directa para la resolución de problemas no lineales.

Por otro lado, se enfatiza que una RNA requiere entrenarse previamente para cada problema, dicho entrenamiento en ocasiones es largo y requiere de un amplio consumo de recursos técnicos. Este entrenamiento se realiza mediante dos posibles metodologías, *Perceptrón* o Redes de Propagación hacia atrás.

1.1.1.5 Redes Neuronales Convolucionales

Las Redes Neuronales Convolucionales (RNC) son una variante del modelo de perceptrón multicapa. Su distinción radica en la capacidad de llevar a cabo operaciones de convolución tanto en los parámetros de la red como en los datos de entrada.

Las RNC son de mayor utilidad en el paradigma de AP, cuando se tiene como objetivo el análisis de imágenes, esto se efectúa porque presentan un mayor desempeño en el procesamiento de datos. Su origen se ubica en las investigaciones de Lecun et al. [12], destacando notablemente en el año 2011 al emplear el uso de GPU para su previo entrenamiento [13].

De acuerdo con [14], se explica que el diseño de las Redes Neuronales Artificiales (RNA) se basa en el funcionamiento de la corteza visual del cerebro, la cual presenta diversas especializaciones para procesar eficientemente la información visual. Se establece que en la corteza visual existen numerosas células receptoras que son capaces de detectar la luz en regiones superpuestas del campo visual, y que estas células receptoras realizan la misma operación de convolución para procesar sus entradas de manera similar.

1.1.2 Enfermedades neurodegenerativas

Las enfermedades neurodegenerativas se definen como aquellas que afectan al sistema nervioso central y resultan en una disfunción progresiva. Estas condiciones incapacitantes e incurables se caracterizan por la pérdida gradual de la función

neuronal y a menudo están asociadas con la atrofia de las estructuras del sistema nervioso afectado. Dentro de este grupo, se destaca un subconjunto importante de enfermedades relacionadas con el envejecimiento y la demencia, siendo la enfermedad de Alzheimer la más prominente, seguida por la Enfermedad de Parkinson (EP).

1.1.2.1 Enfermedad de Parkinson

En particular, la EP es la segunda enfermedad neurodegenerativa más común desarrollada en adultos [15]. Padecimiento caracterizado por una pérdida significativa de neuronas dopaminérgicas en la sustancia negra, así como por la presencia de inclusiones citoplasmáticas que consisten en agregados proteicos insolubles. Estos cambios patológicos conducen a un trastorno progresivo del movimiento, manifestándose con síntomas como temblores, bradicinesia (movimientos lentos) y rigidez. La edad promedio de inicio de la enfermedad se sitúa entre los 50 y 60 años, aunque puede variar en casos en los que la EP tiene un componente hereditario.

1.1.3 Software

Acorde con Pressman [16], el concepto de software se refiere a "programas independientes que abordan una necesidad específica de negocios". Las aplicaciones en este ámbito procesan datos comerciales o técnicos de manera que facilitan las operaciones comerciales o la toma de decisiones administrativas o técnicas.

El Software o Software de aplicación se utiliza para gestionar diversos aspectos del negocio en tiempo real, tales como transacciones, procesos, análisis y toma de decisiones de acuerdo con las necesidades de su implementación, así mismo se conoce como software a la medida, a aquella solución que precisa su diseño, desarrollo y mantenimiento al conjunto de requerimientos establecidos por los usuarios o clientes del producto.

Un concepto importante a considerar es que el software, durante su proceso de elaboración, requiere establecerse a través de una metodología o proceso de desarrollo, con el objetivo de identificar las fases de construcción en un plazo

determinado, entre ellas se conocen desde los requerimientos iniciales, hasta las pruebas y mantenimiento final del producto.

El modelo base del ciclo de vida del software se establece en cuatro etapas principales:

- **Análisis**, donde se evalúan temas como los requerimientos del sistema, áreas de impacto, actores involucrados, planeación de recursos y comprensión de los procesos operacionales.
- **Desarrollo**, donde se lleva a cabo todo el proceso de construcción del producto final, tomando en cuenta métricas de calidad y mantenibilidad del sistema.
- **Implementación**, fase que tiene como objetivo la puesta en marcha del sistema para su evaluación en un entorno productivo.
- **Mantenimiento**, como fase final, donde la prioridad es brindar soporte adicional al producto terminado para mejorar la usabilidad y procesos operativos.

1.1.4 Lenguajes de programación

1.1.4.1 Python

Python es un lenguaje de programación multiparadigma que se caracteriza por su rápida curva de aprendizaje y su gran potencia de procesamiento de datos. Este lenguaje utiliza estructuras de alto nivel de gran eficiencia y un óptimo sistema de programación orientado a objetos. Su sintaxis elegante y su tipado dinámico, frente a una naturaleza de interpretación, lo convierten en un lenguaje ideal para el scripting y el desarrollo rápido de aplicaciones, enfocado en la diversificación de plataformas y áreas de interés [17].

Además, Python demuestra ser una opción destacada en el campo de la inteligencia artificial y el AP. Su amplia gama de bibliotecas y marcos de trabajo especializados, como TensorFlow, Keras y PyTorch, permiten a los investigadores y desarrolladores implementar eficientemente algoritmos de aprendizaje automático y redes neuronales. Estas herramientas aprovechan al máximo las capacidades de Python en el procesamiento de grandes conjuntos de datos y en la construcción de modelos de

inteligencia artificial sofisticados. La comunidad activa y en constante evolución de Python asegura su relevancia continua en el ámbito de la inteligencia artificial y el AP [17].

1.1.5 Bibliotecas de Aprendizaje Profundo

1.1.5.1 Scikit-Learn

Scikit-Learn es una biblioteca de código abierto diseñada para facilitar la implementación rápida y sencilla de modelos de aprendizaje automático en Python. Esta herramienta representa una opción destacada en el ámbito de la inteligencia artificial y el análisis de datos debido a su amplia gama de algoritmos disponibles. Gracias a su integración directa con los modelos de clasificación de instancias, Scikit-Learn proporciona una solución eficiente para abordar problemas de clasificación en diferentes dominios.

La biblioteca destaca por su enfoque en la eficiencia y optimización de procesos de análisis. Aprovecha la capacidad de integrar código en C a través de la biblioteca Cython, lo que contribuye a mejorar el rendimiento de los algoritmos y acelerar el procesamiento de grandes volúmenes de datos [18].

1.2 Situación tecnológica, económica y operativa de la empresa

El Instituto Tecnológico de Orizaba es una institución pública que forma parte del Tecnológico Nacional de México y está bajo la dependencia de la Secretaría de Educación Pública SEP. Se encuentra ubicado en Oriente 9, Colonia Emiliano Zapata, en la ciudad de Orizaba, Veracruz. Esta destacada institución ofrece una amplia gama de programas educativos que incluyen carreras de licenciatura, maestría y doctorado.

Dentro de las maestrías que se ofrecen en el Instituto Tecnológico de Orizaba, se encuentran la Maestría en Ingeniería Electrónica, Maestría en Ingeniería Industrial, Maestría en Ciencias en Ingeniería Química, Maestría en Ingeniería Administrativa y la Maestría en Sistemas Computacionales. Asimismo, también se imparte el programa de Doctorado en Ciencias de la Ingeniería.

1.3 Planteamiento del problema

En relación con la Organización Mundial de la Salud (OMS), la EP es una enfermedad neurodegenerativa crónica y progresiva que afecta al sistema nervioso. Se caracteriza comúnmente por síntomas como la rigidez muscular, dificultades en el movimiento, alteraciones en la coordinación de los movimientos y temblores. Si bien el temblor es el síntoma más reconocido y asociado con esta enfermedad, existen otros síntomas que a menudo resultan más incapacitantes, como los síntomas no motores. Estos pueden incluir trastornos del control, problemas cognitivos y alteraciones del sueño, entre otros.

De acuerdo con la C. Marras et al. [19], más de 10 millones de personas en todo el mundo viven con este padecimiento. Aunque la incidencia de esta enfermedad aumenta con la edad, aproximadamente el 4% de las personas con Parkinson experimentan síntomas antes de los 50 años. Además, se estima que los hombres tienen una probabilidad 1.5 veces mayor de desarrollar Parkinson en comparación con las mujeres. Se prevé que para el año 2040, el Parkinson se convertirá en la enfermedad grave más común, con un incremento significativo en el número de personas afectadas, pasando de 7 millones a 12 millones según estudios recientes. En el caso de México, el Instituto Nacional de Neurología y Neurocirugía estima una prevalencia de 50 nuevos casos de Parkinson por cada 100,000 habitantes al año.

Por otra parte, se encontró que el uso de tecnologías como Internet de las Cosas (*IoT*) en la salud, permiten monitorizar en tiempo real las variables biomédicas de los pacientes a través de wearables, dicha información está disponible para el médico de forma remota. Además, las técnicas de Aprendizaje Automático son un gran aliado en la detección de enfermedades como el Parkinson, como se demuestra en [20]. El Aprendizaje Automático también se aplica en las recomendaciones médicas para pacientes con este tipo de enfermedades.

1.4 Objetivo general y objetivos específicos

1.4.1 Objetivo general

Desarrollar un módulo de software para la detección temprana de la enfermedad de Parkinson mediante técnicas de Deep Learning

1.4.2 Objetivos específicos

- Analizar las tecnologías de la información, algoritmos y conjuntos de datos referentes a la detección temprana de la enfermedad de Parkinson.
- Identificar los parámetros relacionados con la detección temprana de la EP mediante los procesos de trazabilidad.
- Diseñar un módulo de software para los procesos de detección temprana de la EP.
- Desarrollar un módulo de software para la detección temprana de la EP mediante técnicas de Aprendizaje Profundo.
- Evaluar el rendimiento del módulo desarrollado mediante diferentes casos de estudio como prueba de concepto.

1.5 Justificación

La alta incidencia de la EP es un problema de salud global que resulta en un deterioro neurodegenerativo progresivo en las personas afectadas. Entre los síntomas característicos se incluyen dificultades para caminar, temblores y alteraciones en la coordinación de los movimientos. Además, los pacientes también experimentan síntomas no motores, como problemas de sueño, depresión, trastornos de control de impulsos y dificultades cognitivas. Estos síntomas adicionales tienen un impacto significativo en la calidad de vida de los pacientes que padecen esta enfermedad.

Por otro lado, el constante avance tecnológico y su aceptación a nivel mundial de integrar soluciones de Tecnologías de la Información en el sector médico, es un factor que representa mayor relevancia día con día, esto resulta porque su uso e

implementación permite diagnosticar y tratar a los pacientes de una manera rápida y eficiente optimizando el uso de recursos en el sistema de salud.

El presente trabajo de tesis provee un impacto relevante para el caso de estudio, de acuerdo con la finalidad de analizar y determinar puntos clave en la detección temprana de la EP mediante características relacionadas con el comportamiento motriz de un paciente, todo esto, destacando el objetivo principal a evaluar que menciona, desarrollar un módulo de software para la detección temprana de la enfermedad de Parkinson mediante técnicas de Aprendizaje Profundo.

Capítulo 2. Estado de la práctica

A continuación, se presenta el resumen de los trabajos de investigación que tienen relación directa o indirecta y/o parcial o total con la tesis.

2.1 Trabajos relacionados

En [21] con el objetivo de establecer métricas simples y objetivas que permitan la diferenciación entre un modelo conductual normal y un modelo patológico, se exploró el uso de un dispositivo de pluma y tableta que busca estudiar las diferencias en el movimiento de la mano y coordinación entre cada grupo de participantes, el presente estudio, se realizó mediante la ejecución de trazos horizontales simples que registran señales sobre la superficie de una almohadilla, las señales obtenidas, en función de su procesamiento y análisis, los resultados buscan calcular datos característicos de cada participante, como lo es la velocidad del trazo, su desviación del plano y entropía de la trayectoria, por mencionar algunos. Con los datos obtenidos, los experimentos mostraron un valor promedio de precisión del 91% durante la predicción y clasificación de los participantes, adicionalmente, el proyecto presenta una solución en el contexto de aplicaciones de telemedicina y seguimiento de síntomas de la enfermedad de Parkinson, utilizando recursos de hardware de bajo costo y ampliamente accesibles.

Por su parte, en [22] los autores frente al deterioro constante en la escritura a mano de los pacientes con enfermedad de Parkinson, reconocieron que las investigaciones previas que proponen métodos basados en micrografía y visión por computadora, presentan dos problemas principales, el sesgo causado en los modelos por datos desequilibrados y la baja tasa de precisión para la clasificación de los pacientes, con base en estos problemas, se propuso el desarrollo de cuatro modelos diferentes de aprendizaje automático, dentro de los cuales se incluyó, un método de sub muestreo aleatorio para equilibrar el proceso de entrenamiento y un método de aprendizaje en cascada para mejorar la precisión, este último método conecta a un modelo Chi2 con impulso adaptativo (Adaboost). Consiguiendo de los datos obtenidos una mejora de rendimiento frente a otros seis sistemas en cascada similares de última generación,

además, representó también una mejora en la resistencia del modelo Adaboost convencional en un 3,3% y reduciendo su complejidad. Así mismo, el sistema en cascada demostró una precisión de clasificación del 76,44%, una sensibilidad del 70,94% y una especificidad del 81,94%.

En [23] el enfoque se dirigió hacia el estudio de los síntomas motores de la enfermedad de Parkinson (EP), esencialmente, la congelación de la marcha (FOG), como uno de los episodios más incapacitantes durante el desarrollo de la enfermedad, es por ello que se propuso un método de aprendizaje profundo para la detección de episodios de FOG en pacientes con EP, el cual consistió en una novedosa estrategia de entrenamiento mediante datos espectrales, los cuales, consideraron información previa y actual de las señales, para su aplicación se consideró una unidad de medición de inercia, colocada en la cintura de 21 pacientes con EP que manifestaron episodios de FOG. Los resultados demuestran superar con éxito a los métodos de vanguardia para la detección automática de FOG, logrando un 90% para la media geométrica entre sensibilidad y especificidad, frente a un 83% obtenido por algunos métodos de última generación durante la ejecución de análisis sobre las mismas métricas.

Mientras tanto, en [24] el objeto de estudio se enfocó en una de las principales afecciones provocadas por la enfermedad de Parkinson, el sistema nervioso central, durante el estudio se desarrolló un algoritmo de detección capaz de clasificar a los sujetos en Parkinson o normales en función de la distribución de la carga durante la marcha, con el beneficio de la detección temprana de dicha patología, su implementación se realizó mediante el análisis de señales espaciales y temporales sobre las fuerzas de reacción vertical del suelo, categorizando marchas equilibradas o desequilibradas con probabilidad de ser relevancia para la clasificación de posibles casos positivos. Con las pruebas efectuadas, el autor expresa el sistema con una precisión de clasificación general de 95%, utilizando un límite de decisión lineal y posicionando la propuesta con fines de evaluación para un programa de rehabilitación.

Así mismo en [25] se realizó un estudio dedicado a la detección de EP con base en los parámetros cinemáticos de los movimientos de las manos, implementando métodos de aprendizaje automático y un sensor *Leap Motion*, para este estudio se realiza la selección de tres tareas motoras basadas en la parte 3 de la escala unificada de la enfermedad de Parkinson modificada por la Sociedad de Trastornos del Movimiento (MDS-UPDRS), dentro de las cuales se integran, golpeteo de dedos (FT), pronación y supinación de la mano (PS) y movimientos de apertura y cierre (OC), estas tareas se analizan a través de la señal recibida por el sensor que calculó 25 parámetros clave y determinó mediante un algoritmo de máximos y mínimos su clasificación binaria, ejecutando este proceso para cada tarea, se procedió a realizar su categorización mediante 4 algoritmos de clasificación, KNN, SVM, Random Forest y Árboles de decisión. Finalmente, este estudio obtuvo como resultado promedio de precisión un valor de 98.3% global, siendo de cada tarea 95.3% para FT, 90.6% para OC y 93.8% para PS.

En [26], se tomó como referencia de partida la estimulación cerebral profunda adaptativa (DBS), la cual es un método para aliviar los efectos secundarios y mejorar la eficacia de estimulación para los trastornos del movimiento, sin embargo, durante el estudio se reconoce que estas técnicas se fundamentaron en el umbral de una sola característica, impidiendo la entrega optimizada de estimulación para un control preciso de los sistemas motores, por ello, los autores de la investigación propusieron un enfoque de aprendizaje automático para la detección de temblores en estado de reposo a partir de potenciales de campo local (LFP) registrados a partir del núcleo subtalámico (STN) en 12 pacientes con EP, los resultados de esta investigación compararon el rendimiento de diversos clasificadores de última generación frente a los biomarcadores basados en LFP, en el cual para la detección de temblores se visualizan oscilaciones de alta frecuencia que demostraron una mejora significativa en respuesta de estímulo, por ello, el método propuesto planteó una aplicación eficiente en el contexto de estimulación y tratamiento para la enfermedad de Parkinson.

Mientras tanto, en [27] se detectó la falta de pruebas remotas, no invasivas y objetivas que apoyen los estudios motores antes mencionados, es por ello que, durante este estudio a través de teléfonos inteligentes, se evaluó información de referencia clínica de los pacientes mediante métricas de la Escala Unificada de Clasificación de Enfermedad de Parkinson (UPDRS), algunos de los datos analizados son, la voz, la postura, la marcha, tiempo de respuesta y la trayectoria táctil, para las pruebas se les solicita a los participantes realizar las tareas de análisis cuatro veces al día durante un mes y cada semana, realizar una visita remota (telemedicina) con un especialista. Durante la aplicación del estudio en un entorno familiar y comunitario, los análisis clasificaron correctamente entre los pacientes con EP y los que no, obteniendo una media de sensibilidad de 96.2% y especificidad de 96.9%, concluyendo con un rango de error de 1.26 puntos en la escala UPDRS y demostrando así, el valor potencial y factibilidad de integrar un teléfono inteligente como herramienta de apoyo en el diagnóstico.

Desde otro punto de vista, en [28] se consideró el diagnóstico de síntomas motores en conjunto con el estudio de los no motores que aumentan paulatinamente con el desarrollo de la enfermedad, uno de ellos es, en esencia, la hiposmia idiopática (IH), asociada como un desencadenante de riesgo para el desarrollo de EP en adultos sanos, es así que, en este trabajo se utilizó un dispositivo portátil de nombre SensFoot V2, cuyo propósito es adquirir datos motores de 90 participantes (30 sanos, 30 con IH y 30 con EP) utilizando la escala MDS-UPDRS III para la evaluación de miembros inferiores, generando así, una matriz de características que permite identificar diferencias destacables entre los distintos grupos. En el análisis de clasificación mediante la aplicación de tres algoritmos de aprendizaje, se demostró la capacidad de distinguir entre los grupos de personas establecidos, demostrando así, que el sistema cuenta con las capacidades necesarias para apoyar al médico en la evaluación objetiva de la EP y la identificación de la IH en conjunto con el seguimiento de parámetros característicos para detectar su desarrollo prematuro.

No obstante, en [29] el punto de enfoque se dirigió a la obtención de información a través de las señales de voz empleando sistemas computacionales, determinando de esta manera, la clasificación de afección de una persona, con el objetivo de analizar el caso relación entre un paciente con EP y el deterioro del habla, en este trabajo de investigación, se utilizó una gama versátil de algoritmos de aprendizaje basados en la clasificación y con diferentes técnicas de reducción de dimensionalidad (DR). A través de estos métodos, se realizó un estudio comparativo de sus precisiones para diferenciar a un humano sano de uno que está afectado por la enfermedad, los resultados permitieron observar la complejidad temporal de los algoritmos para comprender el efecto de las técnicas de DR, adicionalmente, se obtuvo un nuevo algoritmo entrenado mediante un conjunto de datos de habla de Parkinson del repositorio de aprendizaje automático del *UCI Machine Learning Repository*.

Wodzinski [30] presentó un enfoque para la detección de la enfermedad de Parkinson usando vocales con fonación sostenida y una arquitectura ResNet, la cual, se dirigió originalmente a la clasificación de imágenes, durante la ejecución, Wodzinski aprovecha las ventajas de la arquitectura ResNet, previamente entrenada por un conjunto de datos para analizar alteraciones en el espectro de grabaciones de audio de los pacientes, durante el estudio, con un grupo de 100 participantes, de los cuales, 50 diagnosticados con la enfermedad de Parkinson y 50 clasificados como pacientes sanos, se realizó un total de 3 registros por persona, concluyendo así el proyecto con una precisión en el conjunto de validación, superior al 90%, considerando así, un modelo comparable con los métodos de clasificación de última generación. Los resultados del presente estudio demostraron que, gracias a sus niveles de precisión, se estima una gran capacidad de detección exitosa de la enfermedad de Parkinson a través del análisis de espectrogramas de frecuencias vocales.

En [31], teniendo en cuenta que, la EP presenta síntomas debido a la interrupción de células cerebrales que producen sustancias para su correcta comunicación, se estima que, con el tiempo se llegan a perder entre el 60 y 80 por ciento de estas dando cabida a los primeros síntomas motores del Parkinson y por ello se considera que la

enfermedad comienza desde años antes de su confirmación, es así que, los investigadores buscaron formas de reconocer los síntomas previos lo antes posible y detener la progresión de la enfermedad, en consecuencia, en este trabajo se propuso un diagnóstico basado en el aprendizaje automático de la EP que consiste en procesos de selección y clasificación de características, utilizando árboles de clasificación y regresión, redes neuronales artificiales y máquinas de vectores de soporte en los experimentos. Como resultado destacable se demostró que las máquinas vectoriales de soporte con eliminación de características recursivas, funcionaron mejor que algunos otros métodos planteados, alcanzando una precisión del 93.84% con el menor número de características de voz para el diagnóstico de Parkinson.

Por otro lado, en [32] se realizó un estudio similar donde también se demostró que más del 60% de las neuronas dopaminérgicas llegan a perderse como causa previa a la manifestación de los síntomas clínicos de la EP, por lo que, se deduce la existencia de una etapa promotora estudiada para su correcto tratamiento, dentro de esta fase se mencionan, entre algunos síntomas, trastornos de comportamiento de sueño por movimiento ocular rápido y pérdida olfativa, en el artículo, se implementó un sistema de clasificación, mediante técnicas como Naive Bayes, SVM, Random Forest y Boosted Trees, donde mediante el estudio de 183 participantes sanos y 401 con EP temprana se observó un rendimiento sustancial en el método SVM, proporcionando un promedio de 96.40% de precisión, 97.03% de sensibilidad, 95.01% de especificidad y 98.88% de área bajo la curva ROC, resultados que infieren sobre el estudio de una combinación de marcadores no motores, mediciones del líquido cefalorraquídeo e imágenes que brindan soporte en el diagnóstico preclínico de la EP.

Haq et al. [33] enfatizaron que el diagnóstico preciso de la enfermedad de Parkinson, específicamente en sus etapas iniciales, es extremadamente complejo y requiere mucho tiempo, por ello, su diagnóstico preciso y eficiente se considera un desafío significativo para los expertos médicos e investigadores. En este estudio se propuso el desarrollo de un sistema de predicción no invasivo basado en Machine Learning y Deep Neural Networks, para su desarrollo, se utilizaron modelos de vectores, regresión

logística y redes neuronales para la clasificación de personas con enfermedad de Parkinson y personas sanas. El sistema provee un conjunto de datos que se divide en un 70% para el entrenamiento y 30% para las pruebas, además, se emplean métricas de evaluación del rendimiento como la precisión de la clasificación, la sensibilidad, la especificidad y el coeficiente de correlación de Matthews para evaluar el rendimiento del modelo, donde los resultados clasificaron eficazmente a los determinados grupos de personas y mantuvieron un rendimiento del modelo Deep Neural Network superior frente a algunos métodos tradicionales de clasificación.

Pahuja y Nagabhushan [34] detectan la necesidad de contar con soluciones de enfoque autónomo, basados en el aprendizaje automático para brindar soporte a los médicos en la detección de la enfermedad de Parkinson en su etapa inicial, a través de este trabajo de investigación se tuvo como objetivo, proporcionar una encuesta perspicaz y comparar las técnicas de inteligencia computacional existentes utilizadas para la detección de casos patológicos, con el fin de identificar el clasificador más eficiente y apropiado para la detección, radicando en su aplicación para el conjunto de datos local, se realizó la discusión de los principales tipos de clasificadores, entre ellos: Multilayer Perceptron, Support Vector Machine y KNN, donde se encontró que los métodos de redes neuronales artificiales (ANN) con el algoritmo de Levenberg-Marquardt son los modelos con mayor precisión de clasificación, y que presentaron un valor promedio de precisión de 95.89% frente a otros modelos de predicción.

En [35], se realizó la investigación para descubrir la relación entre las propiedades de la señal del habla y las puntuaciones UPDRS en el diagnóstico de la EP, donde empleando de forma intensiva las técnicas de aprendizaje automático supervisado en la predicción, se estima que la mayoría de las soluciones desarrolladas por estos métodos no admiten las actualizaciones incrementales de datos, además, las técnicas supervisadas estándar no permiten utilizarse en una situación incremental para la predicción de enfermedades y, por lo tanto, es requerido volver a calcular todos los datos de entrenamiento para construir los modelos de predicción, en este artículo, se desarrolló un nuevo método para la predicción de UPDRS, utilizando principalmente

técnicas de máquinas de vectores de soporte incremental, donde los resultados del análisis experimental demostraron que el modelo propuesto es eficaz en la predicción de la EP mediante la estimación UPDRS durante la atención médica.

Así mismo, en [36], se estudió cómo la EP presenta síntomas gracias a la pérdida de células productoras de dopamina en el cerebro, a medida que su producción va disminuyendo, los síntomas de la enfermedad comienza a ser cada vez más notorios y las personas presentan dificultades para caminar o hablar, frente a la necesidad de contar con un método de detección temprana de la enfermedad a través del análisis de síntomas predictivos, en este estudio se realizó la comparación de diferentes métodos de clasificación, tales como la regresión logística, máquinas de vectores de soporte, árboles de decisión, aumento gradual y bosques aleatorios, para realizar el análisis de cada uno de los métodos, inicialmente se utiliza un total de 1208 conjuntos de datos del habla, entre los cuales, se cuentan con 26 características obtenidas de pacientes clasificados con la EP y pacientes sanos, posteriormente, el conjunto de entidades clasificado se expande a través de mapas de correlación construidos con base en las características obtenidas mediante un análisis de componente principales (PCA), la ganancia de información (IG), entre algunas otras características relacionadas, finalmente, se demostró que los resultados de clasificación por características expandidas, superan los métodos de clasificación tradicionales.

Por otro lado, en [37] con el análisis de conjuntos de datos médicos mediante técnicas de minería de datos, se propuso un nuevo sistema basado en el conocimiento para la predicción de enfermedades con el apoyo de técnicas de agrupamiento, eliminación de ruido y predicción de datos, proporcionadas mediante métodos de Árboles de Clasificación y Regresión (CART), los cuales se utilizaron para generar reglas difusas en la base del conocimiento, este método comprueba a través de varios conjuntos de datos médicos públicos de telemonitoreo, tales como, Pima Indian Diabetes, Mesothelioma, WDBC y StatLog, por mencionar algunos. Los estudios mostraron una mejora notable en los resultados de precisión durante la predicción de enfermedades, así mismo, se obtuvo que la propuesta CART a través de los procesos de difusión,

eliminación y agrupación de datos es un método efectivo en la predicción de enfermedades a partir de conjuntos de datos del mundo real y como método analítico clínico.

Mientras tanto en [38], el enfoque se dirigió a la necesidad insatisfecha de pruebas predictivas que faciliten la detección temprana y la caracterización de distintos subtipos de enfermedades, así como también, obtener predicciones mejoradas e individualizadas del curso de la enfermedad mediante la detección de patrones de comportamiento en conjuntos de datos complejos y multidimensionales, para ello, en esta investigación se utilizaron enfoques de aprendizaje automático para la identificación de subtipos de pacientes y predicción de la progresión de la enfermedad, posteriormente, los modelos obtenidos se validaron en una cohorte estadística independiente y clínicamente clasificada por el Programa de Biomarcadores de la Enfermedad de Parkinson (PDBP), de esta manera se distinguieron tres subtipos de enfermedad distintos con tasas de progresión altamente predecibles, lentos, moderados y rápidos, los resultados demostraron la deconstrucción de heterogeneidad dentro de las cohortes de pacientes, permitiendo tener implicaciones inmediatas para los ensayos clínicos en la mejora de la detección de resultados clínicos significativos, así mismo, se logró mejorar el asesoramiento al paciente, el diseño de ensayos clínicos, la asignación de recursos de atención médica y, en última instancia, la atención clínica individualizada.

En [39] se detectaron eficazmente las deficiencias vocales durante las primeras etapas de la EP, es así que los sistemas de diagnóstico basados en trastornos vocales mediante la implementación de tecnologías de redes neuronales, están a la vanguardia en los estudios de detección de la EP, durante el proyecto se propuso emplear dos marcos de trabajo basados en redes neuronales convolucionales para la clasificación de la EP utilizando conjuntos de características vocales, aunque ambos se emplearon para la combinación de características de registros, presentaron diferencias en los términos a emplear, se presentó un primer modelo que caracteriza el estudio en datos CNN de 9 capas y un segundo marco que realiza su estudio de forma paralela entre

los datos de entrada y las capas de convolución, posteriormente los modelos se entrenaron mediante conjuntos de datos del repositorio de aprendizaje automático de UCI, donde demostraron rendimientos prometedores, con capacidad de aprendizajes profundos de cada conjunto de características a través del segundo marco de trabajo, ya que las características extraídas no solo consiguen tener éxito en la distinción de pacientes con EP, sino que también, mostraron efectividad en aumentar el poder discriminativo de los clasificadores.

Por otro lado, en [40] se identificó la falta de existencia de terapias modificadoras establecidas para la EP y la predicción de los resultados dentro del área de investigación, donde el sistema actual, se basó en las imágenes del sistema de dopamina con imágenes de tomografía computarizada y la presencia de síntomas clave. Este estudio tuvo como objetivo, desarrollar un enfoque basado en el aprendizaje profundo para predecir resultados de los pacientes con EP utilizando datos clínicos longitudinales que contienen información diversa de los estudios realizados al paciente, inicialmente se extrajeron los datos clínicos basados en el aprendizaje profundo, y posteriormente, se combinaron para predecir el rendimiento motor en el año 4, una vez obtenidos los resultados, el desempeño se evaluó a través de una validación cruzada de 10 pasos, dicha validación se realizó sobre la base del error absoluto medio (MAE), el cual, durante el estudio se presentó con un valor de 4.33±3.36 mediante el estudio de imágenes, 3.71±2.91 en caracteres y 3.22±2.71 en datos combinados, demostrando de esta manera que el enfoque propuesto incorporando los datos combinados, presentó una promesa significativa para la predicción de resultados en pacientes con EP.

2.2 Análisis comparativo

En la Tabla 2.1 se muestra de manera comparativa, la información de cada uno de los artículos relacionados, su finalidad es, observar las similitudes y diferencias de mayor relevancia.

Tabla 2.1. Análisis comparativo, estado de la práctica

Artículo	Problema	Contribución	Tecnología	Resultados	Estado
Kotsavasiloglou et al. [21]	Falta de métricas de diferenciación de modelos conductuales	Herramienta de estudio de métricas en la ejecución de trazos simples	Aprendizaje Profundo	Promedio de precisión del 91% en la predicción y clasificación de pacientes	Terminado
Ali et al. [22]	Problemas de estudio en el deterioro constante en la escritura a mano	Modelos de aprendizaje automático para estudios basados en micrografía y visión por computadora	Aprendizaje Profundo	Se logró un modelo con precisión de clasificación del 76.44%, sensibilidad del 70.94% y 81.94% de especificidad	Terminado
Camps et al. [23]	Optimización de métodos de aprendizaje de síntomas motores de la EP como la congelación de la marcha (FOG)	Método de aprendizaje profundo para la detección de episodios FOG	Aprendizaje Profundo	Se consiguió un modelo de mejora logrando un valor promedio de 90% de sensibilidad y especificidad, frente a un 83% presentado en modelos similares	Propuesta

Artículo	Problema	Contribución	Tecnología	Resultados	Estado
Alkhatib et al. [24]	Falta de análisis en las afecciones provocadas por la EP en el sistema nervioso central	Algoritmo para la detección y clasificación de personas con EP	Aprendizaje Profundo	Durante los estudios se obtuvo un promedio de clasificación de 95%	Propuesta
Moshkova et al. [25]	Falta de métodos para el estudio de la EP con base en parámetros cinemáticos de los movimientos de las manos	Método de aprendizaje automático a través de un sensor *Leap Motion* y uso de 4 algoritmos de clasificación	Aprendizaje Profundo Leap Motion	Se obtuvo un resultado promedio de precisión global de 98.3%, diversificado en 4 tareas principales	Terminado
Yao et al. [26]	Falta de métodos para la óptima estimulación cerebral profunda	Enfoque de aprendizaje automático para la detección de temblores a partir del núcleo subtalámico	Aprendizaje Profundo	Se obtuvo una aplicación eficiente para la estimulación y tratamiento de la EP	Terminado

Artículo	Problema	Contribución	Tecnología	Resultados	Estado
Arora et al. [27]	Falta de pruebas remotas, no invasivas y objetivas para estudios motores	Sistema de evaluación de información de referencia clínica	Aprendizaje Profundo Smartphone	Sistema de evaluación métrica que logró una media de sensibilidad de 96.2% y 96.9% de especificidad	Terminado
Rovini et al.[28]	Falta de herramientas de estudio para síntomas de la EP	Análisis de clasificación comparativa de los estudios realizados	Aprendizaje Profundo	Se demostró que el sistema cuenta con las capacidades necesarias para apoyar al personal médico	Propuesta
Anand et al. [29]	Falta de herramientas de análisis de voz	Herramienta de análisis para estudio del deterioro del habla	Aprendizaje Profundo	Se obtuvo un nuevo algoritmo entrenado mediante conjuntos de datos de la UCI	Propuesta
Wodzinski et al. [30]	Falta de herramientas de análisis fonético	Herramienta de análisis y clasificación de espectro fonético de pacientes con EP	Aprendizaje Profundo	Se obtuvo un algoritmo de predicción promedio del 90% de asertividad	Terminado

Artículo	Problema	Contribución	Tecnología	Resultados	Estado
Karapinar et al. [31]	Falta de métodos para la detección temprana de la EP	Análisis de algoritmos de clasificación y evaluación del porcentaje de precisión	Aprendizaje Profundo	Se obtuvo el algoritmo de máquinas vectoriales como el de mayor precisión obteniendo 93.84%	Terminado
Prashant et al. [32]	Falta de métodos para la detección de la EP en etapa pre motora	Sistema de clasificación mediante algoritmos de análisis de datos	Aprendizaje Profundo	Se obtuvo un sistema con valores promedio de 96.04% de precisión y 97.03% de sensibilidad	Terminado
Haq et al. [33]	Complejidad de diagnóstico de la EP en etapas iniciales	Sistema predictivo no invasivo basado en Machine Learning para diagnóstico temprano	Aprendizaje Profundo Machine Learning	Se logró obtener un conjunto de datos que funge, 70% como datos de entrenamiento y 30% en pruebas para algoritmos de clasificación	Propuesta

Artículo	Problema	Contribución	Tecnología	Resultados	Estado
Pahuja et al. [34]	Falta de soluciones autónomas para brindar soporte médico	Modelo de comparación de técnicas de análisis y clasificación de datos	Aprendizaje Profundo	Se obtuvo que las redes neuronales artificiales son la mejor opción al presentar 95.89% de precisión	Terminado

En conclusión, se observó que el diseño del módulo de análisis para la detección temprana de la EP, demuestra que las tecnologías de AP representan una propuesta de alto impacto en el contexto presentado, obteniendo resultados positivos y ampliamente implementados a través de la literatura relacionada.

2.3 Solución propuesta

Teniendo en cuenta las ventajas, desventajas, costos y beneficios de las diferentes tecnologías de desarrollo disponibles para cada elemento del módulo, se diseñó la siguiente propuesta como la solución más adecuada. A continuación, se describen cada una de las tecnologías consideradas en cada nivel del módulo.

2.3.1 Tecnologías del entorno de desarrollo y despliegue

2.3.1.1 Visual Studio Code

Visual Studio Code es un editor de código fuente desarrollado por Microsoft, es popular entre los desarrolladores de software debido a su amplia gama de características y funcionalidades. Este editor es gratuito, de código abierto y altamente personalizable, lo que permite configurar su entorno de trabajo de acuerdo con necesidades del proyecto. Además, Visual Studio Code es compatible con múltiples lenguajes de programación, marcos de trabajo y herramientas, lo que lo convierte en una herramienta muy versátil y de gran utilidad para el desarrollo del proyecto [41].

2.3.1.2 Docker

Docker es una plataforma de software que permite crear, probar y desplegar aplicaciones en contenedores. Los contenedores son unidades de software que encapsulan todas las dependencias y configuraciones necesarias para ejecutar una aplicación de manera eficiente y consistente en diferentes entornos. Docker permite crear imágenes de contenedor personalizadas y compartir fácilmente estas imágenes a través de un registro de imágenes, lo que hace que el proceso de implementación de aplicaciones sea mucho más rápido y eficiente, lo que lo convierte en una tecnología esencial para el encapsulamiento y despliegue de los servicios del proyecto [42].

2.3.2 Tecnologías de la capa de presentación e integración

En esta sección se presentan las tecnologías propuestas con enfoque en el desarrollo eficiente de interfaces gráficas y mejoras de la experiencia del usuario. Esta capa incluye tecnologías que permiten el desarrollo de interfaces interactivas, dinámicas y atractivas, así como también, herramientas de integración que permiten la comunicación con los servicios del módulo.

2.3.2.1 TypeScript

Typescript es un lenguaje de programación de código abierto con base de desarrollo en *JavaScript* que incorpora la adición de diversas características avanzadas. *TypeScript* permite procesos como la detección de errores en tiempo de compilación, lo que mejora significativamente la calidad de software al reducir errores de producción. Además, ofrece mayor escalabilidad en proyectos de gran tamaño, así como también, amplio soporte por parte de la comunidad de desarrolladores [43].

2.3.2.2 Angular

Angular es un marco de trabajo construido en *TypeScript* dedicado al desarrollo de aplicaciones web dinámicas. *Angular* ofrece una amplia gama de características enfocadas en optimizar la eficiencia en los procesos de desarrollo, entre los que se incluyen, modularidad de componentes, técnicas de inyección de dependencias y vinculación de datos bidireccional. Además, se presenta como una opción popular en

su ámbito gracias a su capacidad de escalabilidad y fácil mantenimiento, contando con soporte por una amplia comunidad en línea [44].

2.3.2.3 IONIC

IONIC es un marco de trabajo construido mediante las tecnologías de *Angular* y *TypeScript*, su enfoque se determina en el desarrollo de aplicaciones híbridas, por lo que proporciona a los desarrolladores una solución eficiente en la creación de proyectos dirigidos a diferentes plataformas utilizando una misma base de código, cualidad que agiliza los procesos de desarrollo y mantenimiento. Además, *IONIC* destaca por contar con un alto grado de integración con nuevas tecnologías y su amplio catálogo de recursos que permiten resultados de apariencia y funcionalidad profesional [45].

2.3.2.4 PHP

PHP es un lenguaje de programación de código abierto utilizado principalmente en la creación de aplicaciones web dinámicas y sitios web. A diferencia de otros lenguajes web, carga su procesamiento en el lado del servidor, por lo que es una alternativa óptima en el procesamiento de formularios web para el envío y recepción de datos. Además, ofrece una amplia variedad de herramientas para la manipulación de información, permitiendo la rápida integración con sistemas gestores de bases de datos, como *MySQL.* Si bien*, PHP* presenta una curva de aprendizaje de nivel promedio, cuenta con una extensa comunidad que provee soporte y diversas herramientas con el fin de facilitar su proceso de integración [46].

2.3.3 Tecnologías de la capa de servicios

En esta sección se mencionan las tecnologías clave enfocadas en la correcta interpretación y clasificación de datos mediante elementos que permiten la integración de servicios altamente escalables y eficientes. Esta capa incorpora las tecnologías necesarias para recibir, procesar y almacenar la información del módulo, así como también la correcta gestión de los procesos de AP requeridos para la clasificación de la EP.

2.3.3.1 Python

Python es un lenguaje de programación multiparadigma que se caracteriza por rápida curva de aprendizaje y su gran potencia de procesamiento de datos, este lenguaje se conforma por estructuras de alto nivel de gran eficiencia y un óptimo sistema de programación orientado a objetos. Su sintaxis elegante y tipado dinámico, frente a una naturaleza de interpretación, lo convierten en un lenguaje ideal para el scripting y el desarrollo rápido de aplicaciones, enfocado en la diversificación de plataformas y áreas de interés [17].

2.3.3.2 Flask

Flask es un marco de trabajo ligero basado en el lenguaje de programación *Python,* el cual, permite la creación de aplicaciones web y servicios de transferencia de estados (RestAPI) de manera rápida y eficiente. Flask permite generar una interfaz de comunicación entre aplicaciones mediante la transferencia de datos en un formato común, lo que facilita la integración con tecnologías de terceros. Además, la comunidad de Flask proporciona una extensa documentación y guías de apoyo con el fin de agilizar la curva de aprendizaje a nuevos desarrolladores [47].

2.3.3.3 Scikit-Learn

Scikit-Learn es una biblioteca de código abierto enfocada en la implementación fácil y rápida de modelos de aprendizaje automático en el lenguaje de programación *Python*, se presenta como herramienta de gran afinidad en el estudio de la EP, gracias a su directa y amplia integración con los modelos de clasificación de instancias, así como también su optimización de procesos de análisis que mejora la eficiencia de los algoritmos mediante la integración de código en C a través de la biblioteca Cython y soporte de variaciones en los métodos SVM [18].

2.3.4 Tecnologías de la capa de persistencia

Esta sección incorpora las tecnologías que permiten el almacenamiento seguro y confiable de la información. Esta capa incluye tecnologías de bases de datos

relacionales que aseguren una alta capacidad de escalabilidad y flexibilidad conforme al crecimiento histórico del proyecto.

2.3.4.1 MySQL

MySQL, es un SGBD de código abierto, propiedad de Oracle® Corporation quien distribuye y brinda soporte al código de la aplicación, MySQL mantiene soporte de estructuras de bases de datos relacionales, desde contextos simples como una lista de registros, como también grandes volúmenes de información corporativa, Así mismo, MySQL Server se caracteriza por presentar una solución de gran rapidez en el procesamiento de datos, gracias a su óptimo rendimiento en el consumo de recursos y su gran diversidad de herramientas que extienden su compatibilidad con diversos sistemas y plataformas de trabajo [48].

2.3.5 Justificación de la solución

La solución propuesta, se enfoca en el análisis de las ventajas y desventajas de las tecnologías disponibles y su idoneidad para cada uno de los elementos del módulo a desarrollar.

En particular, para la capa de presentación y capa de integración, se ha decidido utilizar *TypeScript* como principal lenguaje de programación que ofrece un paradigma orientado a objetos y tipado estático que compila a *JavaScript*, lo que garantiza un alto nivel de rendimiento y estabilidad en el código. Por otro lado, se implementa mediante los marcos de trabajo *Ionic* y *Angular* debido a que estas tecnologías ofrecen un alto nivel de rendimiento y escalabilidad, además de brindar las herramientas necesarias para el desarrollo de aplicaciones web amigables y de fácil interacción para el usuario final. Además, su gran capacidad en la gestión de datos es uno de los conceptos principales que determinan a *PHP* como eficaz herramienta de comunicación entre los servicios y el sistema de persistencia

Por otro lado, enfocado en la capa de servicios, debido a su amplia adopción y popularidad en el ámbito de la IA, se optó por utilizar *Python* como lenguaje de programación principal. Junto con *Flask*, una herramienta altamente versátil y fácil de

usar que permite la integración con diversas herramientas, lo que la hace ideal para procesar grandes cantidades de datos y consultas como servicio *RestAPI*. Por otro lado, *Scikit-Learn* se eligió como biblioteca de AP debido a su capacidad para soportar una amplia gama de algoritmos de clasificación, lo que lo convierte en una solución segura y confiable para almacenar los clasificadores de información.

Finalmente, se seleccionó *MySQL* como sistema gestor de base de datos principal, debido a su alta escalabilidad, fiabilidad y rendimiento en gestión de grandes volúmenes de datos con optima gestión de recursos. Herramientas ideales para garantizar la integración y disponibilidad de la información.

Capítulo 3. Aplicación de la metodología

El presente capítulo muestra el desarrollo del módulo y la aplicación de las tecnologías propuestas mediante el seguimiento de cuatro fases principales. A continuación, se describe cada una de ellas:

- **Análisis:** Expresa los requerimientos establecidos para el desarrollo del módulo y los casos de uso determinados para el funcionamiento del módulo.
- **Diseño:** Incorpora el diseño de base de datos, la propuesta de interfaz gráfica y el diseño arquitectónico para la implementación del módulo.
- **Desarrollo:** Muestra el modelo de interfaces desarrolladas y describe el módulo de clasificación implementado.
- **Pruebas:** Representa mediante casos de estudio los resultados de funcionalidad y rendimiento del módulo implementado.

En la Figura 3.1 se representa gráficamente el flujo de trabajo a seguir durante el desarrollo de la metodología.

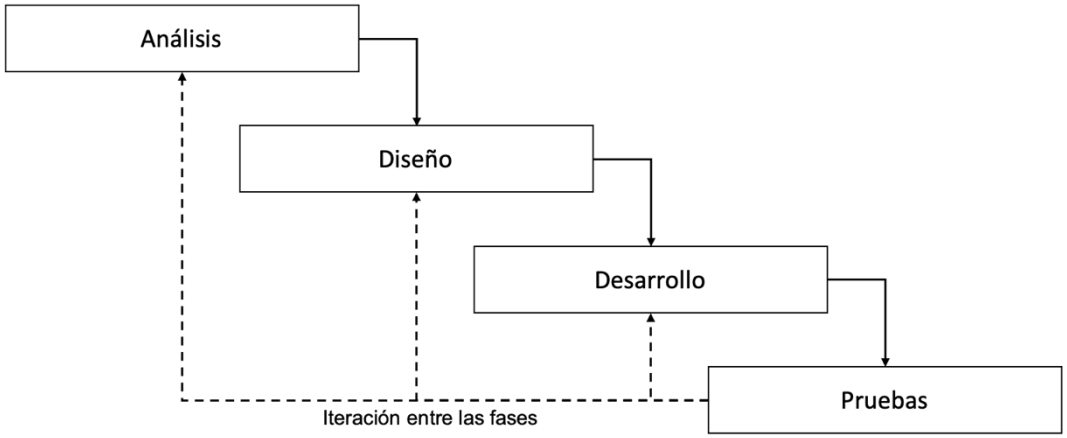


Figura 3.1 Metodología de desarrollo

3.1 Análisis

En esta sección, se describe un análisis detallado de los conceptos base principales para el desarrollo de la solución propuesta, para ello, se determinaron las características necesarias para el correcto funcionamiento del módulo, estableciendo

así, los requisitos funcionales, no funcionales y casos de uso para la interacción con el módulo.

3.1.1 Requerimientos

En esta sección se describen los requisitos funcionales contemplados para el desarrollo del proyecto.

Acceso

- Para ingresar, el usuario autentica su sesión mediante el ingreso de sus credenciales de acceso a través de una pantalla de inicio de sesión.
- Para finalizar la sesión, el usuario cuenta con un botón de salir en el menú principal.

Pruebas

- El usuario cuenta con la funcionalidad de generar nuevas pruebas para su aplicación.
- El usuario tiene la funcionalidad de enviar / registrar una prueba una vez finalizada su aplicación.
- El usuario posee acceso a la funcionalidad de consultar las pruebas realizadas por los pacientes.
- El sistema automatiza el procesamiento de las pruebas recibidas para su clasificación con el modelo de análisis.
- El sistema automatiza el registro de resultados en el modelo de persistencia una vez obtenido el resultado de análisis.

Ejercicios

- Durante la consulta de pruebas el sistema cuenta con el acceso a la visualización de cada uno de los ejercicios realizados incluyendo los resultados obtenidos.
- Durante la aplicación de pruebas el sistema muestra una batería de ejercicios en pantalla para su aplicación

Pacientes

- El usuario dispone de acceso a consultar la lista de pacientes asignados a su servicio.
- El usuario está habilitado para consultar el historial de pruebas realizadas por el paciente y sus detalles generales.

Perfil

- El usuario cuenta con la funcionalidad de modificar su contraseña de acceso al módulo.
- El usuario posee acceso para visualizar la información de su perfil.
- El usuario tiene acceso a modificar sus datos de contacto en su perfil.
- El usuario dispone de la funcionalidad de seleccionar el idioma de preferencia para mostrar la interfaz del módulo.

3.1.2 Casos de uso

A continuación, se muestran los diagramas de casos de uso para cada contexto contemplado en el desarrollo del proyecto.

Ejercicios

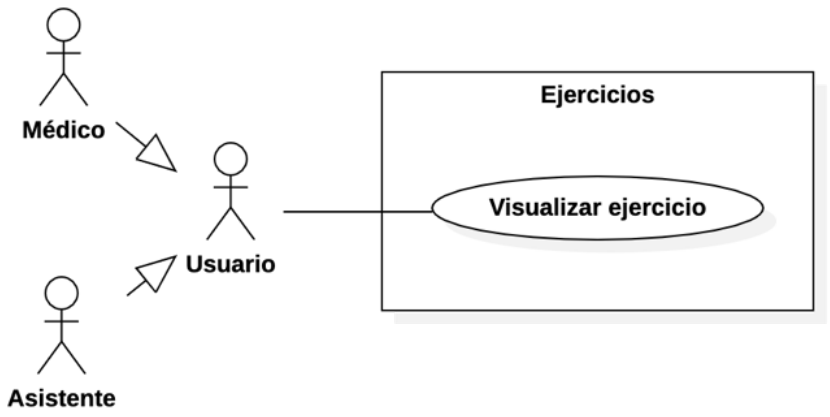


Figura 3.2 Diagrama de casos de uso del contexto Ejercicios

Caso de uso	Visualizar ejercicio	Contexto	Ejercicios
Descripción	Permite al actor visualizar los datos de un ejercicio registrado en el sistema		
Actor	Usuario		
Relaciones	Caso de uso: [Pruebas.Consultar]		
Precondiciones	Los resultados se muestran con relación a las pruebas registradas en el sistema		
Postcondiciones	NA		

General

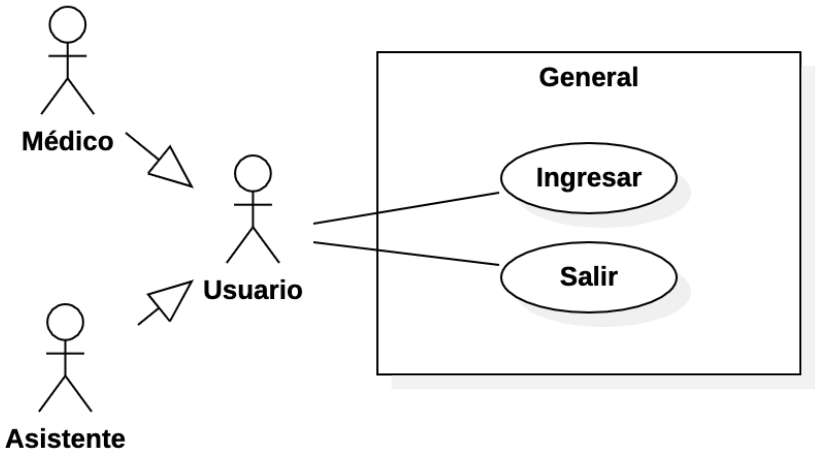


Figura 3.3 Diagrama de casos de uso del contexto General

Caso de uso	Ingresar	Contexto	General
Descripción	Permite al actor ingresar al módulo mediante sus credenciales de acceso		
Actor	Usuario		
Relaciones	NA		
Precondiciones	Registro previo de credenciales de acceso para el usuario		
Postcondiciones	El sistema almacena la sesión activa para permitir la navegación en el módulo		

Caso de uso	Salir	Contexto	General
Descripción	Permite al actor finalizar su sesión en el módulo		
Actor	Usuario		
Relaciones	Caso de uso: [General.Ingresar]		
Precondiciones	Sesión de usuario activa en el sistema		
Postcondiciones	El sistema finaliza la sesión activa y restringe el acceso al módulo hasta un nuevo ingreso		

Pacientes

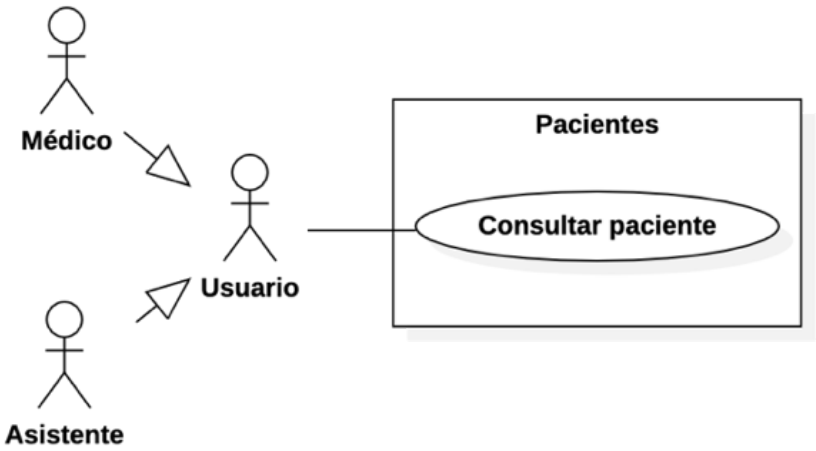


Figura 3.4 Diagrama de casos de uso del contexto Pacientes

Caso de uso	Consultar paciente	Contexto	Pacientes
Descripción	Permite al actor consultar la información de un paciente para la revisión o ejecución de pruebas		
Actor	Usuario		
Relaciones	NA		
Precondiciones	El usuario cuenta con la relación paciente-médico o paciente-asistente establecida para la consulta de información		
Postcondiciones	NA		

Pruebas

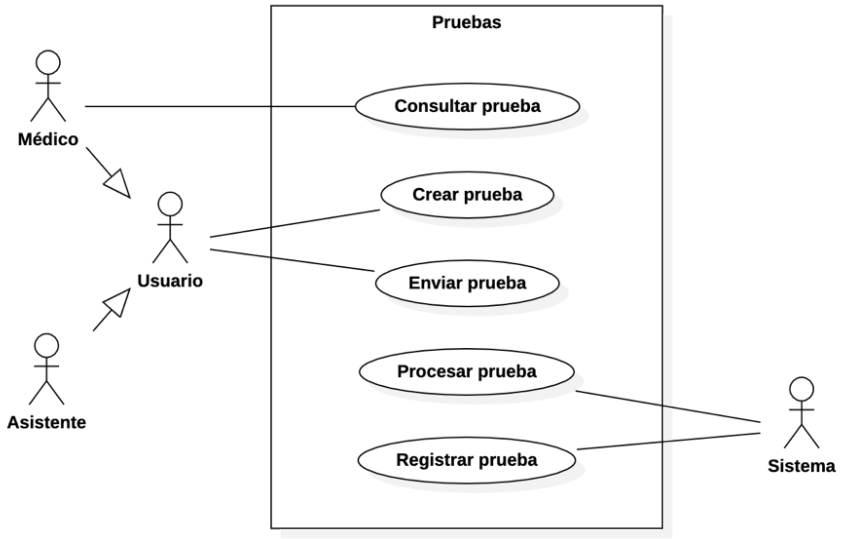


Figura 3.5 Diagrama de casos de uso del contexto Pruebas

Caso de uso	Consultar prueba	Contexto	Pruebas
Descripción	Permite al actor consultar el registro de pruebas realizadas por el paciente		
Actor	Médico		
Relaciones	Caso de uso: [Pacientes.Consultar]		
Precondiciones	La prueba a consultar se encuentra correctamente registrada y el usuario cuenta con el acceso al expediente correspondiente		
Postcondiciones	NA		

Caso de uso	Crear prueba	Contexto	Pruebas
Descripción	Permite al actor realizar una nueva prueba al paciente		
Actor	Usuario		
Relaciones	Caso de uso: [Pacientes.Consultar]		
Precondiciones	El paciente a realizar la prueba se encuentra previamente registrado en el sistema		
Postcondiciones	NA		

Caso de uso	Enviar prueba	Contexto	Pruebas
Descripción	Permite al actor enviar los resultados generados durante la prueba al servidor de análisis		
Actor	Usuario		
Relaciones	Caso de uso: [Pruebas.crear], [Pruebas.procesar]		
Precondiciones	Cada uno de los ejercicios presentados fueron completados de acuerdo con el proceso establecido		
Postcondiciones	El sistema procede a realizar el procesamiento de datos		

Caso de uso	Procesar prueba	Contexto	Pruebas
Descripción	Permite al actor procesar cada uno de los ejercicios recibidos en la prueba para su interpretación		
Actor	Sistema		
Relaciones	Caso de uso: [Pruebas.enviar], [Pruebas.registrar]		
Precondiciones	La prueba se encuentra completada y enviada al módulo de procesamiento		
Postcondiciones	Los resultados cuentan con su registro correspondiente en el modelo de persistencia		

Caso de uso	Registrar prueba	Contexto	Pruebas
Descripción	Permite al actor registrar los resultados obtenidos de la prueba realizada		
Actor	Sistema		
Relaciones	Caso de uso: [Pacientes.procesar]		
Precondiciones	Ejecución correcta de la prueba determinando los resultados de clasificación.		
Postcondiciones	El estado de la prueba se representa como: FINALIZADO		

Usuarios

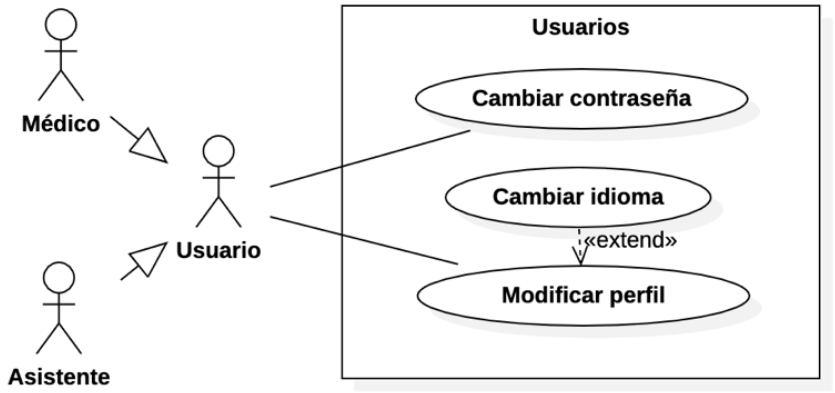


Figura 3.6 Diagrama de casos de uso del contexto Usuarios

Caso de uso	**Cambiar contraseña**	**Contexto**	**Usuarios**
Descripción	Permite al actor modificar su contraseña de acceso		
Actor	Usuario		
Relaciones	NA		
Precondiciones	NA		
Postcondiciones	NA		

Caso de uso	**Modificar perfil**	**Contexto**	**Usuarios**
Descripción	Permite al actor modificar información de su perfil personal		
Actor	Usuario		
Relaciones	NA		
Precondiciones	NA		
Postcondiciones	NA		

Caso de uso	Cambiar idioma	Contexto	Usuarios
Descripción	Permite al actor cambiar el idioma en que se mostrará la interfaz gráfica		
Actor	Usuario		
Relaciones	Caso de uso: [Pacientes.Modificar perfil]		
Precondiciones	NA		
Postcondiciones	La interfaz gráfica del módulo presenta su diseño en el idioma seleccionado		

3.1.3 Características de la aplicación

Con respecto a la selección de tecnologías para el desarrollo de la aplicación web y con base en la propuesta realizada, se determinó el uso del marco de trabajo *IONIC* en conjunto con la tecnología de desarrollo *Angular*. Este marco de trabajo destaca por ser una herramienta de fácil uso y que provee amplio soporte para su rápida integración en el desarrollo de interfaces gráficas de usuario. Además, hace posible el despliegue de aplicaciones híbridas, por lo que proporciona el acceso a nuevas áreas de oportunidad como el uso de plataformas móviles [49], [45]. Así mismo, se utilizó Flask, un micro marco de trabajo desarrollado en el lenguaje Python y que se determinó gracias a su simplicidad, flexibilidad y amplia capacidad de integración con nuevas tecnologías. Punto clave que lo convierte en una herramienta ideal para la construcción del núcleo de la aplicación [47].

Mientras tanto, para el módulo de Inteligencia, dedicado a la distribución de algoritmos de clasificación, se optó por utilizar Scikit-Learn, una biblioteca de Aprendizaje Profundo basada en el lenguaje de programación Python. Esta biblioteca proporciona herramientas simples y eficientes para el análisis y la minería de datos [18]. A través de Scikit-Learn, se integraron diversos algoritmos de clasificación, realizando su comparativa mediante métricas de rendimiento que permitieron la determinación de la opción más óptima para el caso de estudio en cuestión.

Por último, se decidió utilizar MySQL como base de conocimiento para el almacenamiento de datos y control de registros. Punto fundamental para el uso de esta herramienta, fue su eficiente optimización de recursos, amplio soporte y fácil integración con otras herramientas de infraestructura del servidor, lo que permite una correcta gestión de la información.

3.1.4 Conjuntos de datos de ejercicios de trazabilidad para la detección de la Enfermedad de Parkinson

Debido a que el módulo trabaja en conjunto con un servicio para el modelo de clasificación de datos, se determinó la necesidad de identificar los conjuntos de datos que brindan mejor apoyo como modelos de entrenamiento y pruebas en la clasificación de ejercicios para la detección de la EP. A continuación, se enlistan los conjuntos seleccionados para su incorporación.

Parkinson's Drawing

Conjunto de datos de imágenes compuestas de trazos en espiral y ondas, correlacionadas en el análisis compuesto presión de pluma de esbozo y representado mediante 2 atributos principales, los cuales, se determinan por el valor de los ejes X y *Y*, el repositorio cuenta con muestras de un grupo de veintiocho pacientes sanos y veintisiete pacientes con padecimiento de EP evaluados en la Escala Unificada de Calificación de la EP. Además, durante las pruebas de implementación en las investigaciones realizadas, el conjunto de datos obtuvo en la métrica de precisión un valor promedio de 80%, sensibilidad de 66.67% y especificidad promedio de 93.33% [50], [51].

Parkinson Disease Spiral Drawings

Base de datos de escritura manual que consiste en 62 personas con Parkinson y quince individuos sanos distribuidos en un conjunto de datos de setenta y siete instancias y seis atributos definidos por: la precisión del trazo en los ejes X y Y, presión ejercida sobre el lienzo, ángulo de agarre, tiempo de ejecución e identificador de prueba; los resultados se clasifican en dos tipos de prueba (estática y dinámica), el

conjunto de datos de estudio se obtuvo mediante una tableta digital y analizado en distintas investigaciones. Del mismo modo, durante las pruebas de implementación en las investigaciones realizadas, se encontraron métricas de rendimiento promedio con valores de 77.50% en términos de precisión, 80.00% de sensibilidad y 75.00% en valores de especificidad [52].

En la Tabla 3.1 se presenta de forma comparativas los 2 conjuntos de datos analizados para su implementación en el proyecto.

Tabla 3.1 Conjuntos de datos de ejercicios de trazabilidad para la clasificación de la enfermedad de Parkinson

#	Conjunto de datos	Atributos	Precisión	Sensibilidad	Especificidad
1	Parkinson's Drawing	2	80.00%	66.67%	93.33%
2	Parkinson Disease Spiral Drawings	6	77.50%	80.00%	75.00%

3.1.5 Selección de Algoritmos de Clasificación

Como parte fundamental del proyecto, se realizó el estudio comparativo de nueve algoritmos de clasificación de AP. Para este proceso, se seleccionaron aquellos que, a través de la literatura, presentaron los mejores resultados de investigación. La prueba se realizó mediante un entorno preparado con un mismo conjunto de datos y métricas de estudio.

Los resultados de la comparación demostraron que el clasificador Random Forest (RF) obtuvo los mejores valores en términos de precisión y especificidad, con un puntaje promedio del 90.0% y 91.3% respectivamente en las pruebas de detección. El segundo clasificador con mejor desempeño fue Gradient Boosting (GB), con una puntuación de precisión del 81.8% y especificidad del 82.6%. En la Tabla 3.2 se detallan los valores obtenidos por cada uno de los clasificadores durante el análisis.

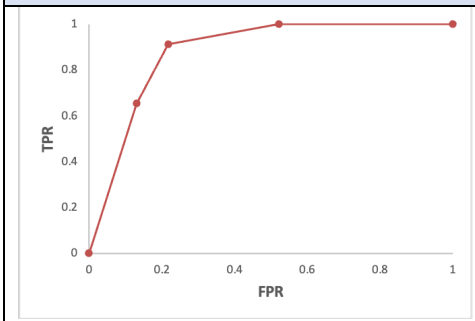
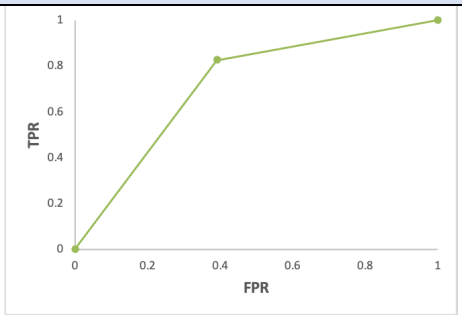
Tabla 3.2 Resultados comparativos de los algoritmos de clasificación

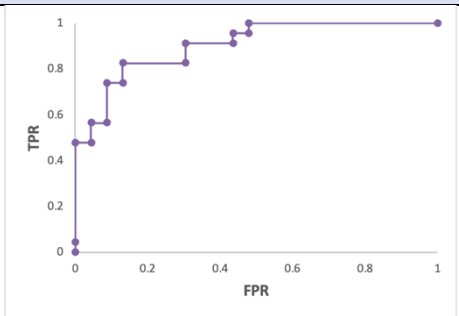
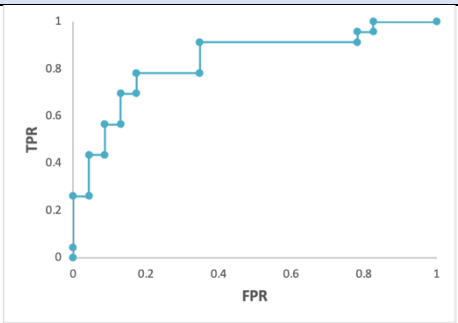
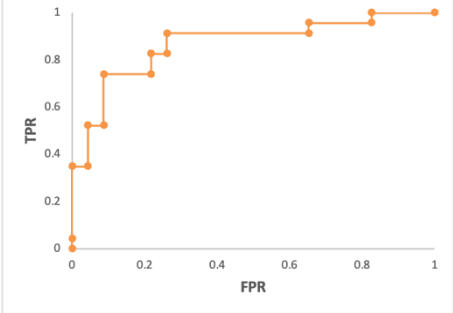
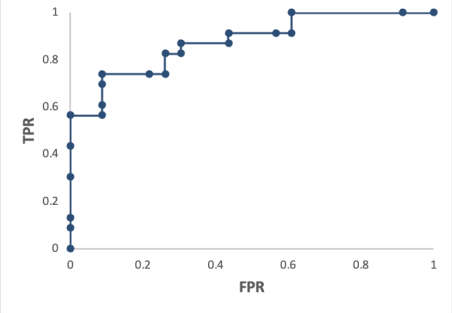
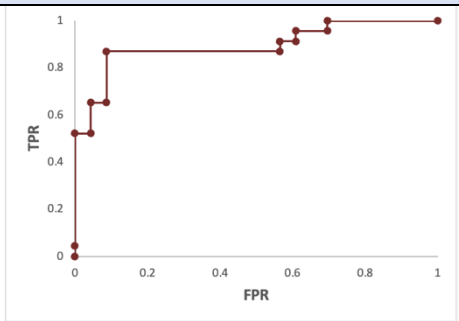
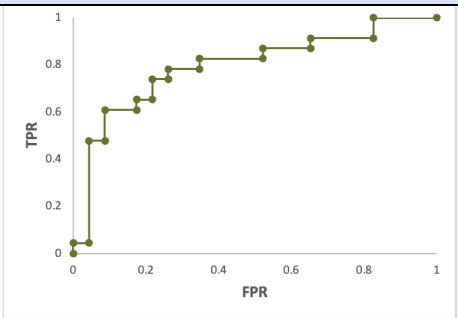
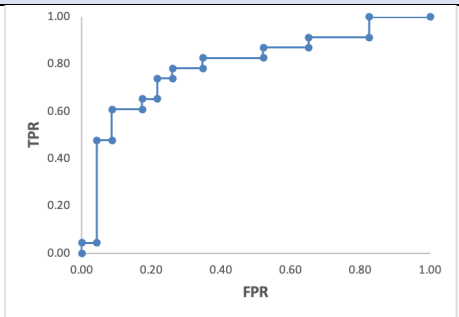
#	Algoritmo	Precisión	Sensibilidad	Especificidad	F1 Score
1	KNN	0.808	0.913	0.783	0.857
2	Decision Tree	0.679	0.826	0.609	0.745
3	SVM	0.792	0.826	0.783	0.809
4	GB	0.818	0.783	0.826	0.800
5	LightGBM	0.792	0.826	0.783	0.809
6	XGBoost	0.760	0.826	0.739	0.792
7	RF	0.909	0.870	0.913	0.889
8	CatBoost	0.739	0.739	0.739	0.739
9	AdaBoost	0.739	0.739	0.739	0.739

Los resultados de la comparación muestran que RF obtuvo los mejores puntajes de rendimiento en la determinación de los clasificadores, lo que lo convierte en el principal modelo a implementar. GB, por otro lado, obtuvo un buen desempeño en términos de precisión y se posiciona como el segundo algoritmo a implementar.

A su vez, se realizaron pruebas comparativas entre los clasificadores mediante el estudio de la curva ROC, la cual representa la relación entre la sensibilidad y especificidad de cada modelo. Los resultados obtenidos durante la prueba confirmaron los valores previos. La Tabla 3.3 muestra los resultados alcanzados durante el análisis.

Tabla 3.3 Resultados de la curva ROC

KNN	Decision Tree
	

SVM	Gradient Boosting
	
Light GBM	**KGBoost**
	
Random Forest	**CatBoost**
	
AdaBoost	
	

Durante el análisis de la curva ROC, se observa que RF continúa representando un mejor balance en el resultado obtenido, seguido por el algoritmo SVM y en tercer instancia GB. El algoritmo KNN, aunque mantiene valores óptimos de precisión y sensibilidad, no proporciona el mejor balance en análisis de la curva ROC y, por lo tanto, no se consideró para su implementación.

En conclusión, posterior a la comparación de los algoritmos de clasificación mediante diversas métricas de rendimiento, se determinó que RF, GB y SVM son los principales clasificadores a implementar en el módulo de detección temprana de la EP mediante técnicas de AP.

3.2 Diseño

En esta sección, se describe el diseño de la arquitectura, así como también, el fundamento teórico y gráfico para el control de la información, el modelo de persistencia y diseño de interfaces gráficas.

3.2.1 Arquitectura del módulo

Esta sección tiene como finalidad, proporcionar una vista general a la arquitectura del módulo para la detección temprana de la EP, expresando la interacción determinada entre el módulo web y el modelo de clasificación entrenado para la identificación de parámetros clave.

El flujo de trabajo comienza desde un cliente web que aloja el módulo en cuestión y permite su ingreso al personal médico en el trabajo de aplicación de pruebas para la EP, Este módulo cuenta con diversos apartados operativos, incluyendo la aplicación de nuevos estudios y el seguimiento de la evolución del paciente enfocado en el historial de pruebas realizadas. Durate la aplicación de nuevos estudios el módulo, en conjunto con las demás capas lógicas, recopila, procesa, interpreta y recupera los resultados obtenidos para su posterior seguimiento y monitorización.

El proceso de obtención de datos en el módulo, es guiado a través de una tableta digitalizadora, la cual permite al paciente realizar una serie de ejercicios de trazos

utilizando plantillas predefinidas. Los datos recolectados se envían a un servidor externo, donde el clasificador realiza las operaciones de procesamiento e interpretación correspondientes. Finalmente, los resultados guardan su registro en la base de conocimiento para su posterior consulta y seguimiento.

El módulo está distribuido en una arquitectura de capas para proporcionar una solución de mayor escalabilidad, robustez y de fácil mantenimiento. Durante el proceso de desarrollo se consideró el uso de diferentes tecnologías para mejorar el rendimiento de los procesos, al igual que la implementación de medidas de seguridad que garanticen la integridad de datos. La Figura 3.7 describe la arquitectura establecida para el desarrollo del módulo, del mismo modo, enumera las etiquetas (1-10) el flujo de trabajo principal “Clasificación de una prueba de trazabilidad”.

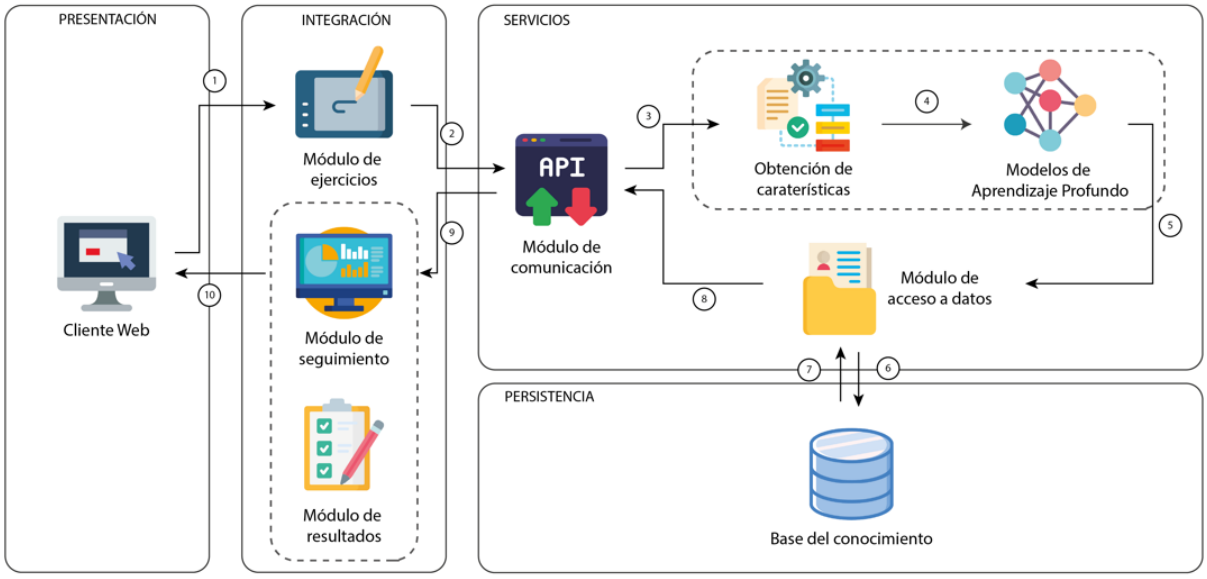


Figura 3.7 Arquitectura del módulo web para la detección temprana de la EP

A continuación, se describe cada una de las capas integradas en la arquitectura.

- **Capa de presentación:** es la interfaz gráfica principal de la aplicación web que permite al usuario (personal médico) recopilar y brindar seguimiento a las

pruebas realizadas. Esta capa está diseñada de manera óptima con el fin de mejorar la usabilidad y proporcionar una correcta experiencia de usuario.

- **Capa de integración:** conformada por diversos módulos encargados de llevar a cabo funciones específicas. El módulo de ejercicios tiene la tarea de recopilar mediante una tableta digitalizadora los trazos realizados por los pacientes durante la ejecución de la prueba, mientras que el módulo de seguimiento incorpora un conjunto de datos estadísticos relacionados con el historial de resultados globales de cada paciente. Por su parte, el módulo de resultados tiene como objetivo presentar de forma clara y concisa los resultados registrados por el clasificador. Finalmente, todos los datos interpretados por esta capa se envían a la capa de servicios a través de una única interfaz de acceso el “módulo de comunicación”.
- **Capa de servicios:** integrada por diversos módulos que definen los casos de uso para la predicción y consulta de estudios. En cuando al proceso de predicción, se establecen el módulo de obtención de características, enfocado en la lectura y análisis de instancias gráficas; y el módulo de AP, encargado de la interpretación y clasificación de resultados mediante algoritmos previamente entrenados para su uso. Por otro lado, se encuentra el módulo de acceso a datos, el cual proporciona una interfaz de acceso a la capa de persistencia para el almacenamiento de datos.
- **Capa de persistencia:** incorpora el sistema gestor de base de datos y la base de datos principal del módulo, encargados del control y almacenamiento de la información necesaria para su posterior análisis y seguimiento.

3.2.2 Modelo de persistencia

En cuanto al diseño e implementación del modelo de persistencia para la integración de los diferentes componentes del proyecto, se siguió una estrategia modular y escalable. Se optó por un diseño distribuido de manera lógica que permitiera una mayor flexibilidad y escalabilidad en el futuro, adaptándose a las necesidades cambiantes del proyecto.

Para lograr esta arquitectura, se dividió el modelo de base de datos en módulos lógicos independientes, cada uno con su propio esquema de contexto limitado, lo que permitió una mayor modularidad y reutilización del esquema en diversos propósitos del proyecto. Además, se utilizó una arquitectura de base de datos relacional, que permitió una mayor facilidad de consulta y manipulación de los datos.

En la Figura 3.8 se presenta el esquema relacional propuesto para el modelo de base de datos, que muestra la estructura de relaciones de entidades involucradas en el proyecto, este modelo asegura un diseño flexible con el fin de asegurar la escalabilidad del proyecto.

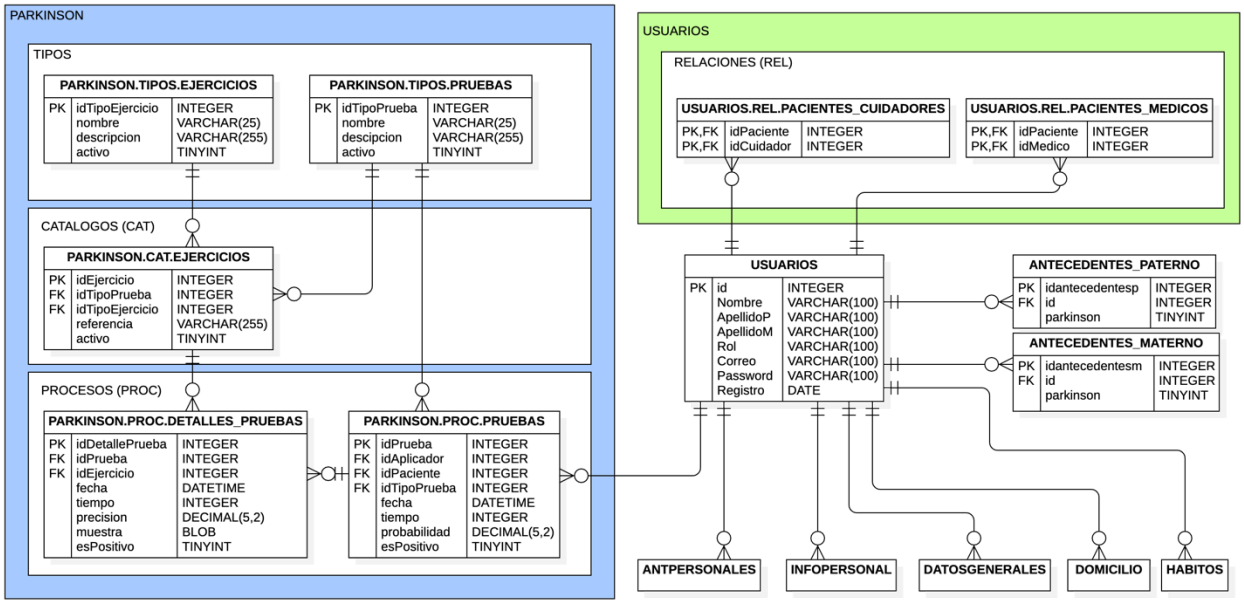


Figura 3.8 Esquema relacional de base de datos

3.2.2.1 Diccionario de datos

En esta sección, se proporciona el diccionario de datos que permitirá identificar y comprender todas las características incluidas en el modelo de persistencia. Este conjunto de información será de gran utilidad para el correcto manejo y gestión de los datos almacenados, asegurando así una correcta integración y funcionamiento del sistema en su totalidad.

- **Entidad:** PARKINSON.TIPOS.EJERCICIOS

Descripción: Entidad que almacena el conjunto de datos para la clasificación de ejercicios de Parkinson

Tabla 3.4 Diccionario de entidad "PARKINSON.TIPOS.EJERCICIOS"

Propiedad	Descripción	Tipo	Longitud	Clave	Valor
idTipoEjercicio	Identificador del registro	INTEGER	default	Primaria	-
nombre	Especifica el nombre del tipo de ejercicio	VARCHAR	25	-	-
descripcion	Especifica los detalles del tipo de ejercicio	VARCHAR	255	-	-
activo	Establece si el registro se encuentra activo	TINYINT	default	-	1

- **Entidad:** PARKINSON.TIPOS.PRUEBAS

  Descripción: Entidad que almacena el conjunto de datos para la clasificación de pruebas de Parkinson

Tabla 3.5 Diccionario de entidad "PARKINSON.TIPOS.PRUEBAS"

Propiedad	Descripción	Tipo	Longitud	Clave	Valor
idTipoPrueba	Identificador del registro	INTEGER	default	Primaria	-
nombre	Especifica el nombre del tipo de prueba	VARCHAR	25	-	-
descripcion	Especifica los detalles del tipo de prueba	VARCHAR	255	-	-
activo	Establece si el registro se encuentra activo	TINYINT	default	-	1

- **Entidad:** PARKINSON.CAT.EJERCICIOS

  Descripción: Entidad que almacena el cátalogo de ejercicios disponibles para si ejecución

Tabla 3.6 Diccionario de entidad "PARKINSON.CAT.EJERCICIOS"

Propiedad	Descripción	Tipo	Longitud	Clave	Valor
idEjercicio	Identificador del registro	INTEGER	default	Primaria	-
idTipoPrueba	Especifica el dato de referencia al tipo de prueba al que pertenece	INTEGER	default	Foránea	-
idTipoEjercicio	Especifica el dato de referencia al ripo de ejercicio al que pertenece	INTEGER	default	Foránea	-
referencia	Especifica la ruta al archivo multimedia de referencia al ejercicio	VARCHAR	255	-	-
activo	Establece si el registro se encuentra activo	TINYINT	default	-	1

- **Entidad:** PARKINSON.PROC.PRUEBAS

  Descripción: Entidad que almacena las pruebas realizadas por los usuarios

Tabla 3.7 Diccionario de entidad "PARKINSON.PROC.PRUEBAS"

Propiedad	Descripción	Tipo	Longitud	Clave	Valor
idPrueba	Identificador del registro	INTEGER	default	Primaria	-
idAplicador	Especifica el dato de referencia al identificador del personal que aplica la prueba	INTEGER	default	Foránea	-
idPaciente	Especifica el dato de referencia al identificador del paciente que realiza la prueba	INTEGER	default	Foránea	-
idTipoPrueba	Especifica el dato de referencia al identificador del tipo de prueba realizada	INTEGER	default	Foránea	-

Propiedad	Descripción	Tipo	Longitud	Clave	Valor
fecha	Especifica el dato de fecha y hora en que se realizó la prueba	DATETIME	default	-	-
tiempo	Especifica el tiempo en segundos que tomó realizar la prueba	INTEGER	default	-	-
probabilidad	Especifica el porcentaje de probabilidad global que resultó durante el análisis de la prueba	DECIMAL	5,2	-	-
esPositivo	Especifica si la prueba realizada tiende a positivo	TINYINT	default	-	-

- **Entidad:** PARKINSON.PROC.DETALLES_PRUEBAS

  Descripción: Entidad que almacena los detalles de las pruebas realizadas por los usuarios

Tabla 3.8 Diccionario de entidad "PARKINSON.PROC.DETALLES_PRUEBAS"

Propiedad	Descripción	Tipo	Longitud	Clave	Valor
idDetallePrueba	Identificador del registro	INTEGER	default	Primaria	-
idPrueba	Especifica el dato de referencia al identificador del ejericio realizado	INTEGER	default	Foranea	-
idEjercicio	Especifica el dato de referencia al identificador del ejercicio realizado	INTEGER	default	Foranea	-
fecha	Especifica el dato de fecha y hora en que se realizó la prueba	DATETIME	default	-	-

Propiedad	Descripción	Tipo	Longitud	Clave	Valor
tiempo	Especifica el tiempo en segundos que tomó realizar el ejercicio	INTEGER	default	-	-
precision	Especifica el porcentaje de precisión que resultó durante el análisis del ejercicio	DECIMAL	5,2	-	-
muestra	Especifica el archivo binario obtenido en la realización del ejercicio	BLOB	default	-	-
esPositivo	Especifica si el ejercicio realizado tiende a positivo	TINYINT	default	-	-

- **Entidad:** USUARIOS.REL.PACIENTES_CUIDADORES

  Descripción: Entidad que almacena las relaciones entre pacientes y cuidadores

Tabla 3.9 Diccionario de entidad "USUARIOS.REL.PACIENTES_CUIDADORES"

Propiedad	Descripción	Tipo	Longitud	Clave	Valor
idPaciente	Especifica el dato de referencia al identificador del usuario paciente	INTEGER	default	Primaria Foránea	-
idCuidador	Especifica el dato de referencia al identificador del usuario cuidador	INTEGER	default	Primaria Foránea	-

Tabla 3.10 Diccionario de entidad "USUARIOS.REL.PACIENTES_MEDICOS"

- **Entidad:** USUARIOS.REL.PACIENTES_MEDICOS

  Descripción: Entidad que almacena las relaciones entre pacientes y médicos

Propiedad	Descripción	Tipo	Longitud	Clave	Valor
idPaciente	Especifica el dato de referencia al identificador del usuario paciente	INTEGER	default	Primaria Foránea	-
idMedico	Especifica el dato de referencia al identificador del usuario médico	INTEGER	default	Foránea	-

3.2.3 Maquetación del módulo

A continuación, se presentan los modelos gráficos principales con el fin de determinar la definición de los aspectos fundamentales para el desarrollo del proyecto.

3.2.3.1 Componentes personalizados

En este apartado se presenta el diseño de los componentes personalizados, generados para la aplicación con el fin de garantizar la coherencia visual, usabilidad de la interfaz de usuario y reutilización de elementos gráficos para lograr una mayor eficacia durante el proceso de desarrollo.

Encabezado (Figura 3.9**):** Este componente representa una barra superior, presente en cada sección del módulo y con la finalidad de orientar al usuario en la aplicación proporcionando información general sobre la sección activa.

Figura 3.9 Mockup - Componente - Encabezado

Menú lateral (Figura 3.10**):** Este componente representa un panel lateral que proporciona el acceso a las diferentes secciones del módulo.

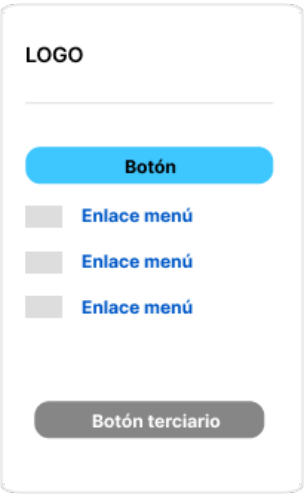


Figura 3.10 Mockup - Componente - Menú lateral

Tarjeta de concepto (Figura 3.11**):** Este componente representa en formato de tarjeta, información rápida sobre un elemento determinado. Es una forma útil de proporcionar una vista previa de los elementos que se encuentran en la aplicación.

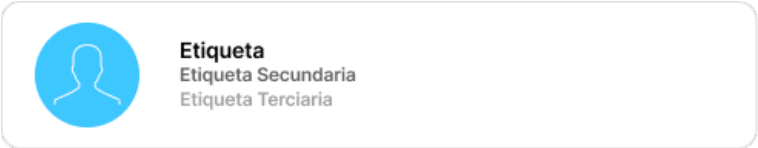


Figura 3.11 Mockup - Componente - Tarjeta de concepto

Tarjeta estadística (Figura 3.12**):** Este componente representa en formato de tarjeta, información estadística sobre un elemento determinado. Es una forma rápida y sencilla de visualizar resultados clave de un dato en el registro.

Figura 3.12 Mockup - Componente - Tarjeta estadística

Tarjeta de gráfico (Figura 3.13**):** Este componente representa en formato de tarjeta, un gráfico estadístico completo que permite al usuario visualizar a detalle un conjunto de registros almacenados para su posterior análisis.

Figura 3.13 Mockup - Componente - Tarjeta de gráfico

Tarjeta de gráfico compacto (Figura 3.14**):** Similar a la “tarjeta de gráfico”, este componente representa en formato de tarjeta, un gráfico estadístico de menor tamaño, útil en la comparación de pequeños conjuntos de datos para su posterior análisis.

Figura 3.14 Mockup - Componente - Tarjeta de gráfico compacto

Tarjeta de datos (Figura 3.15**):** Este componente representa complejos conjuntos de datos de la aplicación en un formato de tabla. Es una opción efectiva para mostrar al usuario datos estructurados y ordenados mediante columnas para su análisis detallado.

Figura 3.15 Mockup - Componente - Tarjeta de datos

3.2.3.2 Inicio de sesión

En la Figura 3.16 se presenta el modelo de acceso inicial al módulo mediante el ingreso por credenciales de cuenta (usuario y contraseña), así mismo, se plantea la opción de permitir restablecer los datos de acceso mediante el correo electrónico registrado.

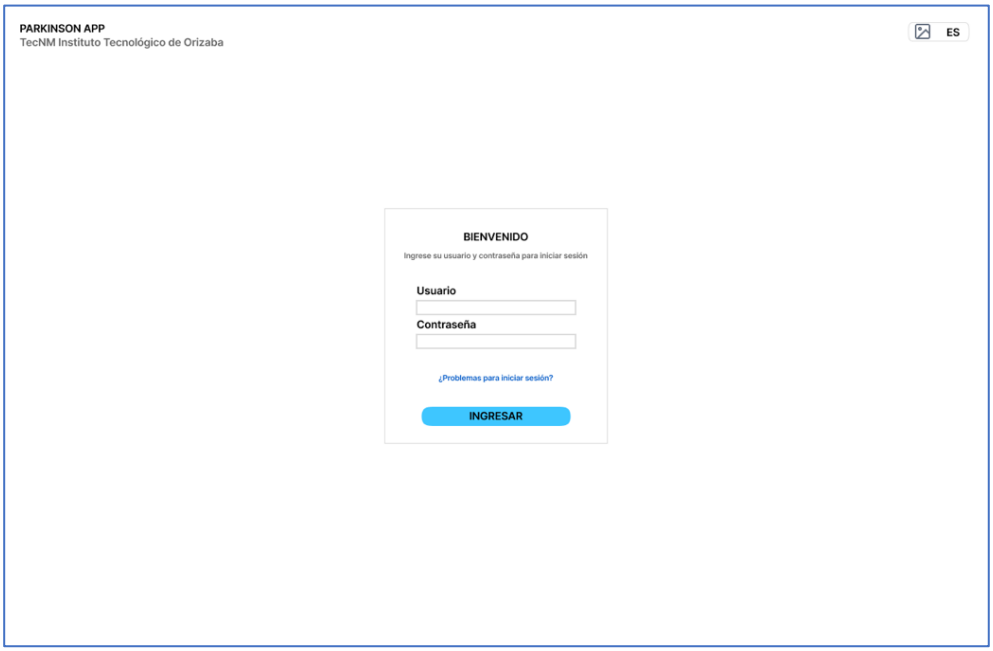


Figura 3.16 Mockup - Pantalla de inicio de sesión

3.2.3.3 Panel estadístico

Enfocado en el diseño presentado en Figura 3.17, Figura 3.18 y Figura 3.19 para las pantallas de “Panel principal”, “Expediente del paciente” y “Detalles de la prueba”, se presenta una interfaz estadística que muestre datos como:

- **Tarjeta informativa**– Muestra información general de la sección, en el caso del panel principal y del paciente, se muestran datos personales del individuo, mientras que, para detalles de las pruebas, se presentan datos de identificación del registro.
- **Tarjeta de conteo** – Muestra información global sobre el registro mostrado, por ejemplo: número de pacientes en atención, porcentaje global de casos, pruebas realizadas, entre otros.
- **Tarjeta de probabilidad** – Muestra un porcentaje global de clasificación indicando la probabilidad del padecimiento con base en todos los estudios realizados.
- **Gráfico histórico –** Muestra un seguimiento histórico del dato presentado con la finalidad de comparar el progreso en los estudios registrados.
- **Gráfico de resultados –** Muestra un comparativo relacional entre clasificadores de información, tales como, número de pacientes por género, por edad o bien, resultados positivos y negativos de una prueba.
- **Tabla de datos –** Muestra múltiples registros realzados en formato de tabla y se muestra un acceso rápido a la información reciente desde el mismo elemento.

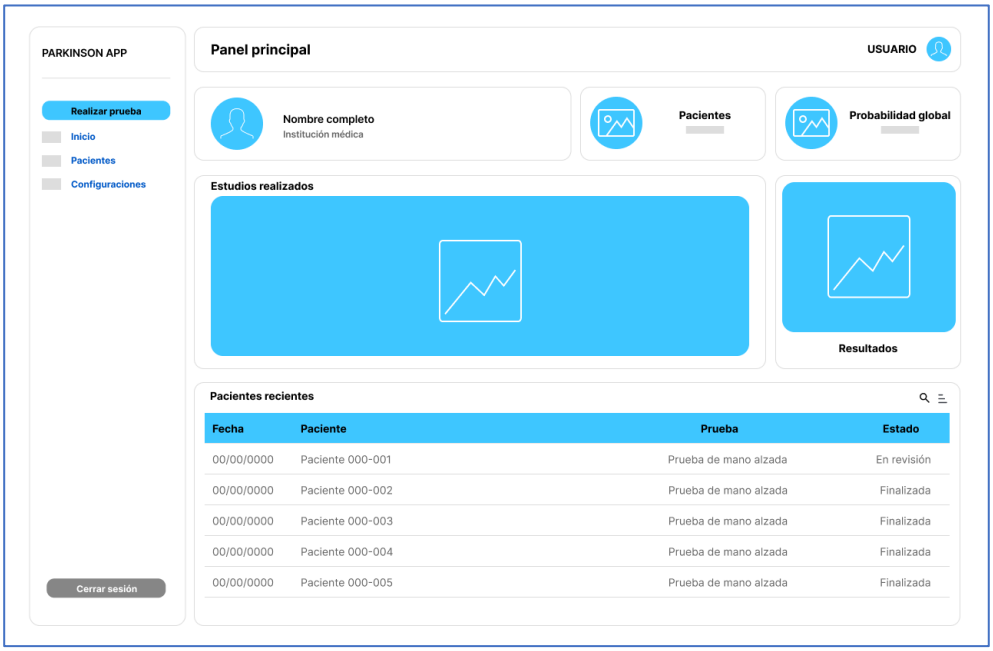


Figura 3.17 Mockup - Pantalla de panel estadístico (Principal)

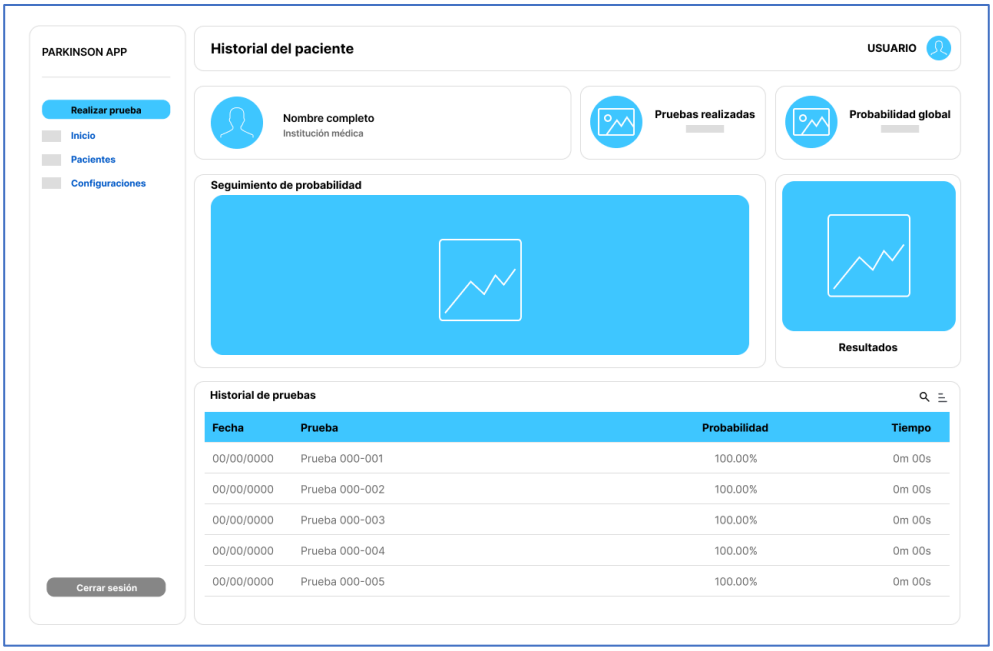


Figura 3.18 Mockup - Pantalla de panel estadístico (Historial del paciente)

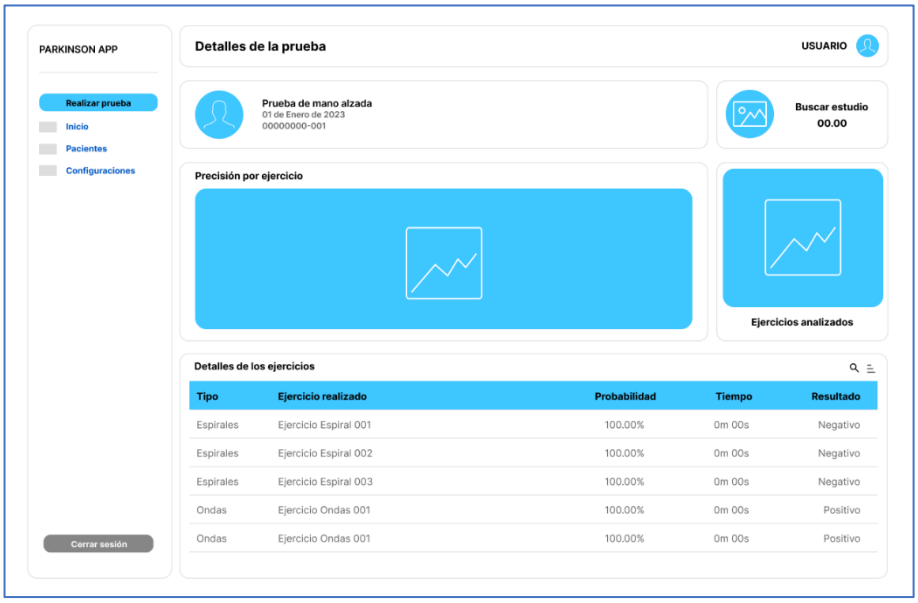


Figura 3.19 Mockup - Pantalla de panel estadístico (Detalles de la prueba)

3.2.3.4 Visualizar ejercicio

En la Figura 3.20 se muestra la pantalla propuesta para la visualización de ejercicios, la cual, provee los detalles de un ejercicio determinado durante la aplicación y análisis de pruebas, la ventana muestra entre, otros datos, identificador del registro, métricas de análisis y el registro realizado por el paciente

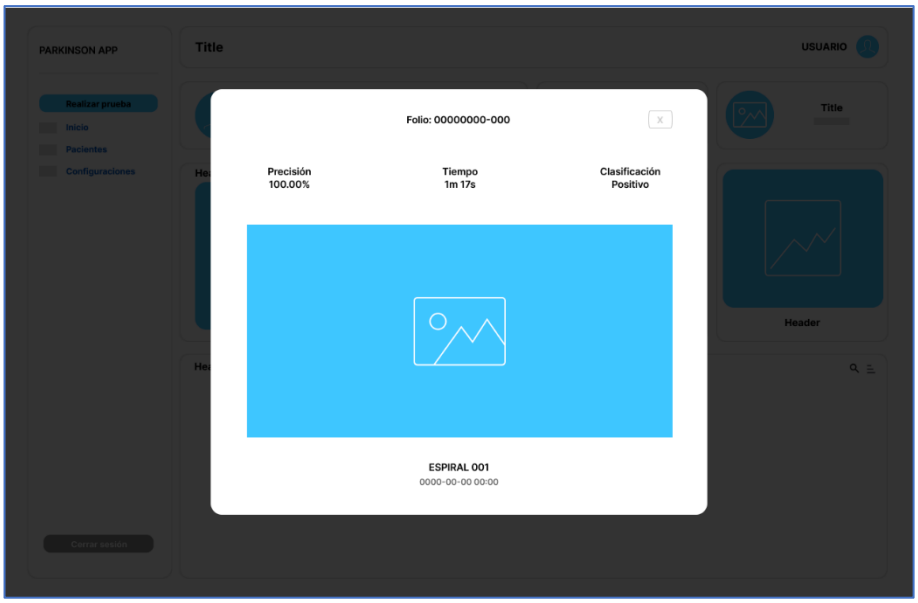


Figura 3.20 Mockup - Pantalla visualizar ejercicio

3.2.3.5 Panel del perfil

A través del diseño mostrado en la Figura 3.21, el módulo presenta al usuario un formulario para administrar sus datos personales y de navegación, las secciones contempladas para este apartado son las siguientes:

- **Datos del usuario –** Datos personales del usuario, nombre, fecha de nacimiento e institución médica afiliada.
- **Datos de contacto –** Medios de comunicación del usuario, tales como, correo electrónico (utilizado para iniciar sesión) y número de teléfono.
- **Preferencias de idioma –** Establecer las preferencias de idioma para visualizar el sistema (Contemplado: español e inglés).
- **Contraseña –** Para acceder al sistema, permite modificar las credenciales de acceso (Se solicitará confirmación antes de guardar el cambio).

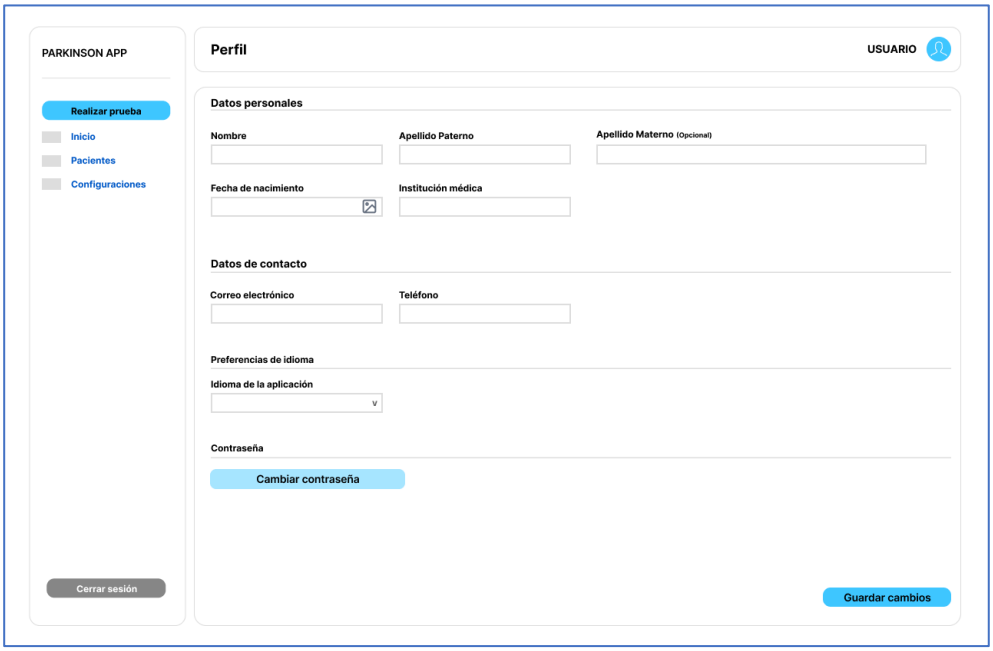


Figura 3.21 Mockup - Pantalla panel del perfil

3.2.3.6 Catálogo de pruebas

Provee al usuario un menú de distintas pruebas disponibles para su análisis, donde se completa una serie de tareas asignadas en pantalla acorde con el modelo de prueba seleccionado, entre ellas: pruebas de mano alzada, pruebas del lenguaje natural o pruebas de movimiento, el diseño para este apartado se muestra en la Figura 3.22

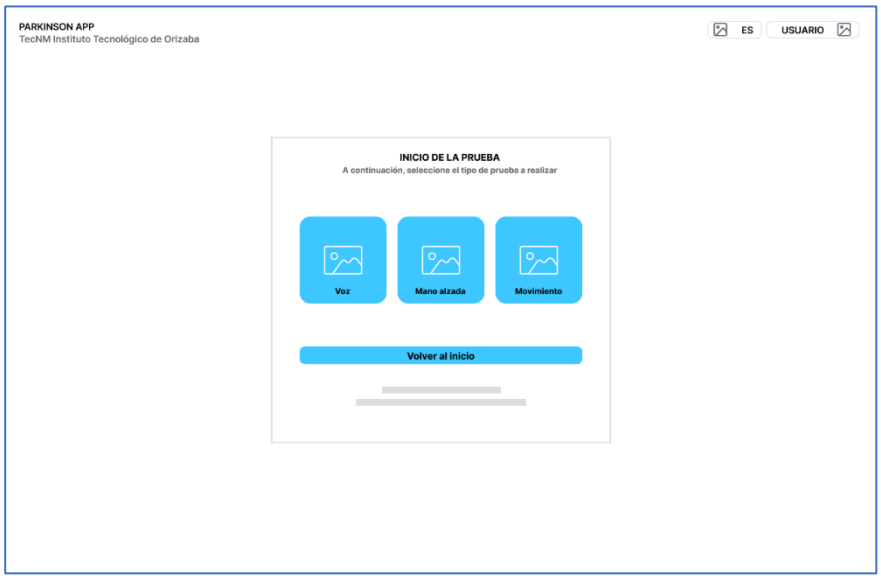


Figura 3.22 Mockup - Pantalla catálogo de pruebas

3.2.3.7 Aplicación de la prueba

Tal como se muestra en el diseño de la Figura 3.23, una vez seleccionado el tipo de prueba a realizar, se presenta al usuario un lienzo en pantalla para la ejecución de los ejercicios presentados, para cada prueba se considera un conjunto de ejercicios de distinta categoría, por ejemplo: ondas, picos y espirares; entre los cuales se muestran secuencialmente hasta finalizar la prueba.

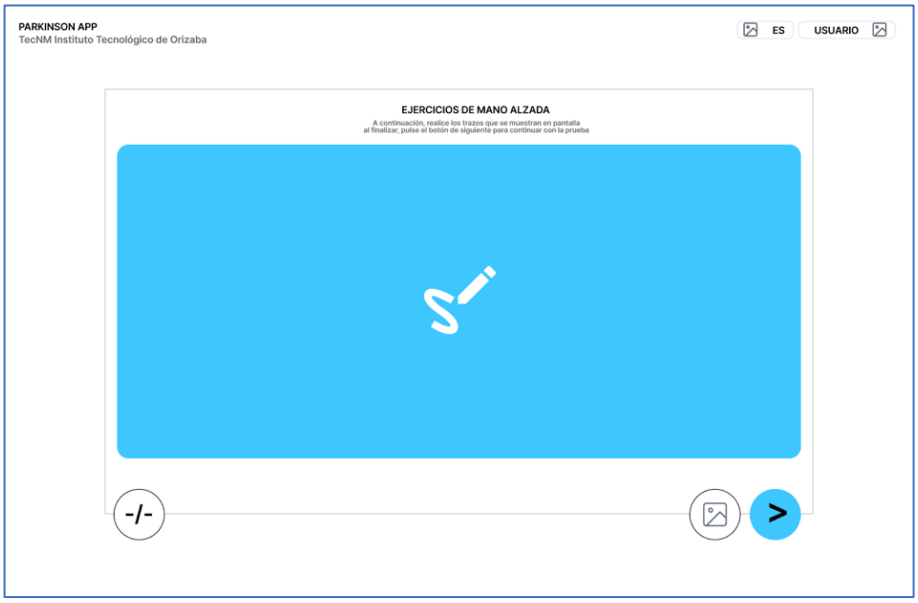


Figura 3.23 Mockup - Pantalla de aplicación de la prueba

3.2.3.8 Resultados de la prueba

Presenta los resultados correspondientes a la prueba, proporcionando al usuario un número de folio de seguimiento para su posterior consulta, los resultados se muestran disponibles en el panel principal o expediente del paciente, una vez procesados por el servidor de AP y registrados en el modelo de persistencia.

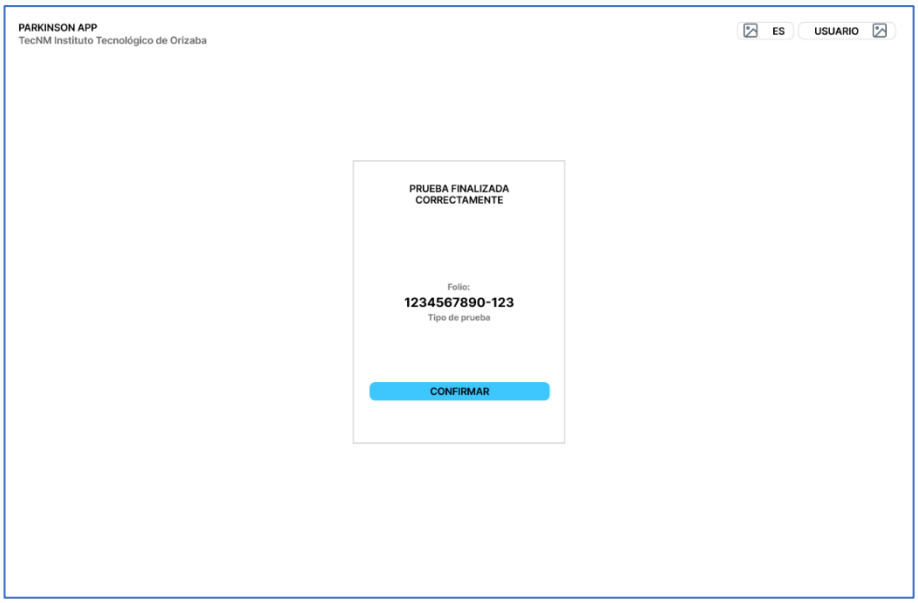


Figura 3.24 Mockup - Pantalla de resultados

3.3 Desarrollo

3.3.1 Implementación de los algoritmos de clasificación

En esta sección se describe el proceso de implementación para los algoritmos de clasificación seleccionados a través de las tecnologías propuestas.

Como punto de partida, se procedió a configurar, entrenar y empaquetar cada uno de los algoritmos seleccionados utilizando la biblioteca *Sickit-Learn*. Como resultado de este proceso, se logró generar los paquetes *.joblib* correspondientes a los modelos RF, SVM y GB, los cuales se encuentran listos para su incorporación como procesos preconfigurados y optimizados para su uso. Este enfoque permite acelerar la implementación de los algoritmos y reducir la complejidad de la configuración del sistema, ya que se cuenta con los modelos ya entrenados y empaquetados previamente lo que contribuyó a mejorar la eficiencia en la implementación del sistema, evitando tener que entrenar los modelos en cada implementación del sistema en un nuevo entorno. El Listado 3.1 presenta el código de implementación para el proceso de entrenamiento de uno de los modelos de clasificación.

Listado 3.1 Código de implementación para el proceso de entrenamiento el modelo de clasificación Random Forest

```
1   def train_model(dataset_path):
2     classifier = RandomForestClassifier(random_state=1)
3     path = dataset_path
4     trainingPath = os.path.sep.join([path, "training"])
5     testingPath = os.path.sep.join([path, "testing"])
6     (trainX, trainY) = load_split(trainingPath)
7     (testX, testY) = load_split(testingPath)
8     le = LabelEncoder()
9     trainY = le.fit_transform(trainY)
10    testY = le.transform(testY)
11    classifier.fit(trainX, trainY)
12    predictions = classifier.predict(testX)
13    cm = confusion_matrix(testY, predictions).ravel()
14    tn, fp, fn, tp = cm
15    accuracy = (tp + tn) / float(cm.sum())
16    sensitivity = tp / float(tp + fn)
17    specificity = tn / float(tn + fp)
18    return classifier, testX, testY, accuracy, sensitivity, specificity
```

Posteriormente, se procedió a implementar una arquitectura *REST* para el servicio *RESTful API* utilizando el marco de trabajo *Flask*, lo que permitió la creación de una interfaz de comunicación estandarizada entre las diversas tecnologías utilizadas en el proyecto. Este submódulo albergó los paquetes *.joblib* previamente generados y accesibles a través de una única interfaz, como una fachada para simplificar su uso.

Así mismo, se estableció un *endpoint* denominado *"/predict"* de tipo *POST*, que recibe como argumentos el nombre del clasificador que se desea utilizar y la imagen a procesar. El Listado 3.2 muestra el código de implementación realizado durante la generación del *endpoint* mencionado.

Listado 3.2 Código de implementación para endpoint POST /predict

```
1   @app.route('/predict', methods=['POST'])
2   def predict():
3       classifier = request.form['classifier']
4       image = request.files['image'].read()
5       features = process_image(image)
6       model = getModel(str(classifier))
7       if(model == False):
8           return jsonify('El clasificador [' + classifier + '] no se encuentra
    registrado'), 400, {'Content-Type': 'application/json'}
10      else:
11          prediction = model.predict_proba([features])
12          result = {'model': classifier, 'result': {'healthy': prediction[0][0],
    'parkinson': prediction[0][1]}}
14          return jsonify(result), 200, {'Content-Type': 'application/json'}
```

El flujo de trabajo comienza con el procesamiento de la instancia, seguido de la detección y extracción de características. Una vez obtenidos los parámetros necesarios de la instancia, se realiza el proceso de clasificación correspondiente al clasificador seleccionado. Finalmente, el proceso culmina al generar una respuesta en formato JSON con el valor numérico obtenido. En la Figura 3.25 se representa de manera general el flujo de trabajo implementado durante el proceso de clasificación de los ejercicios de trazabilidad.

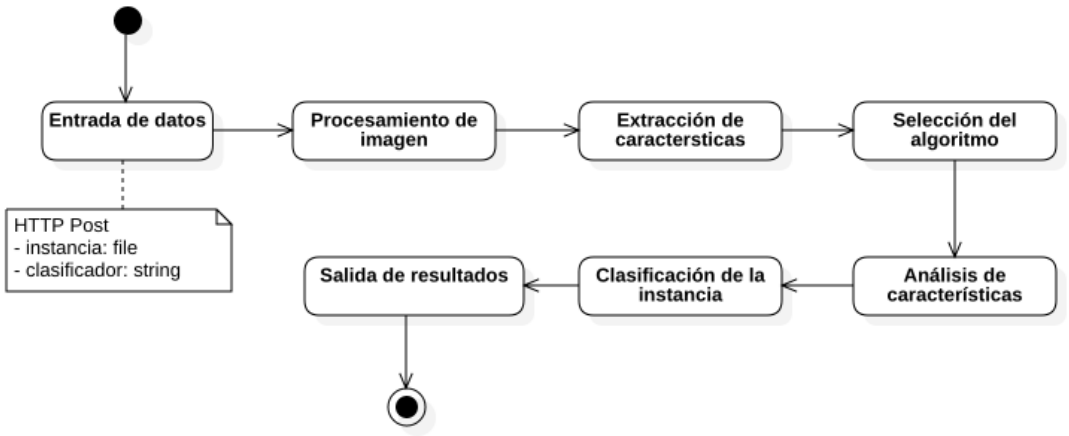


Figura 3.25 Flujo operacional para la clasificación de ejercicios

Finalmente, para garantizar una solución de configuración, despliegue, y fácil implementación en un servidor remoto, se decidió empaquetar el servicio en un contenedor. Con ello, se logró encapsular cada una de las dependencias necesarias, mejorando así, la portabilidad del servicio y simplificando el proceso de migración a otros equipos.

3.3.2 Implementación del módulo

En esta sección se presenta la interfaz de usuario del módulo web, desarrollada con base en proporcionar una experiencia de usuario intuitiva y sencilla en la aplicación del módulo de clasificación. La interfaz ofrece una serie de funcionalidades, incluyendo la aplicación de pruebas, la visualización del resultado de la clasificación y el seguimiento del historial del paciente. Además, se implementaron técnicas de validación de componentes para garantizar la calidad e integridad de los datos recibidos y procesados en el servidor.

La Figura 3.26 representa la ventana de acceso principal al módulo, destacando principalmente por la solicitud de credenciales de acceso al módulo, así como también opciones de recuperación de contraseña.

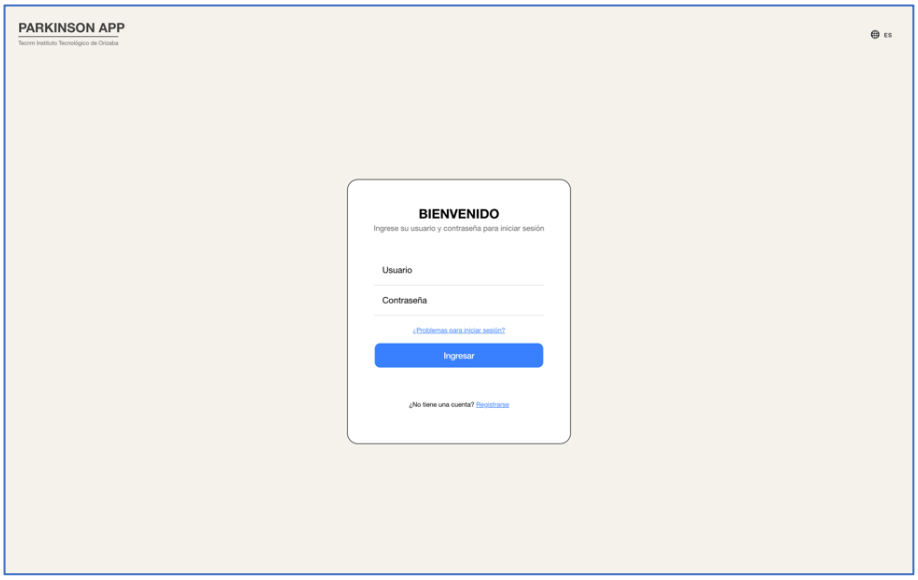


Figura 3.26 Pantalla de inicio de sesión

En la Figura 3.27 se muestra la pantalla principal de la aplicación, considerando como principal funcionalidad, representar información estadística sobre los casos analizados por el médico que ha ingresado, se cuenta con diversos datos informativos, tales como: número de pruebas realizadas, porcentaje de probabilidad global del padecimiento, historial de resultados y su clasificación por género, así como también, acceso rápido a las últimas pruebas ejecutadas.

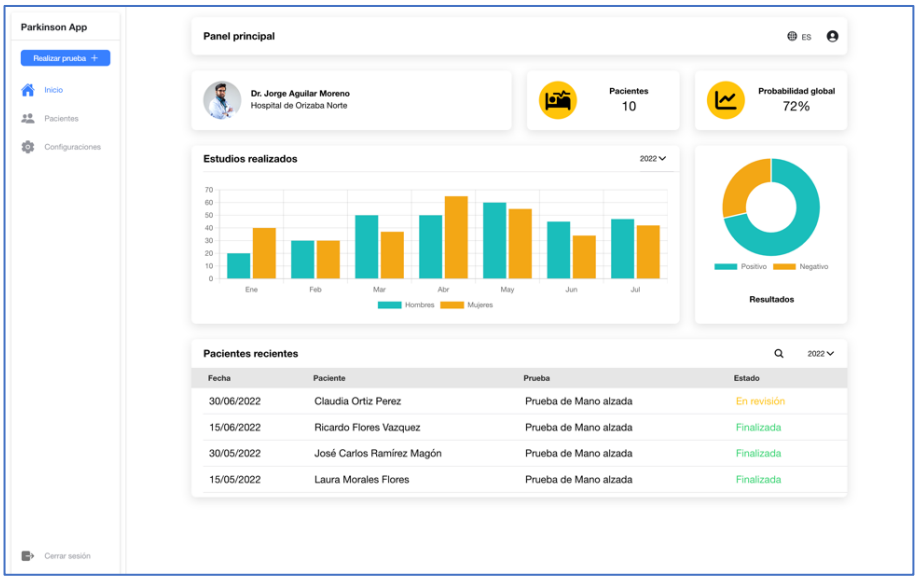


Figura 3.27 Pantalla de panel principal

Para la revisión individual, se muestra en la Figura 3.28 el panel estadístico enfocado al registro y seguimiento de cada paciente, los controles que destacan a través de esta pantalla son: el historial de pruebas realizadas, resultados globales del paciente y seguimiento por resultado, por mencionar algunos.

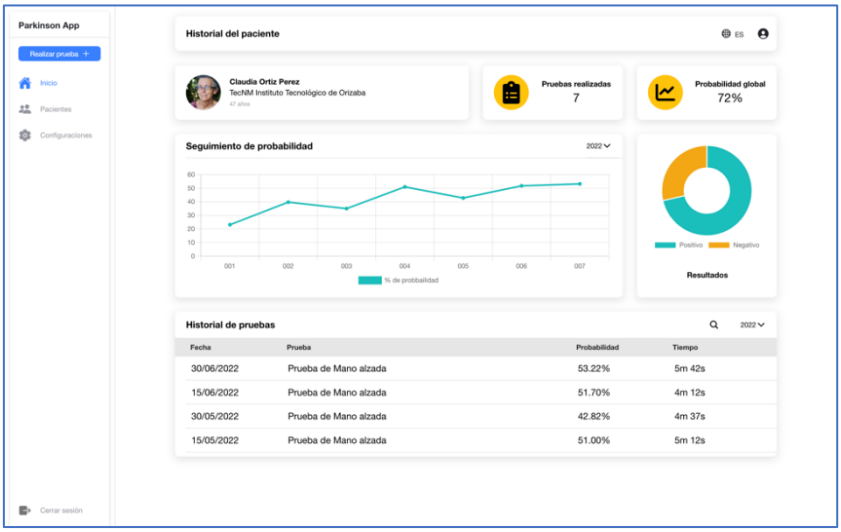


Figura 3.28 Pantalla de historial del paciente

Seguido, en la pantalla de la Figura 3.29, el usuario cuenta con la posibilidad de realizar la consulta detallada de cada prueba generada por el paciente y visualizar, entre los datos mostrados, el resultado obtenido por cada uno de los ejercicios aplicados, su nivel de precisión y tiempo de ejecución.

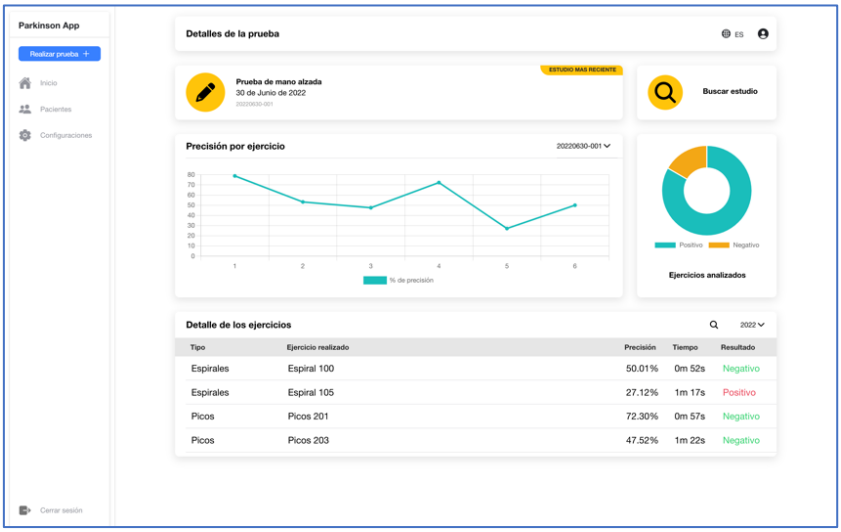


Figura 3.29 Pantalla de detalles de la prueba

Para consultar el ejercicio realizado y sus métricas obtenidas durante la clasificación, se proporciona la ventana de la Figura 3.30, que representa visualmente cada dato obtenido durante el ejercicio, tal como, precisión obtenida, tiempo, resultado de clasificación y muestra del trazo analizado.

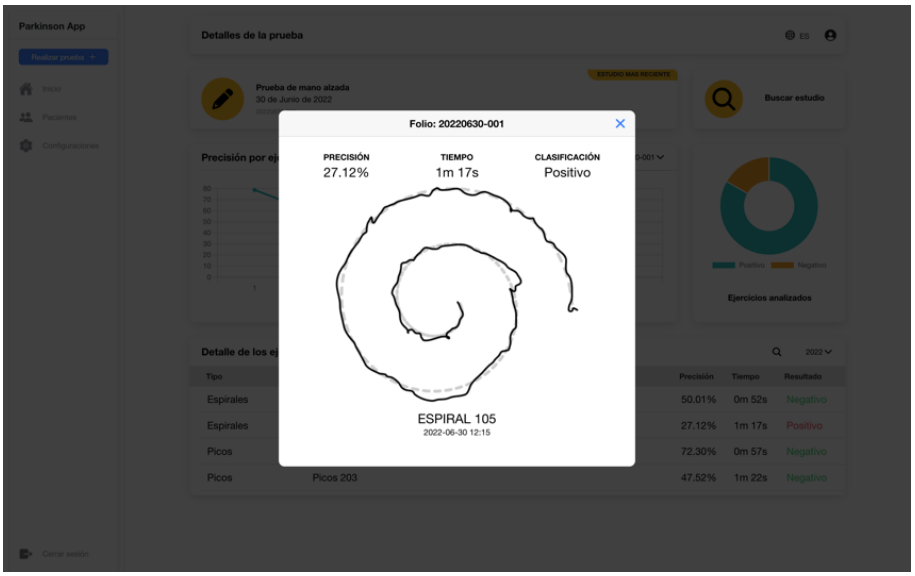


Figura 3.30 Pantalla de detalles del ejercicio

Enfocado en la aplicación de pruebas para la detección de la EP, se desarrolló la interfaz que apertura la realización de pruebas mediante distintos métodos, la pantalla del catálogo de pruebas (Figura 3.31). Además de ofrecer pruebas convencionales, se ha incluido la posibilidad de realizar pruebas de voz y movimiento, ampliando así el alcance de la aplicación y permitiendo una evaluación más completa del paciente.

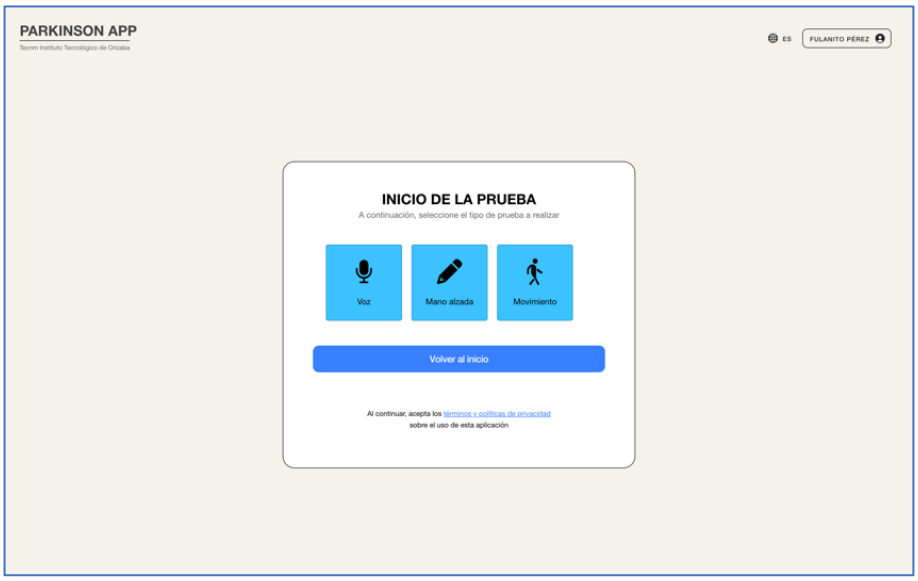


Figura 3.31 Pantalla de catálogo de pruebas

En la Figura 3.32, el usuario cuenta con un lienzo de trabajo para realizar con asistencia de un tercero, el ejercicio mostrado en pantalla, los controles de seguimiento en pantalla son: limpiar lienzo, siguiente ejercicio y finalizar la prueba, este último habilitado únicamente durante el último ejercicio.

Figura 3.32 Pantalla de aplicación de la prueba

En la conclusión de la prueba, se asigna un identificador de registro para su seguimiento, tal como se muestra en la Figura 3.33, la prueba realiza su procesamiento en el servidor y registra los resultados para su posterior consulta.

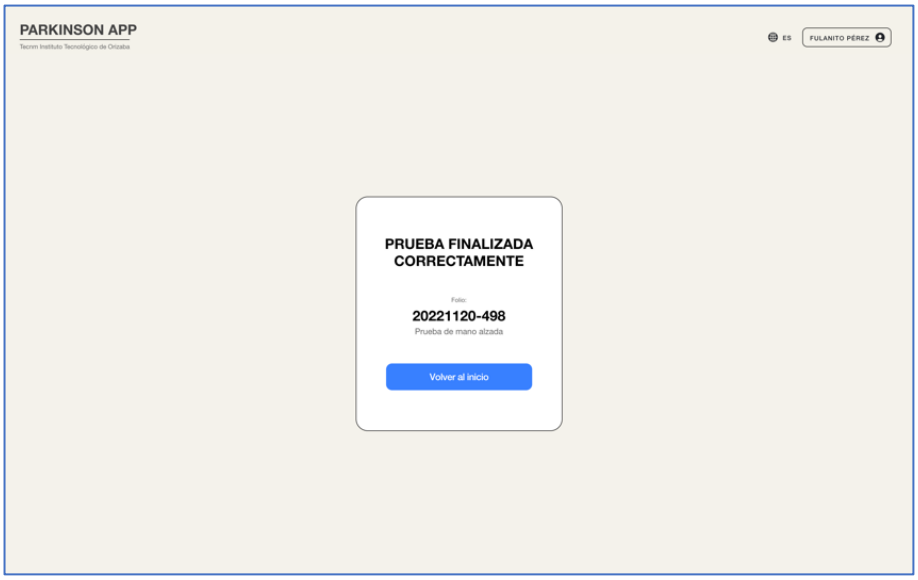


Figura 3.33 Pantalla de conclusión de la prueba

Finalmente, se proporciona también, una pantalla de configuración del perfil, como se observa en la Figura 3.34, el usuario cuenta con la posibilidad de visualizar su información personal, datos de contacto, modificar su contraseña de acceso, al igual que, ajustar el idioma de preferencia en el módulo de la plataforma.

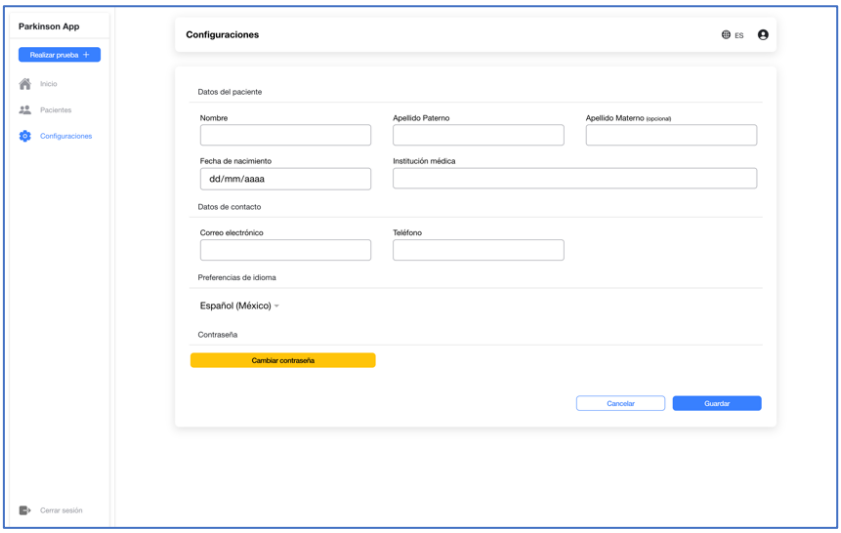


Figura 3.34 Pantalla de panel del perfil

Capítulo 4. Resultados

En este capítulo se describen los casos de estudios realizados con el propósito de validar la solución propuesta en el proyecto. Los casos analizados evaluaron el funcionamiento y desempeño del módulo web, así como también el módulo de clasificación mediante técnicas de AP y los resultados obtenidos.

4.1 Casos de estudio: Identificación temprana de síntomas de la Enfermedad de Parkinson

En el presente apartado se expone un ejemplo práctico que tiene como finalidad comprobar la eficacia del módulo y establecerlo como una herramienta en línea que contribuya en el proceso de la identificación temprana de la EP, además de aplicar las recomendaciones médicas elaboradas por los profesionales en la materia.

Como punto de partida se realizó un estudio con tres pacientes previamente verificados clínicamente por personal médico, los cuales, contribuyeron en el análisis y seguimiento de la EP a través de la aplicación web. Este proceso se realizó durante un lapso de dos meses, comprendido del 10 de enero del 2023 al 10 de marzo de 2023. Cada uno de los pacientes contó con su registro en el sistema y aplicación de las diferentes pruebas de trazabilidad implementadas. A continuación, se detalla el contexto estudiado por cada paciente

4.1.1 Caso de estudio 1: Paciente con diagnostico positivo

El primer paciente, una persona de 56 años del género masculino, quien presenta síntomas notorios de la enfermedad tales como dificultad motriz, el diagnóstico clínico presentado es positivo a la EP.

4.1.2 Caso de estudio 2: Paciente con diagnostico negativo

El segundo paciente, una persona de 51 años del género femenino, quien no presenta síntomas al momento del estudio, sin embargo, se mantiene en sospecha de padecimiento, por lo que cuenta con tratamientos en progreso por parte del centro de salud, su diagnóstico clínico a la EP es actualmente negativo.

4.1.3 Caso de estudio 3: Paciente con diagnostico negativo

El tercer paciente, corresponde a una persona de 54 años del género masculino, sin la presencia de síntomas o sospechas de padecimiento, así como también verificado por diversos estudios realizados en el centro de salud, el cual, determinó un diagnóstico negativo a la EP.

4.1.4 Datos iniciales

En el estudio, se solicitó a los usuarios. que, con apoyo del personal médico, interactuaran en la aplicación web a través del submódulo de pruebas de trazabilidad a lo largo del periodo de evaluación. Este proceso se realizó con el fin de obtener información específica sobre las variaciones de trazos de cada uno de los pacientes corroborando los resultados del clasificador.

Durante el proceso, cada semana se aplicó a los pacientes un examen de trazabilidad completo. Esto permitió al sistema obtener datos precisos de seguimiento en el comportamiento de cada uno de los pacientes, permitiendo al médico detectar patrones y tendencias en la información recopilada, así como también, verificar los resultados previstos de cada paciente.

A continuación, se presentan algunas de las muestras obtenidas por los pacientes durante la evaluación:

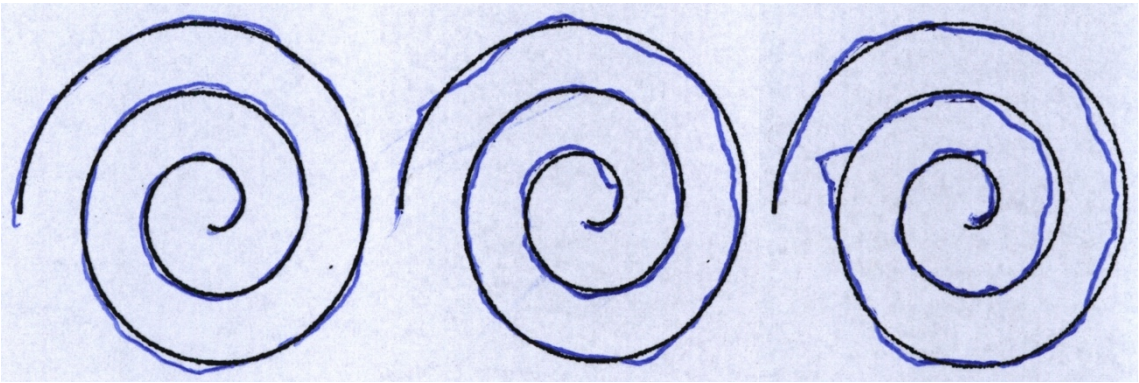


Figura 4.1 Muestras de trazabilidad del caso de estudio 1 - Positivo

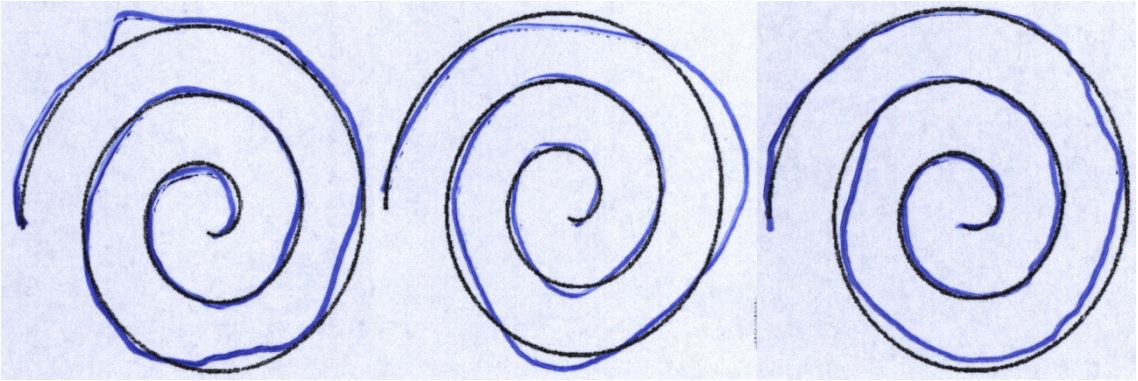


Figura 4.2 Muestras de trazabilidad del caso de estudio 2 – Negativo

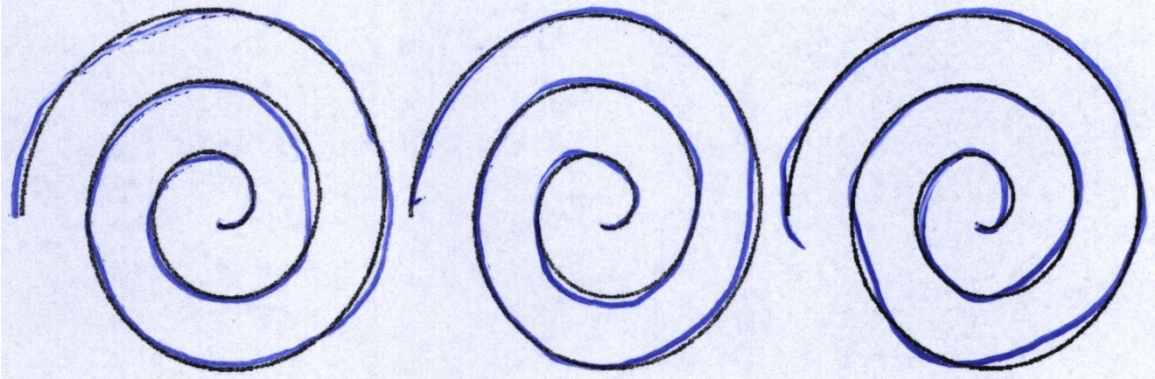


Figura 4.3 Muestras de trazabilidad del caso de estudio 3 - Negativo

4.1.5 Clasificación automática de datos

Basado en los resultados estadísticos obtenidos por el componente de clasificación de la EP en conjunto con los datos recopilados de la aplicación web y la tableta digitalizadora. Se aplicaron las reglas previamente diseñadas para la detección de la EP en conjunto con los valores asociados en la Tabla 4.1.

Atributo: “Probabilidad de Parkinson”

Tabla 4.1 Valores de referencia para la identificación de Parkinson

Tipo de dato	Rango	Resultado	Descripción
Numérico	0.00 – 0.40	Poco Probable	Los resultados demostraron una baja probabilidad de padecimiento
Numérico	0.41 – 0.80	Probable	Los resultados demostraron probabilidad de padecimiento
Numérico	0.81 – 1.00	Altamente Probable	Los resultados demostraron una alta probabilidad de padecimiento

4.1.6 Resultados

En la Figura 4.4 se presenta el panel de uno de los pacientes participantes en el caso de estudio. Para acceder al módulo de ejercicios, es necesario hacer clic en el botón “Realizar prueba” y posteriormente confirmar el paciente con el que se encuentra trabajando el personal médico.

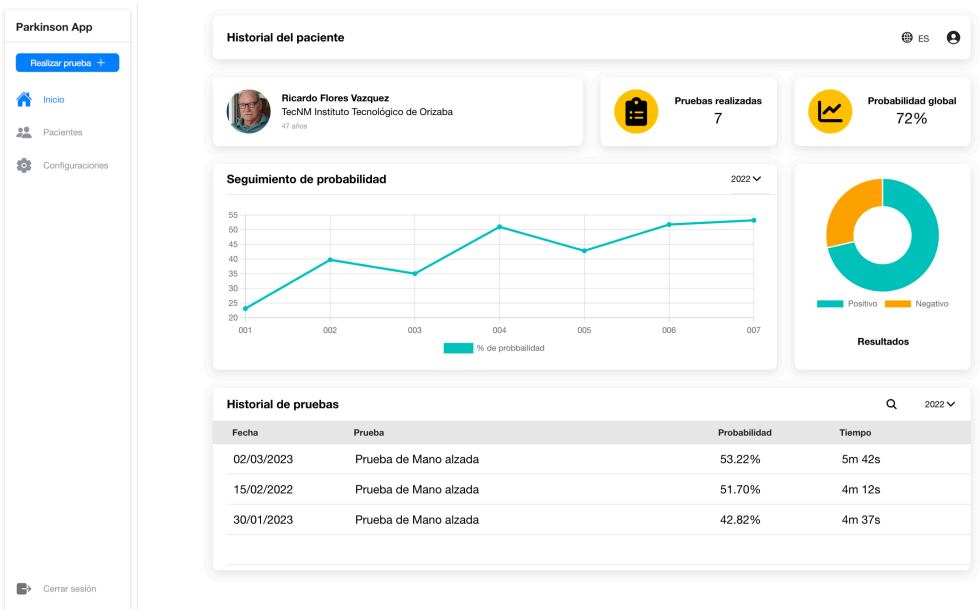


Figura 4.4 Módulo web - Panel del expediente del paciente

Una vez realizada la confirmación del paciente, se muestra en pantalla el catálogo de pruebas posibles para realizar un diagnóstico de clasificación, para el caso de estudio

de estudio actual, se seleccionó el proceso de ejercicios de trazabilidad, correspondiente al botón “Mano alzada”, tal como se muestra en la Figura 4.5 .

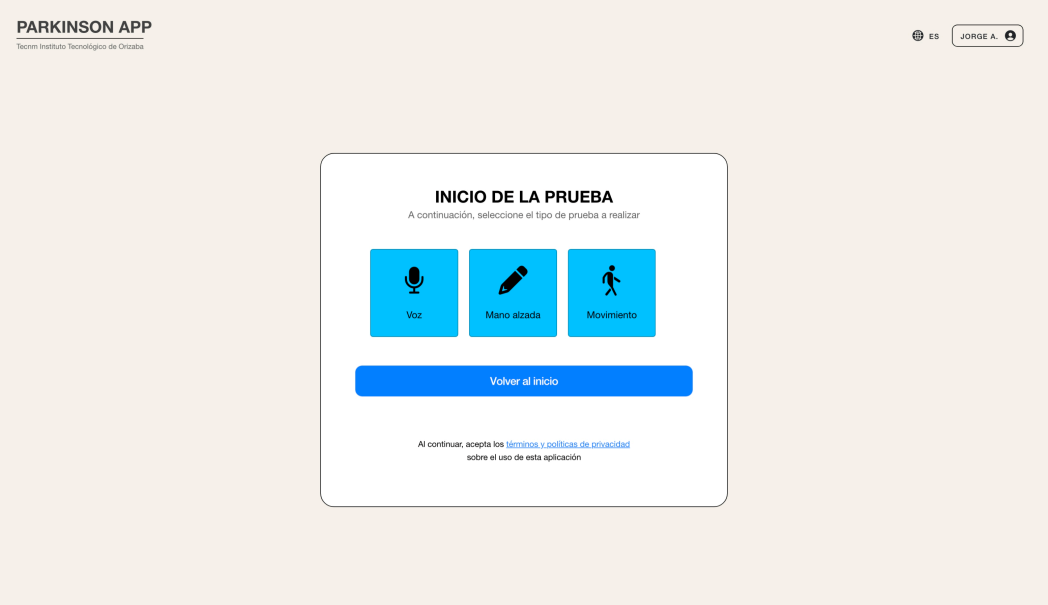


Figura 4.5 Módulo web - Catálogo de pruebas

Esta interfaz da comienzo a la aplicación de pruebas mediante la presentación de un lienzo, el cual, representa en pantalla una plantilla predefinida para que el paciente realice el trazo lo más acertado posible a la línea punteada. La Figura 4.6 ilustra el comienzo de la prueba con el primer ejercicio, mientras que en la Figura 4.7 se aprecia su ejecución por parte del paciente.

Figura 4.6 Módulo web - Aplicación de la prueba – Lienzo

Figura 4.7 Módulo web - Aplicación de la prueba – Realizado

Finalizada la prueba, se notifica en pantalla donde se aprecia también un folio de seguimiento, en este punto, cada uno de los ejercicios se reciben en el componente de clasificación, el cual, se encarga de registrar los resultados obtenidos, posteriormente, el registro cuenta con acceso de consulta desde el panel principal y

expediente del paciente como parte del “Historial de pruebas”, en la Figura 4.8 se aprecia uno de los resultados obtenidos durante la prueba.

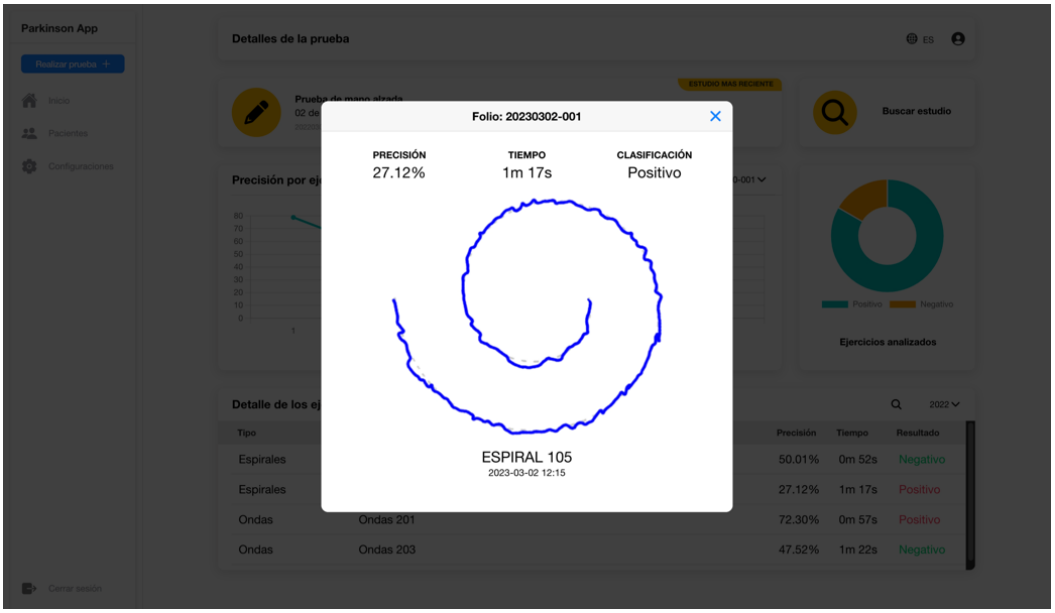


Figura 4.8 Módulo web - Resultados de la prueba

Mediante este enfoque, se lleva a cabo de manera efectiva la clasificación para la identificación y prevención de la EP mediante ejercicios de trazabilidad, brindando así al personal médico más información para el diagnóstico de su estado de salud y posterior toma de decisiones.

Por otro lado, es importante mencionar que el valor de clasificación obtenido a través del módulo es una herramienta complementaria en la toma de decisiones médicas y no un diagnóstico definitivo. Este valor representa una probabilidad de padecimiento basada en los resultados obtenidos durante el estudio, por ello, es fundamental complementar el diagnóstico mediante la interpretación y seguimiento médico correspondiente. En resumen, los resultados obtenidos, tienen como finalidad permitir al personal médico brindar una atención más oportuna y personalizada, ya que cuentan con nuevas herramientas para el diagnóstico y seguimiento oportuno de la EP.

Capítulo 5. Conclusiones

El enfoque en la detección temprana de la EP mediante técnicas de AP y el desarrollo de un módulo web que permita incorporar los procesos de detección mediante ejercicios de trazabilidad, demostraron ser una herramienta efectiva, permitiendo a profesionales de salud, evaluar y clasificar con mayor precisión a los pacientes que presentan esta enfermedad. Así mismo, la precisión obtenida por los algoritmos implementados (RF, SVM, y GB) permiten considerar al modelo como un recurso de gran importancia en el diagnóstico y seguimiento de la EP.

Como conclusiones se destacó que la utilización de técnicas de AP permite obtener resultados de mayor precisión y eficacia en comparación con métodos tradicionales. Además, el módulo desarrollado resultó ser una herramienta de gran utilidad para la detección temprana de la EP, así como también para el seguimiento en la evolución de los pacientes a lo largo del tiempo.

Finalmente, cabe mencionar que la incorporación de tecnologías de la información y técnicas de aprendizaje profundo en el sector de la salud, representan una herramienta valiosa para la detección de enfermedades, lo que brinda apertura y expansión a futuro de mejoras que permitan agilizar los procesos de diagnóstico de diversas enfermedades neurodegenerativas.

5.1 Recomendaciones

En cuanto a trabajo a futuro, se prevé expandir la capacidad del modelo presentado, mejorando su rendimiento para realizar estudios de clasificación a través de nuevos sistemas de información, tales como, el análisis de datos del lenguaje natural y la supervisión de la actividad motriz, así mismo, el modelo presentado se mantendrá como una plataforma enfocada en la identificación, diagnóstico y seguimiento de enfermedades crónicas y neurodegenerativas.

Productos académicos

Como parte importante del trabajo esta tesis, se realizó la escritura y publicación de los siguientes artículos académicos.

Análisis comparativo de algoritmos y conjuntos de datos para la detección de la enfermedad de Parkinson mediante técnicas de Deep Learning.

Universidad Nacional Autónoma de México
Facultad de Estudios Superiores Cuautitlán
Departamento de Ingeniería

Otorgan la presente

Constancia

A: Jaime Hernández Ramírez, Giner Alor Hernández, Nancy Aracely Cruz Ramos, José Luis Sánchez Cervantes and Lisbeth Rodríguez Mazahua

Por su participación como ponente del tema

Análisis comparativo de algoritmos y conjuntos de datos para la detección de la enfermedad de Parkinson mediante técnicas de Deep Learning

En el 5° Congreso Estudiantil de Inteligencia Artificial Aplicada a la Ingeniería y Tecnología (CEIAAIT), realizado del 9 al 11 de noviembre de 2022.

"POR MI RAZA HABLARÁ EL ESPÍRITU"
Cuautitlán Izcalli, Estado de México, noviembre de 2022.

Folio: 2022004G2

Dr. David Quintanar Guerrero
Director

Autores: Jaime Hernández Ramírez, Giner Alor Hernández, Nancy Aracely Cruz Ramos, José Luis Sánchez Cervantes, Lisbeth Rodríguez Mazahua,

Palabras clave: Aprendizaje profundo, Enfermedades neurodegenerativas, Parkinson

Presentado en: 5to Congreso Estudiantil de Inteligencia Artificial Aplicada a la Ingeniería y Tecnología, CEIAAIT 2022

Detección Temprana de la Enfermedad de Parkinson Mediante Técnicas de Aprendizaje Profundo.

Autores: Jaime Hernández Ramírez, Giner Alor Hernández, José Luis Sánchez Cervantes, Lisbeth Rodríguez Mazahua, Nancy Aracely Cruz Ramos

Palabras clave: Aprendizaje profundo, Enfermedades neurodegenerativas, Inteligencia Artificial, Parkinson

Presentado en: XV Congreso Mexicano de Inteligencia Artificial, COMIA 2023

Derechos de Autor

Registro de Derechos de Autor ante en INDAUTOR

INDAUTOR
INSTITUTO NACIONAL DEL DERECHO DE AUTOR

.

- Componente de software para la identificación temprana de la enfermedad de Parkinson mediante técnicas de Deep Learning.

Bibliografía

[1] Haugeland J., *Mind Design II*. The MIT Press, 1997. doi: 10.7551/mitpress/4626.001.0001.

[2] Wooldridge M.J. and Veloso M., *Artificial Intelligence Today*, vol. 1600. in Lecture Notes in Computer Science, vol. 1600. Berlin, Heidelberg: Springer Berlin Heidelberg, 1999. doi: 10.1007/3-540-48317-9.

[3] Fritzsche Paula, *Tools in Artificial Intelligence*, 1st ed., vol. 1. London: InTech, 2008. doi: 10.5772/74.

[4] Russell S.J. and Norvin P., *Artificial intelligence, a modern approach*, 4th ed. Prentice Hall, 1995. doi: 10.5555/1671238.

[5] E. Bonabeau, M. Dorigo, and G. Theraulaz, *Swarm Intelligence from natural to artificial systems.* New York: Oxford University Press, 1999. doi: 10.1093/oso/9780195131581.001.0001.

[6] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*, 1st ed. MIT Press, 2016. Accessed: Oct. 08, 2022. [Online]. Available: http://www.deeplearningbook.org

[7] B. García Navarro, “Implementación de técnicas de deep learning,” 2015. Accessed: Nov. 12, 2022. [Online]. Available: http://riull.ull.es/xmlui/handle/915/1409

[8] P. Kim, *MATLAB Deep Learning*. Berkeley, CA: Apress, 2017. doi: 10.1007/978-1-4842-2845-6.

[9] L. Deng, “Deep Learning: Methods and Applications,” *Foundations and Trends® in Signal Processing*, vol. 7, no. 3–4, pp. 197–387, 2014, doi: 10.1561/2000000039.

[10] J. Heaton, “Ian Goodfellow, Yoshua Bengio, and Aaron Courville: Deep learning,” *Genet Program Evolvable Mach*, vol. 19, no. 1–2, pp. 305–307, Jun. 2018, doi: 10.1007/s10710-017-9314-z.

[11] P. Ponce, *Inteligencia artificial con aplicaciones a la ingeniería*, Primera Edición. Alfaomega, 2010.

[12] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner, “Gradient-based learning applied to document recognition,” *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998, doi: 10.1109/5.726791.

[13] Claude Sammut and Godofredo I. Webb, *Encyclopedia of Machine Learning and Data Mining*, Second Edition. New York: Springer, 2017. doi: 10.1007/978-1-4899-7687-1.

[14] J. Hearty, *Advanced Machine Learning with Python*, First Edition. Birmingham: Packt Publishing Ltd, 2016.

[15] L. Bertram, “The genetic epidemiology of neurodegenerative disease,” *Journal of Clinical Investigation*, vol. 115, no. 6, pp. 1449–1457, Jun. 2005, doi: 10.1172/JCI24761.

[16] S. Pressman Roger, *Ingeniería de Software. Un enfoque práctico*. Mc GrawHill, 2014.

[17] M. Lutz, *Learning Python*, 5th ed. O’Reilly, 2013.

[18] F. Pedregosa *et al.*, “Scikit-Learn: Machine Learning in Python,” *J. Mach. Learn. Res.*, vol. 12, no. null, pp. 2825–2830, Nov. 2011.

[19] C. Marras *et al.*, “Prevalence of Parkinson’s disease across North America,” *NPJ Parkinsons Dis*, vol. 4, no. 1, p. 21, Dec. 2018, doi: 10.1038/s41531-018-0058-0.

[20] E. Abdulhay, N. Arunkumar, K. Narasimhan, E. Vellaiappan, and V. Venkatraman, “Gait and tremor investigation using machine learning techniques for the diagnosis of Parkinson disease,” *Future Generation Computer Systems*, vol. 83, pp. 366–373, Jun. 2018, doi: 10.1016/j.future.2018.02.009.

[21] C. Kotsavasiloglou, N. Kostikis, D. Hristu-Varsakelis, and M. Arnaoutoglou, “Machine learning-based classification of simple drawing movements in Parkinson’s disease,” *Biomed Signal Process Control*, vol. 31, pp. 174–180, Jan. 2017, doi: 10.1016/j.bspc.2016.08.003.

[22] L. Ali, C. Zhu, N. A. Golilarz, A. Javeed, M. Zhou, and Y. Liu, “Reliable Parkinson’s Disease Detection by Analyzing Handwritten Drawings: Construction of an Unbiased Cascaded Learning System Based on Feature Selection and Adaptive Boosting Model,” *IEEE Access*, vol. 7, pp. 116480–116489, 2019, doi: 10.1109/ACCESS.2019.2932037.

[23] J. Camps *et al.*, “Deep learning for freezing of gait detection in Parkinson’s disease patients in their homes using a waist-worn inertial measurement unit,” *Knowl Based Syst*, vol. 139, pp. 119–131, 2018, doi: 10.1016/j.knosys.2017.10.017.

[24] R. Alkhatib, M. O. DIab, C. Corbier, and M. el Badaoui, “Machine Learning Algorithm for Gait Analysis and Classification on Early Detection of Parkinson,” *IEEE Sens Lett*, vol. 4, no. 6, Jun. 2020, doi: 10.1109/LSENS.2020.2994938.

[25] A. Moshkova, A. Samorodov, N. Voinova, E. Ivanova, and E. Fedotova, “Parkinson’s Disease Detection by Using Machine Learning Algorithms and Hand Movement Signal from LeapMotion Sensor,” in *Conference of Open Innovations Association (FRUCT)*, 26th ed.2020, pp. 321–327. doi: 10.23919/FRUCT48808.2020.9087433.

[26] Lin Yao, Peter Brown, and Mahsa Shoaran, “Resting Tremor Detection in Parkinson’s Diseasewith Machine Learning and Kalman Filtering,” pp. 1–4, 2018, doi: 10.1109/BIOCAS.2018.8584721.

[27] S. Arora *et al.*, “Detecting and monitoring the symptoms of Parkinson’s disease using smartphones: A pilot study,” *Parkinsonism Relat Disord*, vol. 21, no. 6, pp. 650–653, Jun. 2015, doi: 10.1016/j.parkreldis.2015.02.026.

[28] E. Rovini, C. Maremmani, A. Moschetti, D. Esposito, and F. Cavallo, “Comparative Motor Pre-clinical Assessment in Parkinson’s Disease Using Supervised Machine Learning Approaches,” *Ann Biomed Eng*, vol. 46, no. 12, pp. 2057–2068, Dec. 2018, doi: 10.1007/s10439-018-2104-9.

[29] A. Anand, A. Haque, J. Sahaya, R. Alex, and N. Venkatesan, “Evaluation of Machine learning and Deep learning algorithms combined with dimentionality reduction techniques for classification of Parkinson’s Disease,” *IEEE International Symposium on Signal Processing and Information Technology (ISSPIT).* , pp. 342–347, Feb. 2018, doi: 10.1109/ISSPIT.2018.8642776.

[30] Marek Wodzinski, Andrzej Skalski, Daria Hemmerling, Juan Rafael Orozco-Arroyave, and Elmar N¨oth, “Deep Learning Approach to Parkinson’s Disease Detection Using VoiceRecordings and Convolutional Neural Network Dedicated to ImageClassification,” *Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, vol. 41, pp. 717–720, 2019, doi: 10.1109/EMBC.2019.8856972.

[31] Z. Karapinar Senturk, “Early diagnosis of Parkinson’s disease using machine learning algorithms,” *Med Hypotheses*, vol. 138, May 2020, doi: 10.1016/j.mehy.2020.109603.

[32] R. Prashanth, S. Dutta Roy, P. K. Mandal, and S. Ghosh, “High-Accuracy Detection of Early Parkinson’s Disease through Multimodal Features and

Machine Learning," *Int J Med Inform*, vol. 90, pp. 13–21, Jun. 2016, doi: 10.1016/j.ijmedinf.2016.03.001.

[33] A. U. Haq *et al.*, "Comparative Analysis of the Classification Performance of Machine Learning Classifiers and Deep Neural Network Classifier for Prediction of Parkinson Disease," 2018. doi: 10.1109/ICCWAMTIP.2018.8632613.

[34] G. Pahuja and T. N. Nagabhushan, "A Comparative Study of Existing Machine Learning Approaches for Parkinson's Disease Detection," *IETE Journal of Research*, vol. 67, no. 1. Taylor and Francis Ltd., pp. 4–14, 2021. doi: 10.1080/03772063.2018.1531730.

[35] M. Nilashi, O. Ibrahim, H. Ahmadi, L. Shahmoradi, and M. Farahmand, "A hybrid intelligent system for the prediction of Parkinson's Disease progression using machine learning techniques," *Biocybern Biomed Eng*, vol. 38, no. 1, pp. 1–15, 2018, doi: 10.1016/j.bbe.2017.09.002.

[36] Enes Celik and Sevinc Ilhan Omurca, *Improving Parkinson's Disease Diagnosis with Machine Learning Methods*. 2019.

[37] M. Nilashi, O. bin Ibrahim, H. Ahmadi, and L. Shahmoradi, "An analytical method for diseases prediction using machine learning techniques," *Comput Chem Eng*, vol. 106, pp. 212–223, 2017, doi: 10.1016/j.compchemeng.2017.06.011.

[38] F. Faghri *et al.*, "Predicting onset, progression, and clinical subtypes of Parkinson disease using machine learning," *bioRxiv*, 2018, doi: 10.1101/338913.

[39] H. Gunduz, "Deep Learning-Based Parkinson's Disease Classification Using Vocal Feature Sets," *IEEE Access*, vol. 7, pp. 115540–115551, 2019, doi: 10.1109/ACCESS.2019.2936564.

[40] K. H. Leung *et al.*, *Using deep-learning to predict outcome of patients with Parkinson's disease*. 2018.

[41] Microsoft, “Visual Studio Code Documentation,” 2022. https://code.visualstudio.com/docs (accessed Dec. 09, 2022).

[42] Docker, “Docker Documentarion,” 2022. https://docs.docker.com/ (accessed Dec. 09, 2022).

[43] TypeScript Community, “The TypeScript Handbook,” *TypeScript*, Apr. 11, 2022. https://www.typescriptlang.org/docs/handbook/intro.html (accessed Dec. 17, 2022).

[44] Angular Commnity, “Angular Docs,” *Angular*, 2022. https://angular.io/docs (accessed Dec. 02, 2022).

[45] C. Griffith, *Mobile App Development with Ionic, Revised Edition: Cross-Platform Apps with Ionic, Angular, and Cordova*. O’Reilly Media, 2017.

[46] The PHP Group, “PHP Official Documentation,” 2022. https://www.php.net/docs.php (accessed Dec. 01, 2022).

[47] M. Grinberg, *Flask Web Development: Developing Web Applications with Python*. O’Reilly Media, 2018.

[48] Oracle, “What is MySQL?,” 2022. https://dev.mysql.com/doc/refman/8.0/en/what-is-mysql.html (accessed Nov. 30, 2022).

[49] C. Griffith, “What is Hybrid Mobile App Development: Hybrid vs Native vs Web - Ionic.” 2021. [Online]. Available: https://ionic.io/resources/articles/what-is-hybrid-app-development/

[50] Scot K, “Parkinson’s Drawings,” 2019. https://www.kaggle.com/datasets/kmader/parkinsons-drawings (accessed Jun. 23, 2022).

[51] P. Zham, D. K. Kumar, P. Dabnichki, S. Poosapadi Arjunan, and S. Raghav, “Distinguishing Different Stages of Parkinson’s Disease Using Composite Index of Speed and Pen-Pressure of Sketching a Spiral,” *Front Neurol*, vol. 8, Sep. 2017, doi: 10.3389/fneur.2017.00435.

[52] Team AI, “Parkinson Disease Spiral Drawings,” 2017. https://www.kaggle.com/datasets/team-ai/parkinson-disease-spiral-drawings (accessed Jun. 23, 2022).