



UNIVERSIDAD DE BURGOS
ESCUELA POLITÉCNICA SUPERIOR
Grado en Ingeniería Informática



TFG del Grado en Ingeniería
Informática

Aplicación web para pacientes
con párkinson



Presentado por Sandra Díaz Aguilar
en Universidad de Burgos — 11 de junio
de 2024

Tutor: Álvar Anaiz González
Cotutora: Alicia Olivares Gil

Resumen

Este proyecto pretende servir de ayuda a pacientes con párkinson, un trastorno neurodegenerativo crónico que lamentablemente afecta a más de 150 000 personas en España.

El proyecto consiste en desarrollar una aplicación web en la que pacientes y médicos pueden acceder a información personal y médica. En concreto a gráficos generados con los datos que mide un sensor que llevan los pacientes incorporado, de forma sencilla y fácil de entender. Además utiliza el aprendizaje automático para tratar de predecir cómo avanzará la enfermedad según la información obtenida de unos vídeos grabados por los pacientes realizando el *finger tapping test*. También permite a los médicos gestionar la información de sus pacientes y añadir nuevos registros del sensor y vídeos de sus pacientes.

La aplicación se ha desplegado y está internacionalizada, de forma que sea accesible para más pacientes.

Descriptores

Enfermedad de Parkinson, sensor, desarrollo web, datos médicos, análisis de series temporales, aprendizaje automático.

Abstract

The aim of this project is to help patients with Parkinson's disease, a chronic neurodegenerative disorder that unfortunately affects more than 150 000 people in Spain. The project consists of developing a web application, where patients and doctors can access personal and medical information. Specifically, graphs generated with the data measured by a sensor that patients have incorporated. It is accessible in a simple and easy way. The app also uses machine learning to try to predict how the disease will progress based on the information obtained from videos recorded by patients performing the finger tapping test. Furthermore, the app also allows doctors to manage their patient information and add new sensor records and videos of their patients.

The application has been deployed and internationalized, so that it is accessible to more patients.

Keywords

Parkinson's disease, sensor, web development, medical data, time series analysis, machine learning.

Índice general

Índice general	iii
Índice de figuras	v
Índice de tablas	vi
1. Introducción	1
1.1. Estructura de los documentos	3
2. Objetivos del proyecto	5
3. Conceptos teóricos	7
3.1. Enfermedad de Parkinson	8
3.2. <i>Machine Learning</i>	9
4. Técnicas y herramientas	13
4.1. Gestión de proyectos	13
4.2. Lenguaje de programación	15
4.3. Diseño web adaptable	19
4.4. Editor de texto	20
4.5. Prototipado	21
4.6. Base de datos	21
4.7. Repositorio	22
4.8. Otros recursos utilizados	23
4.9. Herramienta <i>Hardware</i>	24
5. Aspectos relevantes del desarrollo del proyecto	25
5.1. Aprendizaje de programación web	25

5.2. Utilización de datos de terceros	26
5.3. Incorporación de un Trabajo de Fin de Grado previamente desarrollado	27
5.4. Aprendizaje de minería de datos	28
5.5. Proceso de predicción de los datos	28
5.6. Análisis de la calidad del código	36
5.7. Despliegue en un servidor real	38
6. Trabajos relacionados	41
6.1. Ámbitos del proyecto	41
6.2. Aplicaciones médicas	42
6.3. Aprendizaje automático aplicado a la medicina	43
6.4. Otros TFGs relacionados con el párkinson	44
6.5. Justificación de la necesidad del proyecto	45
7. Conclusiones y Líneas de trabajo futuras	47
7.1. Conclusiones	47
7.2. Líneas de trabajo futuras	49
Bibliografía	53

Índice de figuras

4.1. Sensor	24
5.1. Predicción paciente 1	30
5.2. Predicción paciente 2	31
5.3. Predicción paciente 3	31
5.4. Predicción paciente 4	31
5.5. Predicción paciente 1 - Estimated	32
5.6. Predicción paciente 1 - Heuristic	33
5.7. Predicción paciente 1 - Legacy-heuristic	33
5.8. Predicción paciente 1 - Known	34
5.9. Predicción paciente 1 - Holt	35
5.10. Análisis del código con SonarQube	37
5.11. Segundo análisis del código con SonarQube	38

Índice de tablas

5.1. Tabla con las predicciones del modelo Holt	36
---	----

1. Introducción

El párkinson es un trastorno neurodegenerativo crónico que afecta a más de siete millones de personas en todo el mundo, incluyendo a 150.000 españoles. Por lo general, el riesgo de desarrollar esta enfermedad aumenta con la edad, siendo más común a partir de los 60 años. Dado el aumento de la esperanza de vida, la SEN (Sociedad Española de Neurología) estima que el número de afectados por la Enfermedad de Parkinson se triplicará en los próximos treinta años [14].

Esta enfermedad es comúnmente conocida por el temblor que causa en las manos, pero presenta una variedad de síntomas relacionados con el movimiento, como rigidez muscular, pérdidas de equilibrio y lentitud en los movimientos. Tiene un impacto significativo en la calidad de vida de los pacientes, por lo que, detectar estos síntomas a tiempo es crucial para proporcionar un tratamiento adecuado y mejorar el manejo de la enfermedad.

Durante el presente proyecto se desarrollará una aplicación web que permita, tanto a las personas con párkinson como a sus médicos, llevar un seguimiento de la enfermedad. Esta herramienta ayudará a los médicos a ajustar los tratamientos y evaluar su efectividad a lo largo del tiempo, y a los pacientes les proporcionará una mayor independencia en la gestión de su enfermedad.

La idea del proyecto surgió durante un congreso de medicina, donde se identificó la necesidad de monitorizar la evolución de los pacientes. Se espera que cumpla con las expectativas y resulte de utilidad para los profesionales del Hospital Universitario de Burgos y los miembros de la Asociación Parkinson Burgos.

La aplicación web permitirá a los médicos acceder a un listado de los pacientes con párkinson a su cargo, donde podrán tanto consultar como

introducir información relevante para el tratamiento de la enfermedad. Principalmente se trabajará con dos tipos de datos: aquellos recogidos por un sensor y los obtenidos del análisis de vídeos de los pacientes.

Los pacientes utilizarán un dispositivo médico con sensores que recogerán datos clave para la enfermedad de Parkinson, como la bradicinesia, la discinesia o características de los pasos. Estos datos, en su formato bruto, no son fácilmente interpretables, por lo que se necesitaba generar gráficas que proporcionaran información de utilidad para los médicos. Son ellos los que introducirán los archivos CSV generados por los sensores en la aplicación, para posteriormente visualizar diferentes gráficas con los datos. Se permite elegir el intervalo de tiempo que se desea visualizar y el tipo de gráfica a realizar.

Entre las opciones disponibles, se encuentran gráficas que muestran la marcha media filtrada, la desviación estándar media de la marcha y el número de pasos considerados («parámetros de bradicinesia»). También hay gráficas que ilustran los episodios de *Freezing of Gait* (FoG), la probabilidad de discinesia y la confianza en la detección de discinesia («parámetros de FoG y discinesia»). Además, se puede acceder a información sobre la longitud, el número, la velocidad y la cadencia de los pasos del paciente a lo largo del tiempo («información de los pasos»). Por último, existen gráficas que presentan el estado motor, la discinesia y la bradicinesia del paciente en intervalos de 10 minutos («parámetros de estado motor, discinesia y bradicinesia a 10 minutos»).

El otro tipo de datos proviene de los vídeos que pueden subir los médicos a la aplicación. Estos vídeos, grabados por los propios pacientes, deben mostrar una de sus manos realizando un movimiento de pinza. Este movimiento se debe realizar con ambas manos para detectar alteraciones en la coordinación motora, lo que puede ser indicativo de la progresión de la enfermedad. Se escogió este movimiento en específico ya que forma parte del formulario de la UPDRS [18] para la evaluación de la enfermedad. La aplicación permitirá visualizar los vídeos del paciente, junto con gráficas que muestren la evolución de las características del movimiento como la amplitud y la lentitud a lo largo del tiempo. La aplicación también incorporará modelos de *machine learning* capaces de predecir el avance de dichas características en el futuro.

En conclusión, este proyecto pretende crear una aplicación web para el seguimiento de la enfermedad de Parkinson, que mejore la calidad de vida de los afectados y optimice los tratamientos administrados por sus médicos. La colaboración con el Hospital Universitario de Burgos y la Asociación

Parkinson Burgos será fundamental para crear una herramienta útil y eficaz en la práctica clínica diaria.

1.1. Estructura de los documentos

La documentación del proyecto se ha dividido en dos documentos: la memoria y los anexos. La memoria es el documento principal, donde se explica el trabajo de manera teórica, dirigido principalmente a los evaluadores del mismo o a cualquier lector interesado en la comprensión global del proyecto. Mientras que los anexos están más orientados a desarrolladores, ingenieros o usuarios finales del software al proporcionar información práctica y técnica de forma detallada.

Se describe a continuación el contenido de cada apartado de ambos documentos.

Memoria

La memoria se divide en los siguientes capítulos:

1. Introducción: describe el contenido del trabajo y de los documentos entregados.
2. Objetivos del proyecto: define las metas a cumplir.
3. Conceptos teóricos: proporciona los conocimientos necesarios para comprender el proyecto.
4. Técnicas y herramientas: explica las técnicas y herramientas de desarrollo utilizadas.
5. Aspectos relevantes del proyecto: se destacan los desafíos encontrados durante la realización del proyecto.
6. Trabajos relacionados: se estudia el contexto del proyecto comentando trabajos realizados dentro del ámbito del proyecto actual.
7. Conclusiones y líneas de trabajo futuras: se comentan las conclusiones obtenidas del resultado del proyecto y funcionalidades que sería interesante añadir.

Anexos

Se han desarrollado los siguientes anexos:

- A. Plan de proyecto *Software*: discute la viabilidad económica y legal del proyecto, así como la planificación temporal del mismo.
- B. Especificación de requisitos: se detallan las características que la aplicación web debe cumplir.
- C. Especificación de diseño: detalla la organización de los datos, el diseño de las interfaces y la interacción entre los componentes del sistema.
- D. Documentación técnica de programación: describe el funcionamiento interno del *software* del proyecto, para facilitar su uso a otros desarrolladores.
- E. Documentación de usuario: describe las funcionalidades de la aplicación web desarrollada, así como toda la información que el usuario debe conocer para poder hacer un correcto uso de esta.
- F. Anexo de sostenibilización curricular: aborda aspectos de sostenibilidad aplicados al Trabajo de Fin de Grado.

2. Objetivos del proyecto

El objetivo principal del proyecto es desarrollar una aplicación web que ofrezca a las personas con párkinson una manera intuitiva de llevar un seguimiento de su enfermedad, permitiéndoles ver y predecir su evolución, además de permitir a los médicos gestionar la enfermedad de sus pacientes.

Para ello se han definido una serie de objetivos, entre los que se pueden distinguir objetivos marcados por los requisitos del *software* y objetivos de carácter técnico:

Objetivos marcados por los requisitos del *software*

- Diseñar una aplicación intuitiva y fácil de utilizar, tanto para los pacientes como para los médicos, utilizando el *framework* Flask.
- Crear funcionalidades en la aplicación web que permitan a los pacientes acceder a gráficas que muestren los datos medidos por el sensor y predecir datos futuros sobre su evolución.
- Crear funcionalidades en la aplicación web que permitan a los médicos gestionar la información de sus pacientes, añadir vídeos y datos a sus historiales y ver sus gráficas y predicciones.
- Diseñar una base de datos en la que almacenar la información que utiliza la aplicación.

Objetivos de carácter técnico

- Utilizar una metodología ágil de tipo Scrum mediante la herramienta Jira para realizar un seguimiento de las actividades realizadas, así

como para organizar temporalmente las tareas que vaya marcando el tutor.

- Realizar la documentación del proyecto completa utilizando L^AT_EX, de forma progresiva a la evolución del mismo, mostrando a personas ajenas todos los aspectos relacionados con el proyecto realizado.
- Lectura y aprendizaje sobre la enfermedad de Parkinson, sus síntomas y tratamientos, así como el alcance de la enfermedad para obtener el contexto médico del proyecto.
- Experimentación con diferentes técnicas de *machine learning*, implementadas mediante diferentes bibliotecas de Python, para realizar las predicciones.
- Experimentación con diferentes técnicas de realización de gráficos para mostrar los datos del sensor y predicciones.
- Mejorar los conocimientos sobre Python, *machine learning*, bases de datos, análisis *software* y dirección de proyectos que se estudian en el grado.
- Adquirir conocimientos sobre desarrollo web.
- Internacionalizar la aplicación web, permitiendo su uso a más personas.

Capítulo 3

Conceptos teóricos

Durante el desarrollo de este proyecto se abordan conceptos que pueden resultar complejos para un ingeniero no especializado en el área de la salud o de la inteligencia artificial. A continuación, se explican de manera sintetizada algunos de estos conceptos para su comprensión.

Como se ha explicado en el apartado anterior, el proyecto se centra en el desarrollo de una aplicación web destinada a asistir a pacientes con la enfermedad de Parkinson. Esta aplicación permite a los pacientes visualizar gráficas con los datos recopilados por un sensor de movimiento que deben de haber llevado durante ciertos periodos de tiempo. Además, facilita a los médicos la posibilidad de cargar vídeos de los pacientes realizando la técnica de «*fingertapping*», la cual es una forma adicional de detectar el párkinson. Estos vídeos se analizan y se muestran gráficas con la información obtenida de ellos, pudiendo obtener predicciones sobre la evolución de estos datos en el futuro.

Por lo tanto, en este apartado se abordarán los siguientes conceptos teóricos:

- La enfermedad de Parkinson, la cual no se limita únicamente a temblores. Se explicará por qué es importante medir los movimientos del paciente con un dispositivo con sensores y tener controlada su progresión. También se explicará la técnica del «*fingertapping*» utilizada para evaluar la progresión de la enfermedad y su relación con la escala UPDRS.
- Conceptos fundamentales de aprendizaje automático empleados en el proyecto para la predicción de las gráficas futuras, incluyendo una ex-

plicación de las series temporales, que es la manera en que se gestionan los datos del sensor.

3.1. Enfermedad de Parkinson

El párkinson [7] es un trastorno neurodegenerativo crónico que afecta principalmente al movimiento. Se debe a una degeneración progresiva de ciertas regiones del cerebro, en particular de los núcleos pigmentados del tronco del encéfalo, conocidos como «sustancia negra». Estos núcleos son los que regulan los movimientos voluntarios del cuerpo. Además aparecen estructuras anormales llamadas cuerpos de Lewy que interfieren con el normal funcionamiento de las células nerviosas. Esta enfermedad afecta a 1 de cada 1000 personas y sus síntomas suelen aparecer a partir de los 60 años.

Comúnmente es conocida por el temblor que causa en las manos de los pacientes, pero también presenta otros síntomas como rigidez muscular, bradicinesia (lentitud en los movimientos), hipocinesia (movimiento reducido) y acinesia (pérdida de movimiento), así como anomalías posturales y pérdidas de equilibrio.

La enfermedad puede tener un impacto significativo en la calidad de vida de los pacientes, afectando a sus actividades diarias y a su independencia. Es importante conocer la capacidad biomecánica del paciente porque estos síntomas pueden variar significativamente de un individuo a otro y con el tiempo. Por ello es que el equipo de SENSE4CARE [17] desarrolló el sensor STAT-ON, el cual mide parámetros como los bloqueos, la marcha bradicinética, discinesia, el estado motor... , que son con los que se trabaja en este proyecto.

La evaluación de estos parámetros no solo proporciona información sobre la gravedad de la enfermedad, sino que también puede ayudar a los médicos a ajustar los tratamientos y evaluar su efectividad a lo largo del tiempo.

La aplicación web desarrollada también permite a estos médicos llevar un control de unos vídeos de los pacientes realizando la técnica de «*finger-tapping*». Esta técnica es utilizada para evaluar la velocidad y la regularidad de los movimientos de los dedos de una persona realizando un movimiento de pinza. Estos vídeos fueron utilizados en el TFG de un alumno del curso anterior¹ para detectar alteraciones en la coordinación motora, lo que puede ser indicativo de la progresión de la enfermedad.

¹<https://github.com/cataand/tfg-paddel>

Este movimiento es parte de un examen clínico [18] diseñado en los años 80 para evaluar la gravedad de los síntomas de la enfermedad de Parkinson. Se utiliza una escala de 0 a 4, donde 0 indica ausencia de discapacidad y 4 indica una discapacidad severa. Este «examen» consta de 45 preguntas, divididas en 4 apartados:

- Estado mental, comportamiento y estado de ánimo: evalúa los cambios emocionales y cognitivos del paciente, sus cambios en el intelecto, los pensamientos, la iniciativa...
- Actividades de la vida diaria: se enfoca en cómo afecta la enfermedad en actividades diarias como hablar, comer, escribir, vestirse, lavarse, caminar...
- Exploración de aspectos motores: incluye la evaluación de los principales síntomas motores en los pacientes. En este apartado encontramos la evaluación del movimiento de pinza que realizan los pacientes en los vídeos que se utilizan en el proyecto. Además se analizan la dificultad en el habla, los temblores, la rigidez y otros movimientos como abrir y cerrar las manos, golpes de talón o levantamientos de la silla.
- Complicaciones del tratamiento: contabiliza el nivel de las discinesias (movimientos involuntarios), qué tan incapacitantes o dolorosas son y cuánto duraron la semana previa al examen. Otras complicaciones causadas por el tratamiento que se evalúan son los periodos «OFF» (intervalos en los que los medicamentos no funcionan de forma efectiva y los síntomas de la enfermedad reaparecen), trastornos del sueño o vómitos.

Esta herramienta, utilizada a nivel mundial, es fundamental para evaluar la enfermedad de Parkinson.

3.2. Machine Learning

El aprendizaje automático, también conocido como *machine learning* [11], es un campo de la inteligencia artificial que se centra en el uso de algoritmos que permiten a los ordenadores aprender a partir de unos datos de igual forma que aprenden los seres humanos.

Existen tres técnicas de aprendizaje: supervisado, no supervisado y por refuerzo:

Aprendizaje supervisado: consiste en entrenar un modelo utilizando un conjunto de datos etiquetados (de los que ya conocemos la respuesta) para estimar el resultado de nuevas entradas. Algunas técnicas utilizadas son las redes neuronales, Naïve Bayes, la regresión lineal... Se puede utilizar por ejemplo para estimar la progresión de la diabetes de un paciente, analizando sus datos en función de otros datos conocidos sobre la enfermedad.

Aprendizaje no supervisado: el modelo se entrena en este caso utilizando datos no etiquetados, tratando de encontrar patrones o estructuras internas en los datos, para posteriormente agruparlos según similitudes en conjuntos llamados *clusters*. Se puede utilizar para buscar patrones anómalos en grandes conjuntos de datos médicos, permitiendo identificar enfermedades de forma temprana.

Aprendizaje por refuerzo: consiste en interactuar con el entorno tratando de maximizar la recompensa acumulada. Un agente observa el entorno, toma acciones que afecten a dicho entorno y recibe una penalización o recompensa. El objetivo del agente es aprender la estrategia que suponga el aumento de las acciones que provocan recompensas basándose en reglas o criterios. Se suele aplicar durante el aprendizaje de juegos como el ajedrez, pero también en la industria.

Los datos del sensor se presentan ordenados cronológicamente, como series temporales (entendiendo como serie temporal un conjunto de datos de una variable cuantitativa medida repetidas veces a través del tiempo). Se recogen los valores de las variables de salida de los diferentes algoritmos por minuto, lo que permite analizar su evolución a lo largo del tiempo, comprendiendo patrones y tendencias en los datos que permitirán crear modelos de predicción que se anticipen a la progresión de la enfermedad, permitiendo ajustar los tratamientos en consecuencia. No se ha realizado esta predicción debido a la pobreza de los datos existentes, con gran cantidad de datos nulos, lo que provocaba predicciones incorrectas.

Los datos de lentitud y amplitud del movimiento de pinza de los pacientes, obtenidos tras el análisis de los vídeos por parte de las neurólogas del hospital, junto con otras características del movimiento se tratan también como series temporales. Se dispone de vídeos grabados cada cierto periodo de tiempo por los pacientes, por lo que tras su análisis obtenemos datos cuantitativos repetidos en el tiempo. Por ello, para la predicción de estos datos en el futuro, se utiliza una técnica propia de las series temporales, el suavizado exponencial de Holt. Este método está diseñado para hacer predicciones (de

valores y pendientes) a corto plazo, para series temporales con una tendencia clara de aumento o disminución pero sin tener en cuenta la estacionalidad (variación periódica de los datos).

4. Técnicas y herramientas

En esta sección de la memoria se presentan las técnicas metodológicas y las herramientas de desarrollo que se han utilizado en los diferentes ámbitos del proyecto.

4.1. Gestión de proyectos

En esta sección se va explicar el proceso de selección de la técnica y herramienta utilizadas para la gestión del proyecto.

Técnica

Durante la gestión del proyecto se ha optado por usar una metodología ágil [6]. Estas metodologías ofrecen un enfoque interactivo e incremental, y se centran en entregar productos de calidad de forma rápida y flexible mediante colaboración activa entre miembros del equipo y clientes.

Se ha elegido esta técnica frente a un enfoque más tradicional (con un plan rígido desde el inicio del proyecto) debido a su flexibilidad, pudiendo adaptarse con el tiempo, respondiendo a los cambios, algo muy a tener en cuenta en un Trabajo de Fin de Grado.

Dentro del marco de las metodologías ágiles, se cuenta con variedad de opciones donde elegir. En el proyecto se ha llevado a cabo la selección entre dos de estas metodologías: Scrum y Kanban.

- Scrum: define roles específicos como el *product owner* o el equipo de desarrollo, que participan en eventos predeterminados como el *daily*

scrum o el *sprint review*. Trabaja con iteraciones de 2 a 4 semanas, perfectamente detalladas, llamadas *sprints*, tras las que se debe entregar un incremento del producto.

- Kanban: no tiene roles ni eventos específicos, ofreciendo más flexibilidad. En vez de trabajar con *sprints*, se opera en un flujo continuo, en el que se agregan y retiran tareas en cualquier momento.

Finalmente se escogió utilizar Scrum, ya que la idea de los *sprints* era la que más conveniente le parecía a la alumna. La división de las tareas en períodos de 2 a 4 semanas puede mitigar la sensación de abrumo ante el trabajo futuro, permitiendo concentrarse en finalizar las tareas para la fecha final del *sprint*. Además las reuniones de fin de cada *sprint* sirven para tener contacto con el tutor, comentando las dificultades encontradas y ajustando la cantidad de trabajo para el siguiente *sprint*.

Herramienta

Para llevar a cabo la planificación temporal del proyecto, utilizando una metodología Scrum, se optó por la herramienta Jira. Otra herramienta usada frecuentemente en los TFGs es ZenHub. A continuación se incluye un resumen de cada alternativa, incluyendo comparativas [15] y justificando la decisión:

- Jira²: herramienta de gestión de proyectos desarrollada por Atlassian. Soporta diferentes metodologías de desarrollo como Scrum o Kanban. Ofrece una versión gratuita con funcionalidades básicas y versiones *premium* de pago.
- ZenHub³: herramienta que se integra directamente con GitHub, ofreciendo funciones de gestión ágil directamente en el entorno de desarrollo. Actualmente es una herramienta de pago, no ofrece una versión gratuita.

Ambas opciones proporcionan herramientas para la planificación de *sprints*, seguimiento de tareas y gestión del flujo de trabajo mediante gráficos e informes de rendimiento, lo cual es precisamente lo que se buscaba en el proyecto. Además ambas opciones ofrecen características que mejoran la

²<https://www.atlassian.com/software/jira>

³<https://www.zenhub.com/>

colaboración del equipo, asignando tareas a cada miembro, comentando problemas... (aunque esta característica no es relevante ya que el equipo de trabajo está formado únicamente por la alumna).

Dado que se está utilizando GitHub para mantener un seguimiento del progreso del proyecto, ZenHub sería una opción interesante y fácil de integrar pero se ha elegido utilizar Jira mayoritariamente por ser una opción gratuita. Además es la herramienta que se utiliza en la asignatura de Gestión de Proyectos, por lo que se tenía experiencia previa.

4.2. Lenguaje de programación

Como lenguaje de programación se ha utilizado Python ya que, además de tener una sintaxis simple y fácil de entender, dispone de herramientas para trabajar con desarrollo web, análisis de datos e inteligencia artificial, que son los campos que se han trabajado durante el proyecto.

Python [12] es un lenguaje de código abierto orientado a objetos. Destaca entre los principiantes gracias a su sintaxis fácil de entender. Se utiliza para prácticamente todo: automatización de industrias, creación de videojuegos, desarrollo *software*, análisis y representación de datos, *machine learning* o *blockchain*.

Al ser un lenguaje de alto nivel, su código fuente (fácil de entender por los programadores) con extensión `.py` debe ser interpretado por el intérprete de Python para convertirlo en *bytecode*, ejecutado por la máquina virtual de Python para generar las salidas por consola o modificaciones de archivos.

A continuación se presentan las bibliotecas de Python utilizadas, así como el *framework* utilizado para la aplicación web y los entornos de desarrollo sobre los que se programó:

Bibliotecas

Durante el proyecto se ha hecho uso de las siguientes bibliotecas de Python:

- Scikit-learn⁴ [1]: librería gratuita (de código abierto) para Python que cuenta con gran variedad de algoritmos de aprendizaje automático como *clustering*, clasificación o regresión. Además permite realizar el preprocesamiento de los datos de forma sencilla y una evaluación

⁴<https://scikit-learn.org/stable/>

de la calidad del modelo tras su realización. Es compatible con otras librerías de Python como NumPy o Matplotlib (de la que se hablará a continuación). Es conocida por su facilidad de uso, con herramientas simples y eficientes, y por la abundante documentación y comunidad activa.

- Matplotlib⁵ [8]: biblioteca gratuita utilizada para crear visualizaciones estáticas, animadas o interactivas en Python. Se utilizó para practicar la creación de gráficas con los datos recogidos por el sensor de los pacientes. Puede crear histogramas, diagramas de barras, circulares... con pocas líneas de código. Permite personalizar el estilo visual y exportar a múltiples formatos de archivo.
- Pandas⁶: herramienta de manipulación y análisis de datos de código abierto, potente, flexible y fácil de usar. Permite leer/escribir datos de archivos CSV, Excel o de bases de datos SQL, así como manipularlos mediante filtrados, agrupaciones, combinaciones...
- Numpy⁷: biblioteca creada para computación científica y numérica. Permite trabajar con grandes volúmenes de datos en forma de *arrays* multidimensionales y realizar operaciones matemáticas con ellos. Combina la flexibilidad de Python y la eficiencia de C. Cabe destacar que es la base sobre la que se construyen otras bibliotecas comentadas anteriormente, como Pandas o Scikit-learn.
- Hashlib⁸: biblioteca de Python que permite utilizar varias funciones de *hash* criptográfico como SHA1, SHA224, SHA384... (durante el proyecto se utilizó SHA256). Estas funciones toman un texto de entrada y lo convierten a una cadena de caracteres de longitud fija, que representa esa entrada. Se utiliza para almacenar las contraseñas de la aplicación de forma segura en la base de datos, de manera que no se pueda obtener la contraseña a partir de la cadena *hash*.

Frameworks

Se explica el *framework* utilizado para desarrollar el proyecto junto con sus extensiones.

⁵<https://matplotlib.org/>

⁶<https://pandas.pydata.org/>

⁷<https://numpy.org/>

⁸<https://docs.python.org/3/library/hashlib.html>

- Flask⁹ [10]: herramienta utilizada para desarrollar la aplicación web del proyecto con el lenguaje Python. Promete facilitar la creación de aplicaciones web con el patrón Modelo Vista Controlador. Incluye un servidor web de desarrollo para poder observar los avances en la aplicación sin necesidad de disponer de un servidor web. Soporta el uso de *cookies* y sesiones. Es *open source* y existe mucha documentación, tanto en su página oficial como por parte de otros usuarios. Existen multitud de extensiones que se han utilizado y se explican a continuación:
 - Flask-Babel¹⁰: se ha utilizado para llevar a cabo la internacionalización de la aplicación web. Es capaz de detectar el idioma preferido del usuario y de generar automáticamente archivos de traducción con todo el texto de la aplicación, facilitando su posterior traducción. Además ofrece funciones para formatear fechas, horas o números según el idioma.
 - Flask-Login¹¹: extensión utilizada para la gestión de sesiones de usuario, manejando tareas como iniciar sesión, recordar sesiones durante determinados periodos de tiempo y cerrar sesión.
 - CSRFProtect¹²: se utiliza para proteger la aplicación web contra ataques CSRF (*Cross-Site Request Forgery*) mediante la generación y validación de tokens únicos para cada sesión de un usuario.

Entornos de desarrollo

Los entornos de desarrollo sobre los que se ha trabajado en el proyecto son Jupyter Notebook, para practicar con librerías de gráficas o *machine learning*, y Visual Studio Code, para programar la totalidad de la aplicación web.

- Jupyter Notebook¹³ es una aplicación web de código abierto que se ha utilizado durante la experimentación y pruebas de las librerías de Python. Permite ejecutar código de forma interactiva en celdas, visualizando el resultado paso a paso. Permite crear y compartir documentos que combinan texto, gráficos interactivos y código ejecutable.

⁹<https://flask.palletsprojects.com/en/3.0.x/>

¹⁰<https://python-babel.github.io/flask-babel/>

¹¹<https://flask-login.readthedocs.io/en/latest/>

¹²<https://flask-wtf.readthedocs.io/en/0.15.x/csrf/>

¹³<https://jupyter.org/>

Resultó muy útil para practicar con la librería de Matplotlib, teniendo el código y los gráficos resultantes en el mismo documento.

- Visual Studio Code (vscode) es un entorno de desarrollo integrado, desarrollado por Microsoft, de código abierto y gratuito. Admite muchos lenguajes de programación: Java, Python, C++, JavaScript... Durante el proyecto se utiliza para programar la aplicación web con Flask (Python). Tiene una interfaz de usuario simple y técnicas como el autocompletado que ayudan a los desarrolladores. Existen muchas extensiones en el mercado que proporcionan nuevos lenguajes o temas. Ofrece integración con GitHub, pudiendo subir directamente los códigos desde vscode. Se había utilizado con anterioridad, por lo que se escogió como entorno de desarrollo sobre el que trabajar.

Otro lenguaje utilizado

Además de Python, se ha utilizado JavaScript. Ha sido necesario para programar los *scripts* de los ficheros .html, cuya función es responder a interacciones del usuario, como clics en botones o envíos de formularios, y realizar acciones en consecuencia, como des-ocultar elementos o llamar a otras funciones.

A continuación se presentan algunas de las bibliotecas de JavaScript utilizadas:

- Leaflet¹⁴: librería de JavaScript de código abierto que se utiliza para crear mapas interactivos en sitios y aplicaciones web. Se ha hecho uso de ella en el proyecto para mostrar la ubicación de la UBU en la aplicación web, integrando los datos de OpenStreetMap. Se ha escogido por su simpleza y compatibilidad con diversos navegadores.
- Flatpickr¹⁵: biblioteca de JavaScript que proporciona una interfaz de usuario de calendario, utilizada para la selección, por parte de los usuarios, de las fechas a mostrar en las gráficas que representan los datos de los registros. Se ha escogido por ser altamente personalizable, pudiendo modificar aspectos como el idioma, color, formato de las fechas, elección de un rango de fechas en vez de una sola fecha, deshabilitar ciertas fechas...

¹⁴<https://leafletjs.com/>

¹⁵<https://flatpickr.js.org/>

- Chartjs¹⁶: biblioteca de JavaScript que permite crear gráficos interactivos para visualizar datos en páginas web. Permite crear gráficos de barras, de líneas, de áreas, de radar... Es fácil de usar y muy personalizable, contando con una página web con ejemplos bien explicados. Durante el proyecto se ha hecho uso de ella para generar los gráficos de los datos recogidos por el sensor y de las características de los vídeos de los pacientes.
- jQuery¹⁷: biblioteca de JavaScript que hace más fácil interactuar con los elementos HTML (eliminando o agregando elementos del DOM en respuesta a eventos), manejar de eventos (como clics de ratón) o usar AJAX (para realizar solicitudes HTTP asíncronas), haciendo que las páginas web del proyecto sean más dinámicas e interactivas.

4.3. Diseño web adaptable

Se presentó la idea de que la aplicación web debía ajustarse automáticamente según el dispositivo en el que se estuviera visualizando (ordenador, móvil, tableta...). Para ello hay que utilizar un *framework* de diseño receptivo, que facilite la creación de interfaces ajustables.

Se han estudiado tres alternativas de herramientas [4, 2]:

- Bootstrap¹⁸: herramienta popular, ampliamente utilizada y de la que existe mucha documentación y ayuda por parte de otros usuarios en la web. Dispone de una rejilla para dividir el contenido en columnas y filas, y cuatro tipos de dispositivos (teléfono, tableta, portátiles pequeños y normales), útiles para crear el diseño adaptable que se busca. Dispone de componentes predefinidos como botones, formularios o desplegables, además de gran variedad de estilos prediseñados que se pueden integrar fácilmente en la aplicación. Es conocido por ser fácil de usar e implementar, incluso para principiantes.
- Materialize CSS¹⁹: está basada en el sistema de Material Design creado por Google, por lo que es fácil de integrar con otros productos de Google. Dispone de estilos modernos y una composición minimalista. Ofrece temas de HTML estático pero son de pago.

¹⁶<https://www.chartjs.org/>

¹⁷<https://jquery.com/>

¹⁸<https://getbootstrap.com/>

¹⁹<https://materializecss.com/>

- ZURB Foundation²⁰: esta herramienta es también bastante popular. Dispone de un enfoque más modular, lo que ofrece una mayor flexibilidad, con componentes personalizables, para tener control total sobre la apariencia de la aplicación. La curva de aprendizaje de esta herramienta es mayor, por lo que no es adecuada para principiantes.

Finalmente se decidió utilizar Bootstrap debido a la falta de experiencia en el ámbito del *Front-end* y las aplicaciones web por parte de la alumna. La abundante documentación disponible en Internet sobre esta herramienta fue un factor determinante. Además, la existencia de estilos y componentes predefinidos en Bootstrap, en contraposición a una mayor flexibilidad como la ofrecida por ZURB Foundation, fue considerada como una ventaja significativa dada esa inexperiencia. Se ha optado por una herramienta gratuita y de fácil implementación en Flask.

4.4. Editor de texto

Para llevar a cabo la documentación del proyecto (la memoria y anexos) se analizaron los siguientes editores de texto:

- Microsoft Word: dispone de una interfaz de usuario intuitiva y fácil de usar. Es ampliamente utilizado en entornos académicos y profesionales. Dispone de funcionalidades avanzadas como tablas, gráficos o revisión de documentos. Tiene menor control sobre el formato.
- L^AT_EX: utilizado para documentos científicos, técnicos o académicos. Dispone de gran capacidad de gestión de fórmulas matemáticas y referencias bibliográficas. Ofrece un control preciso sobre el formato del documento. Es menos intuitivo y más complicado de aprender, ya que utiliza comandos y código.

Finalmente se optó por utilizar L^AT_EX porque, aunque se tuvo que invertir tiempo en aprender a manejar el editor, se obtuvo un mejor resultado visual gracias a las plantillas disponibles para realizar la documentación.

Se utilizó Overleaf²¹ como herramienta para editar los documentos de L^AT_EX de forma colaborativa en la nube.

²⁰<https://get.foundation/>

²¹<https://es.overleaf.com>

4.5. Prototipado

Para realizar el diseño de interfaces de la aplicación web se ha utilizado la herramienta Pencil²².

Se barajó la idea de utilizar Adobe XD²³, otra herramienta de diseño y prototipado que permite diseñar experiencias de usuario interactivas tanto para web como para móvil. Integra funciones de diseño y prototipado. Permite importar recursos de otras aplicaciones como Adobe Photoshop y se puede exportar el proyecto una vez finalizado. Se debe iniciar sesión para utilizarla.

Se optó por Pencil Project ya que esta herramienta se había utilizado previamente en la asignatura de Interacción Hombre-Máquina, por lo que la alumna estaba familiarizada con su interfaz. Es una herramienta muy sencilla, con varios paquetes de diseño predeterminados, pero ampliable con colecciones encontradas en la web. Es *open source* y no se debe iniciar sesión para usarla.

Para realizar los diagramas del proyecto se ha utilizado Draw.io²⁴, una herramienta *online* gratuita que permite crear diagramas y mapas mentales de forma intuitiva. Existe también una versión de escritorio.

Los usuarios deben arrastrar y soltar elementos que se encuentran organizados según el tipo de diagrama. Permite crear diversos diagramas (diagrama de flujo, UML, de red, organigramas, mapas conceptuales...) mediante gráficos predefinidos como bloques, clases, atributos, actores, conectores, etc. Posteriormente se pueden exportar en diversos formatos (PNG, PDF, SVG...).

Un dato interesante, aunque no se ha utilizado en el proyecto, es que permite la colaboración en tiempo real con otros usuarios.

Se ha utilizado para crear todos los diagramas del proyecto debido a su simplicidad y a que la alumna ya lo había utilizado en otras ocasiones.

4.6. Base de datos

Fue necesario escoger una herramienta para organizar los datos de los pacientes recolectados por el sensor y los obtenidos de los vídeos, así como

²²<https://pencil.evolus.vn/Features.html>

²³<https://helpx.adobe.com/es/xd/user-guide.html>

²⁴<https://app.diagrams.net/>

los datos que serán utilizados por la aplicación web. Se estudiaron dos alternativas de sistemas gestores de bases de datos relacionales de código abierto [3]: MariaDB y PostgreSQL.

MariaDB es una bifurcación de MySQL, creada por sus desarrolladores originales, después de que Oracle adquiriera MySQL. En cambio, PostgreSQL ha evolucionado de forma más independiente. Ambas utilizan SQL estándar como lenguaje de consulta, por lo que resultarán conocidas para la alumna tras haber estudiado Bases de Datos durante el grado.

Ambas garantizan la integridad de los datos gracias a sus propiedades ACID y admiten extensiones para aumentar su funcionalidad.

A rasgos generales son muy parecidas, pero las implementaciones específicas pueden variar. Se decidió usar MariaDB por tener una comunidad activa que ofrece mucha ayuda en la web, además de por recomendación del tutor.

Como herramienta para crear la base de datos se ha utilizado HeidiSQL [9], una herramienta gratuita de código abierto con la que se pueden administrar bases de datos. Permite crear, modificar o eliminar bases de datos, así como sus tablas y campos, con una interfaz intuitiva.

4.7. Repositorio

La plataforma elegida para llevar el control de versiones y el *hosting* del repositorio ha sido GitHub²⁵.

Es una plataforma de desarrollo colaborativo basada en la nube que permite gestionar proyectos *software*, actualizar versiones de códigos fuentes y facilitar la colaboración entre desarrolladores.

Se ha creado un repositorio en GitHub para almacenar el código fuente y otros documentos del proyecto. Con su sistema de control de versiones se puede ir viendo la evolución del proyecto a lo largo del tiempo. Si hubiera varios desarrolladores podrían trabajar de manera colaborativa fácilmente creando ramas y posteriormente fusionándolas, comparando los cambios realizados.

Se ha optado por hacer el repositorio privado durante la realización del proyecto, invitando únicamente a los tutores, para posteriormente hacerlo público, dejando el proyecto al alcance de otros desarrolladores que podrían estar interesados en él.

²⁵<https://github.com/about>

La herramienta utilizada para llevar a cabo este control de versiones sobre el código es la aplicación gratuita GitHub Desktop²⁶. Se utiliza para realizar comandos de Git, como *commits* o *push*, mediante una interfaz gráfica, en lugar de mediante la línea de comandos.

4.8. Otros recursos utilizados

- Mockaroo²⁷: herramienta en línea que genera datos ficticios, de forma aleatoria, en múltiples formatos (CSV, Json, Excel, XML...). Se ha utilizado para crear los nombres de usuario, contraseñas, direcciones de correo, nombres, apellidos, fechas de nacimiento, direcciones y teléfonos de más de 50 pacientes para la base de datos de la aplicación, ya que no se podía hacer uso de los datos personales reales de los pacientes del estudio.
- Generador de caras aleatorias²⁸: se trata de una página web que utiliza una inteligencia artificial para generar fotos falsas de personas que no existen, de todas las edades, etnias o géneros. Se ha utilizado para rellenar la base de datos de los pacientes y médicos, ya que no se podía acceder a fotos reales de dichos usuarios.

Para ello, se realizó un *script* que accedía a la página web, descargaba y guardaba la imagen de forma local y escribía la ruta de dicha imagen en la base de datos. Este proceso se ejecuta por cada paciente/médico que no tenga imagen en la base de datos.

Al crear imágenes de personas aleatorias, aparecían fotos de menores de edad (que no eran realistas para la aplicación), por lo que se realizó una posterior depuración manual de estas.

- SonarQube²⁹: plataforma de código abierto diseñada para gestionar la calidad del código. Sus herramientas de análisis revisan el código para detectar problemas de seguridad, mantenibilidad, duplicados... Tiene soporte para muchos lenguajes de programación diferentes y ofrece una interfaz intuitiva en la que revisar los informes detallados que realiza.

²⁶<https://docs.github.com/es/desktop/overview/about-github-desktop>

²⁷<https://www.mockaroo.com/>

²⁸<https://thispersondoesnotexist.com/>

²⁹<https://www.sonarsource.com/products/sonarqube/>

4.9. Herramienta *Hardware*

Hasta este punto, se han explorado principalmente las herramientas *software* utilizadas durante el proyecto. Sin embargo, es esencial destacar que la base del trabajo descansa en una herramienta *hardware*: el sensor STAT-ON, desarrollado por SENSE4CARE [17] en marzo de 2020 y que se muestra en la Figura 4.1.

Este dispositivo, dotado de acelerómetros y otros instrumentos, es el encargado de recopilar datos como la marcha bradicinética, discinesia, el estado motor, los bloqueos de la marcha, parámetros de la marcha, transiciones posturales y detalles sobre la postura del paciente. Con estos datos, genera archivos CSV que se utilizan en el proyecto para generar las gráficas.

Los archivos CSV resultantes almacenan los datos procesados por diversos algoritmos a una tasa de frecuencia por minuto. Organizados en forma de matriz, cada columna representa las variables de salida de los algoritmos, mientras que cada fila corresponde al valor de estas variables por minuto. En esencia, este conjunto de datos proporciona una visión de los síntomas motores y la movilidad del paciente.



Figura 4.1: Sensor

5. Aspectos relevantes del desarrollo del proyecto

En el presente apartado se recogen los aspectos más interesantes del desarrollo del proyecto, comentados por los autores del mismo. En este contexto, se tratarán los siguientes temas:

- Aprendizaje de programación web, comenzando sin ningún conocimiento previo, y el esfuerzo que supuso.
- Utilizar extensos conjuntos de datos de terceros.
- La incorporación de un TFG previamente desarrollado por otro alumno.
- El aprendizaje de técnicas de minería de datos para realizar predicciones.
- El proceso de encontrar un modelo que se adapte a los datos disponibles.
- El análisis de la calidad del código.
- Despliegue en un servidor real.

5.1. Aprendizaje de programación web

En primer lugar, se va a destacar el aprendizaje de programación web realizado por la alumna, quien comenzó sin experiencia previa en el campo. Durante el grado en Ingeniería Informática y su vida personal previa, no se habían adquirido conocimientos sobre programación web, por lo que

este proyecto exigió aprender desde cero. Se enfrentó al desafío de sentarse frente al ordenador sin ningún conocimiento de la materia y buscar de forma independiente información en diferentes fuentes: la documentación de Flask, videotutoriales, foros. . .

Se comenzó creando páginas HTML simples, formadas exclusivamente por texto, incrementando la complejidad gradualmente. Primero, se añadieron botones para navegar entre páginas. Posteriormente, se incorporaron formularios y se fusionó la aplicación con la base de datos del proyecto. Se avanzó con el control de sesiones y las interacciones con el usuario final, resultando en la aplicación web obtenida.

Además, se aplicaron algunos de los conceptos aprendidos en la asignatura de «Interacción Hombre-Máquina», los cuales ayudaron a ubicar de manera que resultara intuitiva ciertos elementos en la páginas web, como botones de envío, de eliminación, el perfil del usuario actual. . . mejorando así la experiencia del usuario.

Este enfoque progresivo permitió a la alumna adquirir una base sólida en programación web mientras desarrollaba la aplicación.

5.2. Utilización de datos de terceros

Otro desafío significativo encontrado fue la interpretación y posterior visualización de los datos de los pacientes. Estos datos, recogidos por el sensor que llevan los pacientes (creado por SENSE4CARE [17]), llegaron a manos de la alumna en forma de archivos CSV divididos en carpetas, formados por millones de números separados por comas. Este extenso conjunto de datos, a pesar de que los creadores del sensor proporcionaron un documento con explicaciones sobre lo que detectaban los sensores, fue inicialmente muy complejo de entender.

En el documento se informaba de que los archivos CSV contenían datos a lo largo del tiempo sobre términos como «bradicinesia» y «discinesia» (indicadores clave de la enfermedad de Parkinson que no tenían sentido para la alumna al comienzo del TFG) o el ancho del paso del paciente (cuya importancia la alumna no entendía).

Sin embargo, con el tiempo y tras documentarse sobre el párkinson, estos términos comenzaron a tener sentido y se comprendió la importancia de medir estos datos específicos para ayudar en el diagnóstico y tratamiento de la enfermedad.

Finalmente, se pudieron generar gráficos útiles a partir de estos datos, que servirán tanto a médicos como a pacientes para llevar un control más preciso de la enfermedad.

5.3. Incorporación de un Trabajo de Fin de Grado previamente desarrollado

La idea del proyecto actual surgió de un congreso de medicina donde se presentó el Trabajo de Fin de Grado de Catalin [5], un compañero del año anterior, que consistía en detectar la presencia de bradicinesia (un síntoma de la enfermedad de Parkinson) aplicando técnicas de minería de datos y visión artificial en unos vídeos grabados por pacientes de la asociación de Parkinson realizando un movimiento de pinza. Los asistentes al congreso comentaron la necesidad de tener un registro, y no solo una evaluación puntual, de la evolución de los pacientes, lo que desembocó en la creación de la aplicación web que es el actual TFG presentado.

La primera concepción del proyecto consistió, entre otras funcionalidades, en permitir a los médicos cargar los vídeos de sus pacientes realizando este movimiento de pinza en la base de datos. Los vídeos, proporcionados por el HUBU en colaboración con la asociación de Parkinson Burgos, fueron los utilizados por el compañero comentado anteriormente para su propio proyecto.

Se diseñó una interfaz para visualizar y reproducir dichos vídeos. Posteriormente, se decidió integrar en la interfaz el trabajo del compañero mencionado, mostrando las características obtenidas tras los análisis de los vídeos.

Para ello, se realizó un estudio previo de su TFG, examinando la documentación contenida en la memoria y los anexos, así como revisando el código disponible en GitHub. Tras comprender el proyecto, y considerando que no se implementaría en su totalidad, sino que la idea era obtener las características de los vídeos que se obtienen en una de las etapas iniciales, se comenzó a adaptar el código existente para integrarlo en el desarrollo del presente proyecto.

A lo largo de este proceso, surgieron problemas relacionados con las diferencias de versiones entre las herramientas utilizadas por el autor anterior y las empleadas en el proyecto actual. Se solucionaron mediante ciertos ajustes en el código, instalando librerías y actualizando o descendiendo versiones, garantizando la coherencia y compatibilidad del código.

A pesar de las dificultades encontradas, la integración exitosa de dicho trabajo mejoró la calidad general del proyecto, ampliando su alcance, y permitiendo la implementación de características adicionales, como la capacidad de predecir la evolución de los movimientos de pinza en el futuro.

5.4. Aprendizaje de minería de datos

Otro aspecto a destacar es el aprendizaje de técnicas de inteligencia artificial aplicadas sobre grandes cantidades de datos, con el objetivo de descubrir tendencias o patrones. Estos conceptos se desarrollan en la asignatura de Minería de Datos, impartida como asignatura optativa en la Universidad de Burgos. Sin embargo, la alumna no pudo cursar esta asignatura por haber estado de Erasmus, por lo que tuvo que aprender estos conceptos de manera autodidacta para poder aplicar estos conocimientos en la realización de predicciones a partir de los datos de los pacientes.

Para ello, la alumna dispuso de los apuntes de dicha asignatura, que fueron proporcionados por su tutor. Además, el tutor la inscribió en un curso sobre series temporales, lo que le permitió adquirir conocimientos adicionales en esta área de la minería de datos. Gracias a estos recursos y a la práctica, la alumna pudo comprender y aplicar estos conceptos en su TFG.

Aunque la calidad de los datos disponibles no permitió generar gráficos con predicciones realmente útiles, las técnicas están correctamente implementadas, por lo que, con datos de mejor calidad, se podrían obtener predicciones valiosas para comprender la evolución de los pacientes.

5.5. Proceso de predicción de los datos

Como se ha comentado en la sección anterior, los datos disponibles no permitieron un buen resultado tras la implementación de diferentes modelos de *machine learning*. Ya que en la aplicación web se aprecia el resultado pero no el trabajo previo, en este apartado se va a comentar el proceso realizado hasta conseguir el resultado actual.

A la alumna le fueron suministrados una gran cantidad de vídeos de diferentes pacientes realizando un movimiento de pinza con las manos. Estos vídeos se analizaron, obteniendo ciertas características que muestran el estado de la enfermedad. Posteriormente se realizaría una predicción de estas características utilizando modelos de inteligencia artificial.

En el nombre de los vídeos se especificaba la fecha en la que fue grabado, la mano que aparecía realizando el movimiento, el sexo del paciente y su edad. Al ser vídeos de pacientes reales no se indicaba a qué paciente pertenecían, por lo que se asignó un par de vídeos, uno de la mano derecha y otro de la mano izquierda a cada paciente ficticio de la base de datos del proyecto.

Para entrenar buenos modelos de predicción, lo ideal hubiera sido tener gran cantidad de vídeos para cada paciente, tomados cada mes o cada ciertos meses, con el fin de ver la evolución de la enfermedad y poder realizar mejores predicciones. Por esta razón, se simuló esta situación para los cuatro primeros pacientes de la base de datos. Al primer paciente se le asignaron vídeos de pacientes hombres cuyas edades estaban comprendidas entre los 62 y 67 años. Se modificaron las fechas de los vídeos para simular el paso del tiempo entre cada vídeo, grabando en el año 2018 los primeros vídeos (con 62 años ficticios) y grabando los últimos vídeos en 2023 (con 67 años ficticios). Se trató de realizar un buen preparado de los vídeos para simular el paso de los años en el paciente pero, al tratarse de pacientes con diferente avance de la enfermedad, no salían resultados muy razonables. Para los siguientes tres pacientes se realizó un proceso parecido: el segundo paciente es una mujer, con vídeos tomados entre los 56 y 60 años, el tercero es un hombre, de entre 76 y 82 años y el cuarto es una mujer de entre 75 y 82 años.

Los datos de amplitud y lentitud de los vídeos son los datos reales, calculados por las neurólogas del HUBU. El resto de características son propias del movimiento de las manos que aparecen en el vídeo, calculadas mediante `tsfresh` (esto se adaptó a partir del TFG previo comentado).

Cada característica se toma como una serie temporal, y se predice de forma independiente utilizando un modelo de *machine learning* para series temporales. Ya que no hay una tendencia clara ni estacionalidad en los datos de los vídeos, se comenzó usando el método de suavizado exponencial simple. Este modelo es capaz de predecir los valores futuros que se le indiquen (se ha escogido predecir 4 valores) y su método es dar más peso a los datos más recientes y menos peso a los más antiguos.

Tras leer el código fuente de la función `SimpleExpSmoothing` se detectaron 5 métodos diferentes: «None», «estimated», «heuristic», «legacy-heuristic» y «known». Se probó la eficacia de cada uno de ellos prediciendo los datos de los pacientes:

SimpleExpSmoothing con «None»

Se aplicó la función del suavizado exponencial simple por defecto, sin indicarle a la función un método de inicialización, en los datos de los cuatro primeros pacientes de la aplicación. El resultado no fue bueno, por lo que se modificaron las fechas de los datos para darles periodicidad. No se quiso alterar mucho las fechas originales, por lo que cada paciente tiene una periodicidad diferente, calculada para no hacer muchas modificaciones. Se muestran a continuación los resultados de las predicciones para cada paciente.

El paciente 1 corresponde con un hombre de entre 62 y 66 años, cuyos vídeos son tomados con 199 días de diferencia. En la Figura 5.1 se muestra el gráfico que genera la aplicación con las características de sus vídeos con la mano izquierda y derecha y cuatro días con datos generados por el modelo.

El estudio de la eficacia del modelo se va a basar en los datos de lentitud y velocidad media. Los datos de lentitud de los vídeos de la mano izquierda del paciente son: 0, 0, 1, 0, 0, 0, 1, 0, 1, 0, 0. La media es 0.27 aproximadamente y el modelo predice una lentitud de 0.203. Los datos de la velocidad media son: 1.605, 1.911, 2.662, 1.631, 5.538, 1.616, 1.296, 0.821, 0.913, 2.553, 1.024. La media es 1.870 aproximadamente y el modelo predice 1.605.

El problema es que da este mismo valor para las 4 fechas a predecir. La razón de este suceso es la mala calidad de los datos, que además de no ser suficientes, no tienen una tendencia clara que el modelo pueda detectar al ser, al fin y al cabo, vídeos de pacientes totalmente diferentes que se encuentran en diferentes etapas de la enfermedad. Figura 5.1.

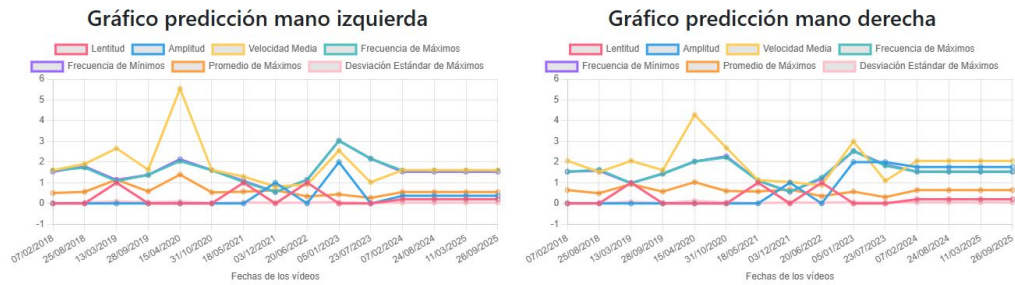


Figura 5.1: Predicción paciente 1

La predicción realizada para el segundo paciente, que es una mujer con una edad de entre 56 y 60 años, con vídeos tomados cada 272 días, se puede observar en la Figura 5.2. El modelo sigue asociando a cada fecha futura el mismo valor.

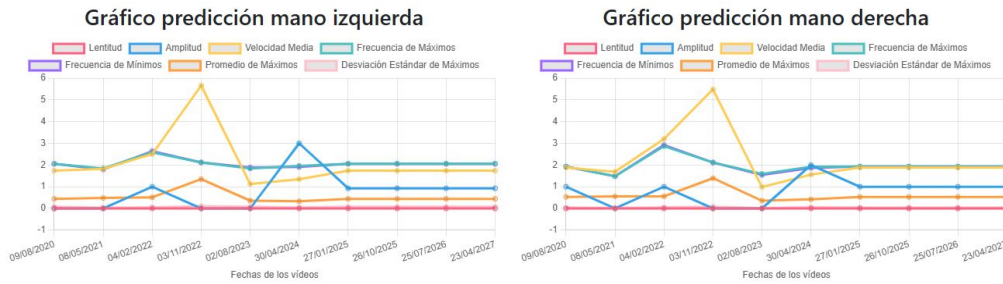


Figura 5.2: Predicción paciente 2

En la Figura 5.3 se aprecia la predicción del tercer paciente, hombre de entre 76 y 82 años, cuyos vídeos fueron grabados con 299 días de diferencia.

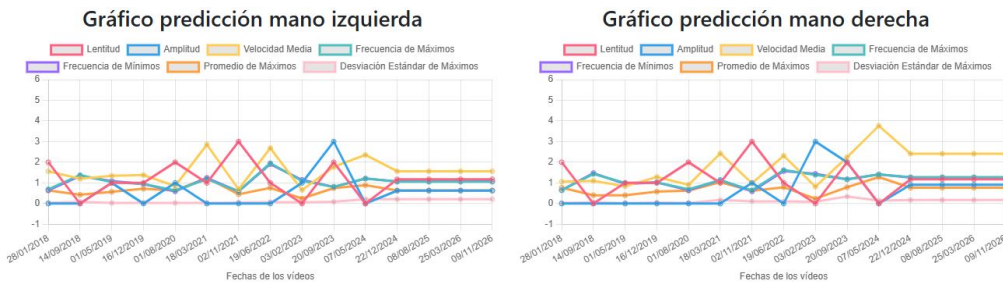


Figura 5.3: Predicción paciente 3

Por último, el gráfico con la predicción de la última paciente (con vídeos grabados con 441 días de diferencia), se encuentra en la Figura 5.4.

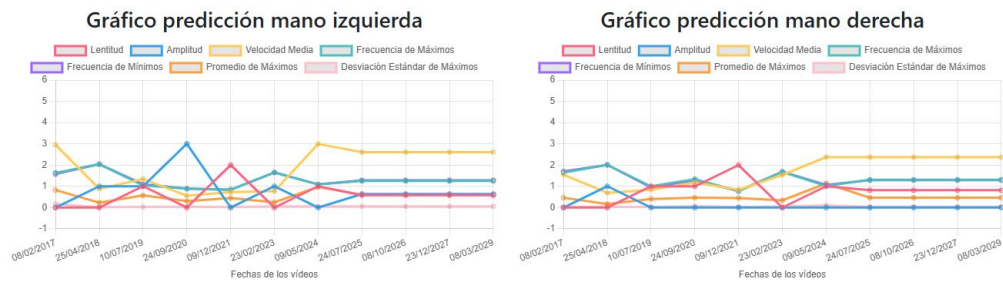


Figura 5.4: Predicción paciente 4

Dado que para los cuatro pacientes se ha obtenido el mismo resultado de linealidad de los datos predichos, se ha probado con el resto de métodos del

modelo. Aunque, a partir de ahora, únicamente se mostrarán los resultados para la mano izquierda del paciente 1.

SimpleExpSmoothing con «estimated»

Con esta opción, los valores iniciales se estiman automáticamente como parte del proceso de ajuste del modelo. En la Figura 5.5 se ve como para la lentitud el modelo predice 0.273 y para la velocidad media predice 1.961, pero sigue siendo siempre el mismo valor para todas las fechas futuras.



Figura 5.5: Predicción paciente 1 - Estimated

SimpleExpSmoothing con «heuristic»

El modelo establece los valores iniciales mediante promedios. Este enfoque es más rápido pero menos preciso que el anterior. La Figura 5.6 muestra el gráfico de predicción que genera, calculando una lentitud de 0.133 y una velocidad media de 2.175 para los cuatro datos a predecir.

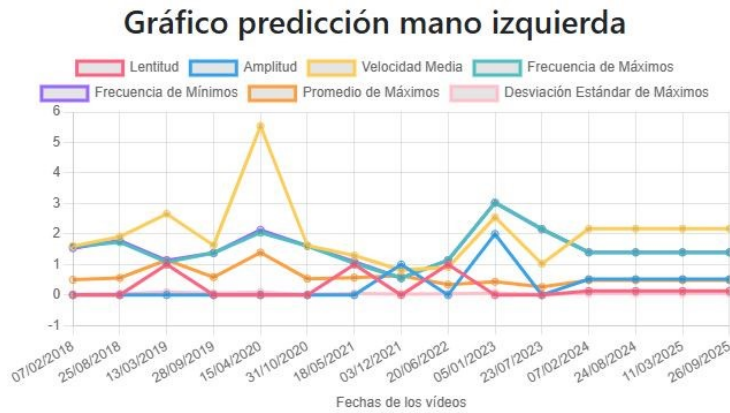


Figura 5.6: Predicción paciente 1 - Heuristic

SimpleExpSmoothing con «legacy-heuristic»

Este enfoque inicializa los valores iniciales mediante estimaciones básicas basadas en los primeros datos de la serie temporal. Para el nivel inicial se utiliza el primer valor de la serie temporal y para la tendencia inicial la diferencia entre los dos primeros valores. Para la lentitud el modelo predice 0.203 y para la velocidad media predice 1.605. Como indica el gráfico de la Figura 5.7, sigue ofreciendo el mismo valor para las diferentes fechas.

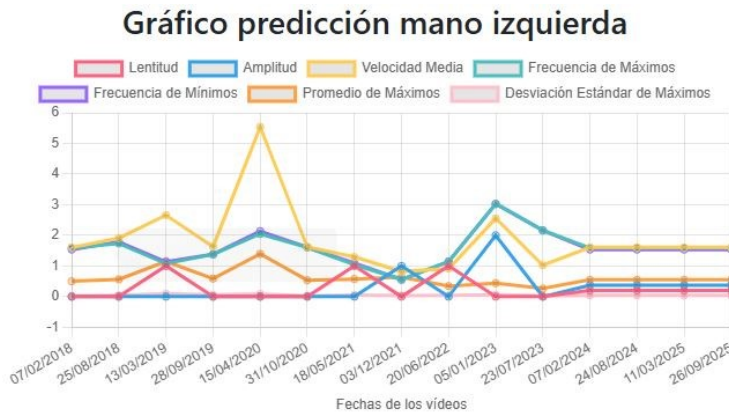


Figura 5.7: Predicción paciente 1 - Legacy-heuristic

SimpleExpSmoothing con «known»

Por último, el modo «known» requiere que el usuario proporcione los valores iniciales: valor inicial, tendencia inicial y estacionalidad inicial. Estos

valores iniciales pueden afectar significativamente al rendimiento del modelo, por lo que encontrar los valores óptimos para los datos utilizados requiere de experimentación. Por esta falta de experimentación, se ha decidido establecer los valores como predeterminados, a cero, obteniendo las predicciones que se observan en la Figura 5.8.

Para la lentitud el modelo predice 0.203 y para la velocidad media predice 1.474, siempre el mismo valor para las diferentes fechas.

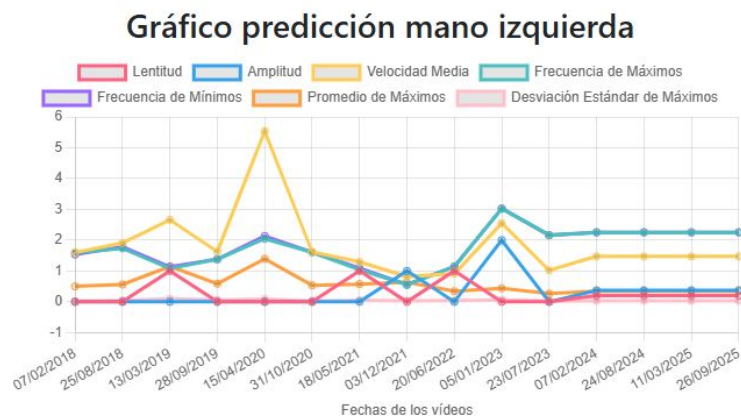


Figura 5.8: Predicción paciente 1 - Known

Como se ha visto, el modelo de suavizado exponencial simple nos proporciona el mismo valor para todas las predicciones futuras (diferentes valores con cada método del modelo). Las posibles razones de este suceso son: la poca variabilidad o escasez de datos, que no permiten al modelo distinguir ningún patrón, la falta de una tendencia clara en los datos o la utilización de un parámetro de suavizado cercano a cero.

Para comprobar esta última, se comprobaron los parámetros de suavizado de cada característica, descubriendo cifras desde 0.16 a 0.76. También se comprobó el resultado de usar parámetros de suavizado mayores o menores, pero seguía apareciendo siempre el mismo valor. Que el parámetro de suavizado o alfa sea cercano a cero significa que la predicción está altamente influenciada por los valores históricos de toda la serie temporal, dando más peso a los valores antiguos.

Por ello se decidió evaluar otro modelo.

Suavizado exponencial de Holt

Este método está diseñado para hacer predicciones (de valores y pendientes) a corto plazo, para series temporales con una tendencia clara de aumento o disminución pero sin tener en cuenta la estacionalidad. El suavizado utiliza dos parámetros, alfa y beta, que poseen valores entre 0 y 1. Cuanto más cercano a cero, menos peso tienen las observaciones más antiguas en la predicción.

Aunque a primera vista no se considere que los datos disponibles tengan una tendencia clara, se decidió probar a utilizar este otro tipo de suavizado para las predicciones, mediante la función `Holt()` del paquete `StatsModels`.

En la Figura 5.9 se observa cómo este modelo ha detectado tendencia en los datos y ha realizado predicciones con pendientes. Para el dato de la lentitud ha predicho un ligero aumento, de 0.274 a 0.32, basándose en los datos disponibles. Para la velocidad media predice un ligero descenso, de 1.243 a 0.602, pudiéndose deber a que, aunque hay picos, la tendencia es descendente.

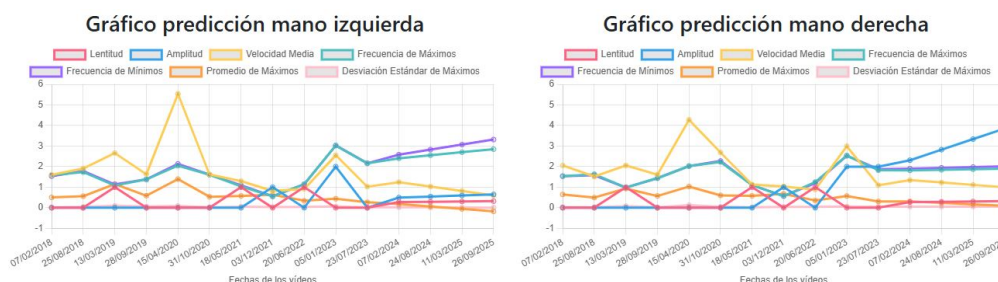


Figura 5.9: Predicción paciente 1 - Holt

El suavizado de Holt también tiene diferentes métodos de inicialización («None», «estimated», «heuristic», «legacy-heuristic» y «known»). Se ha probado con todos ellos, se adjuntan en la tabla 5.1 los datos predichos de lentitud y velocidad media para la mano izquierda del paciente 1.

Tras haber comprobado que este modelo sí es capaz de encontrar cierta tendencia en los datos, se trató de evaluar un modelo más avanzado, el suavizado exponencial de Holt-Winter. Este modelo está diseñado para series temporales con una tendencia y una estacionalidad claras. Utiliza tres parámetros, alfa, beta y gamma para estimar el valor, la pendiente y la estacionalidad de la serie.

Método	Lentitud	Velocidad media
«None»	de 0.274 a 0.32	de 1.243 a 0.602
«estimated»	de 0.327 a 0.355	de 1.337 a 1.025
«heuristic»	de 0.497 a 0.588	de 1.559 a 1.33
«legacy-heuristic»	de 0.274 a 0.32	de 1.243 a 0.602
«known»	de -0.091 a -0.693	de 0.978 a 0.449

Tabla 5.1: Tabla con las predicciones del modelo Holt

Debido a que la función solicita introducir la tendencia y la estacionalidad (y el número de periodos estacionales) y los datos de los vídeos del paciente no tienen estacionalidad, no se pudo probar. Por ello, el modelo elegido para realizar la predicción en la aplicación es el suavizado exponencial de Holt.

5.6. Análisis de la calidad del código

Para garantizar la calidad del código y detectar posibles problemas de seguridad, mantenibilidad o estilo, se ha utilizado la herramienta SonarQube. Se trata de una plataforma de código abierto diseñada para evaluar y mejorar continuamente la calidad del código fuente. Es capaz de encontrar vulnerabilidades, errores, duplicaciones y otras malas prácticas en el código.

El análisis se puede realizar enlazando la herramienta con el repositorio de GitHub o en local, que es la opción escogida. Se instaló y configuró SonarQube y SonarScanner siguiendo las instrucciones proporcionadas en la documentación oficial. Además se creó un fichero con propiedades, necesario para indicar los archivos a analizar y el token del usuario que realiza el escáner. El escaneo se realiza con el comando `sonar-scanner` y los resultados se pueden consultar en el navegador.

El resumen del primer análisis realizado con SonarQube se encuentra en la Figura 5.10.

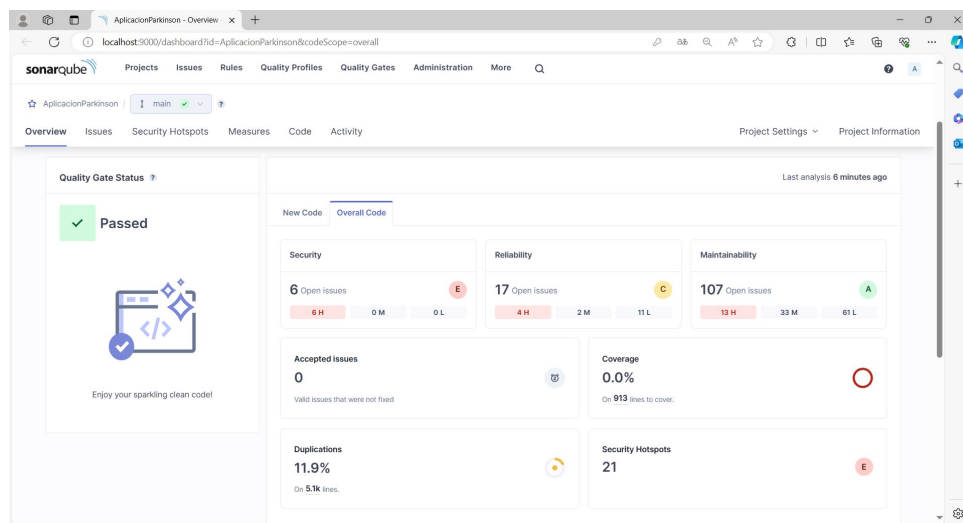


Figura 5.10: Análisis del código con SonarQube

El análisis inicial de SonarQube dio por aprobado el código pero identificó varios problemas de seguridad, entre ellos la falta de protección CSRF (*Cross-Site Request Forgery*) y la exposición de claves secretas y credenciales de la base de datos en el código. Por ello, se realizaron los siguientes cambios en el código:

- Se implementaron *tokens* CSRF para proteger las solicitudes POST contra ataques maliciosos, asegurando rutas sensibles como los formularios.
- Se movió la configuración de la base de datos a un archivo de configuración externo (`config.py`). Este archivo no se debería incluir en el repositorio, con el fin de proteger la contraseña y los datos sensibles de los pacientes.
- También se incluyó en dicho archivo de configuración la clave secreta con la que se firman las sesiones y el CSRF.

Además, indicó algunas incidencias de fiabilidad (*reliability*) y mantenibilidad que, aunque no suponían errores graves, se decidieron solventar. Entre estos cambios se encuentran:

- Cambiar las variables del código que contengan la letra «ñ» con el fin de mejorar la compatibilidad internacional. Para desarrolladores

que no hablen español puede resultar un carácter extraño y ciertos sistemas y herramientas no manejan bien los caracteres no ASCII.

- SonarQube ayudó a detectar variables no utilizadas y código comentado, el cual se eliminó para mejorar la mantenibilidad del código en un futuro.
- Se editaron los descriptores de las imágenes de la aplicación, eliminando la palabra «imagen», ya que SonarQube los consideraba redundantes.
- Se añadieron descriptores explicativos a las tablas de la aplicación.

Tras realizar los cambios pertinentes en el código, se volvió a analizar el código, obteniendo un mejor resultado (Figura 5.11):

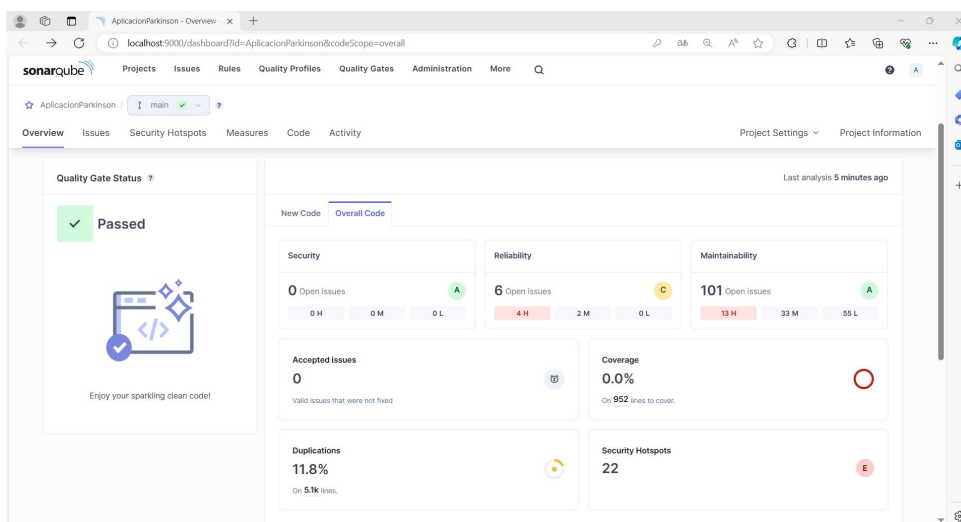


Figura 5.11: Segundo análisis del código con SonarQube

SonarQube ha resultado ser una herramienta muy útil para evaluar y mejorar el código del proyecto, proporcionando recomendaciones prácticas para un código seguro, eficiente y mantenible, que la alumna no hubiera sido capaz de detectar por sí misma.

5.7. Despliegue en un servidor real

El último aspecto a destacar del proyecto es el despliegue exitoso de la aplicación en un servidor real, con un sistema operativo Linux.

El primer paso fue disponer de una máquina real para ponerlo en marcha. Se optó por utilizar un servidor proporcionado por la Universidad de Burgos, de manera que la aplicación estuviera en ejecución y disponible en todo momento de manera confiable y pudiendo satisfacer las demandas de uso.

Tras conseguir desplegar la aplicación en el servidor mediante el servidor de desarrollo integrado de Flask, únicamente destinado a desarrollo y pruebas, se decidió desplegarlo en un servidor de aplicaciones en producción, como establecen las buenas prácticas.

El despliegue en un servidor real como Gunicorn ofrece las siguientes ventajas: permite manejar mayores volúmenes de tráfico que el servidor de desarrollo de flask, es capaz de mantener un alto nivel de rendimiento y proporciona un entorno más robusto.

Tras el despliegue, la aplicación está disponible introduciendo la dirección <http://10.168.168.34:8000/> en el navegador, siempre y cuando se encuentre conectado a la intranet de la UBU. Esto hará más sencilla la conexión a usuarios que quieran probar su funcionamiento, sin necesidad de tener que ejecutarlo en su propio ordenador de manera local.

6. Trabajos relacionados

En este apartado se comentarán trabajos y proyectos realizados dentro del ámbito del proyecto en curso, lo cual proporcionará contexto sobre la situación actual en el mercado, permitiendo explicar la contribución específica de este proyecto.

Se abordarán los siguientes puntos:

- Ámbitos del proyecto.
- Aplicaciones relacionadas con la gestión de datos médicos y el seguimiento de pacientes.
- Aprendizaje automático aplicado a la medicina.
- Otros Trabajos de Fin de Grado (TFGs) relacionados con el párkinson.
- Justificación de la necesidad del proyecto.

6.1. Ámbitos del proyecto

El proyecto fusiona el campo de la informática y la medicina, relacionando la creación de aplicaciones web y el uso de *machine learning* con necesidades médicas, específicamente centradas en la enfermedad de Parkinson.

El campo de la medicina se caracteriza por su privatización, aunque se dispone de una amplia variedad de artículos accesibles en fuentes fiables como bibliotecas universitarias, centros de investigación o revistas científicas.

Algunas de estas fuentes incluyen la Organización Mundial de la Salud³⁰, la PMC³¹, que ofrece revistas biomédicas y artículos de la Biblioteca Nacional de EE.UU., revistas digitales como Journal of Parkinson's Disease³², que publica investigaciones relacionadas con la enfermedad de Parkinson o la propia biblioteca de la UBU³³.

El campo de la informática y la programación web está en constante evolución, con aplicaciones web desempeñando un papel crucial en la vida cotidiana, incluyendo el trabajo, los estudios, el ocio (juegos, redes sociales o comercio electrónico) o la atención médica. Existen multitud de aplicaciones web en el mercado, pero el apartado se centrará en aquellas que permiten visualizar datos médicos a pacientes y personal sanitario.

6.2. Aplicaciones médicas

Es destacable resaltar como tras la época de restricciones debidas a la pandemia, se fomentó el uso de aplicaciones médicas para recibir atención sanitaria, así como para consultar registros médicos o gestionar citas de forma remota.

Se presentan a continuación algunas de estas aplicaciones:

- 75health³⁴: permite a los médicos gestionar historias clínicas y emitir recetas a sus pacientes. Por su parte, los pacientes tienen acceso a sus datos médicos y pueden contactar con su especialista. El problema de esta aplicación es que solo está disponible en Estados Unidos e India, además de que, si se quiere disponer de todas sus funcionalidades, se debe obtener una versión de pago, lo que la hace poco accesible.
- EpicCare EMR³⁵: la aplicación está dirigida a profesionales sanitarios que trabajan en grandes instituciones médicas, proporcionándoles herramientas para administrar la documentación y optimizar su trabajo. A su vez, ofrece a los pacientes la posibilidad de comunicarse con sus médicos y programar citas, incluso mediante videoconferencias, así como acceder a su información e historial médico. Sin embargo,

³⁰<https://www.who.int/es/news-room/fact-sheets/detail/parkinson-disease>

³¹<https://www.ncbi.nlm.nih.gov/pmc/?term=parkinson+disease>

³²<https://www.journalofparkinsonsdisease.com/>

³³https://ubucat.ubu.es/discovery/search?vid=34BUC_UBU:VU1

³⁴<https://www.75health.com/>

³⁵<https://www.emrsystems.net/epic-emr-software/>

no cuenta con una versión gratuita, aunque se puede solicitar una demostración. Asimismo, está disponible únicamente en inglés.

- MyChart³⁶: otra aplicación similar a las mencionadas anteriormente. Permite la visualización de resultados de laboratorio, historial médico, programación de citas y comunicación con el médico, entre otras funcionalidades. Aunque, nuevamente, no se encuentra disponible en España.
- PatientsLikeMe³⁷: en este caso se trata de una aplicación web destinada a pacientes con enfermedades crónicas como es el párkinson. Se utiliza mayoritariamente para compartir experiencias y obtener apoyo del resto de usuarios. Además, permite el seguimiento de la enfermedad y los tratamientos. Por desgracia, esta aplicación está igualmente solo disponible en inglés.

6.3. Aprendizaje automático aplicado a la medicina

El campo del *machine learning* o aprendizaje automático aplicado a la medicina es muy innovador. Existen proyectos que analizan grandes cantidades de datos médicos para realizar predicciones que ayudan con diagnósticos y tratamientos.

Un ejemplo es un proyecto publicado en la base de datos científica de la Universidad de La Rioja, donde se analizaron 13 métodos de *machine learning* para comprobar cuál era más útil en la predicción de la diabetes mellitus tipo 2 en pacientes mayores de edad [13]. Los resultados indicaron que el modelo LightGBM demostró mejores resultados de precisión, sensibilidad, tasa de clasificación errónea, etc. Dado que la diabetes es una de las diez primeras causas de mortalidad entre la población adulta, la capacidad de realizar diagnósticos tempranos de forma rápida gracias a la inteligencia artificial puede ser crucial para prevenir futuras complicaciones.

Otro buen proyecto relacionado con el uso del aprendizaje automático para predecir enfermedades es el Trabajo de Fin de Grado entregado por Javier Pérez Córdova en la Universidad de Cataluña [16]. En este estudio se tratan de aplicar modelos de *machine learning* basados en árboles de decisión para detectar el cáncer de mama de forma menos invasiva que mediante

³⁶<https://www.mychart.org/Features>

³⁷<https://www.patientslikeme.com/>

una mamografía convencional. Se evaluaron diversos métodos, dando lugar a resultados más y menos óptimos, que sugieren la necesidad de disponer de más datos para mejorar la eficacia de los modelos y conseguir mejores resultados.

Resulta interesante considerar la inversión en proyectos de este tipo ya que pueden contribuir a la detección temprana de enfermedades como el cáncer de mama, cuya incidencia está aumentando en España, ofreciendo un método de diagnóstico menos doloroso para los pacientes.

6.4. Otros TFGs relacionados con el párkinson

Otro ejemplo del uso de *machine learning* para asistir, en este caso, a pacientes con párkinson, es el TFG realizado por un compañero de la carrera el curso pasado. Debido al acuerdo de colaboración entre la UBU y la Asociación Parkinson Burgos, se han llevado a cabo varios Trabajos de Fin de Grado relacionados con esta enfermedad. En este caso se aborda el proyecto de Catalin Andrei Cacuci, estudiante de Ingeniería Informática, quien presentó el curso pasado un TFG titulado «Identificación de Parkinson por visión artificial»³⁸.

El proyecto se centró en la creación de un sistema capaz de detectar, mediante visión artificial, la presencia de bradicinesia (un síntoma presente en personas con párkinson que se manifiesta como ralentización del movimiento). Los individuos debían grabar un vídeo haciendo un movimiento de pinza con los dedos índice y pulgar, y el sistema detectaba cualquier alteración en el movimiento.

Este proyecto podría ser muy beneficioso para facilitar un primer diagnóstico, especialmente tras el aumento del uso de la telemedicina durante la pandemia.

Existen más Trabajos de Fin de Grado realizados en la UBU en colaboración con esta asociación, pero se destaca el realizado por Sara González Bárcena, estudiante de Ingeniería de la Salud, quien presentó el curso pasado un TFG titulado «Detección de la actividad muscular de las personas con enfermedad de Parkinson»³⁹.

³⁸<https://github.com/cataand/tfg-paddel>

³⁹https://github.com/saragonzalezbarcelona/TFG_Deteccion_Activ_Muscular

Este proyecto se incluye en la sección de trabajos relacionados debido a que describe el desarrollo de un sensor diseñado para analizar la marcha de pacientes con párkinson, similar al sensor desarrollado por SENSE4CARE [17] en 2020 que genera los archivos CSV de datos utilizados en este proyecto. Específicamente, el sensor desarrollado por la alumna es capaz de analizar la duración del ejercicio, el número de bloqueos durante el período de actividad y detectar desequilibrios entre ambos lados del cuerpo del paciente. Incorpora sensores inerciales (acelerómetro y giroscopio) para detectar la información.

Estos proyectos son muy importantes, ya que diseñan dispositivos muy específicos que, aunque son poco comunes y costosos, son muy útiles para ayudar al personal sanitario a diseñar terapias más precisas y personalizadas, mejorando la calidad de vida de los pacientes.

6.5. Justificación de la necesidad del proyecto

Después de exponer el contexto actual relacionado con los temas abordados en el proyecto, se procede a justificar resumidamente la necesidad de este.

Como se ha podido observar, las aplicaciones y plataformas web están a la orden del día. Son utilizadas diariamente por personas de todas las edades y en ámbitos de la vida muy variados. En el ámbito médico en específico, se empezaron a popularizar durante la cuarentena vivida en el año 2020, cuando las restricciones impidieron el acceso físico a los servicios de salud, lo que resultó en la detección tardía de enfermedades o monitorizaciones deficientes de las enfermedades de los pacientes, afectando gravemente a su salud.

Herramientas como la desarrollada en este proyecto, que permiten a pacientes y médicos llevar un seguimiento de la enfermedad de forma remota, e incluso predecir la tendencia de la enfermedad a lo largo del tiempo, podrían haber sido de mucha utilidad tanto durante la pandemia como en la actualidad.

Aunque existen y se han mostrado aplicaciones disponibles actualmente en el mercado que ofrecen funcionalidades similares, la mayoría de ellas están disponibles solo en inglés y son de pago. Por lo tanto, existe una necesidad de desarrollar aplicaciones web como la de este proyecto, accesibles para pacientes de habla hispana, en concreto para pacientes con párkinson, y

gratuitas, resultando asequibles para personas de cualquier nivel socio-económico.

7. Conclusiones y Líneas de trabajo futuras

En este último apartado de la memoria se abordan las conclusiones derivadas del desarrollo del proyecto. En concreto se comentan conclusiones relacionadas con los resultados del proyecto y un conjunto de conclusiones técnicas y personales.

Además, tras las conclusiones, se comentan ideas de mejora del proyecto. Estas ideas pueden servir de guía de trabajo a nuevos estudiantes que decidan continuar con el proyecto en el futuro.

7.1. Conclusiones

A continuación se describen las conclusiones relacionadas con los resultados del proyecto, así como conclusiones técnicas y personales.

Resultados del proyecto

La aplicación web desarrollada cumple con todas las especificaciones que fueron solicitadas por el Hospital. Es completamente funcional y ha sido diseñada según los requerimientos planteados inicialmente para permitir un seguimiento continuo de los pacientes con párkinson. Cabe destacar que la idea original del proyecto fue una necesidad expresada durante un congreso de medicina, con el objetivo de monitorizar la evolución de los pacientes, no fue una concepción de la alumna.

Aunque la aplicación no destaca por su diseño estético, ya que ha sido desarrollada utilizando Flask y HTML en lugar de plataformas como

WordPress que facilitan la creación de interfaces más atractivas, cumple su propósito principal de manera efectiva. Además de la idea inicial, a lo largo del desarrollo surgieron algunas propuestas para mejorar la aplicación y darla por finalizada, las cuales se detallarán en la próxima sección.

Si el hospital decide finalmente utilizar esta aplicación web con pacientes reales, con sus datos y vídeos reales recogidos a lo largo de cierto periodo de tiempo, se espera que tenga un impacto significativo tanto para la asociación de Parkinson Burgos como para toda la comunidad involucrada.

Conclusiones técnicas

La conclusión técnica más relevante de este proyecto es que los modelos de inteligencia artificial puede ofrecer predicciones con mucha relevancia en el campo de la medicina, ya que conocer la evolución de los pacientes puede facilitar la toma de decisiones a los médicos. Además el uso de dispositivos con sensores que detecten alteraciones en el movimiento, junto con una herramienta para mostrar dichos datos de forma que sean comprensibles, es vital en el seguimiento de los pacientes.

Cabe destacar que la aplicación se ha desplegado en un servidor real, propiedad de la Universidad de Burgos. Además es importante resaltar que se han implementado medidas de seguridad para proteger la integridad y confidencialidad de los datos de los pacientes, como métodos de autenticación y control de accesos o el cifrado de datos sensibles como la contraseña de los usuarios en la base de datos.

Conclusiones personales

El proceso de desarrollo del proyecto ha sido una experiencia de aprendizaje significativa para la alumna. Se han repasado y aplicado los conocimientos adquiridos a lo largo de su carrera universitaria, desde aspectos fundamentales como bases de datos y programación, hasta el diseño y gestión de proyectos.

La implementación de la aplicación ha supuesto la adquisición de conocimientos en desarrollo web, un área en la que la alumna no tenía experiencia previa. Además, demuestra su capacidad de aprendizaje y adaptación a nuevos entornos tecnológicos.

La implementación de técnicas de inteligencia artificial, en particular en el contexto de las series temporales, ha permitido a la alumna profundizar en este campo más allá de lo que se había cubierto en sus estudios. La

exposición a ejemplos reales y la dedicación necesaria para comprender y aplicar estas técnicas mejoró su comprensión en este ámbito.

El proceso de desarrollo del proyecto ha requerido una dedicación constante y disciplinada. Durante los primeros meses, el proyecto avanzaba poco, por lo que en las etapas finales se tuvieron que intensificar los esfuerzos para cumplir con los plazos establecidos. Con disciplina se consiguió trabajar a diario en el proyecto, compaginándolo con las prácticas en empresa, aunque supusiera un esfuerzo extra y una reducción notable del tiempo personal. Como conclusión personal, el proyecto ha resaltado la importancia de la gestión del tiempo y el compromiso personal desde el principio para conseguir los objetivos.

7.2. Líneas de trabajo futuras

En este apartado se pretende realizar informe crítico indicando cómo se puede mejorar el proyecto, o cómo se puede continuar trabajando en la línea del proyecto realizado.

Subida asíncrona de los vídeos

Actualmente, el proceso de subida de vídeos una vez que el médico ha rellenado el formulario de subida, implica añadir el vídeo a la base de datos y procesarlo mediante funciones de análisis para extraer sus características relevantes. Este proceso de análisis puede tardar varios minutos, en los que el usuario de la aplicación debe esperar. Son pocos minutos debido a que la aplicación se encuentra en un entorno local, sin embargo, al desplegar la página web en un entorno de producción, existe el riesgo de que este proceso supere los límites de tiempo establecidos y resulte en un error de *timeout*.

Para mitigar este problema, se propone como mejora del proyecto implementar la subida de vídeos de forma asíncrona. Esto permitiría que el proceso de subida y análisis se realice en segundo plano, sin bloquear la interfaz de usuario principal, garantizando una experiencia fluida para los usuarios y evitando posibles errores de *timeout*.

Conseguir mejores datos

Otra mejora significativa sería la adquisición de conjuntos de datos más robustos y específicos. Esto incluiría recopilar múltiples vídeos reales de un mismo paciente a lo largo del tiempo, realizando la técnica de *fingertapping*,

para muchos pacientes. Esto permitiría evaluar con mayor precisión la eficacia de la aplicación al monitorizar y predecir la progresión de dicha técnica en un mismo individuo. Actualmente, la aplicación se evalúa con vídeos de personas diferentes debido a la falta de disponibilidad de datos específicos, aunque se ha intentado mantener la homogeneidad en términos de sexo y edad entre los pacientes.

Además, se debería mejorar la calidad de los datos recopilados por el sensor que llevan incorporados los pacientes. Se descartó la idea de implementar inteligencia artificial para predecir el desarrollo de estos datos en el futuro debido a la presencia de una gran cantidad de valores nulos, lo que haría que el modelo no pudiera identificar buenos patrones de predicción. Con datos de mayor calidad, se abriría la posibilidad de ampliar las funcionalidades de la aplicación, incluyendo la predicción con IA no solo en las características de los vídeos, sino también en los datos recogidos por el sensor.

Funcionalidades del UPDRS

Se plantea en un futuro añadir a la aplicación web más funcionalidades para evaluar la progresión de la enfermedad. En el presente proyecto no se han añadido porque no lo requiso así el cliente.

Actualmente el proyecto se ha centrado, además de en visualizar los datos de los sensores, en determinar la gravedad de los síntomas de la enfermedad mediante vídeos de pacientes realizando un golpeteo con los dedos. Esta funcionalidad corresponde con la evaluación número 26, dentro del apartado de aspectos motores de la escala UPDRS [18]. El paciente golpea el pulgar con el índice en rápida sucesión y con la mayor amplitud posible con ambas manos y el médico califica el movimiento en una escala de 0 a 4, donde 0 indica ausencia de discapacidad y 4 indica una discapacidad severa que prácticamente impide realizar la acción.

Como se explica en la sección de conceptos teóricos, el UPDRS es una herramienta clínica ampliamente utilizada en todo el mundo para evaluar la gravedad de la enfermedad de Parkinson. Consta de 45 preguntas, divididas en cuatro apartados (estado mental, actividades de la vida diaria, aspectos motores y complicaciones del tratamiento). El proyecto presentado se centra en una única pregunta del apartado de aspectos motores, pero se plantea la posibilidad de incluir otras evaluaciones.

La aplicación web podría centrarse en el apartado de aspectos motores, desarrollando funciones como la creada por Catalin [5] para detectar temblor o rigidez en ciertos movimientos de los pacientes, quienes podrían grabar

vídeos realizando el movimiento desde sus propias casas y subirlos a la aplicación, dónde sus médicos podrían analizar los resultados. Dentro del apartado existen varias preguntas sobre la cantidad de temblores en ciertas ocasiones o sobre la realización de movimientos que involucran diferentes partes del cuerpo (abrir y cerrar las manos, movimientos de pronación-supinación de las manos, golpes de talón, levantamientos de una silla. . .). Todos ellos podrían analizarse mediante visión artificial.

Otra opción podría ser llevar un registro de los informes de los pacientes, con todos los apartados medidos con la escala por sus médicos a lo largo del tiempo, e incluir alguna herramienta capaz de compararlos y visualizar el avance de la enfermedad.

Comunicación médico-paciente

Como línea de trabajo futura, se propone la implementación de una funcionalidad adicional que surgió por parte de la alumna durante la fase de desarrollo del proyecto: la creación de un apartado que permita la comunicación entre médicos y pacientes.

Este apartado permitirá a los médicos establecer una comunicación directa con sus pacientes a través de un sistema de mensajería, pudiendo, por ejemplo, comentar y analizar los datos obtenidos por el sensor o las conclusiones obtenidas a partir de los vídeos y sus predicciones. Además, servirá como canal para solicitar citas presenciales o realizar consultas rápidas en tiempo real mediante videollamadas.

Con este apartado se busca mejorar la continuidad de la atención médica, ofreciendo un seguimiento más cercano y personalizado al paciente, incluso fuera del ámbito clínico. Además, se pretende simplificar y agilizar el proceso de solicitud y realización de citas para los pacientes. Mediante videollamadas, se elimina la barrera geográfica para aquellos que viven en áreas remotas sin acceso fácil al transporte o para aquellos con movilidad reducida, facilitando así su acceso a la atención médica.

Sección de ayuda para el día a día

Una idea adicional que se considera es la creación de una sección más centrada en el aspecto humano. Aquí, los pacientes encontrarían recursos para sobrellevar mejor el párkinson en su vida diaria, como consejos prácticos respaldados por artículos científicos o recomendaciones de productos útiles para manejar los temblores y otros síntomas asociados con la enfermedad.

Por ejemplo, se podría proporcionar información sobre técnicas de ejercicio, terapias complementarias y hábitos de vida saludables que hayan demostrado beneficios en la gestión del párkinson, así como dispositivos novedosos que ayuden a controlar los temblores.

En relación a esa idea, se plantea la creación de un foro comunitario donde los pacientes puedan intercambiar mensajes de apoyo y motivación entre ellos, así como cualquier otro contenido que promueva el bienestar emocional y el sentimiento de comunidad entre pacientes con párkinson. Para garantizar un entorno seguro y respetuoso, se implementaría un filtro mediante inteligencia artificial que moderaría los mensajes, evitando la presencia de ciertas palabras inapropiadas o contenido perjudicial.

Estas iniciativas más humanas, tienen como objetivo principal mejorar la calidad de vida de los pacientes.

Realizar conexión segura entre la aplicación y el servidor

Al acceder a la aplicación web «TremorTrack» desde el navegador aparece el mensaje «No es seguro» o un candado rojo junto a la URL, lo que no aporta mucha confianza al usuario que quiera hacer uso de la aplicación. Esto se debe a que la conexión entre el navegador y el servidor no está protegida mediante un certificado SSL válido.

Se trató de usar `openssl` para hacer la conexión segura pero no fue posible por no disponer de permisos de administrador en el servidor en el que fue desplegada la aplicación.

Es importante que la aplicación sea segura, ya que se trata de una aplicación médica en la que se van a manejar datos sensibles de los pacientes. Además, garantizaría la integridad y confidencialidad de los datos, evitando que los datos personales transmitidos entre el navegador y el servidor sean adquiridos o modificados por terceros, gracias a su encriptación.

Se ha incluido esta última sección en el apartado de líneas de trabajo futuras para fomentar la realización de este cambio. La idea barajada es utilizar el servidor web/Proxy NGINX junto con un certificado SSL para configurar una conexión segura HTTPS⁴⁰.

⁴⁰<https://a4u.medium.com/deploy-flask-app-with-gunicorn-ssl-16460ec14c24>

Bibliografía

- [1] U de Alcalá. Scikit-Learn, herramienta básica para el Data Science en Python. *Recuperado el*, 29, 2019.
- [2] Altexsoft. Most Popular Responsive CSS Frameworks: Bootstrap, Foundation, Materialize, Pure, and more, 2018. [Internet; descargado 15-diciembre-2023].
- [3] AWS. ¿Cuál es la diferencia entre MariaDB y MySQL?, 2023. [Internet; descargado 13-enero-2024].
- [4] Viviana Bruno. Alternativas a Bootstrap, 2019. [Internet; descargado 15-diciembre-2023].
- [5] Catalin Andrei Cacuci. Identificación de Parkinson por visión artificial, 2023.
- [6] José H Canós, Patricio Letelier, and M^a Carmen Penadés. Metodologías ágiles en el desarrollo de software. *Universidad Politécnica de Valencia, Valencia*, pages 1–8, 2003.
- [7] Marsden CD. Parkinson’s disease. 1994.
- [8] DataScientest. Matplotlib: todo lo que tienes que saber sobre la librería Python de Dataviz, 2022. [Internet; descargado 17-diciembre-2023].
- [9] Mauricio Developer. HeidiSQL: La herramienta imprescindible para administrar bases de datos MySQL, 2023. [Internet; descargado 13-enero-2024].
- [10] Epitech. Qué es Flask (Python) y cuáles son sus principales ventajas, 2021. [Internet; descargado 17-diciembre-2023].

- [11] IBM. ¿Qué es machine learning? [Internet; descargado 20-marzo-2024].
- [12] Pablo Londoño. Qué es Python, para qué sirve y cómo se usa (+ recursos para aprender), 2023. [Internet; descargado 20-diciembre-2023].
- [13] Jaime Yelsin Rosales Malpartida. Predicción de diabetes mellitus tipo 2 utilizando atributos médicos del Policlínico Leo SAC de San Juan de Lurigancho mediante el enfoque de Machine Learning. *TecnoHumanismo*, 2(4):1–19, 2022.
- [14] Pharma Market. La SEN estima que el número de personas con enfermedad de Parkinson se triplicará en los próximos 30 años en España, 2022. [Internet; descargado 27-mayo-2024].
- [15] Devin Partida. Jira vs Zenhub (2023): What are the main differences?, 2023. [Internet; descargado 17-diciembre-2023].
- [16] Javier Pérez Córdova. Técnicas de machine learning aplicadas a la búsqueda de biomarcadores de cáncer de mama. 2021.
- [17] SENSE4CARE. Experts in the detection and analysis of human movement patterns applied to healthcare. [Internet; descargado 22-mayo-2024].
- [18] Dr. Stanley Fahn y Dr. Ronald Elton. Unified Parkinson’s disease rating scale (UPDRS), 1987. [Internet; descargado 22-mayo-2024].