



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 — 5 de abril de 2024

Tutor: Álgvar Anaiz González

Cotutora: Alicia Olivares Gil



UNIVERSIDAD DE BURGOS
ESCUELA POLITÉCNICA SUPERIOR
Grado en Ingeniería Informática



D. Álvar Arnaiz González, profesor del departamento de Ingeniería Informática, área de Lenguajes y Sistemas Informáticos.

Expone:

Que el alumno D. Sandra Díaz Aguilar, con DNI 71309532X, ha realizado el Trabajo final de Grado en Ingeniería Informática titulado Aplicación web para pacientes con Parkinson.

Y que dicho trabajo ha sido realizado por el alumno bajo la dirección del que suscribe, en virtud de lo cual se autoriza su presentación y defensa.

En Burgos, 5 de abril de 2024

Vº. Bº. del Tutor:

Vº. Bº. del co-tutor:

D. nombre tutor

D. nombre co-tutor

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. Se ha desarrollado una aplicación web en la que pacientes y médicos puedan acceder a gráficos 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 predecir, según estos datos, cómo avanzará la enfermedad. La aplicación también permite a los médicos gestionar otra información y vídeos de sus pacientes.

Descriptores

Enfermedad de Parkinson, sensor, desarrollo web, aplicación web, datos médicos, análisis de datos, 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. A web application has been developed, where patients and doctors can access graphics with data that was measured by a sensor that patients have incorporated. It is accessible in a simple and easy way. The app also uses machine learning to predict, based on this data, how the disease will progress. Furthermore, the app also allows doctors to manage other information and videos of their patients.

Keywords

Parkinson's disease, sensor, web development, web application, medical data, data análisis, machine learning,...

Índice general

Índice general	iii
Índice de figuras	iv
Índice de tablas	v
1. Introducción	1
2. Objetivos del proyecto	3
3. Conceptos teóricos	5
3.1. Enfermedad de Parkinson	6
3.2. <i>Machine Learning</i>	7
4. Técnicas y herramientas	9
5. Aspectos relevantes del desarrollo del proyecto	19
6. Trabajos relacionados	21
7. Conclusiones y Líneas de trabajo futuras	27
Bibliografía	29

Índice de figuras

Índice de tablas

1. Introducción

Descripción del contenido del trabajo y del estructura de la memoria y del resto de materiales entregados.

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 a 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 entender, 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 mediante MariaDB 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, que muestre 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 por la biblioteca Scikit-learn 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.

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 de objetivos, 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, así como obtener predicciones sobre la evolución de estos datos en el futuro. Además, facilita a los médicos la posibilidad de cargar vídeos de los pacientes realizando la técnica de “*finger tapping*”, la cual es una forma adicional de detectar el párkinson.

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

- La enfermedad de Parkinson, la cual no se limita únicamente a temblores, y se explicará por qué es importante medir los movimientos del paciente con un sensor y tener controlada su progresión. También se explicará la técnica del “*finger tapping*” utilizada para evaluar la progresión de la enfermedad.
- Conceptos fundamentales de aprendizaje automático empleados en el proyecto para la predicción de las gráficas futuras, incluyendo una explicación de las series temporales, que es la manera en que se gestionan los datos del sensor.

3.1. Enfermedad de Parkinson

El párkinson [6] 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 50 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 de movimiento) 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 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 desarrolló el sensor STAT-ON, el cual mide parámetros como los bloqueos, la marcha bradicinética, discinesia, el estado motor, etc., 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. Además, la predicción de estos parámetros a futuro puede significar una ayuda extra.

La aplicación web desarrollada también permite a estos médicos llevar un control de los vídeos de los pacientes realizando la técnica de “*fingertapping*”. 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>

3.2. Machine Learning

El aprendizaje automático, también conocido como *machine learning* [10], 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 semisupervisado:

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 semisupervisado: combina las dos anteriores, resolviendo el problema de no tener suficientes datos etiquetados para realizar un aprendizaje supervisado.

En este proyecto se utilizan técnicas de aprendizaje supervisado para predecir la evolución de los datos del sensor del paciente en el futuro, utilizando datos previamente recopilados y etiquetados con la progresión de la enfermedad. El proceso que se lleva a cabo para entrenar el modelo es el siguiente:

1. Recopilación de datos: se recopilan los datos del sensor que se tengan de ese paciente, junto con otra información de la evolución de la enfermedad.
2. Preparación de datos: estos datos se preparan para el entrenamiento del modelo. Esto incluye la limpieza de datos erróneos, la normalización, etc. Además se dividen en dos conjuntos, uno de entrenamiento y otro de pruebas.

3. Entrenamiento del modelo: se ha escogido X como algoritmo de aprendizaje supervisado y se procede a entrenarlo utilizando el conjunto de entrenamiento. Durante el entrenamiento, el modelo trata de minimizar la diferencia entre sus predicciones y los valores reales observados. Posteriormente, con el conjunto de prueba, se realiza la evaluación de la exactitud del modelo con métricas como el error cuadrático medio que mide el promedio de errores obtenidos.
4. Predicciones: si se obtiene un buen modelo, este va a ser utilizado en la aplicación para predecir la evolución de los datos del sensor a partir de los datos actuales.

Los datos del sensor se presentan ordenados cronológicamente, como series temporales. Se recogen los datos de cada apartado cada 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 precisos que se anticipen a la progresión de la enfermedad permitiendo ajustar los tratamientos en consecuencia.

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.

Gestión de proyectos

Técnica

Durante la gestión del proyecto se ha optado por usar una metodología ágil [5]. 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, permitiéndote concentrarte 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 [13] 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 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

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

³<https://www.zenhub.com/>

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.

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 [11] 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

- Scikit-learn⁴ [1]: librería gratuita (código abierto) para Python que cuenta con gran variedad de algoritmos de aprendizaje automático como *clustering*, clasificación o regresión. Se utilizará durante el proyecto para la predicción de la evolución de los pacientes. Además permite realizar el preprocesamiento de los datos de forma sencilla y una evaluación 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 [7]: biblioteca gratuita utilizada para crear visualizaciones estáticas, animadas o interactivas en Python. Se utilizará para mostrar

⁴<https://scikit-learn.org/stable/>

⁵<https://matplotlib.org/>

gráficamente 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, etc
- 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.
- Leaflet⁸: 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.

Frameworks

- Flask⁹ [9]: 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

⁶<https://pandas.pydata.org/>

⁷<https://numpy.org/>

⁸<https://leafletjs.com/>

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

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

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.

Dentro de Flask se estudió el diseño web adaptable (*responsive*):

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* receptivo, que facilite la creación de interfaces ajustables.

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

- Bootstrap¹³: 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 4 clases (teléfono, tableta, portátiles pequeños y normales), útiles para crear el diseño adaptable que se busca. Dispone de componentes predefinidos como botones, formularios o desplegados, 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 CSS¹⁴: 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.

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

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

¹³<https://getbootstrap.com/>

¹⁴<https://materializecss.com/>

- ZURB Foundation¹⁵: 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.

Entornos de desarrollo

- Jupyter Notebook¹⁶ 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. 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.

¹⁵<https://get.foundation/>

¹⁶<https://jupyter.org/>

Editor de texto

Para llevar a cabo la documentación del proyecto (la memoria y anexos) se escogió entre 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 Overleaf como herramienta para editar los documentos de L^AT_EX de forma colaborativa en la nube.

Prototipado

Para realizar el diseño de interfaces de la aplicación web se ha utilizado la herramienta Pencil¹⁷.

Se barajó la idea de utilizar Adobe XD¹⁸, 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.

¹⁷<https://pencil.evolus.vn/Features.html>

¹⁸<https://helpx.adobe.com/es/xd/user-guide.html>

Para realizar los diagramas del proyecto se ha utilizado Draw.io¹⁹, 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.

Base de datos

Se necesitó escoger una herramienta para organizar los datos de los pacientes recolectados por el sensor, así como 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 [8], 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.

¹⁹<https://app.diagrams.net/>

Repositorio

La plataforma elegida para llevar el control de versiones y el *hosting* del repositorio ha sido GitHub²⁰.

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.

Otros recursos utilizados

- Mockaroo²¹: 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 aleatorias²²: 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

²⁰<https://github.com/about>

²¹<https://www.mockaroo.com/>

²²<https://thispersondoesnotexist.com/>

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.

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 en marzo de 2020.

Este dispositivo es el encargado de recopilar los datos del paciente que lo lleva y generar archivos CSV que contienen la información que se utiliza en el proyecto para generar las gráficas.

Los archivos CSV resultantes almacenan 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.

Alguno de los datos medidos por el sensor son: la marcha bradicinética, discinesia, estado motor, bloqueos de la marcha, parámetros de la marcha, transiciones posturales y detalles sobre la postura del paciente.

5. Aspectos relevantes del desarrollo del proyecto

Este apartado pretende recoger los aspectos más interesantes del desarrollo del proyecto, comentados por los autores del mismo. Debe incluir desde la exposición del ciclo de vida utilizado, hasta los detalles de mayor relevancia de las fases de análisis, diseño e implementación. Se busca que no sea una mera operación de copiar y pegar diagramas y extractos del código fuente, sino que realmente se justifiquen los caminos de solución que se han tomado, especialmente aquellos que no sean triviales. Puede ser el lugar más adecuado para documentar los aspectos más interesantes del diseño y de la implementación, con un mayor hincapié en aspectos tales como el tipo de arquitectura elegido, los índices de las tablas de la base de datos, normalización y desnormalización, distribución en ficheros³, reglas de negocio dentro de las bases de datos (EDVHV GH GDWRV DFWLYDV), aspectos de desarrollo relacionados con el WWW... Este apartado, debe convertirse en el resumen de la experiencia práctica del proyecto, y por sí mismo justifica que la memoria se convierta en un documento útil, fuente de referencia para los autores, los tutores y futuros alumnos.

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 Finales de Grado(TFGs) relacionados con el párkinson.
- Justificación de la necesidad del proyecto.

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 Salud²³, la PMC²⁴, que ofrece revistas biomédicas y artículos de la Biblioteca Nacional de EE.UU., revistas digitales como Journal of Parkinson's Disease²⁵, que publica investigaciones relacionadas con la enfermedad de Parkinson o la propia biblioteca de la UBU²⁶.

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.

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:

- 75health²⁷: 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 EMR²⁸: 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 y historial médico. Sin embargo,

²³<https://www.who.int/es/news-room/fact-sheets/detail/parkinson-disease>

²⁴<https://www.ncbi.nlm.nih.gov/pmc/?term=parkinson+disease>

²⁵<https://www.journalofparkinsonsdisease.com/>

²⁶https://ubucat.ubu.es/discovery/search?vid=34BUC_UBU:VU1

²⁷<https://www.75health.com/>

²⁸<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.

- MyChart²⁹: 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.
- PatientsLikeMe³⁰: 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.

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 [12]. 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 (TFG) entregado por Javier Pérez Córdova en la Universidad de Cataluña [14]. 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 una mamografía convencional. Se evaluaron diversos métodos, dando lugar a resultados más y menos óptimos, que sugieren la necesidad de

²⁹<https://www.mychart.org/Features>

³⁰<https://www.patientslikeme.com/>

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.

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”³¹.

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”³².

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 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

³¹<https://github.com/cataand/tfg-paddel>

³²https://github.com/saragonzalezbarcelona/TFG_Deteccion_Activ_Muscular

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.

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

Todo proyecto debe incluir las conclusiones que se derivan de su desarrollo. Éstas pueden ser de diferente índole, dependiendo de la tipología del proyecto, pero normalmente van a estar presentes un conjunto de conclusiones relacionadas con los resultados del proyecto y un conjunto de conclusiones técnicas. Además, resulta muy útil realizar un informe crítico indicando cómo se puede mejorar el proyecto, o cómo se puede continuar trabajando en la línea del proyecto realizado.

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] 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.
- [6] Marsden CD. Parkinson’s disease. 1994.
- [7] DataScientest. Matplotlib: todo lo que tienes que saber sobre la librería python de dataviz, 2022. [Internet; descargado 17-diciembre-2023].
- [8] Mauricio Developer. Heidisql: La herramienta imprescindible para administrar bases de datos mysql, 2023. [Internet; descargado 13-enero-2024].
- [9] Epitech. Qué es flask (python) y cuáles son sus principales ventajas, 2021. [Internet; descargado 17-diciembre-2023].
- [10] IBM. ¿qué es machine learning? [Internet; descargado 20-marzo-2024].

- [11] Pablo Londoño. Qué es python, para qué sirve y cómo se usa (+ recursos para aprender), 2023. [Internet; descargado 20-diciembre-2023].
- [12] 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.
- [13] Devin Partida. Jira vs zenhub (2023): What are the main differences?, 2023. [Internet; descargado 17-diciembre-2023].
- [14] Javier Pérez Córdova. Técnicas de machine learning aplicadas a la búsqueda de biomarcadores de cáncer de mama. 2021.