



UNIVERSIDAD DE BURGOS  
ESCUELA POLITÉCNICA SUPERIOR  
Grado en Ingeniería Informática



**TFG del Grado en Ingeniería  
Informática**

**Aplicación web para pacientes  
con párkinson  
Documentación Técnica**



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

Tutor: Álvar Anaiz González  
Cotutora: Alicia Olivares Gil



---

# Índice general

---

<b>Índice general</b>	<b>i</b>
<b>Índice de figuras</b>	<b>iii</b>
<b>Índice de tablas</b>	<b>vi</b>
<b>Apéndice A Plan de Proyecto Software</b>	<b>1</b>
A.1. Introducción . . . . .	1
A.2. Planificación temporal . . . . .	1
A.3. Estudio de viabilidad . . . . .	27
<b>Apéndice B Especificación de Requisitos</b>	<b>37</b>
B.1. Introducción . . . . .	37
B.2. Objetivos generales . . . . .	38
B.3. Catálogo de requisitos . . . . .	38
B.4. Especificación de requisitos . . . . .	41
<b>Apéndice C Especificación de diseño</b>	<b>55</b>
C.1. Introducción . . . . .	55
C.2. Diseño de datos . . . . .	56
C.3. Diseño de interfaces . . . . .	60
C.4. Diseño procedimental . . . . .	74
C.5. Diseño arquitectónico . . . . .	79
<b>Apéndice D Documentación técnica de programación</b>	<b>81</b>
D.1. Introducción . . . . .	81
D.2. Estructura de directorios . . . . .	81

D.3. Manual del programador . . . . .	84
D.4. Compilación, instalación y ejecución del proyecto . . . . .	86
D.5. Pruebas del sistema . . . . .	89
<b>Apéndice E Documentación de usuario</b>	<b>103</b>
E.1. Introducción . . . . .	103
E.2. Requisitos de usuarios . . . . .	103
E.3. Instalación . . . . .	104
E.4. Manual del usuario . . . . .	104
<b>Apéndice F Anexo de sostenibilización curricular</b>	<b>125</b>
F.1. Introducción . . . . .	125
F.2. Envejecimiento saludable . . . . .	125
F.3. Materiales de los sensores . . . . .	126
F.4. <i>Software</i> . . . . .	127
<b>Bibliografía</b>	<b>129</b>

---

# Índice de figuras

---

A.1. Gráfico <i>burndown</i> - <i>Sprint 1</i> . . . . .	3
A.2. Informe de estado - <i>Sprint 1</i> . . . . .	3
A.3. Gráfico <i>burndown</i> - <i>Sprint 2</i> . . . . .	4
A.4. Informe de estado - <i>Sprint 2</i> . . . . .	5
A.5. Gráfico <i>burndown</i> - <i>Sprint 3</i> . . . . .	6
A.6. Informe de estado - <i>Sprint 3</i> . . . . .	6
A.7. Gráfico <i>burndown</i> - <i>Sprint 4</i> . . . . .	7
A.8. Informe de estado - <i>Sprint 4</i> . . . . .	7
A.9. Gráfico <i>burndown</i> - <i>Sprint 5</i> . . . . .	8
A.10. Informe de estado - <i>Sprint 5</i> . . . . .	9
A.11. Gráfico <i>burndown</i> - <i>Sprint 6</i> . . . . .	10
A.12. Informe de estado - <i>Sprint 6</i> . . . . .	10
A.13. Gráfico <i>burndown</i> - <i>Sprint 7</i> . . . . .	11
A.14. Informe de estado - <i>Sprint 7</i> . . . . .	12
A.15. Gráfico <i>burndown</i> - <i>Sprint 8</i> . . . . .	13
A.16. Informe de estado - <i>Sprint 8</i> . . . . .	13
A.17. Gráfico <i>burndown</i> - <i>Sprint 9</i> . . . . .	14
A.18. Informe de estado - <i>Sprint 9</i> . . . . .	14
A.19. Gráfico <i>burndown</i> - <i>Sprint 10</i> . . . . .	16
A.20. Informe de estado - <i>Sprint 10</i> . . . . .	16
A.21. Gráfico <i>burndown</i> - <i>Sprint 11</i> . . . . .	18
A.22. Informe de estado - <i>Sprint 11</i> . . . . .	18
A.23. Gráfico <i>burndown</i> - <i>Sprint 12</i> . . . . .	20
A.24. Informe de estado - <i>Sprint 12</i> . . . . .	20
A.25. Gráfico <i>burndown</i> - <i>Sprint 13</i> . . . . .	22
A.26. Informe de estado - <i>Sprint 13</i> . . . . .	22
A.27. Gráfico <i>burndown</i> - <i>Sprint 14</i> . . . . .	24

A.28.Informe de estado - <i>Sprint 14</i>	24
A.29.Gráfico <i>burndown</i> - <i>Sprint 15</i>	26
A.30.Informe de estado - <i>Sprint 15</i>	26
A.31.Diagrama de flujo acumulado total	27
B.1. Diagrama de casos de uso	42
C.1. Diagrama entidad-relación	57
C.2. Diagrama relacional de la base de datos	58
C.3. <i>Mockup</i> - Página principal	61
C.4. <i>Mockup</i> - Página principal (info)	62
C.5. <i>Mockup</i> - Página principal (contacto)	62
C.6. <i>Mockup</i> - Página principal (idioma)	63
C.7. <i>Mockup</i> - Inicio sesión	64
C.8. <i>Mockup</i> - Bienvenida paciente	64
C.9. <i>Mockup</i> - Evolución paciente	65
C.10. <i>Mockup</i> - Predicción paciente	66
C.11. <i>Mockup</i> - Bienvenida médico	66
C.12. <i>Mockup</i> - Listado pacientes	67
C.13. <i>Mockup</i> - Información personal paciente	68
C.14. <i>Mockup</i> - Subir vídeo paciente	69
C.15. <i>Mockup</i> - Subir datos paciente	69
C.16. <i>Mockup</i> - Evolución médico	70
C.17. <i>Mockup</i> - Predicción médico	70
C.18. <i>Mockup</i> - Bienvenida admin	71
C.19. <i>Mockup</i> - Gestión usuarios	72
C.20. <i>Mockup</i> - Modificar usuario	72
C.21. <i>Mockup</i> - Eliminar usuario	73
C.22. <i>Mockup</i> - Crear usuario	73
C.23. Diagrama de secuencia - Iniciar Sesión	75
C.24. Diagrama de secuencia - Mostrar Datos Sensor	76
C.25. Diagrama de secuencia - Subir Vídeo	78
C.26. Diagrama de secuencia - Predicción	79
C.27. Diagrama de despliegue	80
E.1. Página principal	105
E.2. Página principal - Contacto 1	105
E.3. Página principal - Contacto 2	106
E.4. Página principal - Idioma	106
E.5. Página principal - Inicio de Sesión	107
E.6. Página principal - Paciente	108

E.7. Página principal - Paciente (editar datos personales) . . . . .	108
E.8. Mostrar Datos Paciente . . . . .	109
E.9. Mostrar Datos Paciente - Seleccionar . . . . .	110
E.10. Mostrar Datos Paciente - Fechas . . . . .	111
E.11. Mostrar Datos Paciente - Miniaturas . . . . .	112
E.12. Mostrar Datos Paciente - Gráfico . . . . .	112
E.13. Mostrar Vídeos . . . . .	113
E.14. Mostrar Vídeos - Gráficas . . . . .	113
E.15. Mostrar Vídeos - Gráficas con predicción . . . . .	114
E.16. Página principal - Médico . . . . .	115
E.17. Listado pacientes Médico . . . . .	115
E.18. Listado pacientes Médico - Información personal . . . . .	116
E.19. Listado pacientes Médico - Subir vídeo . . . . .	117
E.20. Listado pacientes Médico - Subir datos sensor . . . . .	117
E.21. Página principal - Administrador . . . . .	118
E.22. Gestión de usuarios . . . . .	119
E.23. Gestión de usuarios - Administradores . . . . .	120
E.24. Gestión de usuarios - Añadir Usuario . . . . .	121
E.25. Gestión de usuarios - Médicos . . . . .	122
E.26. Gestión de usuarios - Pacientes . . . . .	123
E.27. Gestión de usuarios - Añadir paciente . . . . .	124

---

# Índice de tablas

---

A.1. Tabla de contribuciones que las empresas pagan por trabajador	29
A.2. Tabla de costes del proyecto . . . . .	30
A.3. Licencias . . . . .	35
B.1. CU-1 Visualizar características vídeos. . . . .	43
B.2. CU-1.1 Realizar predicción. . . . .	44
B.3. CU-2 Visualizar evolución. . . . .	45
B.4. CU-3 Administrar pacientes. . . . .	46
B.5. CU-3.1 Cambiar datos paciente. . . . .	47
B.6. CU-3.2 Subir datos sensor. . . . .	48
B.7. CU-3.3 Subir vídeos. . . . .	49
B.8. CU-4 Gestionar sesión. . . . .	50
B.9. CU-5 Administrar usuarios. . . . .	51
B.10.CU-5.1 Crear usuario. . . . .	52
B.11.CU-5.2 Modificar usuario. . . . .	53
B.12.CU-5.3 Eliminar usuario. . . . .	54
C.1. Diccionario de datos. Tabla administrador. . . . .	58
C.2. Diccionario de datos. Tabla médico. . . . .	59
C.3. Diccionario de datos. Tabla paciente. . . . .	59
C.4. Diccionario de datos. Tabla registros. . . . .	59
C.5. Diccionario de datos. Tabla vídeos. . . . .	60
D.1. CP1: Control de acceso . . . . .	92
D.2. CP2: Añadir usuario . . . . .	93
D.3. CP3: Eliminarse a sí mismo . . . . .	94
D.4. CP4: Eliminar paciente con vídeos . . . . .	95
D.5. CP5: Eliminar médico con pacientes . . . . .	96
D.6. CP6: Modificar cero cambios . . . . .	97

D.7. CP7: Editar correo electrónico . . . . .	98
D.8. CP8: Predicción con diferente periodicidad . . . . .	99
D.9. CP9: Predicción al eliminar vídeo . . . . .	100
D.10.CP10: Seleccionar fechas sin registros . . . . .	100
D.11.CP11: Subir vídeos con mismo nombre . . . . .	101
D.12.CP12: Fecha del vídeo incoherente . . . . .	102



## *Apéndice A*

---

# **Plan de Proyecto Software**

---

## **A.1. Introducción**

El Trabajo de Fin de Grado se comienza con la planificación. Esta primera fase es esencial para cumplir todos los objetivos y plazos del proyecto en el futuro.

En este primer anexo se divide la planificación inicial en planificación temporal y estudio de viabilidad:

- En la primera parte, se detallará la planificación temporal escogida, tan importante en el desarrollo *software*. Se ha optado por una metodología ágil de tipo Scrum (estudiada durante el grado), dividiendo el calendario de trabajo en ciclos cortos llamados *sprints*.
- El estudio de viabilidad, por su parte, se divide en viabilidad económica y legal. La viabilidad económica consiste en aproximar los costes y beneficios asociados con el desarrollo del proyecto, mientras que en la viabilidad legal se requiere investigar sobre la legislación que pueda estar relacionada con el proyecto.

## **A.2. Planificación temporal**

Se ha escogido una metodología Scrum para llevar a cabo la planificación temporal del proyecto, aunque no se ha podido seguir al 100% ya que no se trata de un proyecto real (con equipo de desarrollo de varias personas, con reuniones diarias, *scrum master*, *product owner*, incrementos entregables en cada *sprint*...), sino de un Proyecto de Fin de Grado.

Los roles de *scrum master* y *product owner* fueron desempeñados por el tutor, quien se encargó de definir las actividades del *product backlog* y asistir al equipo con las prácticas de Scrum. Por su parte, la alumna asumió el rol del equipo de desarrollo, siendo responsable de convertir los elementos del *product backlog* en incrementos al final de cada *sprint*.

El proyecto comenzó en septiembre y concluyó en junio. Durante este periodo, se dividió el tiempo en *sprints* de una, dos o tres semanas, adaptándose a las tareas requeridas o a las circunstancias personales del equipo de desarrollo. Tras cada sprint, se llevaban a cabo reuniones entre el tutor y la alumna, con el propósito de abordar dudas sobre las actividades realizadas y definir las tareas a realizar en el próximo ciclo, asegurándose de estar avanzando conforme a las necesidades del proyecto y de poder cumplir los plazos de entrega establecidos.

Las actividades dentro de cada *sprint* se clasificaban según su importancia, priorizando generalmente aquellas con mayor prioridad. Además, se utilizaba la técnica de Scrum de medición en *story points* para estimar el esfuerzo necesario para completar una tarea. Los *story points* no representan simplemente el tiempo necesario, sino que tienen en cuenta la complejidad de las tareas.

Las estimaciones de *story points* y prioridad, así como la duración de los *sprints*, fueron mejorando con el transcurso de los meses, basándose en lo vivido en *sprints* anteriores.

La herramienta seleccionada para llevar a cabo la planificación temporal fue Jira. En un primer momento, se consideró la posibilidad de utilizar ZenHub, una herramienta de GitHub usada frecuentemente en los Trabajos de Fin de Grado. Sin embargo, dado que ZenHub actualmente es de pago, se decidió optar por Jira.

A continuación se presentan los *sprints* realizados, comentando sus objetivos generales, las tareas realizadas y el tiempo empleado. Además se muestran informes realizados por Jira que muestran el avance de las actividades a lo largo del tiempo de cada *sprint*.

## ***Sprint 1 - Comienzo***

El proyecto comenzó tras una primera reunión con el tutor, durante la cual se presentaron diferentes ideas de proyecto. Después de esta reunión, se seleccionó el tema que se discutiría más detalladamente en el siguiente encuentro, comentando cual sería el objetivo del proyecto y las actividades a realizar durante el primer *sprint*.

El primer *sprint* se contabilizó desde esa primera reunión, por lo que dura varias semanas. Durante este periodo, se llevó a cabo una investigación sobre el parkinson, la enfermedad que aborda el proyecto. Además, la alumna se comenzó a familiarizar con las herramientas que se utilizarían posteriormente durante el proyecto (GitHub, Jira, Python, entornos virtuales...).

Se comenzó la documentación del proceso en Jira tarde, por lo que el gráfico *burndown* tiene el aspecto que se observa en la Figura A.1. Las tareas finalizadas se muestran en el informe de la Figura A.2.

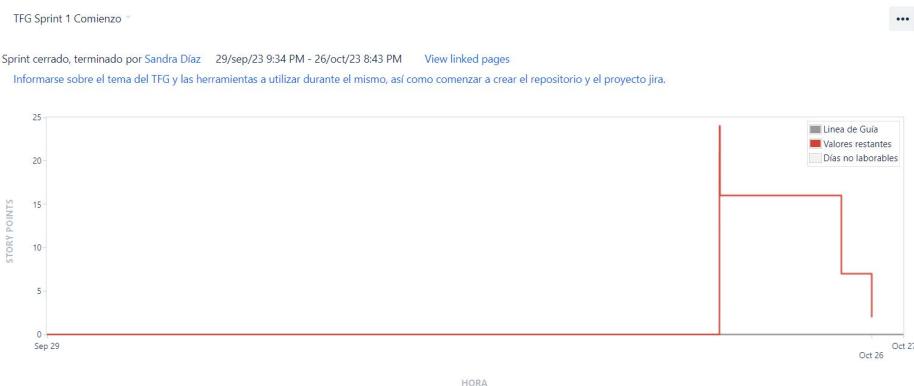


Figura A.1: Gráfico *burndown* - *Sprint 1*

Informe de estado						* Incidencia agregada al sprint después de la fecha de comienzo
						Ver en el navegador de incidencias
Incidencias terminadas						
Clave	Resumen	Tipo de Incidencia	Prioridad	Estado	Story Points (19 - 22)	
TFG-1 *	Leer artículos sobre el Parkinson	Historia	↑ High	FINALIZADA	5	
TFG-2 *	Recordar funcionamiento de GitHub	Historia	= Medium	FINALIZADA	2 - 5	
TFG-3 *	Conseguir licencia Zenhub gratuita	Historia	▼ Low	FINALIZADA	2	
TFG-4 *	Investigar sobre virtual environments en Python	Historia	= Medium	FINALIZADA	2	
TFG-7 *	Descargar plantillas e información del TFG en ubuvirtual	Historia	↑ High	FINALIZADA	2	
TFG-8 *	Recordar funcionamiento de Jira	Historia	▼ Low	FINALIZADA	2	
TFG-9 *	Primera reunión con el tutor	Historia	↑ High	FINALIZADA	2	
TFG-10 *	Reunión de fin del sprint con el tutor	Historia	↑ High	FINALIZADA	2	

Incidencias Sin Completar						Ver en el navegador de incidencias
Clave	Resumen	Tipo de Incidencia	Prioridad	Estado	Story Points (2)	
TFG-6 *	Acceder a la VPN de la UBU	Historia	↙ Lowest	TAREAS POR HACER	2	

Figura A.2: Informe de estado - *Sprint 1*

## Sprint 2 - Usos de Python

El segundo *sprint* tuvo una duración de tres semanas, durante las cuales se adquirieron habilidades en el uso de varias herramientas y bibliotecas de Python. Se aprendió a utilizar Jupyter Notebook para la programación en

Python, así como Flask para la creación de aplicaciones web. Además, se exploraron dos importantes bibliotecas de Python que serían fundamentales para el desarrollo del proyecto:

- Matplotlib: se utilizará para generar gráficos y otras representaciones visuales a partir de los datos de los pacientes.
- Scikit-learn: esta biblioteca cuenta con una variedad de algoritmos de *machine learning*, tanto supervisados como no supervisados, útiles para realizar clasificaciones o predicciones con los datos proporcionados.

Para adquirir competencias en aprendizaje automático, se repasó el temario de las asignaturas de Minería de datos y Sistemas inteligentes, ya que la alumna no las había cursado.

El gráfico de *burndown* y el informe del *sprint* 2 se muestran en la Figura A.3 y Figura A.4.



Figura A.3: Gráfico *burndown* - *Sprint 2*

Informe de estado					
Incidencias terminadas				Ver en el navegador de incidencias	
Clave	Resumen	Tipo de Incidencia	Prioridad	Estado	Story Points (24)
TFG-15	Investigar el funcionamiento de la herramienta Jupyter Notebook	Historia	High	FINALIZADA	2
TFG-17	Aprender a usar Matplotlib	Historia	High	FINALIZADA	2
TFG-18	Conocer el funcionamiento de Scikit-learn	Historia	Medium	FINALIZADA	2
TFG-19	Aprender y coger soltura a Flask	Historia	High	FINALIZADA	8
TFG-20	Mirar apuntes de Minería de datos	Historia	Low	FINALIZADA	5
TFG-21	Mirar apuntes de Sistemas inteligentes	Historia	Low	FINALIZADA	5

Incidencias Sin Completar				Ver en el navegador de incidencias	
Clave	Resumen	Tipo de Incidencia	Prioridad	Estado	Story Points (10)
TFG-14	Recordar el lenguaje de programación Python	Historia	Highest	EN CURSO	8
TFG-16	Familiarizarse con el código existente	Historia	Medium	EN CURSO	2

Figura A.4: Informe de estado - *Sprint 2*

### Sprint 3 – Memoria y Flask

La duración de este *sprint* fue de diez días, los cuales se concentraron en la implementación de un prototipo de aplicación web utilizando Flask, una herramienta previamente introducida pero no estudiada en profundidad en el *sprint* anterior.

El objetivo principal del *sprint* fue crear un prototipo funcional de la aplicación, aplicando los conceptos aprendidos de la guía de Flask.

Se desarrolló una aplicación web que dispone de una página principal con un formulario para que los pacientes ingresen su nombre e *id* (campos obligatorios, con restricciones y preparados contra ataques). Con esos datos, la página redirige a los pacientes a otra ventana donde se muestran los datos de dicho paciente en forma de tabla (extraídos de un archivo CSV de prueba).

Además, existe una ventana para manejar el error 404, con un botón de redirección a la página principal. Durante este *sprint*, también se exploraron conceptos como *cookies* y sesiones.

Además del desarrollo de la aplicación, se inició el proceso de documentación del proyecto (memoria y anexos). Se comenzó a completar el apéndice A (plan de proyecto *software*) de los anexos, que consiste en la planificación temporal (en la que nos encontramos) y el estudio de viabilidad, tanto económica como legal. De la memoria se realizan el resumen, los descriptores y los objetivos del proyecto.

A continuación se adjuntan el gráfico (Figura A.5) e informe (Figura A.6) generados por Jira, aunque es importante destacar que las tareas de do-

cumentación no se cerraron antes de que finalizara el *sprint* ya que no se pudieron completar todas las partes que se plantearon en la reunión.



Figura A.5: Gráfico burndown - *Sprint 3*

Informe de estado						
Incidencias terminadas						
Clave	Resumen	Tipo de Incidencia	Prioridad	Estado	Story Points (18)	Ver en el navegador de incidencias
TFG-16	Familiarizarse con el código existente	Historia	= Medium	FINALIZADA	2	<a href="#">Ver en el navegador de incidencias</a>
TFG-24	Crear aplicación web con Flask	Historia	↗ High	FINALIZADA	16	<a href="#">Ver en el navegador de incidencias</a>
Incidencias Sin Completar						
Clave	Resumen	Tipo de Incidencia	Prioridad	Estado	Story Points (16)	Ver en el navegador de incidencias
TFG-22	Comenzar documentación memoria	Historia	= Medium	EN CURSO	8	<a href="#">Ver en el navegador de incidencias</a>
TFG-23	Comenzar documentación anexos	Historia	= Medium	EN CURSO	8	<a href="#">Ver en el navegador de incidencias</a>

Figura A.6: Informe de estado - *Sprint 3*

## **Sprint 4 – Apéndices B y C**

El cuarto *sprint*, con una duración de dos semanas, se vio afectado por el puente de diciembre, lo que limitó el tiempo dedicado al proyecto por parte de la alumna.

Tras la reunión anterior con el tutor, se comprendieron los requerimientos que debía cumplir la aplicación a desarrollar. Con estos conocimientos, se realizó el apéndice B de los anexos que consiste en:

- Definir los objetivos generales del proyecto.
- Definir los requisitos del cliente (el tutor en este caso) y crear casos de uso en consecuencia. Posteriormente generar un diagrama general de casos de uso con los actores involucrados y rellenar tablas con información sobre cada caso de uso.

Continuando con el apéndice C de los anexos, se elaboraron los diagramas de entidad-relación y el diagrama relacional para estructurar la futura base de datos. Se incluye también en este apéndice la realización de un *mockup* en Pencil para diseñar las interfaces de la aplicación.

También se completó el apartado de «Trabajos relacionados» de la memoria, que proporciona un contexto en el campo del proyecto en curso. Se investigaron aplicaciones y artículos relacionados, y se incluyeron TFGs de compañeros de años anteriores relacionados con el proyecto.

El gráfico *burndown* y el informe del proyecto se ilustran en la Figura A.7 y Figura A.8. Las tareas se subdividen en subtareas que se fueron completando progresivamente.



Figura A.7: Gráfico *burndown* - *Sprint 4*

#### Informe de estado

Incidencias terminadas						<a href="#">Ver en el navegador de incidencias</a>
Clave	Resumen	Tipo de Incidencia	Prioridad	Estado	Story Points (56)	
TFG-25	Apéndice B: Especificación de Requisitos	Historia	High	FINALIZADA	32	
TFG-26	Apéndice C: Especificación de diseño	Historia	High	FINALIZADA	16	
TFG-33	Rellenar apartado de trabajos relacionados	Historia	Low	FINALIZADA	8	

Figura A.8: Informe de estado - *Sprint 4*

## ***Sprint 5 – Correcciones memoria y flask***

El quinto *sprint* duró una semana, desde el 14 de diciembre hasta el inicio de las vacaciones de Navidad, momento en el cual se llevó a cabo una reunión de final de *sprint* de forma *online* con el tutor.

Los objetivos del *sprint* fueron demasiados para el corto periodo de tiempo, por lo que no se pudieron completar todos. Se realizaron correcciones sobre el trabajo realizado en el *sprint* anterior, entre las que se encuentran:

- Ordenar los requisitos funcionales según su importancia y mejorar la forma de expresarlos.
- Corregir el diagrama de casos de uso y el de entidad-relación.
- Crear ventanas para el administrador en el *mockup* elaborado en Pencil e introducir en los anexos un apartado para el diseño de interfaces.

Se realizó el apartado 4 de la memoria (Técnicas y herramientas), detallando las herramientas utilizadas hasta el momento y justificando por qué se eligieron frente a otras opciones disponibles.

Los objetivos que no se lograron cumplir fueron los relacionados con la base de datos y la aplicación web. Se tenía la intención de crear las tablas principales de la base de datos, como las tablas para usuarios, pacientes o médicos. Además, se pretendía comenzar con el desarrollo de la aplicación web definitiva, implementando un formulario de inicio de sesión funcional que redirigiera a diferentes ventanas según el tipo de usuario, así como la página principal con las funcionalidades básicas que aparecen en la barra superior.

El gráfico y el informe que se muestran en la Figura A.9 y Figura A.10 muestran las incidencias terminadas y las pendientes.



Figura A.9: Gráfico *burndown* - *Sprint 5*

## Informe de estado

Incidencias terminadas						Ver en el navegador de incidencias
Clave	Resumen	Tipo de Incidencia	Prioridad	Estado	Story Points (18)	
TFG-38	Corregir requisitos y casos de uso	Historia	High	FINALIZADA	2	
TFG-39	Corregir diagramas de la bbdd	Historia	High	FINALIZADA	2	
TFG-40	Crear ventanas faltantes en Pencil	Historia	Low	FINALIZADA	2	
TFG-41	Investigar sobre diseño web adaptable	Historia	Medium	FINALIZADA	4	
TFG-44	Apartado de técnicas y herramientas (memoria)	Historia	Low	FINALIZADA	8	

Incidencias Sin Completar						Ver en el navegador de incidencias
Clave	Resumen	Tipo de Incidencia	Prioridad	Estado	Story Points (44)	
TFG-42	Comenzar con la página principal en flask	Historia	High	TAREAS POR HACER	16	
TFG-43	Comenzar con el inicio de sesión en flask	Historia	High	TAREAS POR HACER	16	
TFG-45	Crear tablas de la bbdd	Historia	Medium	TAREAS POR HACER	4	
TFG-46	Recordar apuntes sobre Bases de Datos	Historia	Medium	TAREAS POR HACER	8	

Figura A.10: Informe de estado - *Sprint 5***Sprint 6 – Flask BBDD LATEX**

En un principio, se había planeado que este *sprint* durara las tres semanas de vacaciones de Navidad. Sin embargo, las reuniones familiares típicas de estas fechas, junto con una posterior gripe, no permitieron a la alumna centrarse en este trabajo.

La duración final del *sprint* fue del 9 al 26 de enero y los objetivos planteados fueron las incidencias que quedaron sin completar en el *sprint* anterior, así como pasar la documentación del proyecto a LATEX.

Se creó la base de datos del proyecto, utilizando HeidiSQL y siguiendo el diagrama relacional realizado en el apéndice C. Se estableció una tabla para cada usuario de la aplicación, así como para los registros y vídeos, con sus correspondientes restricciones y claves foráneas. Se llenó una fila de cada tabla para probar el acceso a la base de datos desde la aplicación.

Además, se trasladaron los documentos del proyecto a LATEX, aprendiendo en el proceso todas las facilidades que ofrece esta herramienta, como la creación de listas, la inclusión de imágenes, referencias, saltos de línea, negritas y otras características.

Se dio inicio al desarrollo de la aplicación web definitiva en Flask, creando una página principal y una de *login*, con un formulario de inicio de sesión. Se implementaron botones con redirecciones entre ventanas, se creó una base para todos los .html, se mostraron imágenes por pantalla, se accedió a los datos de la base de datos y se creó un formulario de inicio de sesión provisional, entre otras tareas.

El gráfico *burndown* del *sprint* 6 se puede ver en la Figura A.11, mientras que el informe de estado con las incidencias del mismo *sprint* se presenta en la Figura A.12.



Figura A.11: Gráfico *burndown* - *Sprint 6*

#### Informe de estado

Incidencias terminadas						Ver en el navegador de incidencias
Clave	Resumen	Tipo de Incidencia	Prioridad	Estado	Story Points (38 → 46)	
TFG-42	Comenzar con la página principal en flask	Historia	High	FINALIZADA	16	
TFG-45	Crear tablas de la bbdd	Historia	Medium	FINALIZADA	4	
TFG-46	Recordar apuntes sobre Bases de Datos	Historia	Medium	FINALIZADA	8	
TFG-47	Investigar sobre LaTeX y Overleaf	Historia	Medium	FINALIZADA	2	
TFG-48	Pasar la documentación a LaTeX	Historia	Medium	FINALIZADA	8 → 16	
Incidencias Sin Completar						Ver en el navegador de incidencias
Clave	Resumen	Tipo de Incidencia	Prioridad	Estado	Story Points (16)	
TFG-43	Comenzar con el inicio de sesión en flask	Historia	High	TAREAS POR HACER	16	

Figura A.12: Informe de estado - *Sprint 6*

## Sprint 7 - Programación con Flask

Tras la reunión de fin de *sprint* con el tutor, se establecieron los objetivos a cumplir en este *sprint*:

- Respecto a la documentación en L<sup>A</sup>T<sub>E</sub>X, se debía citar con BibTeX en un documento nuevo llamado `bibliografia.bib`. Además debían corregir los errores lingüísticos o propios de L<sup>A</sup>T<sub>E</sub>X resaltados por el tutor.
- En cuanto a la base de datos, se espera llenar las tablas. Se debía incluir los vídeos y archivos CSV proporcionados a la alumna, y llenar el resto de los datos de pacientes y médicos con información aleatoria.

- La parte importante de este *sprint* sería la programación web. Se debían crear todas las ventanas que fueron diseñadas en Pencil en la aplicación, utilizando Bootstrap para el redimensionamiento y otros aspectos. También se debían emplear nuevas herramientas como `chart.js` o `bootstrap-table` para realizar gráficos o tablas.

Durante las dos semanas de duración del *sprint*, se siguió el recordatorio del tutor de realizar *commits* en GitHub cada vez que se realizara un cambio en la aplicación, no solo para documentar el trabajo realizado, sino también como control de versiones para la alumna, lo que facilitaría la resolución de fallos futuros.

Durante este período, por cada cambio realizado en la aplicación (barra de navegación, *footer*, página de contacto, página de sobre nosotros, página de listado de pacientes, colores, cambios de aspecto...) se realizó un *commit* y el posterior *push* al repositorio de la alumna. Aunque no fue posible crear todas las ventanas que se diseñaron en el *mockup* de Pencil, se avanzó significativamente con la programación web.

Se citó la bibliografía de la documentación y se realizaron los cambios señalados por el tutor, como distinguir palabras en inglés, formas de poner las URL, los números o incluso la manera de citar el propio L<sup>A</sup>T<sub>E</sub>X.

Además, la base de datos se llenó con datos aleatorios utilizando la herramienta *online* Mockaroo. La alumna no disponía de los vídeos de los pacientes, por lo que se incluirán en la base de datos en el siguiente *sprint*.

Como se puede ver en la Figura A.13, se presenta el gráfico *burndown* del *sprint* 7. Asimismo, el informe de estado se muestra en la Figura A.14.



Figura A.13: Gráfico *burndown* - *Sprint 7*

Informe de estado					
Incidencias terminadas					
Clave	Resumen	Tipo de Incidencia	Prioridad	Estado	Story Points (60 – 56)
TFG-49	Citar bibliografía BibTeX	Historia	Low	FINALIZADA	4
TFG-50	Añadir vídeos y csv en la BBDD	Historia	Medium	FINALIZADA	4
TFG-51	Rellenar BBDD con pacientes y médicos aleatorios	Historia	Low	FINALIZADA	8 – 4
TFG-52	Investigar sobre Bootstrap	Historia	Highest	FINALIZADA	8
TFG-53	Crear barra superior Flask	Historia	Medium	FINALIZADA	16
TFG-54	Cambiar aplicación usando Bootstrap	Historia	High	FINALIZADA	16
TFG-57	Hacer commits en GitHub con cada cambio	Historia	Highest	FINALIZADA	4

Incidencias Sin Completar					
Clave	Resumen	Tipo de Incidencia	Prioridad	Estado	Story Points (40)
TFG-55	Usar chart.js para los gráficos	Historia	Medium	TAREAS POR HACER	8
TFG-56	Crear todas las ventanas de Pencil en Flask	Historia	High	EN CURSO	32

Figura A.14: Informe de estado - *Sprint 7*

## Sprint 8 - Flask y corregir

Este *sprint* tiene una duración de una única semana, ya que la alumna posteriormente se va a un proyecto europeo, por lo que no se han incluido todas las actividades que se comentaron con el tutor en la reunión de fin de *sprint*. Se han solucionado las siguientes incidencias:

- Rellenar el apartado de los informes de *sprint* anterior y actual.
- Corregir la documentación de L<sup>A</sup>T<sub>E</sub>X: especificar rutas, cambiar fecha de la bibliografía, dividir los apartados en subsecciones para facilitar la lectura y cambiar la forma de expresarse a un lenguaje más impersonal.
- Utilizar *bootstrap-table* para mostrar el listado de pacientes de cada médico.
- Editar la barra de navegación para tener una única barra universal en el *layout*.
- Añadir los vídeos de los pacientes en la base de datos.
- Realizar el apartado de la aplicación que permite a los médicos subir los vídeos y archivos de datos de sus pacientes.
- Continuar con la creación de más ventanas de las generadas inicialmente en Pencil.

El gráfico *burndown* presenta la forma que se observa en la Figura A.15 al final del sprint ya que, además de no haberse terminado la incidencia sobre la creación de todas las ventanas de la aplicación, la alumna olvidó

cerrar el sprint en Jira por las prisas del viaje, lo que hizo que el tiempo del sprint siguiera corriendo.

El informe del *sprint* se muestra en la Figura A.16.



Figura A.15: Gráfico burndown - Sprint 8

#### Informe de estado

Incidencias terminadas						<a href="#">Ver en el navegador de incidencias</a>
Clave	Resumen	Tipo de Incidencia	Prioridad	Estado	Story Points (32)	
TFG-6	Acceder a la VPN de la UBU	Historia	Lowest	FINALIZADA	2	
TFG-58	Usar bootstrap-table para mostrar pacientes	Historia	High	FINALIZADA	8	
TFG-59	Corregir documentación de LaTeX	Historia	Low	FINALIZADA	2	
TFG-62	Informe de sprint anterior de los anexos	Historia	High	FINALIZADA	4	
TFG-63	Barra de navegación universal	Historia	High	FINALIZADA	8	
TFG-64	Subida de archivos a la bbdd desde flask	Historia	Medium	FINALIZADA	8	

Incidencias Sin Completar						<a href="#">Ver en el navegador de incidencias</a>
Clave	Resumen	Tipo de Incidencia	Prioridad	Estado	Story Points (32)	
TFG-56	Crear todas las ventanas de Pencil en Flask	Historia	High	EN CURSO	32	

Figura A.16: Informe de estado - Sprint 8

## Sprint 9 - Estamos de vuelta

Tras el proyecto europeo, casi un mes más tarde se continuó con el *sprint* 9, cuya duración fue de dos semanas y media. Consistió en realizar las incidencias que no se incluyeron en el anterior *sprint* por falta de tiempo.

Se avanzó con la documentación, realizando los apartados de conceptos teóricos y de técnicas y herramientas de la memoria, así como el apartado actual y algunas correcciones.

Se realizó un *script* para llenar la base de datos de pacientes y médicos con imágenes generadas por inteligencia artificial, ya que no se tenía acceso a las fotografías reales de estos usuarios.

Respecto a la aplicación web se programó el formulario de inicio de sesión definitivo, comprobando los usuarios y contraseñas en la base de datos, y llevando un posterior control de sesiones. Se incluyó en la barra de navegación un apartado extra con el perfil del usuario, permitiéndole cerrar su sesión, y se modificaron las páginas de bienvenida de cada tipo de usuario de la aplicación.

También se comenzó con la generación de los gráficos de los pacientes a partir de los registros del sensor. El usuario elige qué gráfico ver y de qué fechas mediante un formulario. Se implementó el código cedido por los creadores del sensor.

El gráfico *burndown* se presenta en la Figura A.17 y el informe en la Figura A.18:



Figura A.17: Gráfico *burndown* - *Sprint 9*

#### Informe de estado

Incidencias terminadas						Ver en el navegador de incidencias
Clave	Resumen	Tipo de Incidencia	Prioridad	Estado	Story Points (62 → 50)	
TFG-43	Inicio de sesión en flask con la bbdd	Historia	~ High	FINALIZADA	8 → 4	
TFG-55	Crear ventana con la gráfica (chart.js)	Historia	~ Low	FINALIZADA	16	
TFG-60	Apartado de conceptos teóricos de la memoria	Historia	= Medium	FINALIZADA	8	
TFG-61	Apartado de técnicas y herramientas de la memoria	Historia	= Medium	FINALIZADA	16 → 8	
TFG-65	Informe del sprint anterior y actual	Historia	~ High	FINALIZADA	2	
TFG-66	Control de sesiones	Historia	= Medium	FINALIZADA	8	
TFG-67	Rellenar bbdd con imágenes de pacientes	Historia	= Medium	FINALIZADA	4	

Incidencias Sin Completar						Ver en el navegador de incidencias
Clave	Resumen	Tipo de Incidencia	Prioridad	Estado	Story Points (8)	
TFG-68	Mejorar subida de vídeos a la bbdd	Historia	= Medium	TAREAS POR HACER	8	

Figura A.18: Informe de estado - *Sprint 9*

## Sprint 10 - Mostrar gráficos

Durante las poco más de dos semanas de duración del *sprint* se llevaron a cabo diferentes acciones sobre el código de la aplicación web:

- Tras la reunión con el tutor, se abandonó la idea inicial de que el usuario administrador pudiera entrar como un médico o paciente. Se programó una ventana para los administradores de gestión de usuarios. En ella se puede acceder a tres tablas, conteniendo todos los administradores, médicos y pacientes. Estos usuarios se pueden editar o eliminar de la base de datos, así como crear usuarios nuevos. Además, sobre los pacientes se pueden realizar las mismas acciones que realizaban ya los médicos.
- Se corrigió el acceso a las ventanas de la aplicación, comprobando que el usuario que ha iniciado sesión tenga permiso para acceder a dicha información.
- Se corrigió también la subida de vídeos y registros a la base de datos, cambiando el nombre de los archivos a uno que definiera su contenido y el momento exacto en el que fue introducido en la base de datos, controlando así la subida de elementos con el mismo nombre.
- Se cambió la forma en la que se guardan los registros en la base de datos, teniendo estos una fecha inicial y una fecha final que no es introducida a mano por el usuario, sino que es calculada por un *script* que obtiene dichos datos del primer y último dato del propio registro.
- El cambio anterior se realizó con el fin de facilitar la creación de un calendario para elegir las fechas del gráfico. En un primer momento la idea era resaltar en verde las fechas que sí tuvieran registros pero, tras investigar las funcionalidades de la biblioteca flatpickr, se optó por deshabilitar directamente las fechas en las que no hubiera registros de ese paciente. Al ser estas muy cuantiosas, se informa al usuario de las escasas fechas disponibles para facilitar la búsqueda.
- Respecto a la gráfica, se trabajó sobre las funciones de los creadores del sensor, adaptándolas al código del proyecto para poder mostrar los gráficos en la propia web, utilizando chart.js.
- Por último, se creó una nueva ventana, disponible en el apartado de acciones sobre los pacientes, en la que se muestran los vídeos disponibles en la base de datos para dicho paciente.

Además de cambios en la aplicación, también se avanzó con la documentación, en concreto con el manual del programador, situado al final de los anexos. Simplemente se investigó la función de dicho apartado y se comenzó a explicar lo que se podía por el momento.

El gráfico e informe del *sprint* se muestran en la Figura A.19 y en la Figura A.20.



Figura A.19: Gráfico burndown - *Sprint 10*

Informe de estado						* Incidencia agregada al sprint después de la fecha de comienzo
						Ver en el navegador de incidencias
Incidencias terminadas						
Clave	Resumen	Tipo de Incidencia	Prioridad	Estado	Story Points (52 → 66)	
TFG-68	Mejorar subida de vídeos a la bbdd	Historia	= Medium	FINALIZADA	8	
TFG-69	Gestión de usuario de los admin	Historia	= Medium	FINALIZADA	8	
TFG-70	Editar datos pacientes	Historia	= Medium	FINALIZADA	4	
TFG-71	Fechar registros bbdd	Historia	↗ Highest	FINALIZADA	4	
TFG-72	Corregir sesiones en incógnito	Historia	= Medium	FINALIZADA	4	
TFG-73	Selección de calendario para el gráfico	Historia	↗ High	FINALIZADA	16	
TFG-74	Mostrar gráfico en la propia ventana	Historia	↗ High	FINALIZADA	8	
TFG-75 *	Comenzar manual del programador	Historia	= Medium	FINALIZADA	8	
TFG-77	Correcciones documentación	Historia	= Medium	FINALIZADA	2	
TFG-78 *	Crear ventana para mostrar videos de pacientes	Historia	= Medium	FINALIZADA	→ 4	

Figura A.20: Informe de estado - *Sprint 10*

## Sprint 11 - Mejora /análisis

La duración de este *sprint* fue de menos de dos semanas. Su objetivo principal, además de realizar mejoras sobre varios aspectos comentados con el tutor en la reunión, era incorporar parte del TFG realizado por el compañero Catalin [2] durante el curso pasado al presente proyecto.

Algunas de estas mejoras fueron:

- Realizar modificaciones en las gráficas. Se cambió su representación, optando por un gráfico que resaltara más los puntos que las líneas, permitiendo centrarse más en las medidas. Además se muestra una miniatura de los gráficos del apartado seleccionado, para ofrecer una visión global al usuario, que posteriormente elige el gráfico a mostrar completo. Respecto al calendario para seleccionar las fechas de las gráficas, se editó para mostrar en primer momento una fecha disponible con registros y para poder seleccionar un solo día en vez de un rango.
- Se realizó el *hash* de las contraseñas de los usuarios de la aplicación para aumentar la seguridad. También se implementó la opción de poder ocultar la contraseña al escribirla en el inicio de sesión.
- Se trató de mejorar la apariencia de la aplicación, agrupando los botones y dándoles colores característicos, centrándose las tablas, mostrando los logos de la UBU y de la asociación de Parkinson Burgos únicamente en la página principal...
- Se implementaron restricciones en los formularios, de forma que el usuario no pueda introducir datos absurdos o textos malintencionados, añadiendo seguridad a la aplicación. Además, se permite limpiar o cancelar los formularios si no se van a enviar.

Respecto a la implementación de lo realizado en el trabajo de fin de grado del estudiante Catalin Andrei Cacuci, se comenzó preparando los datos. Se crearon los campos de amplitud y velocidad en la base de datos de los vídeos. Estos campos serán llenados por los médicos especializados en el momento en el que suban un nuevo vídeo, siendo 0 un buen resultado y 4 uno que denota mayor severidad de parkinson.

Cuando se añade un nuevo vídeo a un paciente, este pasa por las funciones creadas por el alumno mencionado, obteniendo ciertas estadísticas, que se guardan en la base de datos y se muestran en la pantalla en la que anteriormente solo se mostraban los vídeos. Como dicho proceso dura unos minutos, se han añadido elementos visuales para indicar al usuario de que el proceso de análisis se está llevando a cabo. Posteriormente, se pueden ver dichos vídeos y sus características, así como también se pueden eliminar si así se desea.

También se avanzó con la documentación, realizando los informes del *sprint* anterior y actual, y comenzando con el manual de usuario, explicando las funcionalidades de las ventanas, pero sin imágenes ni entrando mucho

en detalle porque ciertos aspectos todavía pueden cambiar hasta la entrega del proyecto.

En el gráfico *burndown*, representado en la Figura A.21, se observa la mejora en la predicción de los *Story Points* de cada incidencia por su semejanza con la línea de Guía. El informe con todas las incidencias del *sprint* terminadas se puede observar en la Figura A.22.



Figura A.21: Gráfico *burndown* - *Sprint 11*

#### Informe de estado

Incidencias terminadas						Ver en el navegador de incidencias
Clave	Resumen	Tipo de Incidencia	Prioridad	Estado	Story Points (42)	
TFG-76	Comenzar manual de usuario	Historia	Medium	FINALIZADA	2	
TFG-79	Informes de sprint	Historia	Medium	FINALIZADA	2	
TFG-80	Modificar gráficas	Historia	Medium	FINALIZADA	8	
TFG-81	Hash de contraseñas	Historia	Medium	FINALIZADA	4	
TFG-82	Mejorar apariencia app	Historia	Medium	FINALIZADA	2	
TFG-83	Formularios con restricciones	Historia	Medium	FINALIZADA	4	
TFG-84	Amplitud y velocidad en los videos	Historia	Medium	FINALIZADA	4	
TFG-85	Implementar TFG de Catalin	Historia	Medium	FINALIZADA	16	

Figura A.22: Informe de estado - *Sprint 11*

## Sprint 12 - IA para vídeos

La duración del presente *sprint* estaba estimada en dos semanas pero se finalizó antes de tiempo. Durante estos días se avanzó con la documentación, se realizaron correcciones en el diseño y navegación de la aplicación y se realizó un estudio del parkinson de los pacientes mediante inteligencia artificial a partir de las características de los videos de los pacientes.

Respecto a la documentación, se comenzaron a redactar apartados nuevos como las conclusiones obtenidas del proyecto y las líneas de trabajo futuras,

es decir, ideas que se pueden implementar en la aplicación en un futuro. Además se incluyó el utilizar el TFG de un compañero como aspecto relevante y se completó información sobre las librerías usadas por el momento.

Las correcciones en la navegación de la aplicación consistieron en generar «migas de pan», permitiendo al usuario ver un rastro de las páginas que ha ido visitando hasta llegar a la página actual, pudiendo acceder a ellas de forma intuitiva.

Respecto al diseño de la aplicación, se han modificado aspectos de la ventana que muestra los gráficos de los datos del sensor. También se ha debido cambiar el aspecto de la ventana que muestra los vídeos, adaptándose al aumento de la cantidad de vídeos por paciente y añadiendo las nuevas funcionalidades. Este aumento de vídeos por paciente es necesario para poder implementar la funcionalidad de predicción mediante inteligencia artificial del desarrollo de la enfermedad, ya que con pocos vídeos esto no sería posible. Este aumento se ha simulado para los 4 primeros usuarios, para poder mostrar el funcionamiento de la aplicación.

El primer paciente tiene vídeos de pacientes hombres que estaban entre los 62 y 66 años. Se han usado los datos de amplitud y lentitud reales, tomadas por las neurólogas del hospital, junto con las características reales obtenidas de las funciones de Catalin [2]. Se han modificado las fechas para simular que se trata del mismo hombre, que ha ido grabando vídeos a lo largo de los años. 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.

Las características de cada vídeo, entre las que se encuentran amplitud, lentitud, velocidad media o frecuencia de máximos/mínimos, se muestran en una gráfica, para tener una idea visual de la evolución del paciente.

Para que la alumna se familiarizara con las series temporales, el tutor la matriculó como estudiante (no oficial) en un curso de series temporales de UBUVirtual. Estos conocimientos se aplicaron para predecir la amplitud y lentitud que tendrían los pacientes ficticios basándose en los datos disponibles. El resultado no fue el deseado debido a la pobreza de los datos, por lo que se pospone la predicción a otro *sprint*.

En la Figura A.23, se puede observar como el ritmo de trabajo de la alumna fue más rápido que el tiempo planificado, representado por la línea de guía, por lo que el *sprint* finalizó cuatro días antes. En la Figura A.24 aparecen todas las incidencias explicadas con anterioridad finalizadas.

Figura A.23: Gráfico burndown - *Sprint 12*

#### Informe de estado

Incidentes terminadas						Ver en el navegador de incidentes	
Clave	Resumen	Tipo de Incidencia	Prioridad	Estado	Story Points (50)		
TFG-86	Estudiar series temporales	Historia	= Medium	FINALIZADA	8		
TFG-87	Aspectos relevantes doc	Historia	= Medium	FINALIZADA	2		
TFG-88	Líneas de trabajo futuras y referenciar	Historia	= Medium	FINALIZADA	4		
TFG-89	Apariencia gráficas	Historia	= Medium	FINALIZADA	2		
TFG-90	Migas de pan	Historia	= Medium	FINALIZADA	8		
TFG-91	Apariencia ventana videos	Historia	= Medium	FINALIZADA	2		
TFG-92	Añadir videos similares a pacientes	Historia	= Medium	FINALIZADA	4		
TFG-93	Mostrar gráfico con las características de los videos	Historia	= Medium	FINALIZADA	4		
TFG-94	Predecir amplitud y lentitud con IA	Historia	= Medium	FINALIZADA	16		

Figura A.24: Informe de estado - *Sprint 12*

## Sprint 13 - Cambios y Docs

Durante una reunión entre el tutor y la alumna se comentaron aspectos de la aplicación que podrían mejorarse para hacerla más homogénea e intuitiva. Algunos de estos cambios fueron:

- Editar las «migas de pan» para hacer la navegación más intuitiva.
- En la ventana de gestión de los usuarios, evitar que los administradores se puedan eliminar a sí mismos y corregir la edición de los usuarios y sus contraseñas.
- Modificar el aspecto de los formularios de la aplicación para hacerlos más concisos, así como añadir avisos obligando a llenar todos los campos obligatorios.

- Añadir una característica a los pacientes, indicando si son zurdos o diestros.
- Fusionar las ventanas de «sobre nosotros» y «contacto» en una única ventana con ambas informaciones.

También se añadió contenido en los siguientes apartados de la documentación:

- Conceptos teóricos: se explicaron las series temporales como subsección a lo previamente explicado sobre *machine learning* y se añadió una explicación sobre la escala UPDRS [7] en relación con el parkinson.
- Aspectos relevantes: se destacaron ciertos desafíos encontrados durante la realización del proyecto, como aprender programación web y minería de datos o usar datos de terceros.
- Conclusiones y líneas futuras: se comentaron las conclusiones obtenidas del resultado del proyecto y ciertas funcionalidades extras interesantes como el añadir más apartados del formulario de la UPDRS.
- Licencias: se compararon las licencias de las librerías utilizadas, con el fin de escoger la más restrictiva como licencia del proyecto.
- Diseño: se realizaron diagramas de secuencia para ciertas funciones del código y el diagrama de despliegue del proyecto.
- F-ODS: se abordaron aspectos sobre la sostenibilidad del proyecto, como cambiar los materiales de los sensores, contribuir al envejecimiento saludable o utilizar modelos de *machine learning* eficientes.
- Introducción: se explicó cuál y cómo es el trabajo realizado.

En la Figura A.25, que representa el gráfico *burndown* se muestra cómo se comenzó con las incidencias de programar de forma rápida y se finalizó más tarde de lo esperado con las relacionadas con la documentación. La Figura A.26 representa el informe con todas las incidencias finalizadas. Se puede apreciar cómo se editó la dificultad en *story points* de la incidencia sobre las migas de pan, y que se añadieron dos incidencias al *sprint* después de la fecha de comienzo.



Figura A.25: Gráfico burndown - Sprint 13

Informe de estado						* Incidencia agregada al sprint después de la fecha de comienzo
						Ver en el navegador de incidencias
Incidencias terminadas						
Clave	Resumen	Tipo de Incidencia	Prioridad	Estado	Story Points (32 – 34)	
TFG-95	Editar migas de pan	Historia	= Medium	FINALIZADA	4 → 2	
TFG-96	Editar gestión de usuarios	Historia	= Medium	FINALIZADA	4	
TFG-97	Aspecto formularios	Historia	= Medium	FINALIZADA	2	
TFG-98	Añadir mano dominante a los pacientes	Historia	= Medium	FINALIZADA	2	
TFG-99	Fusionar "contacto" y "sobre nosotros"	Historia	= Medium	FINALIZADA	2	
TFG-100	Aspecto ventanas de mostrar datos y videos	Historia	= Medium	FINALIZADA	2	
TFG-101	Conceptos teóricos	Historia	= Medium	FINALIZADA	2	
TFG-102	Aspectos relevantes	Historia	= Medium	FINALIZADA	2	
TFG-103	Conclusiones y líneas futuras	Historia	= Medium	FINALIZADA	4	
TFG-104	Licencias	Historia	= Medium	FINALIZADA	2	
TFG-105	Diseño	Historia	= Medium	FINALIZADA	4	
TFG-106	F_ODS	Historia	= Medium	FINALIZADA	2	
TFG-107 *	Informe de Sprint	Historia	= Medium	FINALIZADA	→ → 2	
TFG-108 *	Introducción	Historia	= Medium	FINALIZADA	→ → 2	

Figura A.26: Informe de estado - Sprint 13

## Sprint 14 - Predicciones

El principal objetivo de este *sprint* es generar unas predicciones con resultados razonables de las características de los vídeos grabados por los pacientes realizando el movimiento de pinza. Las predicciones se realizarán para todas las características, utilizando una gráfica por mano. Anteriormente se intentó con un modelo pero no se obtenían buenos resultados, por lo que ahora, además de editar los datos de los pacientes para simular una periodicidad de fechas entre los vídeos, se probará con diferentes modelos. Este proceso hasta conseguir las predicciones deseadas se documentó en la memoria, en el apartado de aspectos relevantes. Tras probar con diferentes modelos y técnicas se decidió usar el modelo de suavizado exponencial de Holt.

Además de las predicciones se realizaron otros cambios en la aplicación comentados en la reunión de final de *sprint*, como ajustes en los formularios y en los datos a mostrar.

La traducción de los textos de la aplicación no se había realizado hasta este momento, ya que se iban añadiendo textos cada semana. Una vez dados por acabados los cambios, se realizó la traducción al inglés y al francés de todos los textos que se muestran en la aplicación, con el fin de internacionalizarla y hacerla accesible para más usuarios.

A medida que avanzaba el *sprint* se fueron descubriendo pequeños fallos en el código que obligaron a reescribir los documentos de las traducciones en varias ocasiones. Algunos de estos fallos fueron: permitir eliminar un paciente aunque tenga vídeos y registros asociados (eliminándolos para acabar con las restricciones de claves foráneas), detectar que no se han realizado cambios en los formularios de edición de los pacientes (evitando enviar el formulario con los mismos datos que ya se encuentran en la base de datos), realizar las traducciones de las alertas al usuario por errores en el formulario...

Para revisar la calidad del código del programa se estudió la herramienta SonarQube. Debido a que es una herramienta de pago, se pidió una licencia de prueba, pero al ser fin de semana no se obtuvo respuesta. A comienzos de la semana, y tras haber configurado y aprendido el funcionamiento de la herramienta, la licencia de prueba fue denegada, por lo que no se pudo ejecutar.

Por último se realizaron correcciones en ciertas partes de la memoria y anexos, como mejorar los diagramas de secuencia, añadir imágenes de las ventanas en el manual de usuario, calcular los beneficios que se obtendrían al añadir publicidad...

En la Figura A.27 se observa cómo se han ido completando las tareas a lo largo del *sprint*. En el informe de la Figura A.28 se aprecian las prioridades dadas a cada incidencia, así como los cambios en las estimaciones de *story points* o las nuevas incidencias agregadas tras la fecha de comienzo.



Figura A.27: Gráfico burndown - Sprint 14

Informe de estado						* Incidencia agregada al sprint después de la fecha de comienzo
						Ver en el navegador de incidencias
Incidencias terminadas						
Clave	Resumen	Tipo de Incidencia	Prioridad	Estado	Story Points (48 → 52)	
TFG-109	Desplegable para elegir médico	Historia	Medium	FINALIZADA	2	
TFG-110	Editar periodicidad datos	Historia	High	FINALIZADA	2	
TFG-111	Mostrar un gráfico por mano	Historia	High	FINALIZADA	4	
TFG-112	Predicciones	Historia	Medium	FINALIZADA	16	
TFG-113	Traducciones	Historia	Low	FINALIZADA	4	
TFG-114	SonarQube	Historia	Low	FINALIZADA	8	
TFG-115	Arreglar comentarios docs	Historia	Medium	FINALIZADA	2	
TFG-116	Diagramas de secuencia	Historia	High	FINALIZADA	4 → 2	
TFG-117	Aspectos relevantes	Historia	Medium	FINALIZADA	4	
TFG-118	Informe Sprint	Historia	Low	FINALIZADA	2	
TFG-119 *	Imágenes manual de usuario	Historia	Medium	FINALIZADA	- → 2	
TFG-120 *	Arreglar errores código	Historia	Medium	FINALIZADA	- → 4	

Figura A.28: Informe de estado - Sprint 14

## Sprint 15 - Despliegue

Este *sprint* corresponde con la semana antes de la entrega, por lo que su objetivo es terminar de perfeccionar la aplicación, completar los apartados de la memoria restantes, mejorar el aspecto del repositorio de GitHub y desplegar el proyecto en el servidor de la UBU.

Respecto a la aplicación, se realizaron las traducciones de los títulos de las gráficas, que previamente no se habían podido traducir. También se revisaron ciertos aspectos de las predicciones.

Además, se trató de revisar el código de nuevo con la herramienta SonarQube, pero con la versión *Community Edition* en vez de la versión para desarrolladores. Esta versión si resultó ser gratuita, por lo que se pudo realizar el análisis del código. Gracias a este análisis se pudieron modificar varias partes del código, que suponían problemas de seguridad.

o mantenibilidad. Estando a tan pocos días de la entrega, cambiar tanto código de la aplicación sin que nada dejara de funcionar supuso un esfuerzo extra para la alumna.

Respecto a la documentación, se desarrollaron los casos de prueba realizados con el fin de probar ciertas funcionalidades de la aplicación que podían fallar. Se realizaron tablas para cada caso de prueba, mostrando el funcionamiento esperado y el funcionamiento que tiene la aplicación.

Además se completaron los apartados de técnicas y herramientas, y aspectos relevantes, incluyendo el proceso y la información de la herramienta SonarQube con la que se analizó el código.

Por último se repasó toda la documentación, arreglando errores y desarrollando más ciertos apartados como el diseño de datos con la base de datos final, los manuales de programador y usuario, el plan de proyecto,...

Una vez finalizados definitivamente los cambios en la aplicación se comenzó con la puesta a punto del repositorio de GitHub. Se fusionó la rama *master* que contenía todo el código de la aplicación con la rama *main* principal. Además se eliminaron documentos y archivos duplicados y se realizó una carpeta en la que se incluyeron las pruebas realizadas al comienzo del TFG para familiarizarse con el lenguaje y ciertas herramientas. Se realizó el archivo `README.md` informativo y se incluyó la licencia elegida para la distribución del código. Además se exportó la configuración de la base de datos y se incluyó en el repositorio. Al finalizar la documentación, se añadirán tanto los .pdf como los documentos .tex propios de L<sup>A</sup>T<sub>E</sub>X.

La parte más tediosa del *sprint* fue la realización del despliegue de la aplicación en un servidor real. La máquina real en la que se puso en marcha fue un servidor proporcionado por la Universidad de Burgos, con un sistema operativo Linux. La idea es que la aplicación estuviera disponible en todo momento desde la intranet de la UBU, facilitando el trabajo de corrección del TFG.

Por último se grabó el tutorial explicando las funcionalidades de la aplicación, se imprimió la memoria y se crearon los 3 pinchos USB con el contenido del repositorio para la entrega física.

En la Figura A.29 se aprecia el ritmo de trabajo llevado en el *sprint*, y en la Figura A.30 el informe con todas las incidencias finalizadas.

Figura A.29: Gráfico burndown - *Sprint 15*

Informe de estado						* Incidencia agregada al sprint después de la fecha de comienzo
						Ver en el navegador de incidencias
Incidencias terminadas						
Clave	Resumen	Tipo de Incidencia	Prioridad	Estado	Story Points (30 → 42)	
TFG-121	Traducir títulos gráficas	Historia	High	FINALIZADA	2	
TFG-122	Revisar predicciones	Historia	High	FINALIZADA	4	
TFG-123	Corregir y terminar documentación	Historia	Medium	FINALIZADA	4	
TFG-124	Casos de prueba	Historia	High	FINALIZADA	4 → 8	
TFG-125	Arreglar GitHub	Historia	High	FINALIZADA	4	
TFG-126	Desplegar aplicación	Historia	Medium	FINALIZADA	8	
TFG-127	Imprimir, preparar USBs y subir proyecto	Historia	Medium	FINALIZADA	2	
TFG-128	Reunión de final de Sprint	Historia	Medium	FINALIZADA	2	
TFG-129 *	Herramienta SonarQube	Historia	Medium	FINALIZADA	- → 8	

Figura A.30: Informe de estado - *Sprint 15*

## Gráfico global final

Al terminar el proyecto, se consultó el diagrama de flujo acumulado, proporcionado por la herramienta Jira.

La Figura A.31 muestra cómo se van acumulando el número de incidencias desde la fecha de inicio del proyecto hasta la fecha actual.

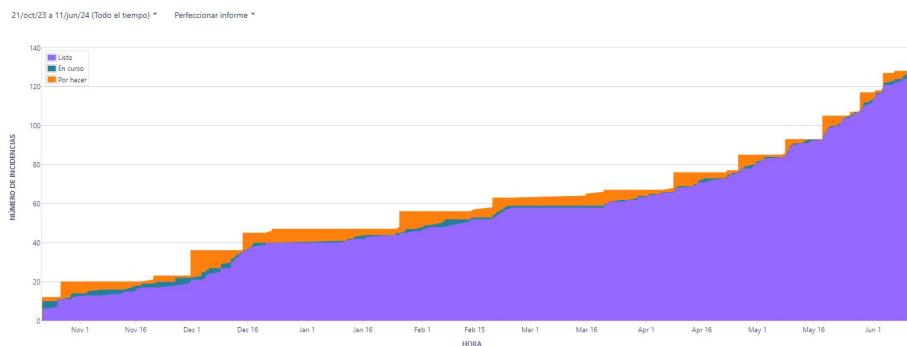


Figura A.31: Diagrama de flujo acumulado total

## A.3. Estudio de viabilidad

El estudio de la viabilidad de un proyecto consiste en llevar a cabo una evaluación exhaustiva del mismo antes de iniciarla, determinando si es rentable o factible. Dado que se trata de un Trabajo de Fin de Grado y no un proyecto real, no se va a analizar de la misma forma. Además, no se pretende obtener beneficios económicos con su realización y no serán necesarios permisos especiales, ya que no se pretende lanzarlo al mercado.

Se va ha realizar una evaluación económica y legal, con el fin de asegurar el éxito y la continuidad del proyecto.

### Viabilidad económica

Se procede a analizar los costes y beneficios del proyecto, para estudiar su rentabilidad. Se tienen en cuenta todos los aspectos que se analizarían si el proyecto fuera real, realizado por una empresa y con un equipo de trabajo.

Es importante tener en cuenta la duración del proyecto. Comenzó el 29 de septiembre y la fecha prevista de entrega al cliente es el 12 de junio, por lo que el proyecto tendrá una duración aproximada de 8 meses.

### Costes

Los costes involucrados en la realización del proyecto se pueden dividir en costes *hardware*, *software*, del personal y otros:

- Costes *hardware*: el único material utilizado ha sido el ordenador portátil de la alumna (DELL XPS 15 9570 con 32 GB de RAM y un procesador Intel® Core™ i5 de octava generación y 2.3 GHz).

Actualmente se encuentra en el mercado por unos 1700 euros<sup>1</sup>, pero el equipo tiene una antigüedad de 4 años, por lo que habría que calcular su amortización en el proyecto [10]:

$$\frac{1700 \text{ €}}{4 \text{ años} \times 12 \text{ meses/año}} = 35.42 \text{ €/mes}$$

$$35.42 \text{ €/mes} \times 8 \text{ meses} = 283.36 \text{ €}$$

- Costes *software*: se entiende como costes *software* los gastos asociados al desarrollo y mantenimiento del *software* a utilizar en el proyecto. Entre ellos se encuentran: licencias, plataformas para el desarrollo o pruebas, formación del equipo de desarrollo, mantenimiento que garantice el funcionamiento del *software* a lo largo de los años...

El principal *software* sobre el cual se han ejecutado las aplicaciones del proyecto, es el sistema operativo de la alumna, en este caso una licencia de Windows 10 Pro, que actualmente se encuentra en el mercado por 259<sup>2</sup> euros. Calculamos su amortización, teniendo en cuenta que la licencia tiene la misma antigüedad que el portátil:

$$\frac{259 \text{ €}}{4 \text{ años} \times 12 \text{ meses/año}} = 5.4 \text{ €/mes}$$

$$5.4 \text{ €/mes} \times 8 \text{ meses} = 43.2 \text{ €}$$

Otras licencias utilizadas durante el proyecto han sido:

- Licencia de Office: utilizada mayoritariamente para realizar la documentación, para el almacenamiento en la nube proporcionado por OneDrive y para las comunicaciones con el tutor a través de correos electrónicos. Siendo estudiante, el coste real de la licencia es de 0 euros, ya que la UBU proporciona este servicio. Si se tratara de una empresa real, disponer de Office 365 tendría un coste de 5.6 € al mes<sup>3</sup>, 44.8 € teniendo en cuenta la duración del proyecto.

---

<sup>1</sup><https://www.worten.es/productos/portatil-dell-xps-15-9570-n5n09-15-6-intel-core-i5-8300h-ram-8-gb-256-gb-ssd-nvidia-geforce-gtx-1050-6841095>

<sup>2</sup><https://www.microsoft.com/es-es/d/windows-11-pro/dg7gmgf0d8h4>

<sup>3</sup><https://www.microsoft.com/es-es/microsoft-365/business/compare-all-microsoft-365-business-products>

- Licencia de GitHub Pro: plataforma de desarrollo colaborativo escogida para desarrollar el *software* del proyecto. Se utiliza una licencia *pro*, cuyo coste sería de 4 euros al mes<sup>4</sup>, constituyendo un total de 32 € durante todo el proyecto. En este caso, la licencia es gratis por ser estudiante.
- Jira: actualmente se utiliza el plan *Free* ya que es gratuito pero, si se tratara de una empresa real, lo mínimo sería utilizar la versión *Standard*, cuyo precio es de 7.75 € al mes<sup>5</sup>, resultando en un total de 62 € para todo el proyecto.
- Costes del personal: se calculan aproximadamente los salarios de los integrantes del equipo (una única desarrolladora y un director de proyecto) en función de la normativa laboral española actual.
  - La alumna es considerada como única integrante del equipo de desarrollo del producto *software*. Dedica las tardes al proyecto, habiendo invertido aproximadamente 500 horas de trabajo repartidas a lo largo de 8 meses. Al seguir estudiando y no contar con experiencia laboral previa en el campo, se le podría considerar programadora *junior*. En España, el sueldo medio de un programador *junior* es de 21384 € al año<sup>6</sup>. Teniendo en cuenta que el tiempo de trabajo anual en España es de 1820 horas<sup>7</sup>, el trabajo de programador *junior* se paga a 11.75 € la hora. Habiendo trabajado 500 horas, el salario bruto de la desarrolladora sería de 5875 € (734 € al mes).

A esta cifra se le deben añadir los impuestos y contribuciones sociales que debe pagar la empresa, recogidos en la tabla A.1 [6]:

<b>Contingencias comunes</b>	23.60 %
<b>Contingencias profesionales</b>	1.50 %
<b>Formación profesional</b>	0.60 %
<b>Desempleo</b>	6.70 %
<b>Fogasa</b>	0.20 %

Tabla A.1: Tabla de contribuciones que las empresas pagan por trabajador

<sup>4</sup><https://github.com/settings/billing/summary>

<sup>5</sup><https://www.atlassian.com/es/software/jira/pricing>

<sup>6</sup>[https://www.glassdoor.es/Sueldos/espaa%C3%BAa-programador-j%C3%A1unior-sueldo-SRCH\\_IL.0,6\\_IN219\\_K07,25.htm](https://www.glassdoor.es/Sueldos/espaa%C3%BAa-programador-j%C3%A1unior-sueldo-SRCH_IL.0,6_IN219_K07,25.htm)

<sup>7</sup><https://www.expansion.com/diccionario-juridico/jornada-laboral.html>

Con estos porcentajes, el coste total para la empresa sería de:

$$\frac{734 \text{ €/mes}}{1 - (0.236 + 0.015 + 0.006 + 0.067 + 0.002)} = 1089.02 \text{ €/mes}$$

$$1089.02 \text{ €/mes} \times 8 \text{ meses} = 8712.17 \text{ €}$$

- El tutor es considerado *product owner* y *scrum master* del proyecto. Sabiendo que el salario promedio anual de ambos empleos ronda los 45000 euros<sup>8</sup>, es decir, 23.08 € la hora. Teniendo en cuenta que el tutor dedica unas 2 horas semanales, cobraría 200 € brutos al mes. Aplicando la fórmula anterior para añadir los impuestos:

$$\frac{200 \text{ €/mes}}{1 - (0.236 + 0.015 + 0.006 + 0.067 + 0.002)} = 296.74 \text{ €/mes}$$

$$296.74 \text{ €/mes} \times 8 \text{ meses} = 2373.89 \text{ €}$$

- Otros costes: si se tratara de un proyecto real realizado por una empresa, un coste a tener en cuenta sería el lugar de trabajo, es decir, el coste de alquiler de una oficina u otro espacio, añadiendo costes como luz, calefacción, Internet... Aproximadamente, alquilar una oficina pequeña en una ciudad como Burgos ronda los 300 euros<sup>9</sup> al mes, suponiendo un coste extra de 2400 €.

El total de los costes se recoge en la tabla A.2:

<b>Costes software</b>	181.97 €
<b>Costes hardware</b>	283.36 €
<b>Costes de personal</b>	11086.06 €
<b>Otros costes</b>	2400 €
<b>Costes totales</b>	13951.39 €

Tabla A.2: Tabla de costes del proyecto

<sup>8</sup><https://es.talent.com/salary?job=product+owner>

<sup>9</sup><https://www.idealista.com/alquiler-oficinas/burgos-burgos/>

## Beneficios

Los beneficios esperados no son monetarios. El objetivo real del proyecto para la alumna es consolidar y ampliar los conocimientos adquiridos durante su carrera universitaria, así como adquirir nuevas habilidades que puedan resultar útiles en su futura vida laboral. Además, busca obtener el título universitario como culminación a su dedicación personal durante estos 4 años.

Además de los aspectos académicos y personales, el proyecto tiene un propósito social, contribuir a mejorar la calidad de vida de los pacientes con parkinson al proporcionarles un acceso más fácil a sus constantes físicas y al facilitar el seguimiento de su enfermedad.

Por ello, los beneficios van más allá del dinero, el proyecto tiene un fin social y personal.

Pero, como se ha estado realizando hasta ahora, se considerarán los posibles beneficios como si el proyecto fuera llevado a cabo por una empresa real.

El proyecto se realiza en colaboración con la Asociación Parkinson Burgos, gracias a los convenios establecidos con la Universidad de Burgos [3]. No se espera obtener ingresos directos por el uso de la aplicación web, pero se explorará la posibilidad de mitigar algunos costos a través de donaciones a la asociación.

No se contempla la implementación de suscripciones a los usuarios, ya que se busca que la aplicación sea accesible para toda la población, por lo que no se espera obtener beneficios monetarios de los pacientes.

Aunque inicialmente no se planeaba la idea de monetizar la aplicación con publicidad, es útil considerar cómo esta podría generar ingresos pasivos. Utilizando servicios como Google AdSense<sup>10</sup>, es posible integrar anuncios sin afectar negativamente la experiencia de los usuarios. Además, para el desarrollador es tan simple como añadir un fragmento de código para que Google muestre automáticamente anuncios relevantes en la aplicación y adaptados a su diseño en cualquier dispositivo.

Mediante su página oficial se puede realizar una estimación de los ingresos anuales que se obtendrían con los anuncios. Se ha de seleccionar la ubicación de los visitantes del sitio web y la categoría del contenido, en el caso del proyecto actual la ubicación es Europa y en el campo de la salud. Indicando el número de veces al mes que las páginas del sitio web se cargan, la página

---

<sup>10</sup><https://adsense.google.com/start/>

calcula los ingresos. El mínimo de visitas al mes que permite la herramienta de estimación es 50000, lo que supondría unos ingresos de \$ 3204, unos 2956 €.

Se estima que la aplicación web del proyecto podría alcanzar los 200 usuarios activos al mes, entre pacientes y médicos, cuando esté en funcionamiento. Si cada usuario visita la aplicación 10 veces al mes, formaría un total de 2000 visitas al mes. Aplicando la regla de tres de la página, supondrían \$ 128.16, unos 118 €.

Esta cantidad es simplemente una estimación, los ingresos reales dependen de diversos factores: la demanda de los anunciantes, la ubicación y el dispositivo de los usuarios, el contenido, la temporada, el tamaño de los anuncios...

La rentabilidad del proyecto determina si los beneficios serán suficientes para cubrir los costes, lo que permite confirmar si el proyecto será viable o no. Como se puede observar tras estudiar la viabilidad económica del proyecto, los costes han sido de 13951.39 € y los beneficios han sido nulos. Típicamente la rentabilidad [5] se calcularía utilizando la fórmula del retorno de la inversión, que divide los beneficios netos menos los costes entre los costes, pero en este proyecto el resultado de esa operación sería una rentabilidad del -100 %.

El resultado nos ofrece que el proyecto ha supuesto pérdidas y no es rentable pero, como se trata de un Trabajo de Fin de Grado, se va a hacer caso omiso del resultado obtenido.

Si se tuviera en cuenta la idea de utilizar publicidad, se obtendrían unos 118 € al mes, 1416 € al año. Calculando entonces la rentabilidad, se obtendría una rentabilidad del -89.8 %:

$$\text{ROI} = \frac{1416 \text{ €} - 13951.39 \text{ €}}{13951.39 \text{ €}} = \frac{-12535.39 \text{ €}}{13951.39 \text{ €}} \approx -0.898$$

Para rentabilizar el proyecto obteniendo esos beneficios anuales, serían necesarios casi 10 años:

$$\text{Años para rentabilizar} = \frac{13951.39 \text{ €}}{1416 \text{ €}} \approx 9.85$$

## Viabilidad legal

El estudio de la viabilidad legal consiste en analizar si existe alguna barra legal que impida realizar el proyecto. Se deben analizar todas las cuestiones

jurídicas que afecten al proyecto: acuerdos de confidencialidad, derechos de propiedad intelectual, legislaciones de la Unión Europea, requisitos legales del sector...

Ya que el proyecto se realiza junto con una universidad y una asociación médica, se deben considerar diversas normativas. Las más relevantes pueden ser:

- Reglamento General de Protección de Datos (RGPD [4]): ya que se trabaja con datos de pacientes reales, residentes en la Unión Europea, hay que seguir estos estrictos estándares para la protección de sus datos.
- Consentimiento informado: según el RGPD se debe contar con el consentimiento de los pacientes para poder hacer uso de su información personal. Se entiende consentimiento como: «la manifestación de voluntad libre, específica, informada e inequívoca por la cual una persona acepta, mediante una clara acción afirmativa, el tratamiento de sus datos personales [1]». Cuando los pacientes accedieron a llevar el sensor con fines de investigación, dieron su consentimiento para que sus datos fueran utilizados en el ámbito universitario para realizar proyectos como el desarrollado.
- Seguridad de los datos: se deben implementar medidas de seguridad que garanticen la confidencialidad e integridad de los datos. Para ello se utiliza un acceso con usuario y contraseña para poder acceder a los datos de cada paciente, acompañado de medidas de ciberseguridad.
- Propiedad intelectual: ya que durante el proyecto se utilizan obras de otros autores (por ejemplo, se utiliza el código de Python utilizado para tratar y mostrar los datos del sensor creado por los creadores de SENSE4CARE [9]) hay que abordar el tema de los derechos de autor.

Además, hay que tener en cuenta las licencias de las librerías de Python utilizadas durante el proyecto. En la tabla A.3 se resumen las licencias de las bibliotecas usadas y sus versiones. Se han obtenido escribiendo el comando «pip-licenses» en la terminal del entorno virtual.

El orden de restrictividad, de la menos a la más restrictiva, de las licencias de las librerías usadas es el siguiente:

1. MIT License

2. BSD License (2-Clause y 3-Clause)
3. PSF License
4. Apache Software License 2.0

Para el proyecto se debe escoger la más restrictiva para asegurar que todas las condiciones de las licencias sean cumplidas, evitando conflictos legales. Es por ello que en el actual proyecto se utilizará una licencia Apache License 2.0<sup>11</sup>, usada por la librería «mediapipe». Es una licencia permisiva y similar a las licencias MIT y BSD, pero se diferencia de ellas ya que incluye además una cláusula de concesión de patente, protegiendo a los usuarios que usen el software de demandas por infracción de patentes por parte de los que contribuyeron en el desarrollo del software.

---

<sup>11</sup><https://www.apache.org/licenses/LICENSE-2.0>

Librería	Versión	Licencia
Flask	3.0.0	BSD-3-Clause License
flask-babel	4.0.0	BSD License
Flask-SQLAlchemy	3.1.1	BSD-3-Clause License
Flask-WTF	1.2.1	BSD-3-Clause License
Jinja2	3.1.2	BSD-3-Clause License
jsonschema	4.19.1	MIT License
matplotlib	3.8.0	PSF License
mediapipe	0.10.11	Apache Software License 2.0
numpy	1.26.1	BSD-3-Clause License
pandas	1.5.3	BSD-3-Clause License
pydantic	1.10.15	MIT License
requests	2.31.0	Apache Software License 2.0
scikit-learn	1.3.2	BSD-3-Clause License
scipy	1.11.3	BSD License
tsfresh	0.20.0	MIT License

Tabla A.3: Librerías utilizadas, junto con sus versiones y licencias.



## *Apéndice B*

---

# **Especificación de Requisitos**

---

## **B.1. Introducción**

En este apéndice se realiza la especificación de los requisitos que la aplicación web debe de cumplir para entregar al cliente un producto acorde con sus requerimientos.

Se siguen los apuntes de la asignatura de Ingeniería del *Software*, cursada durante el segundo año del grado en Ingeniería Informática, para realizar el análisis de requisitos del sistema siguiendo el estándar IEEE 830/1993 (*IEEE Recomended Practice for Software Requeriments*).

Se utiliza UML (*Unified Modelling Language*) como sistema de modelado para capturar dichos requisitos en un diagrama de casos de uso. Se usarán como referencia los apuntes de la asignatura de Análisis y Diseño de Sistemas.

La fase del modelado es fundamental para reducir la complejidad del *software* y estructurar las ideas. Se comienza con una reunión del equipo de desarrollo (en este caso la alumna) con el cliente (en este caso el tutor), que especifica las funcionalidades que se espera que tenga la aplicación. Estas funcionalidades se corresponden con los requisitos.

El anexo se divide en:

- Objetivos generales: se resumen los objetivos que se pretenden cumplir con el proyecto.
- Catálogo de requisitos: se detallan las características que la aplicación web debe cumplir, analizando sus requisitos funcionales y no funcionales.

- Especificación de requisitos: en este último apartado se definen los actores que forman parte del sistema y los casos de uso que realizan. Se crea un diagrama general de casos de uso presentando las relaciones y la situación general. Además se especifican las características de cada caso de uso específico mediante tablas.

## B.2. Objetivos generales

Se procede a numerar los principales objetivos del proyecto:

1. Ofrecer a las personas con parkinson una manera rápida e intuitiva de llevar un seguimiento de su enfermedad.
2. Desarrollar una aplicación web utilizando el *framework* Flask, que permita a los pacientes ver y predecir su evolución, además de permitir a los médicos gestionar a sus pacientes. Podrán acceder a todas las funcionalidades desde su navegador sin necesidad de descargar nada en su equipo.
3. Estas predicciones se pretenden conseguir gracias a técnicas de *machine learning* mediante bibliotecas Python.
4. Almacenar los datos de los pacientes de forma estructurada, en una base de datos relacional como MariaDB.

## B.3. Catálogo de requisitos

Los requisitos son utilizados durante el desarrollo *software* para especificar las características y el comportamiento que debe tener el sistema. El objetivo de realizar un catálogo de requisitos es definir de forma clara, completa y verificable las funcionalidades (y restricciones) del sistema que se quiere elaborar.

Existen dos tipos: requisitos funcionales y no funcionales. Los funcionales se centran en qué debe hacer el sistema desde el punto de vista de los usuarios, es decir, los servicios que debe proporcionar la página, mientras que los no funcionales se centrarán en cómo se debe comportar, sus características generales, como usabilidad, mantenimiento, rendimiento...

Tras múltiples entrevistas con el cliente (en este caso el tutor), y tras sucesivas versiones de este documento, se han aprobado los siguientes requisitos:

## Requisitos funcionales

- **RF1:** El sistema debe **gestionar los datos** medidos por el sensor que llevan incorporado los pacientes.
  - **RF1.1:** Debe permitir subir datos del sensor.
  - **RF1.2:** Debe mostrar gráficas que representen los datos del sensor.
    - **RF1.2.1:** Se debe poder elegir el rango de tiempo que se quiere mostrar de la gráfica de datos.
    - **RF1.2.2:** Se debe poder elegir la gráfica que se quiere mostrar, mediante una lista de preselección.
- **RF2:** El sistema debe **gestionar los datos** obtenidos tras el análisis de los vídeos de los pacientes.
  - **RF2.1:** El sistema debe mostrar gráficas que representen las características obtenidas de los vídeos.
  - **RF2.2:** El sistema debe ser capaz de **predecir** datos futuros a partir de los datos históricos de los vídeos de los pacientes.
- **RF3:** La aplicación tiene que ser capaz de mostrar un **listado** con todos los pacientes que un médico tiene a su cargo.
- **RF4:** La aplicación tiene que permitir la subida y eliminación de **vídeos** de la base de datos.
- **RF5:** El sistema debe permitir cambiar **información personal** de los usuarios como domicilio, número de teléfono, médico asociado...
  - **RF5.1:** El campo del médico asociado consistirá en una lista de preselección, mostrándose los médicos disponibles.
- **RF6:** El sistema debe mostrar el **nombre y foto** de perfil del usuario en la esquina superior derecha.
- **RF7:** El sistema debe permitir **iniciar sesión** en la aplicación con nombre de usuario y contraseña.
  - **RF7.1:** El campo de la contraseña debe poderse mostrar y ocultar mediante un botón.
  - **RF7.2:** El campo nombre debe aceptar caracteres alfabéticos, numéricos y especiales.

- **RF8:** El sistema debe permitir a los usuarios **cerrar sesión**.
- **RF9:** El sistema debe soportar la **gestión de sus usuarios** (creación, eliminación y modificación).
- **RF10:** El sistema debe mostrar el **logo** de la Universidad de Burgos y el logo de la Asociación de Parkinson de Burgos, redireccinando a sus páginas web.
  - **RF10.1:** El sistema debe mostrar el logo de la propia aplicación, redireccinando a la página principal.
- **RF11:** El sistema debe mostrar **información** y contexto sobre los creadores de la aplicación web.
- **RF12:** La aplicación debe tener una opción que ponga en **contacto** a los usuarios con los creadores de esta, mediante correo electrónico u otros medios, para responder posibles dudas sobre su uso.
- **RF13:** La aplicación web debe soportar el cambio de **idioma**.

## Requisitos no funcionales

- **RNF1: Usabilidad:** el sistema debe ser intuitivo, resultando fácil de usar para médicos y pacientes.
- **RNF2: Internacionalización:** el sistema debe poder ser utilizado por personas de cualquier nacionalidad.
- **RNF3: Rendimiento:** el sistema debe tener un tiempo de respuesta adecuado, siendo capaz de manejar los grandes volúmenes de datos recogidos en la base de datos del sensor, así como ofreciendo una espera al usuario de menos de 60 segundos en mostrar las gráficas o en subir y analizar los vídeos.
- **RNF4: Seguridad:** la aplicación web debe garantizar la protección de la información sensible de los usuarios como datos médicos, datos personales o contraseñas. Las contraseñas se almacenan de forma segura utilizando técnicas de cifrado. Se utilizan técnicas para prevenir ataques como csrf (*cross-site request forgery*) en los formularios.
- **RNF5: Mantenibilidad:** el código fuente de la aplicación web debe estar bien estructurado así como comentado, para resultar fácil de mantener y actualizar en un futuro.

- **RNF6: Disponibilidad:** la aplicación debe estar disponible el 99 % del tiempo para los usuarios.
- **RNF7: Compatibilidad:** la aplicación debe ser compatible con los navegadores web más utilizados (Chrome, Edge, Firefox... ) y con cualquier dispositivo electrónico (ordenador, teléfono móvil, tableta... ).
- **RNF8: Respuesta a errores:** el sistema debe ofrecer mensajes de error claros y amigables para ser comprendidos por todo tipo de usuarios, así como detalles técnicos para el equipo de soporte.

## B.4. Especificación de requisitos

Se procede a realizar la especificación de requisitos de la aplicación web del proyecto, basándose en el catálogo de requisitos desarrollado en el apartado **B.3**.

Antes de determinar los casos de uso, o interacciones con el sistema, se deben definir los actores que realizarán dichos casos de uso.

Se entiende por actor a cualquier elemento que intercambia información en el sistema. Pueden ser usuarios directos del sistema, responsables de su uso o mantenimiento, u otros sistemas. En este proyecto, el sistema corresponde con la aplicación web desarrollada y los actores son los siguientes:

- Paciente: usuario directo de la aplicación. Tras iniciar sesión con su nombre de usuario y contraseña, estos usuarios pueden visualizar sus gráficas de evolución, editar sus datos personales y predecir sus evoluciones futuras en el movimiento de pinza. Corresponden físicamente a los pacientes con parkinson que llevan el sensor de movimiento y forman parte del proyecto. Al acabar pueden cerrar su sesión.
- Médico: usuario directo de la aplicación web. Tras iniciar sesión de igual manera que los pacientes, estos usuarios disponen de una lista con todos los pacientes a su cargo. Pueden cambiar los datos personales de sus pacientes, subir los datos medidos por el sensor que llevan incorporado o subir vídeos de cada paciente. Al igual que los pacientes, pueden acceder a la visualización de gráficas con los datos del sensor y a las predicciones de la evolución del movimiento de pinza de cada paciente a su cargo. Corresponden físicamente con los médicos del Hospital de Burgos relacionados con el proyecto y que tratan a los pacientes.

- Administrador: responsable del sistema y de administrar los usuarios que acceden a él, creando y eliminando cuentas de usuario cuando sea necesario. Tienen acceso a todas las funcionalidades.

Tras comentar resumidamente los actores existentes y sus principales actividades, se presentan los casos de uso que describen el comportamiento del sistema. Se entiende por casos de uso a las interacciones entre los usuarios y el sistema informático, que capturan los requisitos funcionales del sistema. Corresponden con las secuencias de acciones que el sistema ejecuta y producen un resultado observable para los actores.

Se ha realizado un diagrama general de casos de uso, en el que se representan las relaciones entre los casos de uso y los actores, tal como se indica en la Figura B.1. Los casos de uso se representan con elipses conteniendo el nombre y los actores se representan con «monigotes» con el nombre del actor:

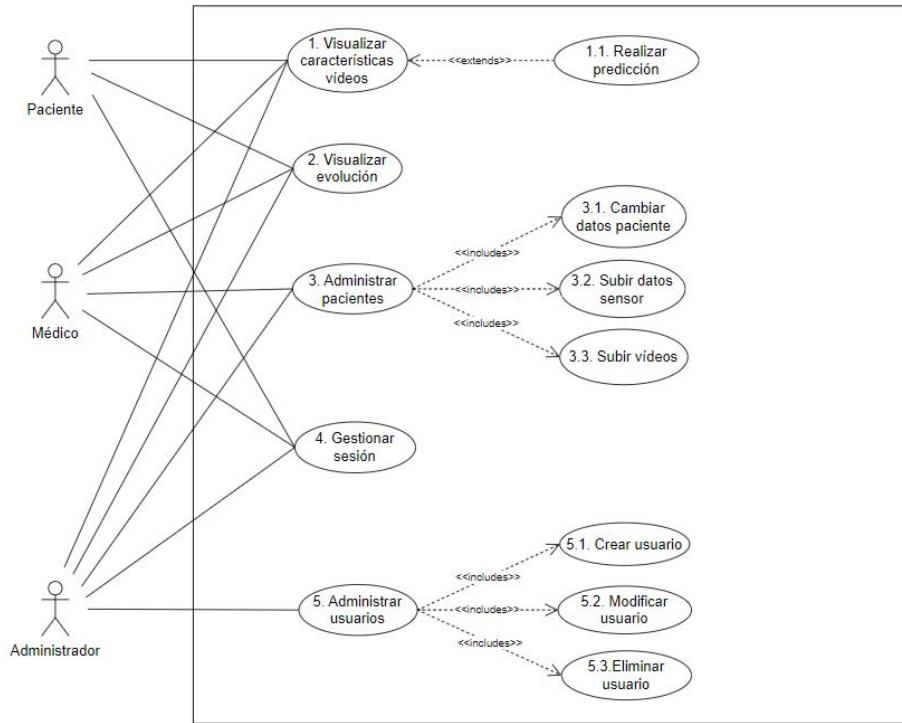


Figura B.1: Diagrama de casos de uso

A continuación se detallan tablas individuales para cada caso de uso, comentando las acciones que se realizan, su importancia, los RF asociados, precondiciones, postcondiciones y excepciones.

<b>CU-1</b>	<b>Visualizar características vídeos</b>
<b>Versión</b>	1.0
<b>Autor</b>	Sandra Díaz Aguilar
<b>Requisitos asociados</b>	RF2, RF1.2.1
<b>Descripción</b>	Permitir a los usuarios visualizar gráficas que muestren las características del movimiento de pinza obtenidas tras el análisis de los vídeos de los pacientes.
<b>Precondición</b>	El usuario debe haber iniciado sesión y ser paciente, médico o administrador. Si es médico o administrador debe haber elegido de qué paciente desea ver la evolución.
<b>Acciones</b>	<ol style="list-style-type: none"> <li>1. El usuario debe bajar en la página hasta encontrar las gráficas.</li> </ol>
<b>Postcondición</b>	Se habrá visualizado un gráfico por cada mano que muestre las características del movimiento.
<b>Excepciones</b>	En el caso de que el paciente no tenga vídeos, aparecerá un mensaje en pantalla.
<b>Importancia</b>	Alta

Tabla B.1: CU-1 Visualizar características vídeos.

<b>CU-1.1</b>	<b>Realizar predicción</b>
<b>Versión</b>	1.0
<b>Autor</b>	Sandra Díaz Aguilar
<b>Requisitos asociados</b>	RF2, RF2.1
<b>Descripción</b>	Realizar gráficas que predigan, utilizando <i>machine learning</i> , la evolución de los pacientes.
<b>Precondición</b>	El usuario debe haber iniciado sesión y debe estar en la pantalla de visualizar los vídeos.
<b>Acciones</b>	<ol style="list-style-type: none"> <li>1. El usuario pulsa el botón que hace que se muestren las predicciones.</li> </ol>
<b>Postcondición</b>	Se habrá visualizado una gráfica por mano, con datos futuros que corresponden con la predicción.
<b>Excepciones</b>	Si no hay suficientes vídeos, el modelo no puede realizar la predicción.
<b>Importancia</b>	Alta

Tabla B.2: CU-1.1 Realizar predicción.

<b>CU-2</b>	<b>Visualizar evolución</b>
<b>Versión</b>	1.0
<b>Autor</b>	Sandra Díaz Aguilar
<b>Requisitos asociados</b>	RF1, RF1.2, RF1.2.1, RF1.2.2
<b>Descripción</b>	Permitir a los usuarios visualizar gráficas que muestran datos de su elección de los medidos por el sensor que llevan los pacientes. Para ello se deben seleccionar los datos a mostrar y el intervalo de fechas que se quiere visualizar.
<b>Precondición</b>	El usuario debe haber iniciado sesión y ser paciente, médico o administrador. Si es médico o administrador debe haber elegido de qué paciente desea ver la evolución. Además, se debe disponer de un registro de datos del paciente seleccionado para las fechas a seleccionar.
<b>Acciones</b>	<ol style="list-style-type: none"> <li>1. El usuario elige los datos a mostrar de la lista de gráficos existentes.</li> <li>2. El usuario selecciona las fechas por medio de un calendario.</li> <li>3. El usuario pulsa el botón que hace que se muestren los datos.</li> </ol>
<b>Postcondición</b>	Se habrá visualizado un gráfico que muestre datos del sensor.
<b>Excepciones</b>	En el caso de que se vaya a indicar una fecha en la que no se tienen registros, el calendario no lo permitirá.
<b>Importancia</b>	Alta

Tabla B.3: CU-2 Visualizar evolución.

<b>CU-3</b>	<b>Administrar pacientes</b>
<b>Versión</b>	1.0
<b>Autor</b>	Sandra Díaz Aguilar
<b>Requisitos asociados</b>	RF3
<b>Descripción</b>	Se permite a los médicos acceder a un listado de todos los pacientes a su cargo, pudiendo editar su información personal, subir vídeos, subir datos y ver su evolución con gráficas del sensor y de los vídeos.
<b>Precondición</b>	El usuario debe haber iniciado sesión y ser un médico.
<b>Acciones</b>	<ol style="list-style-type: none"> <li>1. Situarse en el listado de pacientes.</li> </ol>
<b>Postcondición</b>	Se mostrará un listado de los pacientes.
<b>Excepciones</b>	En el caso de que el médico no tenga pacientes a su cargo no aparecerá nada.
<b>Importancia</b>	Alta

Tabla B.4: CU-3 Administrar pacientes.

<b>CU-3.1</b>	<b>Cambiar datos paciente</b>
<b>Versión</b>	1.0
<b>Autor</b>	Sandra Díaz Aguilar
<b>Requisitos asociados</b>	RF5, RF5.1
<b>Descripción</b>	Permite a un médico editar datos personales de los pacientes a su cargo, tales como dirección, teléfono, médico asociado...
<b>Precondición</b>	El usuario debe haber iniciado sesión y ser un médico. Además, debe situarse en el perfil del paciente sobre el cual se quieren editar datos personales.
<b>Acciones</b>	<ol style="list-style-type: none"> <li>1. El médico entra al listado de sus pacientes.</li> <li>2. Selecciona el paciente</li> <li>3. Pulsa el botón que muestra la información personal del paciente.</li> <li>4. Se editan los datos elegidos.</li> <li>5. Se debe confirmar la edición de los datos.</li> </ol>
<b>Postcondición</b>	Los cambios realizados se verán reflejados en la base de datos del paciente.
<b>Excepciones</b>	Cada campo tiene sus propias excepciones: la fecha de nacimiento debe corresponder con una fecha pasada válida, el médico asociado debe corresponder con uno existente en el registro, la dirección debe ser válida, el número de teléfono debe tener la longitud adecuada...
<b>Importancia</b>	Media

Tabla B.5: CU-3.1 Cambiar datos paciente.

<b>CU-3.2</b>	<b>Subir datos sensor</b>
<b>Versión</b>	1.0
<b>Autor</b>	Sandra Díaz Aguilar
<b>Requisitos asociados</b>	RF1, RF1.1
<b>Descripción</b>	Permite a un médico (o administrador) subir a la base de datos de la aplicación los datos recogidos por el sensor de un paciente.
<b>Precondición</b>	El usuario debe haber iniciado sesión y ser un médico (o administrador). Además, debe situarse en el perfil del paciente sobre el cual se quieren subir datos.
<b>Acciones</b>	<ol style="list-style-type: none"> <li>1. El médico entra al listado de sus pacientes.</li> <li>2. Selecciona el paciente.</li> <li>3. Pulsa el botón que permite subir nuevos datos.</li> <li>4. El médico es redirigido a sus archivos, donde debe escoger el archivo de datos que desea subir.</li> <li>5. Aparecen los datos escogidos y se debe confirmar la subida.</li> </ol>
<b>Postcondición</b>	Los datos añadidos se verán reflejados en la base de datos del paciente.
<b>Excepciones</b>	En caso de no haber seleccionado un formato de archivo adecuado, el sistema no dejará subir los datos, apareciendo un mensaje de error.
<b>Importancia</b>	Alta

Tabla B.6: CU-3.2 Subir datos sensor.

<b>CU-3.3</b>	<b>Subir vídeos</b>
<b>Versión</b>	1.0
<b>Autor</b>	Sandra Díaz Aguilar
<b>Requisitos asociados</b>	RF4
<b>Descripción</b>	Permite a un médico (o administrador) añadir un vídeo a la lista de vídeos asociados a un paciente.
<b>Precondición</b>	El usuario debe haber iniciado sesión y ser un médico (o administrador). Además, debe situarse en el perfil del paciente al que se quiere asociar dicho vídeo. Los pacientes deben haber grabado previamente esos vídeos, que deben estar a disposición del médico.
<b>Acciones</b>	<ol style="list-style-type: none"> <li>1. El médico entra al listado de sus pacientes.</li> <li>2. Selecciona el paciente.</li> <li>3. Pulsa el botón que permite subir un vídeo.</li> <li>4. El médico es redirigido a sus archivos, donde debe escoger el archivo de vídeo que desea subir.</li> <li>5. Se deben indicar algunas características del vídeo en cuestión.</li> <li>6. Aparecen el vídeo y los datos escogidos, y se debe confirmar la subida.</li> </ol>
<b>Postcondición</b>	El vídeo debe haberse añadido a los registros de la base de datos del paciente.
<b>Excepciones</b>	En caso de no haber seleccionado un formato de vídeo adecuado, el sistema no dejará subir el vídeo, mostrando un mensaje de error.
<b>Importancia</b>	Media

Tabla B.7: CU-3.3 Subir vídeos.

CU-4	Gestionar sesión
<b>Versión</b>	1.0
<b>Autor</b>	Sandra Díaz Aguilar
<b>Requisitos asociados</b>	RF7, RF7.1, RF7.2, RF8
<b>Descripción</b>	Permite a un usuario iniciar sesión para acceder a sus funcionalidades de la aplicación web y cerrar posteriormente dicha sesión.
<b>Precondición</b>	Para iniciar sesión: No debe haber una sesión ya iniciada, si es así se debe primero cerrar la sesión anterior. Debe existir el perfil al que se desea acceder. Para cerrar sesión: Se debe tener una sesión activa.
<b>Acciones</b>	<p>Para iniciar sesión:</p> <ol style="list-style-type: none"> <li>1. Entrar a la página principal de la aplicación web.</li> <li>2. Seleccionar el botón arriba a la derecha de «Iniciar sesión», que redirigirá a la ventana de inicio de sesión.</li> <li>3. El usuario debe introducir el nombre de usuario asociado a su cuenta.</li> <li>4. El usuario debe introducir la contraseña asociada a la cuenta.</li> <li>5. Se debe pulsar el botón de «Iniciar sesión».</li> </ol> <p>Para cerrar sesión:</p> <ol style="list-style-type: none"> <li>1. El usuario debe pulsar el botón de «Cerrar sesión» que se encuentra arriba a la derecha, junto a su nombre y foto de perfil.</li> <li>2. Se debe confirmar que se desea cerrar sesión.</li> </ol>
<b>Postcondición</b>	Al iniciar sesión el usuario será redirigido a la página principal de su cuenta dependiendo del tipo de usuario. Al cerrar sesión el usuario será redirigido a la página principal y la sesión anterior se habrá cerrado.
<b>Excepciones</b>	Si el usuario introduce un nombre o una contraseña incorrectas se le avisará con un error.
<b>Importancia</b>	Alta

Tabla B.8: CU-4 Gestionar sesión.

<b>CU-5</b>	<b>Administrar usuarios</b>
<b>Versión</b>	1.0
<b>Autor</b>	Sandra Díaz Aguilar
<b>Requisitos asociados</b>	RF9
<b>Descripción</b>	Permite a un administrador gestionar los usuarios que acceden a la aplicación.
<b>Precondición</b>	El usuario debe haber iniciado sesión y ser administrador.
<b>Acciones</b>	<ol style="list-style-type: none"> <li>1. El administrador accede a la ventana de administración de usuarios, donde puede elegir qué tipo de usuario mostrar.</li> <li>2. Elige el tipo de usuario y aparecerá una tabla con los administradores, médicos o pacientes.</li> </ol>
<b>Postcondición</b>	Aparecerá un listado de los usuarios de la aplicación.
<b>Excepciones</b>	No hay excepciones.
<b>Importancia</b>	Alta

Tabla B.9: CU-5 Administrar usuarios.

<b>CU-5.1</b>	<b>Crear usuario</b>
<b>Versión</b>	1.0
<b>Autor</b>	Sandra Díaz Aguilar
<b>Requisitos asociados</b>	RF9
<b>Descripción</b>	Permite a los administradores añadir un nuevo usuario al sistema.
<b>Precondición</b>	El usuario debe haber iniciado sesión y ser administrador, así como estar situado en la lista de usuarios.
<b>Acciones</b>	<ol style="list-style-type: none"> <li>1. Pulsar en el botón que añade un usuario al sistema.</li> <li>2. En el formulario que aparece se deben llenar los datos de nombre de usuario, contraseña, nombre, foto de perfil...</li> <li>3. Pulsar en el botón que lo añade y realizar la confirmación.</li> </ol>
<b>Postcondición</b>	El usuario creado se verá reflejado en la base de datos de usuarios.
<b>Excepciones</b>	Cada campo del formulario tiene sus propias excepciones: nombre de usuario nuevo y con longitud adecuada, contraseña suficientemente segura, etc.
<b>Importancia</b>	Alta

Tabla B.10: CU-5.1 Crear usuario.

<b>CU-5.2</b>	<b>Modificar usuario</b>
<b>Versión</b>	1.0
<b>Autor</b>	Sandra Díaz Aguilar
<b>Requisitos asociados</b>	RF9
<b>Descripción</b>	Permite a los administradores modificar la información de los usuarios al sistema.
<b>Precondición</b>	El usuario debe haber iniciado sesión y ser administrador, así como estar situado en la lista de usuarios.
<b>Acciones</b>	<ol style="list-style-type: none"> <li>1. Pulsar sobre el usuario que se desea modificar.</li> <li>2. Pulsar en el botón de modificar usuario.</li> <li>3. Editar el campo o campos que se deseen.</li> <li>4. Aceptar confirmación de modificación.</li> </ol>
<b>Postcondición</b>	Los datos del usuario aparecerán modificados en la base de datos.
<b>Excepciones</b>	No hay excepciones.
<b>Importancia</b>	Baja

Tabla B.11: CU-5.2 Modificar usuario.

<b>CU-5.3</b>	<b>Eliminar usuario</b>
<b>Versión</b>	1.0
<b>Autor</b>	Sandra Díaz Aguilar
<b>Requisitos asociados</b>	RF9
<b>Descripción</b>	Permite a los administradores eliminar un usuario del sistema.
<b>Precondición</b>	El usuario debe haber iniciado sesión y ser administrador, así como estar situado en la lista de usuarios.
<b>Acciones</b>	<ol style="list-style-type: none"> <li>1. Pulsar sobre el usuario que se desea eliminar.</li> <li>2. Pulsar en el botón de eliminar usuario.</li> <li>3. Aceptar confirmación de borrado.</li> </ol>
<b>Postcondición</b>	El usuario eliminado no aparecerá en la base de datos de usuarios, por lo que, al intentar iniciar sesión con esos datos, ocurrirá un error.
<b>Excepciones</b>	No se puede eliminar un médico que tenga pacientes asignados, ya que dichos pacientes quedarían sin médico.
<b>Importancia</b>	Baja

Tabla B.12: CU-5.3 Eliminar usuario.

## *Apéndice C*

---

# Especificación de diseño

---

## C.1. Introducción

El objetivo de la especificación de diseño es crear una guía para el desarrollo de la aplicación web, describiendo la estructura y organización de los datos que los desarrolladores deben seguir para producir un *software* robusto.

Además, se describirá el procedimiento interno que utiliza el sistema para mostrar los datos a los usuarios y para el resto de funcionalidades, así como la arquitectura elegida con la que se comunican los integrantes del producto *software*.

Este apéndice se divide en:

- Diseño de datos: se presentan el diagrama entidad-relación, el diagrama relacional y el diccionario de datos realizados al crear la base de datos.
- Diseño de interfaces: se muestra el *mockup* de la aplicación, cada una de las ventanas diseñadas en Pencil.
- Diseño procedimental: se realizan diagramas de secuencia para mostrar la interacción de los componentes del sistema durante cada una de las funcionalidades de la aplicación mediante pasos ordenados.
- Diseño arquitectónico: se realiza un diagrama de despliegue para exponer las relaciones entre el cliente, el servidor, la base de datos...

Se han seguido los apuntes de la asignatura de Ingeniería del *Software* para realizar dichos diagramas.

## C.2. Diseño de datos

### Modelo entidad-relación

En el diagrama entidad-relación de la Figura C.1 se representan las entidades existentes en la base de datos del proyecto, así como las relaciones entre ellas. Ayuda a comprender los datos con los que se va a trabajar en el proyecto y a identificar errores.

Existen tres tipos diferentes de usuarios: pacientes, médicos y administradores. Como todos ellos comparten ciertas características básicas, se ha indicado mediante un usuario «Usuario» general, que se relaciona con los tres tipos mediante una relación ISA. Las características comunes para todos los usuarios de la aplicación son: identificador, nombre, apellido, foto de perfil, correo electrónico, nombre de usuario y contraseña.

Los pacientes tienen además otros campos como: fecha de nacimiento, dirección, teléfono, lateralidad (que puede ser diestro o zurdo) y sensor (que puede tener o no tener sensor.)

Los pacientes se relacionan con los médicos mediante una relación 1 a muchos, en la que un paciente es atendido por un único médico, pero un médico se encarga de atender a muchos pacientes.

A su vez, los pacientes se relacionan con otras dos entidades (registros y vídeos) mediante relaciones 1 a muchos, ya que un paciente puede tener muchos registros medidos por el sensor o muchos vídeos, pero cada registro/vídeo solo puede corresponder con un paciente.

Los registros tienen un identificador de registro, campos para la fecha inicial y final del registro, calculadas por un *script* del programa, y un campo llamado datos en crudo en el que se almacena la ubicación del registro en vez del registro en sí.

Los vídeos tienen también un identificador, y los campos fecha, mano dominante, lentitud y amplitud que son introducidos por el médico o administrador al añadir el vídeo a la base de datos. El campo contenido contiene la ruta al vídeo y el resto de características son obtenidas tras el análisis de los vídeos al subirlos.

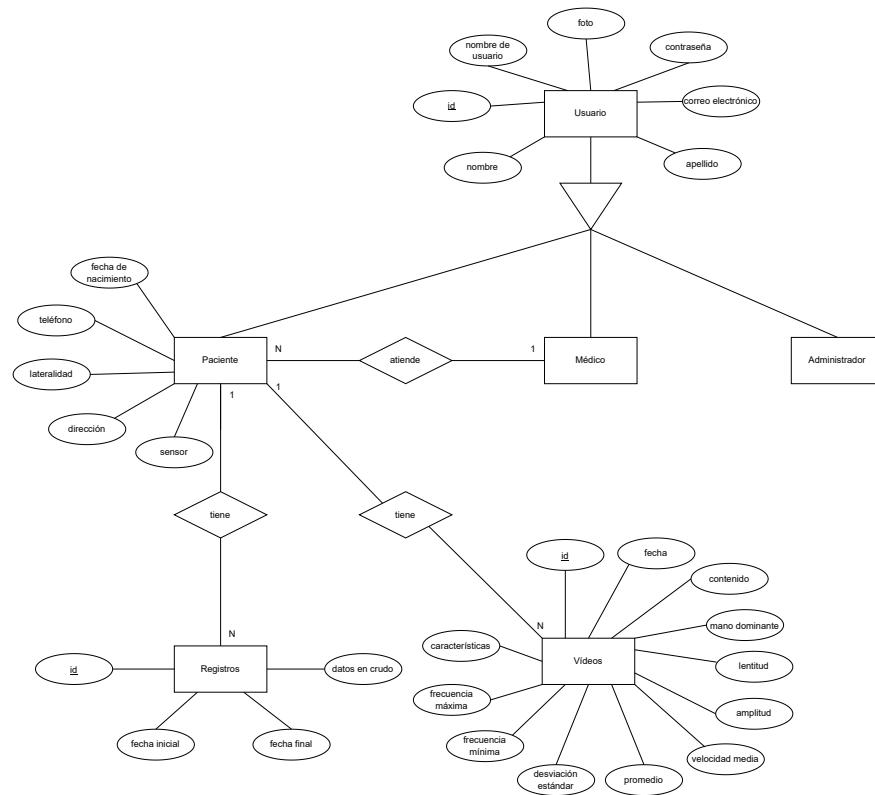


Figura C.1: Diagrama entidad-relación.

## Diagrama relacional

Tras realizar el diagrama ER se ha realizado el diagrama relacional, que proporciona una representación más detallada de la base de datos y se espera que sea de utilidad para la futura mantenibilidad de la base de datos. Se adjunta en la Figura C.2.

Como no existe posibilidad de que exista alguien que sea usuario de la aplicación sin que sea paciente, médico o administrador, no se genera tabla para el supertipo de la relación ISA (usuario). Se crea una tabla para cada subtipo, en la que se incluyen los atributos comunes.

La tabla de pacientes tiene además las características adicionales comentadas anteriormente y se relaciona con las tablas de registros, videos y de médicos.

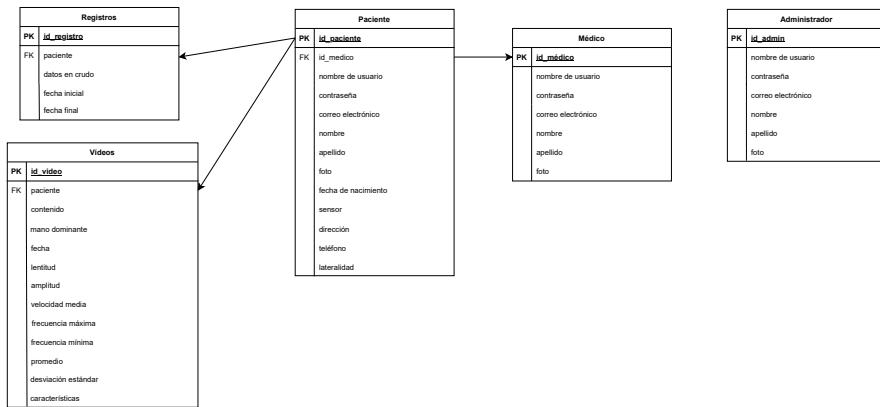


Figura C.2: Diagrama relacional de la base de datos.

## Diccionario de datos

Se adjunta a continuación el diccionario de datos para cada una de las cinco tablas de la base de datos. En el diccionario se proporciona una descripción de la finalidad de cada campo de cada tabla, además del tipo de campo y restricciones.

La tabla C.1 contiene la información de la tabla de administradores.

En la tabla C.2 se desarrolla la finalidad de los campos de la tabla de médicos, semejante a la tabla de administradores.

La extensa tabla C.3 muestra la distribución de la tabla utilizada para almacenar a los pacientes de la aplicación.

En la tabla C.4 se describen los campos de la tabla registros.

Nombre	Tipo	Columna	Descripción
<u>id_admin</u>	INTEGER	<u>PK</u>	Identificador único del administrador. Generado automáticamente.
nombre_de_usuario	VARCHAR(50)	UNIQUE NOT NULL	Nombre de usuario del administrador.
contraseña	VARCHAR(64)	NOT NULL	Contraseña del administrador.
correo_electronico	VARCHAR(255)	UNIQUE NOT NULL	Correo electrónico del administrador.
nombre	VARCHAR(50)	NOT NULL	Nombre del administrador.
apellido	VARCHAR(50)	NOT NULL	Apellido del administrador.
foto	VARCHAR(255)		Ruta de la imagen de perfil del administrador.

Tabla C.1: Diccionario de datos. Tabla administrador.

Nombre	Tipo	Columna	Descripción
<u>id_medico</u>	INTEGER	<u>PK</u>	Identificador único del médico. Generado automáticamente.
nombre_de_usuario	VARCHAR(50)	UNIQUE NOT NULL	Nombre de usuario del médico.
contraseña	VARCHAR(64)	NOT NULL	Contraseña del médico.
correo_electronico	VARCHAR(255)	UNIQUE NOT NULL	Correo electrónico del médico.
nombre	VARCHAR(50)	NOT NULL	Nombre del médico.
apellido	VARCHAR(50)	NOT NULL	Apellido del médico.
foto	VARCHAR(255)		Ruta de la imagen de perfil del médico.

Tabla C.2: Diccionario de datos. Tabla médico.

Nombre	Tipo	Columna	Descripción
<u>id_paciente</u>	INTEGER	<u>PK</u>	Identificador único del paciente. Generado automáticamente.
nombre_de_usuario	VARCHAR(50)	UNIQUE NOT NULL	Nombre de usuario del paciente.
contraseña	VARCHAR(64)	NOT NULL	Contraseña del paciente.
correo_electronico	VARCHAR(255)	UNIQUE NOT NULL	Correo electrónico del paciente.
nombre	VARCHAR(50)	NOT NULL	Nombre del paciente.
apellido	VARCHAR(50)	NOT NULL	Apellido del paciente.
foto	VARCHAR(255)		Ruta de la imagen de perfil del paciente.
fecha_de_nacimiento	DATE	NOT NULL	Fecha de nacimiento del paciente.
sensor	ENUM('SI', 'NO')		Indicador de si el paciente usa sensor o no.
direccion	VARCHAR(255)	NOT NULL	Dirección del paciente.
telefono	VARCHAR(15)	NOT NULL	Teléfono del paciente.
id_medico	INTEGER	FK	Identificador del médico asignado al paciente.
lateralidad	ENUM('diestro', 'zurdo')		Lateralidad del paciente.

Tabla C.3: Diccionario de datos. Tabla paciente.

Nombre	Tipo	Columna	Descripción
<u>id_registro</u>	INTEGER	<u>PK</u>	Identificador único del registro. Generado automáticamente.
paciente	INTEGER	FK NOT NULL	Identificador del paciente al que pertenece el registro.
datos_en_crudo	VARCHAR(250)	NOT NULL	Ruta del registro.
fecha_inicial	DATE		Fecha de inicio del registro.
fecha_final	DATE		Fecha de finalización del registro.

Tabla C.4: Diccionario de datos. Tabla registros.

Nombre	Tipo	Columna	Descripción
<b>id_video</b>	INTEGER	<u>PK</u>	Identificador único del vídeo. Generado automáticamente.
<b>paciente</b>	INTEGER	FK NOT NULL	Identificador del paciente al que pertenece el vídeo.
<b>fecha</b>	DATE	NOT NULL	Fecha del vídeo.
<b>contenido</b>	VARCHAR(250)	NOT NULL	Nombre del archivo de vídeo.
<b>mano_dominante</b>	ENUM('derecha', 'izquierda')	NOT NULL	Mano que aparece en el vídeo.
<b>lentitud</b>	ENUM('0', '1', '2', '3', '4')		Escala de lentitud del movimiento del vídeo.
<b>amplitud</b>	ENUM('0', '1', '2', '3', '4')		Escala de amplitud del movimiento del vídeo..
<b>velocidad_media</b>	VARCHAR(50)		Velocidad media registrada en el vídeo.
<b>frecuencia_max</b>	VARCHAR(50)		Frecuencia máxima registrada en el vídeo.
<b>frecuencia_min</b>	VARCHAR(50)		Frecuencia mínima registrada en el vídeo.
<b>promedio_max</b>	VARCHAR(50)		Promedio máximo registrado en el vídeo.
<b>desv_estandar_max</b>	VARCHAR(50)		Desviación estándar máxima registrada en el vídeo.
<b>diferencia_ranurada_min</b>	VARCHAR(50)		Diferencia ranurada mínima registrada en el vídeo (no utilizado).
<b>diferencia_ranurada_max</b>	VARCHAR(50)		Diferencia ranurada máxima registrada en el vídeo (no utilizado).
<b>características</b>	TEXT		Características adicionales del vídeo (no utilizado).

Tabla C.5: Diccionario de datos. Tabla vídeos.

Por último, en la tabla C.5 se encuentra el diccionario de datos de la tabla utilizada para almacenar los vídeos del paciente.

### C.3. Diseño de interfaces

Se procede a mostrar el diseño de interfaces realizado por la alumna tras las reuniones con el tutor para definir las funcionalidades que debe soportar la aplicación. Se corresponde con el *mockup* que crearía el equipo del proyecto tras las reuniones iniciales con el cliente para, posteriormente, enseñárselo a éste para que dé su opinión.

Se ha elegido la herramienta de Pencil para realizar el diseño ya que, durante el grado, se utiliza esta herramienta en varias asignaturas.

En la página principal mostrada en la Figura C.3 aparece el logo y el nombre de la aplicación. El logo se corresponde con un tulipán ya que es un símbolo de la enfermedad de Parkinson [8]. El nombre de la aplicación, «*Tremor Track*», es un juego de palabras que en inglés significa «realizar un seguimiento del temblor», por lo que se ajusta perfectamente al objetivo del proyecto.

Debajo, aparecen resumidas las utilidades de la aplicación para los dos tipos de usuario que van a hacer uso de ella.

El logo de la Universidad de Burgos y de la Asociación de Parkinson de Burgos aparecen en la parte baja de la ventana, permitiendo al usuario acceder a sus páginas oficiales si pincha en ellos.



Figura C.3: *Mockup - Página principal*

Además, en la parte de arriba aparecen 4 funcionalidades que se explican a continuación:

- Si se pulsa en «Sobre nosotros» se desplegará una sección (como se muestra en la Figura C.4) con información sobre los creadores de la aplicación (el tutor y la alumna) y el objetivo de esta.



Figura C.4: *Mockup - Página principal (info)*

- Si se pulsa en el apartado de contacto, se mostrarán los correos electrónicos de los creadores de la *app*, así como la ubicación en la que se les puede encontrar, sirviendo como soporte a los usuarios en el caso de que tengan dudas sobre el funcionamiento de la aplicación web. Se puede apreciar esta funcionalidad en la Figura C.5.



Figura C.5: *Mockup - Página principal (contacto)*

- Existe una opción para realizar el cambio de idioma (Figura C.6). De forma predeterminada la aplicación se muestra en el idioma preferido por el navegador del usuario (normalmente español), pero existen versiones en inglés y francés.



Figura C.6: *Mockup - Página principal (idioma)*

- Si se pulsa en «Iniciar sesión», se redirige al usuario a la ventana representada por la Figura C.7, en la que debe introducir sus credenciales para realizar el inicio de la sesión.

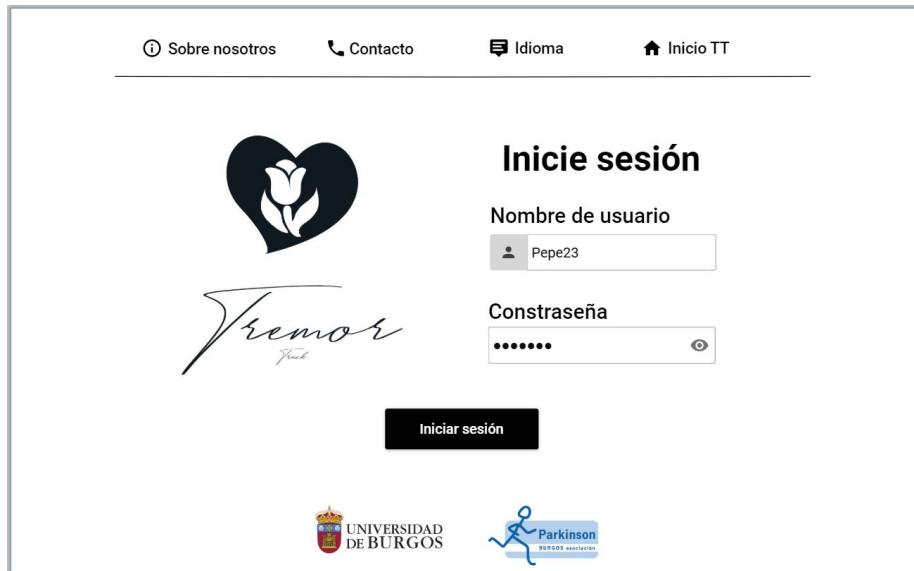


Figura C.7: *Mockup - Inicio sesión*

Una vez se ha iniciado sesión, dependiendo del rol del usuario aparecerán diferentes funcionalidades.

Si el usuario es un paciente aparecerá una ventana de bienvenida basada en la Figura C.8.

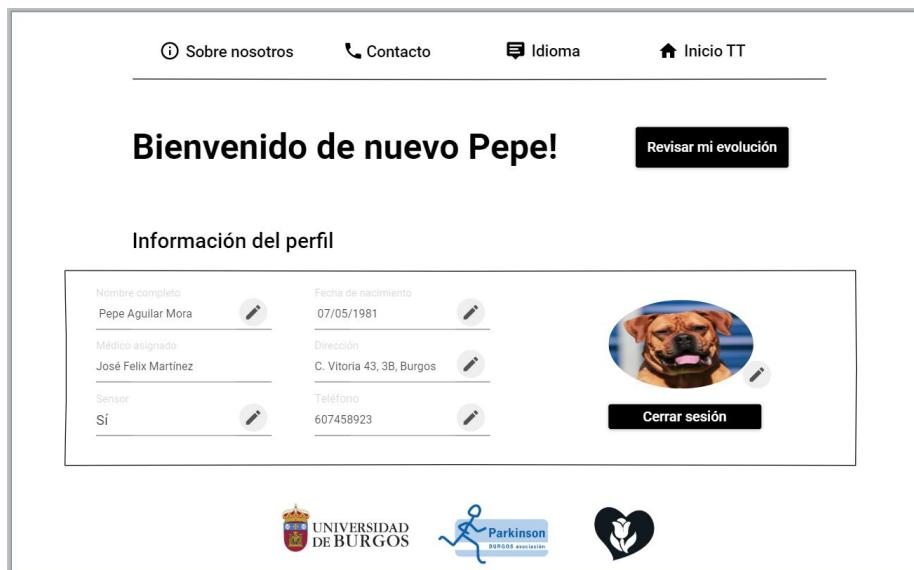


Figura C.8: *Mockup - Bienvenida paciente*

- Puede editar sus datos personales y su fotografía si así lo desea, mediante el lápiz que hay junto a cada campo (el médico asignado no se lo puede cambiar el propio paciente).
- Puede cerrar su sesión, lo que le redirigirá a la página principal.
- Si pulsa en «Revisar mi evolución» se le redirigirá a la ventana que se ilustra en la Figura C.9.



Figura C.9: *Mockup - Evolución paciente*

En ella deben seleccionar qué datos de los medidos por el sensor desean visualizar entre los disponibles. También deben indicar la fecha en la que fueron tomados dichos datos y dar al botón de «Mostrar datos». Esto hará que aparezca una gráfica mostrando la evolución de los datos pedidos durante las fechas seleccionadas.

Aparece arriba a la derecha el nombre y foto del usuario, indicando de quién es la sesión abierta y con posibilidad de cerrarla.

Si se indican fechas futuras, de las que no existen registros, el sistema utilizará inteligencia artificial para predecir una posible evolución (mostrada en la Figura C.10), generando un gráfico con las partes predichas en otro color.

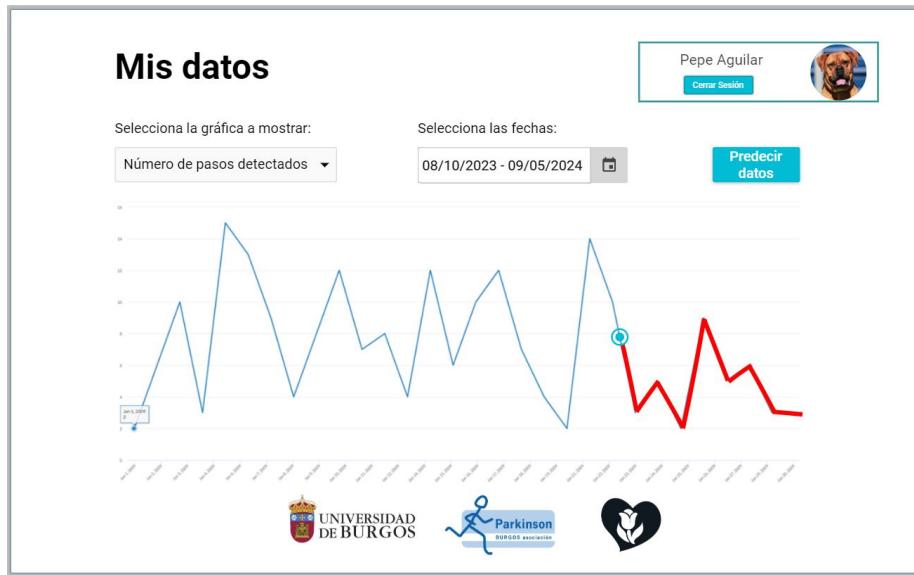


Figura C.10: *Mockup - Predicción paciente*

Si el usuario es un médico aparecerá la ventana de bienvenida que se muestra en la Figura C.11:

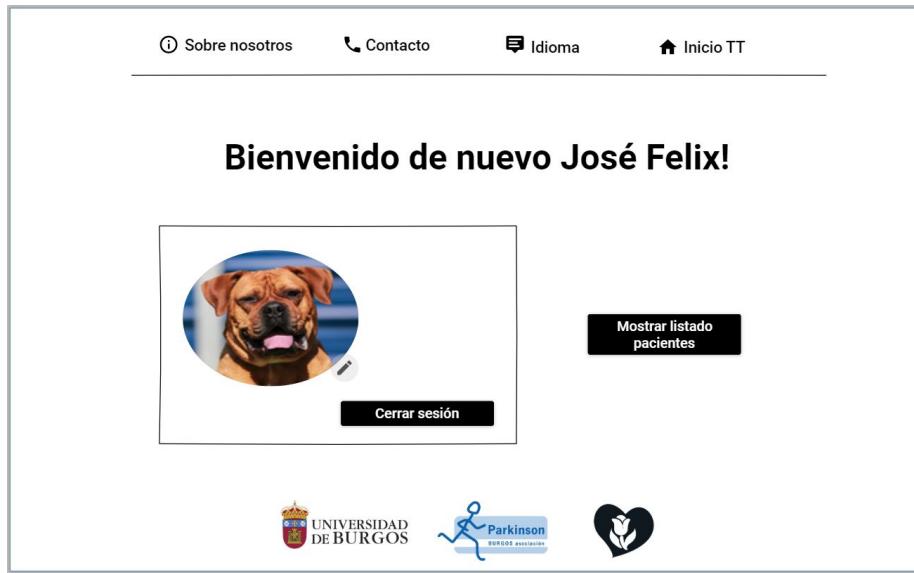


Figura C.11: *Mockup - Bienvenida médico*

- Puede editar su fotografía si así lo desea, mediante el lápiz que hay junto a ella.

- Puede cerrar su sesión, lo que le redirigirá a la página principal.
- Si pulsa en «Mostrar listado pacientes» se le redirigirá a la ventana de la Figura C.12.



Figura C.12: Mockup - Listado pacientes

En ella aparecerán todos los pacientes que tengan a dicho médico asociado. Junto al perfil del paciente aparecen 4 botones con acciones que puede realizar el médico.

- Si pulsa en «Información personal» se desplegarán los datos personales del paciente, como se muestra en la Figura C.13. Estos datos podrán ser editados.



Figura C.13: *Mockup - Información personal paciente*

- Si pulsa en «Subir vídeo», el médico podrá asociar más vídeos al paciente, mediante una interfaz similar a la de la Figura C.14. Si pulsa en el símbolo de añadir vídeo, será redirigido a sus archivos, donde debe escoger el vídeo/vídeos que desea subir. Además se debe indicar la fecha de grabación del vídeo y si corresponde con un vídeo de la mano derecha o de la mano izquierda.



Figura C.14: Mockup - Subir vídeo paciente

- Si pulsa en «Subir datos sensor» podrá subir, desde sus archivos, un archivo de datos de los creados por el sensor del paciente. Se desplegará la opción como en la Figura C.15. Se debe indicar la fecha inicial de dichos datos.



Figura C.15: Mockup - Subir datos paciente

- Si pulsa en «Mostrar datos sensor» se le redirigirá a una ventana parecida a la que tienen los pacientes para ver su evolución, de acuerdo con la Figura C.16.



Figura C.16: *Mockup* - Evolución médica

Los médicos también pueden hacer uso de la funcionalidad de predicción, según se observa en la Figura C.17.



Figura C.17: *Mockup* - Predicción médica

Si el usuario es un administrador, se mostrará la ventana de bienvenida que se ilustra en la Figura C.18.



Figura C.18: *Mockup - Bienvenida admin*

Los administradores pueden acceder a las mismas funcionalidades que los médicos y pacientes para comprobar el correcto funcionamiento de las mismas. Pueden cerrar su sesión al igual que el resto de los pacientes y disponen de un acceso a la gestión de los usuarios de la aplicación, tal como se indica en la Figura C.19.



Figura C.19: Mockup - Gestión usuarios

- Desde aquí, los administradores pueden modificar los datos de usuarios existentes. Este proceso se muestra en la Figura C.20.



Figura C.20: Mockup - Modificar usuario

- También pueden eliminarlos. Este proceso se ilustra en la Figura C.21.



Figura C.21: Mockup - Eliminar usuario

- Puede crear usuarios nuevos introduciendo sus datos. Este proceso se visualiza en la Figura C.22.

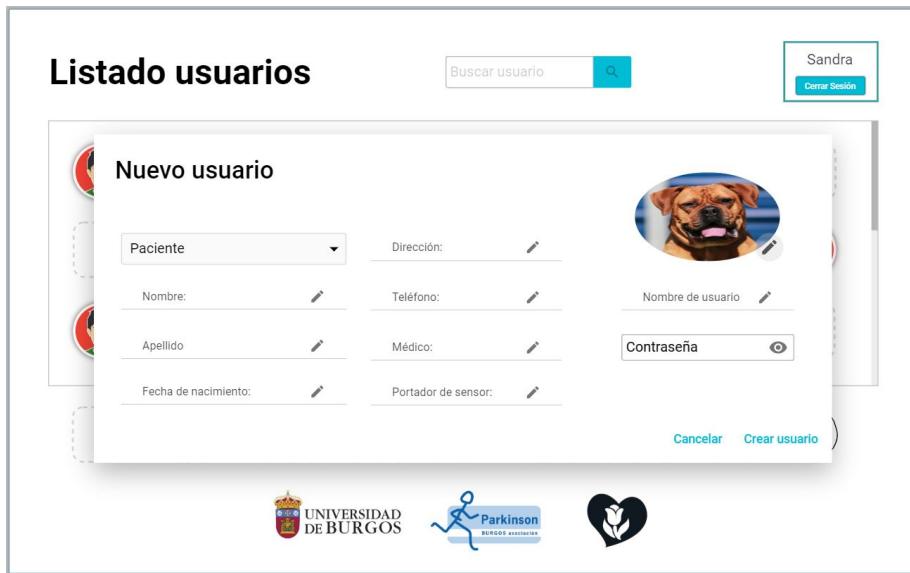


Figura C.22: Mockup - Crear usuario

## C.4. Diseño procedimental

El diseño procedimental durante un desarrollo *software* consiste en identificar los pasos necesarios para llevar a cabo una función específica del programa. Estos pasos representan pequeñas tareas que, ejecutadas de manera secuencial, logran el objetivo deseado.

Este método es útil porque facilita la organización del código, haciendo más sencillo su desarrollo, mantenimiento y comprensión, ayudando incluso en la identificación y solución de problemas.

Durante el proyecto se han utilizado diagramas secuenciales, creados con la herramienta draw.io<sup>1</sup>, para representar la interacción entre los diferentes componentes del sistema durante ciertas funcionalidades de la aplicación web, con el fin de facilitar su comprensión. Los componentes del sistema son:

- Usuario: representado por un «monigote», simboliza a los usuarios que utilizan la aplicación, que pueden ser pacientes, médicos o administradores.
- Navegador: se refiere al *Frontend*, a la interfaz de usuario de la aplicación. Se ha trabajado con tecnologías como HTML o JavaScript.
- Servidor web: se refiere al *Back-end*, al código de la aplicación desarrollado con Flask.
- Base de Datos: como su nombre indica representa a la base de datos usada durante el proyecto, trabajada mediante la biblioteca de Python SQLAlchemy.

Se muestran los diagramas de secuencia para las funciones que permiten iniciar sesión, mostrar los datos del sensor de un paciente mediante gráficas, subir vídeos de los pacientes al sistema y realizar predicciones.

### Iniciar Sesión

Al acceder un usuario a la aplicación web de TremorTrack lo primero que le aparece es la página principal. Para iniciar sesión existe una página en la que se deben introducir las credenciales. Dependiendo de las credenciales

---

<sup>1</sup><https://app.diagrams.net/>

introducidas, la aplicación reaccionará de diferente manera, representado en el diagrama de la Figura C.23 por una alternativa.

Si se introducen unas credenciales incorrectas la aplicación avisa al usuario con un mensaje de error, pero si se introducen los datos correctos, existen tres tipos de comportamiento: si el usuario inicia como administrador se le devuelve la página principal de los administradores, si es un médico la de los médicos y si es un paciente la de los pacientes.

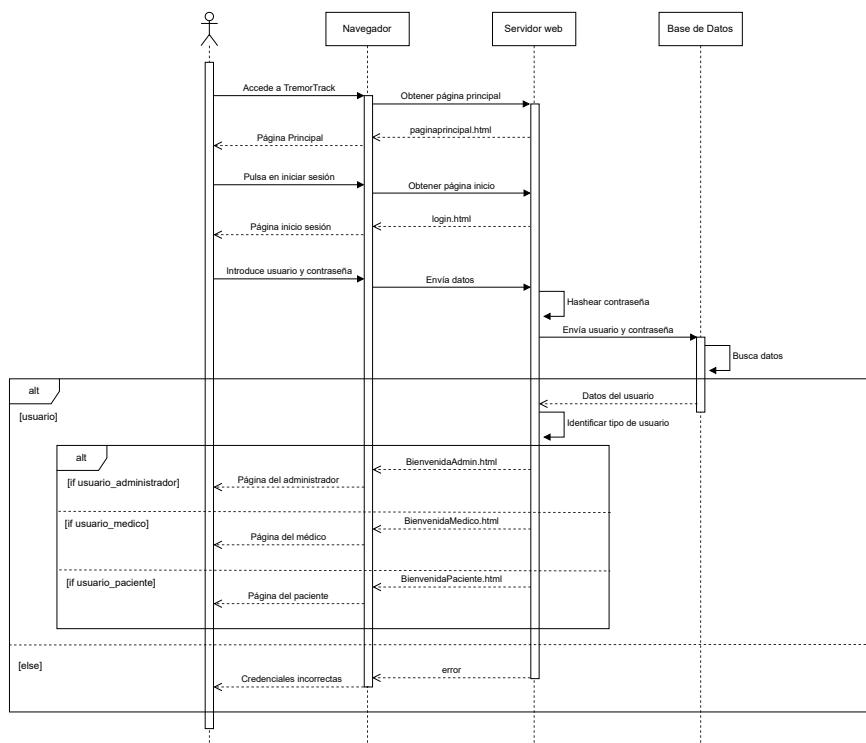


Figura C.23: Diagrama de secuencia - Iniciar Sesión

## Mostrar Datos Sensor

A esta funcionalidad pueden acceder tanto los pacientes (únicamente para sus propios datos), como médicos (para los datos de sus pacientes) o administradores (para los datos de todos los pacientes).

Cuando el usuario pulsa en el botón de «Mostrar Datos Sensor» se le redirecciona a una página con un formulario para elegir la gráfica a mostrar. Una vez elegidos el tipo de gráfica y las fechas, al darle al botón de «Mostrar Datos» las elecciones pasan al *back-end*.

Se accede a la base de datos para obtener todos los registros del paciente, que serán filtrados posteriormente por las fechas elegidas por el usuario. Se dispone de una función `crearGrafico()` que es la encargada de generar los datos para crear el gráfico en el *frontend*. En ella se utiliza la función `plot3Axis()`, creada inicialmente por los desarrolladores del sensor y adaptada por la alumna al código del proyecto, que a su vez llama a la función `returnByDatas()`. A la función se le pasan diferentes datos en función de lo elegido en el formulario y devuelve un JSON con los datos del gráfico preparados para ser utilizados.

Posteriormente, en el *frontend* se utiliza ese JSON para generar los gráficos con `chart.js`. Cada selección genera tres o cuatro gráficos, que se muestran como miniaturas y se pueden pulsar para verse en grande.

La Figura C.24 representa el diagrama de secuencia del proceso de mostrar una gráfica.

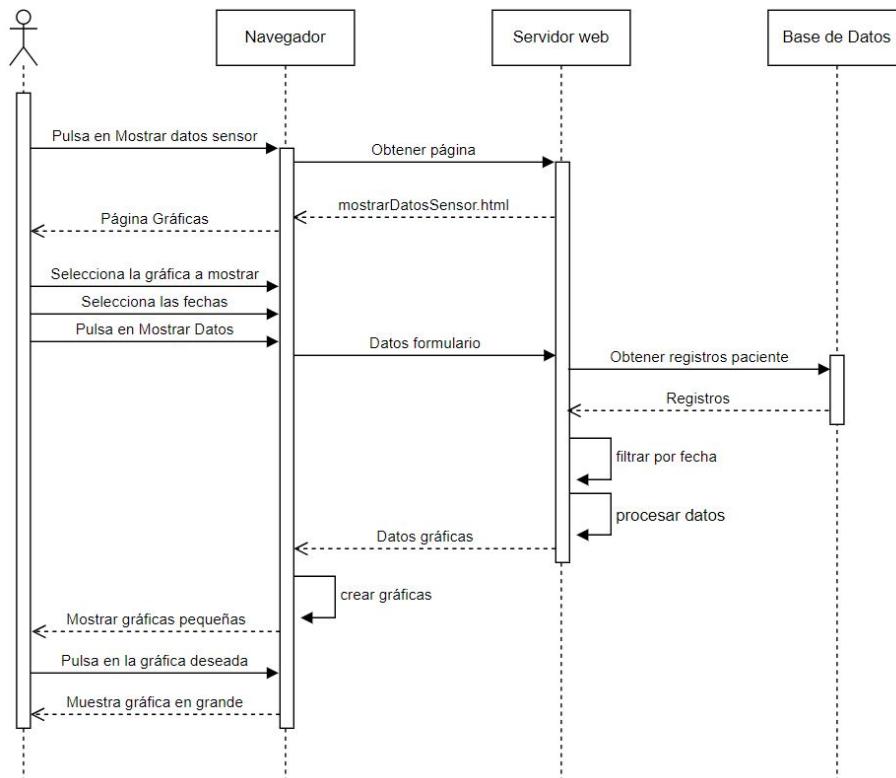


Figura C.24: Diagrama de secuencia - Mostrar Datos Sensor

## Subir Vídeo

Como se muestra en la Figura C.25 únicamente los administradores y los médicos pueden subir vídeos de pacientes al sistema. Los administradores acceden a esta función desde su página de gestión de usuarios, seleccionando la tabla con todos los pacientes del sistema y el paciente deseado. Los médicos acceden desde su página con el listado de pacientes a su cargo, seleccionando el paciente deseado.

Al pulsar en «Subir Video» se muestra un formulario en el que se debe introducir el vídeo, la fecha en la que fue grabado, la mano que aparece realizando el movimiento y la amplitud y lentitud del movimiento de pinza, normalmente calculados por un neurólogo. Al enviar el formulario lo primero que se realiza es una validación de los datos introducidos que, si son correctos, pasan al *back-end*.

Se comienza guardando el vídeo en la carpeta del paciente seleccionado, con un nombre que lo defina. Se guarda la ubicación y los datos del vídeo en la base de datos. Posteriormente se analiza el vídeo para obtener ciertas características extras del movimiento.

Si todo este proceso funciona correctamente se avisa al usuario de la subida exitosa del vídeo.

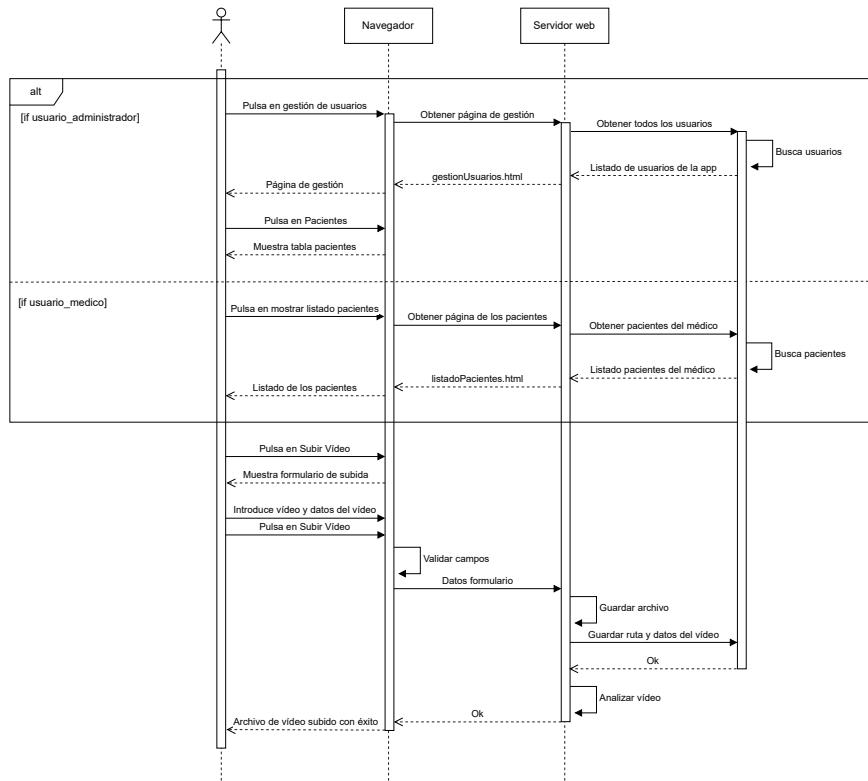


Figura C.25: Diagrama de secuencia - Subir Vídeo

## Predicción

La funcionalidad de predicción se encuentra en la página que muestra los vídeos del paciente y sus características, permitiendo predecir la evolución de esas características en el futuro. A esta ventana pueden acceder tanto los pacientes (únicamente para sus propios vídeos), como médicos (para los vídeos de todos los pacientes a su cargo) o administradores (para los vídeos de todos los pacientes).

En la ventana se muestra un listado con los vídeos de los pacientes, que permite reproducirlos o eliminarlos. Posteriormente se muestran dos gráficas, una para cada mano, con la evolución de las características de los vídeos de cada mano. Junto a las gráficas existe un botón «Predecir».

Al pulsar en el botón, como se muestra en la Figura C.26 se envían las características actuales de los vídeos a la función `predecirVideo()`. En esta función se utiliza el suavizado exponencial de Holt para calcular una predicción de 4 días con la evolución del paciente. Esta predicción, junto

con los datos de antes, se envía al .html, dónde se generan dos gráficas con las predicciones, una para cada mano.

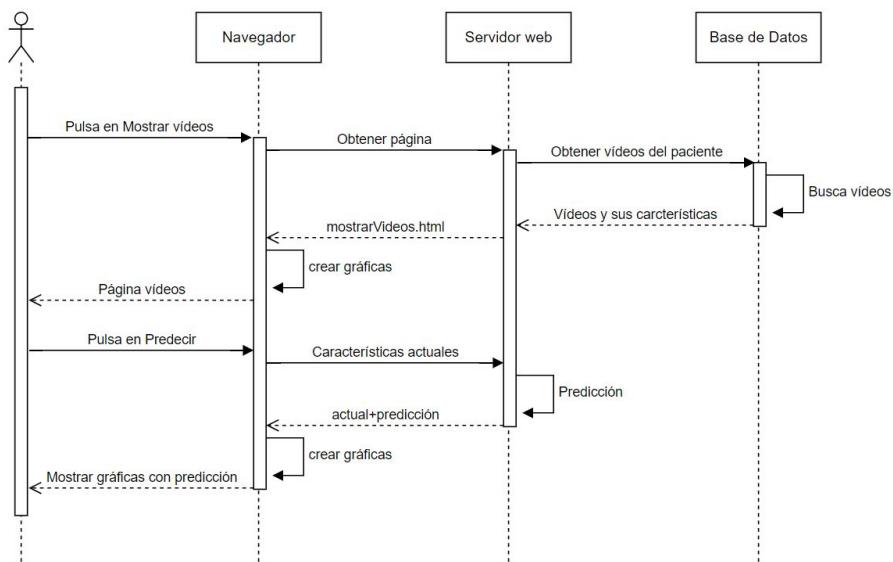


Figura C.26: Diagrama de secuencia - Predicción

## C.5. Diseño arquitectónico

Para explicar la arquitectura de la aplicación se utiliza el diagrama de despliegue de la Figura C.27. Un diagrama de despliegue es un tipo de diagrama UML que muestra la disposición física de los componentes software del sistema.

No se ha desplegado la aplicación por tener ficheros muy pesados, tanto los vídeos como los datos de los sensores de los pacientes, por lo que el apartado de *Web Server* no es verídico ya que la lógica de la aplicación se encuentra en local.

El *Data Base Server* es el que gestiona la capa de persistencia del proyecto. Se ha utilizado MariaDB y Heidi SQL como herramienta de gestión de la base de datos.

El usuario o cliente es el controlador de la vista de usuario, interactuando con las páginas HTML desde su navegador.

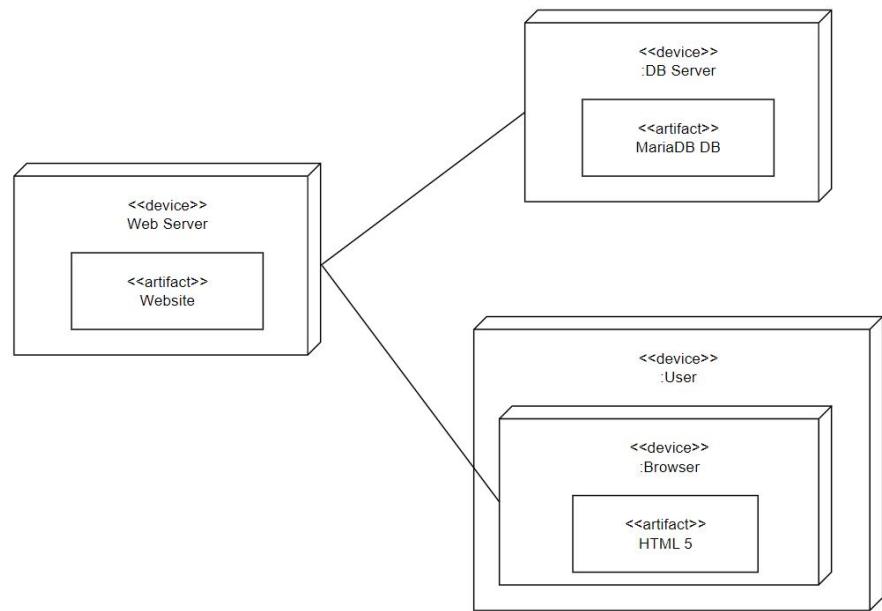


Figura C.27: Diagrama de despliegue

## *Apéndice D*

---

# **Documentación técnica de programación**

---

## **D.1. Introducción**

Esta sección contiene la documentación técnica de programación, es decir, el conjunto de materiales que describen el funcionamiento interno del *software* del proyecto. Puede ser de utilidad para otros programadores que pretendan usar, mantener o modificar el *software*.

Se incluyen aspectos como la arquitectura del sistema, la estructura del código fuente, los requisitos del sistema, las pruebas o los pasos necesarios para compilar, instalar y ejecutar el proyecto.

## **D.2. Estructura de directorios**

El repositorio de GitHub del proyecto<sup>1</sup> se divide en:

- doc: contiene la documentación del proyecto, formada por la memoria y anexos realizados en L<sup>A</sup>T<sub>E</sub>X. Además contiene el resultado final del *mockup* de la aplicación.
- src: contiene las imágenes necesarias para el archivo README: los logos de la aplicación realizada, de la UBU y de la Asociación Parkinson Burgos.

---

<sup>1</sup><https://github.com/SandraDiaz22/TFG>

- Pruebas: contiene las pruebas realizadas al comienzo del proyecto para familiarizarse con el lenguaje Python, la herramienta Jupyter Notebook y las librerías Matplotlib y Scikit-learn.
- App:
  - `app.py`: archivo principal de la aplicación, que contiene el código necesario para su ejecución. Define las rutas y las acciones que la aplicación realiza en respuesta a las solicitudes del usuario, así como las importaciones de los módulos necesarios.
  - Otros archivos .py: se trata de pequeños *scripts* que realizan tareas específicas: `fechasRegistros.py` se utiliza para actualizar y obtener las fechas inicial y final de los registros, `personasAleatorias.py` rellena la base de datos con imágenes creadas por inteligencia artificial, `analizarVideos.py` extrae características de los vídeos de los pacientes, `hashContrasenias.py` encripta las contraseñas de los usuarios en la base de datos y `modelosbdd.py` contiene la estructura de la base de datos.
  - *templates*: contiene todos los .html que corresponden a cada una de las ventanas de la aplicación. Existe un documento llamado `layout.html` que es la base del resto, definiendo los estilos o la barra de navegación. El resto de documentos heredan de él.
  - `messages.pot` y `babel.cfg`: ficheros generados automáticamente necesarios para las traducciones. En concreto, `messages.pot` contiene todos los textos de la aplicación y `babel.cfg` indica sobre qué archivos se realizan las traducciones.
  - *translations*: formado por tres carpetas utilizadas para la traducción de la aplicación a los diferentes idiomas disponibles (español, inglés y francés). En cada una de ellas encontramos `messages.po`, donde se fueron traduciendo a mano los textos de `messages.pot` y `messages.mo`, obtenido tras compilar.
  - *paddlel*: contiene las carpetas con el código necesario para el análisis de los vídeos de los pacientes, obtenido del TFG de un alumno del curso pasado [2].
  - Documento `.gitignore`: necesario para que GitHub ignore toda la carpeta *Static*, que contiene las fotos de los usuarios o logos, los vídeos y los registros de los pacientes. De lo contrario, se superaría el espacio de almacenamiento del repositorio. Además evita subir al repositorio la carpeta del entorno virtual con todas las librerías.

- Documento `sonar-project.properties`: contiene la configuración necesaria para realizar el escaneo de la calidad del código mediante la herramienta SonarQube.
- `pycache`: directorio creado automáticamente en Python que almacena los archivos de caché compilados de los módulos de Python que se han importado en el programa, como el programa que rellena las fechas de los registros o el archivo con la estructura de la base de datos.
- `LICENSE`: contiene información sobre la licencia escogida para la distribución del proyecto, licencia Apache versión 2.0.
- `README`: documento que presenta la información del repositorio y ofrece una vía de contacto con los desarrolladores.
- `baseDeDatosParkinson.sql`: contiene el código a ejecutar para crear una base de datos como la utilizada en el proyecto, pero sin datos en su interior.
- `requirements.txt`: indica las librerías utilizadas en el proyecto y sus versiones necesarias para poder ejecutar la aplicación sin problemas.

Además, se ha utilizado una base de datos llamada «parkinson». Durante el desarrollo se ha utilizado la herramienta HeidiSQL para administrar la base de datos, utilizando MariaDB como sistema gestor de base de datos. La base de datos está formada por cinco tablas:

- `administrador`: contiene los datos de los administradores de la aplicación, entre los que se encuentran: nombre, apellido, nombre de usuario y contraseña para la aplicación, correo electrónico y foto para el perfil.
- `medico`: contiene los datos (mismos campos que para los administradores) de una veintena de médicos de ejemplo, que trabajarían en el proyecto haciendo el seguimiento de los pacientes.
- `paciente`: además de los campos de médicos y administradores, para los pacientes también se almacena su número de teléfono, dirección, fecha de nacimiento, si han participado llevando un sensor en el proyecto, si es diestro o zurdo y el médico que les corresponde.
- `registros`: almacena la ubicación dentro de la carpeta `static` de los registros medidos por el sensor. Además se guardan el paciente al que corresponde y la fecha inicial y final de dicho registro.

- videos: almacena los vídeos grabados por ciertos pacientes realizando el movimiento de pinza. Se guarda la fecha en la que fue grabado, de qué mano (derecha o izquierda) y de qué paciente se trata. Además se almacenan la lentitud y amplitud, y las características obtenidas tras analizar el vídeo: velocidad media, frecuencia de máximos y mínimos, promedio de máximos...

### D.3. Manual del programador

Se describen a continuación las herramientas que fueron necesarias para el desarrollo de la aplicación:

- **Sistema Operativo:** la desarrolladora ha utilizado Windows ya que es su entorno de trabajo habitual, pero se podría desarrollar en Linux si se prefiere.
- **Python:** se ha utilizado la versión 3.10.2. Se recomienda como mínimo utilizar dicha versión para garantizar la compatibilidad. La última versión de Python se puede descargar desde su página oficial<sup>2</sup>.
- **GitHub Desktop:** aplicación para el control de versiones. Se utiliza para realizar comandos de Git como *commits* o *push* al repositorio de GitHub mediante una interfaz gráfica de usuario en lugar de mediante línea de comandos. La descarga se puede realizar desde la página de GitHub<sup>3</sup>. Se puede realizar por comandos si así se prefiere.
- **Visual Studio Code:** utilizado como editor del código fuente. Es compatible con cualquier sistema operativo<sup>4</sup>. Se podría utilizar otro editor de código si así se prefiere.

Para programar y ejecutar el proyecto, la desarrolladora utilizaba Visual Studio Code, pero se puede utilizar el entorno de desarrollo que se prefiera.

El primer paso fue crear un entorno virtual, en el que se instalaron todas las librerías necesarias para el proyecto. Tener las mismas librerías con las mismas versiones es completamente necesario para evitar fallos debidos a versiones.

---

<sup>2</sup><https://www.python.org/downloads/>

<sup>3</sup><https://desktop.github.com/>

<sup>4</sup><https://code.visualstudio.com/download>

Para simplificar este paso a futuros programadores se ha creado un documento `requirements.txt` en el que se indica la versión de cada una de las librerías utilizadas. Si se instalasen más librerías posteriormente se deberían añadir al documento. Para conocer las versiones de las librerías instaladas se puede consultar mediante `pip list`. Para añadir las nuevas librerías en el documento se deben introducir de la forma «librería==versión».

Para crear el entorno virtual se instaló la herramienta `virtualenv` mediante el siguiente comando: `pip install virtualenv`. Posteriormente se creó el entorno mediante: `virtualenv env`. Una vez que tenemos el entorno, se debe activar para trabajar sobre él. Se activa mediante el fichero `.\activate` que se encuentra en la carpeta `Scripts` dentro de `env`. Una vez activado debería salir «(env)» al principio de la línea del terminal.

Para instalar las librerías indicadas en el documento de requerimientos, es tan simple como ejecutar el siguiente comando: `pip install -r .\requirements.txt`.

Con todas las versiones necesarias ya se puede ejecutar el proyecto, volviendo a la carpeta de la aplicación y ejecutando `python app.py`. Al ejecutar dicho comando se podrá acceder a la aplicación de forma local en la ubicación «`http://127.0.0.1:5000`».

Si un nuevo programador realiza cambios en el proyecto y se crean nuevos textos que vayan a ser mostrados en la aplicación se deben actualizar los archivos para las traducciones. El texto que necesite internacionalizarse se debe escribir entre «`trans`» y «`endtrans`» de la siguiente manera:

```
{% trans %}Texto de ejemplo{% endtrans %}
```

Posteriormente se debe ejecutar el siguiente comando: `pybabel extract -F babel.cfg -o messages.pot ..`, que se encarga de llenar el fichero `messages.pot` con el texto de la aplicación que aparezca de la manera anteriormente comentada. Se indica el documento y la línea en la que aparece el texto, ofreciendo una plantilla que hay que usar en los ficheros `messages.po` de cada idioma, traduciendo el texto que aparece como «`msgid`» en el campo «`msgstr`». Una vez realizadas las traducciones se deben compilar, mediante el comando `pybabel compile -d translations`, obteniendo los ficheros `messages.mo`.

## D.4. Compilación, instalación y ejecución del proyecto

En la presente sección se comentan los pasos seguidos para realizar el despliegue del proyecto en el servidor.

El primer paso es conectarse en remoto al servidor que se encuentra en la Universidad de Burgos. Como la alumna se encontraba en su casa, hizo uso de una VPN para emular estar conectada a la intranet de la UBU. Esta VPN se realiza mediante GlobalProtect a `vpn.ubu.es`. Es necesario que un tutor u otro miembro de la universidad te de acceso para poder utilizar este recurso. Para poder posteriormente acceder a uno de los servidores también es necesario que te den acceso con un usuario y contraseña.

Una vez conectada la VPN, es posible conectarse al servidor mediante SSH (*Secure Shell*). Se trata de un protocolo de red destinado a la conexión por remoto a un servidor mediante línea de comandos. Se debe abrir el «CMD» o intérprete de comandos del sistema operativo utilizado (en el caso de la alumna Windows) y escribir el comando:

```
ssh usuario@ip_del_servidor
```

Se deberá introducir la contraseña del usuario para acceder.

Una vez organizado todo el contenido del proyecto en el repositorio de GitHub se pudo clonar en el servidor mediante el comando:

```
git clone <url_repositorio>
```

La carpeta «static», en la que se almacenan las fotos de los usuarios y logos, los registros de los sensores y los vídeos de los pacientes, no se pudo añadir en el repositorio porque se excedía la capacidad permitida. Por ello fue necesario agregar esta carpeta al servidor por separado.

Para transferir archivos entre el ordenador de la alumna y el servidor se utiliza el comando SCP en el ordenador personal. Al tratarse de una carpeta con todo su contenido se realizó de la siguiente manera:

```
scp -r "C:\...\static" usuario@ip_del_servidor:~/TFG/App/
```

En el repositorio hay un documento `baseDeDatosParkinson.sql` que contiene los «CREATE TABLE» necesarios para crear una copia de la

base de datos, pero no contiene los «INSERTS», creando tablas vacías. Para poder probar la aplicación en el servidor se creó un documento **baseDeDatosConInserts.sql** que incluye los datos que se han utilizado en el desarrollo. Este documento se pasó al servidor mediante SCP.

Para crear la base de datos en el servidor, lo primero que fue necesario fue instalar SQL. No se pueden descargar librerías en el servidor, no se tienen permisos de administrador, ya que podría afectar a otros proyectos que se encuentren alojados también en él, por lo que se ha instalado Miniconda en el usuario específico de la alumna del servidor. Miniconda permite crear un entorno aislado, para poder instalar otras versiones de Python o más librerías específicas sin afectar al servidor general o al resto de usuarios. Se ha descargado el instalador de la página oficial y se ha instalado en el servidor mediante el comando:

```
bash Miniconda3-latest-Linux-x86_64.sh
```

Al ejecutar el comando `which python` se aprecia como se está utilizando la versión de Python de Miniconda en vez de la del servidor.

Los pasos seguidos para crear un entorno en conda llamado «proyecto», instalar MySQL, inicializar la base de datos y el servidor MySQL, cambiar la contraseña por defecto por la misma que se usaba en local, crear la base de datos llamada «parkinson» e importar el contenido de la base de datos del proyecto se detallan a continuación:

```
conda create --name EntornoProyecto python=3.10.2
conda activate EntornoProyecto
conda install -c conda-forge mysql
mkdir -p /home/sda1003/mysql_data
mysqld --initialize --datadir=/home/sda1003/mysql_data
mysqld --datadir=/home/sda1003/mysql_data &
mysql -u root -p
ALTER USER 'root'@'localhost' IDENTIFIED BY 'maria';
exit;
mysql -u root -p -e "CREATE DATABASE parkinson;" 
cd TFG
mysql -u root -p parkinson < baseDeDatosConInserts.sql
```

En local se tuvo que crear un entorno virtual en el que instalar las versiones de las librerías correctas, pero en el servidor, al haber creado un

entorno conda, no ha sido necesario crear otro. Se han instalado las librerías del documento `requirementsBase.txt` (nuevo documento con una versión más baja de la librería tsfresh debido a problemas con la versión de la gráfica del servidor) y se ha debido instalar también otra librería para las contraseñas con `conda install cryptography`.

Por último, se han realizado un par de cambios en `app.py` para poder ser ejecutado en el servidor:

- Cambiar la configuración de las *cookies* de idioma, relajando las restricciones de seguridad, cambiando a `False` las siguientes configuraciones:

```
respuesta.set_cookie('idioma', value=idioma,
max_age=86400, secure=False, httponly=False)
```

- Cambiar la configuración de la ejecución de Flask para escuchar en el host y puerto correctos, con el fin de poder acceder desde cualquier ordenador conectado a la red y no solo localmente:

```
app.run(host='0.0.0.0', port=5000, debug=False)
```

Con las versiones y cambios necesarios, teniendo iniciada la base de datos, ya se puede ejecutar el proyecto, ejecutando: `python app.py`. Se puede acceder a la aplicación desde el navegador de un ordenador conectado a la intranet de la UBU mediante la dirección `http://10.168.168.34:5000/`.

El despliegue realizado es el que no recomienda Flask, porque se debe usar únicamente para `debug` y pruebas. No se debería desplegar la aplicación Flask con su propio servidor. Por ello se ha utilizado la herramienta Gunicorn para realizar otro despliegue, en el puerto 8000. Se detallan a continuación las acciones que fueron necesarias. Instalar Gunicorn mediante `pip install gunicorn`. Crear un archivo `wsgi.py` que ejecute la aplicación:

```
from app import app

if __name__ == "__main__":
    app.run()
```

Ejecutar Gunicorn en el puerto escogido mediante :

```
gunicorn -w 4 -b 0.0.0.0:8000 wsgi:app
```

Ahora se puede acceder a la aplicación, desde la intranet de la UBU, mediante la dirección <http://10.168.168.34:8000/>.

Aparece el mensaje «No es seguro» junto a la URL, lo que significa que la conexión a la aplicación no es segura. Esto se debe a que la conexión entre el navegador y el servidor no está protegida mediante un certificado SSL válido.

Se trató de utilizar el servidor web/Proxy NGINX junto con un certificado SSL para configurar una conexión segura HTTPS, pero no fue posible instalar NGINX por no tener permisos de administrador en el equipo. Se detalla este problema en el apartado de «Líneas de trabajo futuras» de la memoria.

## D.5. Pruebas del sistema

En este apartado se han realizado pruebas del sistema o pruebas *software* para comprobar el correcto funcionamiento de la aplicación. Se llevan a cabo mediante casos de prueba, procedimientos utilizados durante el desarrollo *software* para verificar que la aplicación cumpla con los requisitos especificados y detectar errores antes del despliegue, reduciendo costes asociados con correcciones posteriores.

Las pruebas básicas no se incluyen en el apartado, ya que se han estado utilizando continuamente durante el desarrollo. Algunas de ellas son:

- Inicio de sesión: si se introducen credenciales de un administrador se accede a las funcionalidades para administradores, si se introducen credenciales de un médico se accede a la página principal de los médicos y si se introducen credenciales de un paciente se accede a su página de bienvenida. Si por el contrario, se introduce el nombre de usuario incorrecto, o la contraseña incorrecta, o ambas incorrectas, aparecerá un mensaje por pantalla diciendo «Credenciales incorrectas. Inténtalo de nuevo.», que no especifica cuál de los dos es incorrecto para no ayudar a posibles intrusos.
- Cierre de sesión: si se pulsa sobre el botón de cierre de sesión estando con una sesión de paciente, médico o administrador activa, la aplicación finalizará la sesión y redirigirá al usuario a la página principal de la aplicación. Se ha realizado un control de acceso, de forma qué antes de cargar las páginas la aplicación comprueba que el usuario está

logueado y tiene acceso a dicha página, evitando que los usuarios puedan acceder a páginas que no deberían poder acceder escribiendo la url directamente (ver caso de prueba 1).

- Añadir usuario a la base de datos: los administradores pueden añadir nuevos administradores, médicos o pacientes a la base de datos mediante un formulario. Como es de esperar, el programa crea correctamente el usuario en la base de datos, con todas las características indicadas por el administrador (previamente comprobadas por el programa). Se ha realizado un caso de prueba en el cual el administrador introduce un usuario con el mismo correo electrónico que otro que ya está en la base de datos (cuando el correo debería ser único) y el programa no deja añadir dicho usuario (ver caso de prueba 2).
- Eliminar usuario de la base de datos: los administradores pueden eliminar administradores, médicos o pacientes de la base de datos mediante el botón «Eliminar Usuario». Antes de eliminarlo el programa pide una confirmación, para evitar imprevistos. Como es de esperar, el programa elimina correctamente el usuario de la base de datos. Se han creado tres casos de prueba: uno en el que el administrador intenta borrarse a sí mismo (ver caso de prueba 3), otro en el que se intenta borrar un paciente que tiene registros y vídeos asociados a él en las tablas de registros y vídeos de la base de datos (ver caso de prueba 4) y otro en el que se intenta borrar a un médico con pacientes a su cargo (ver caso de prueba 5).
- Edición de usuarios: los administradores pueden editar todos los datos de los tres tipos de usuarios, los médicos pueden editar ciertos datos de sus pacientes y un paciente puede editar ciertos datos personales propios. El programa comprueba que los nuevos datos introducidos sean correctos y realiza los cambios en la base de datos. Si le dan a modificar datos sin haber cambiado ninguno, el programa avisa de que no hay cambios para guardar (ver caso de prueba 6) y si intentan asignar un correo electrónico que ya tiene otro paciente salta un mensaje de error (ver caso de prueba 7).
- Realizar predicciones: el programa trata de predecir lo mejor posible el avance de los datos de los vídeos en el futuro. Los vídeos disponibles tienen una periodicidad fija. Se ha realizado un caso de prueba en el que se añade un vídeo cuya fecha no tenga periodicidad con el resto (ver caso de prueba 8) y otro en el que se elimina un vídeo de los disponibles (ver caso de prueba 9).

- Generación de gráficos: una vez elegidos el tipo de gráfica a mostrar y las fechas, el programa es capaz de generar y mostrar las gráficas solicitadas mostrando la información recogida por el sensor para las fechas seleccionadas. Si el usuario quisiera seleccionar unas fechas en las que el paciente no tiene registros, el calendario no le dejaría (ver caso de prueba 10).
- Subida de registros y vídeos a la base de datos: para subir un registro de un paciente a la base de datos simplemente se debe llenar el formulario que solicita un archivo de tipo CSV, y el programa sube con éxito el archivo, calculando las fechas a las que corresponde el registro automáticamente. Para subir un vídeo se deben aportar más datos en el formulario y, posteriormente, el programa analizará el vídeo y lo subirá junto con las características extraídas del mismo a la base de datos. Si el usuario sube dos vídeos con el mismo nombre no ocurren problemas en la base de datos (no se sobrescribe el anterior) porque la aplicación genera un nombre único para cada archivo antes de subirlo (ver caso de prueba 11). Además, el programa no permite al usuario indicar que la fecha del vídeo es posterior a la actual (ver caso de prueba 12).
- Cambio de idioma: cuando un paciente elige el idioma de la aplicación mediante el desplegable de la barra de navegación, se cambian todos los textos que aparecen en las pantallas y pasan a estar en el idioma elegido.

Únicamente se muestran los casos de prueba más relevantes, que podrían llegar a dar fallos en la aplicación, y que se han probado en diferentes navegadores:

---

Identificador: CP1

Prioridad: Alta

Fecha de ejecución: 05/06/2024

Realización: Manual

Objetivo: Probar el acceso a las ventanas

Precondiciones: No estar logueado como administrador

Postcondiciones: No acceder y ser redirigido a la página principal

<b>Acción</b>	<b>Entrada</b>	<b>Salida Esperada</b>	<b>Salida Real</b>	
Hacer clic sobre el botón «Cerrar sesión»		Estar en la página principal sin sesión activa	Estar en la página principal sin sesión activa	✓
Poner manualmente la URL para acceder a la gestión de usuarios		Estar en la página principal	Estar en la página principal	✓

Tabla D.1: CP1: Control de acceso

---

Identificador: CP2

Prioridad: Media

Fecha de ejecución: 05/06/2024

Realización: Manual

Objetivo: Probar a añadir usuario con mismo correo electrónico

Precondiciones: Estar logueado como administrador

Postcondiciones: Aviso al administrador del error al agregar el usuario

Acción	Entrada	Salida Esperada	Salida Real	
Acceder al panel de gestión de usuarios		Estar en el panel de gestión de usuarios	Estar en el panel de gestión de usuarios	✓
Hacer clic sobre el botón «Médicos»		Aparece la lista de todos los médicos de la aplicación	Aparece la lista de todos los médicos de la aplicación	✓
Hacer clic sobre el botón «Añadir Médico»		Aparece el formulario de creación de usuario	Aparece el formulario de creación de usuario	✓
Introducir información sobre el nuevo médico	Nombre; Apellido; Nombre de Usuario; Contraseña; Correo Electrónico: igual que el de otro médico; Foto	Formulario rellenado	Formulario rellenado	✓
Hacer clic sobre el botón «Agregar usuario»		Mensaje de error al agregar el usuario	Mensaje de error al agregar el usuario	✓

Tabla D.2: CP2: Añadir usuario

---

Identificador: CP3

Prioridad: Alta

Fecha de ejecución: 05/06/2024

Realización: Manual

Objetivo: Probar a eliminarse a sí mismo de la base de datos

Precondiciones: Estar logueado como administrador

Postcondiciones: Aviso de que no se puede eliminar a sí mismo

Acción	Entrada	Salida Esperada	Salida Real
Acceder al panel de gestión de usuarios		Estar en el panel de gestión de usuarios	Estar en el panel de gestión de usuarios ✓
Hacer clic sobre el botón «admins»		Aparece la lista de todos los administradores de la aplicación	Aparece la lista de todos los administradores de la aplicación ✓
Hacer clic sobre el botón «Eliminar Usuario» de tu propio usuario		Aparece el mensaje de error de no poderse eliminar a sí mismo	Aparece el mensaje de error de no poderse eliminar a sí mismo ✓

---

Tabla D.3: CP3: Eliminarse a sí mismo

---

Identificador: CP4

Prioridad: Media

Fecha de ejecución: 06/06/2024

Realización: Manual

Objetivo: Probar a borrar un paciente con vídeos y registros

Precondiciones: Estar logueado como administrador

Postcondiciones: Paciente y sus vídeos eliminados

Acción	Entrada	Salida Esperada	Salida Real
Acceder al panel de gestión de usuarios		Estar en el panel de gestión de usuarios	Estar en el panel de gestión de usuarios ✓
Hacer clic sobre el botón «Pacientes»		Aparece la lista de todos los pacientes de la aplicación	Aparece la lista de todos los pacientes de la aplicación ✓
Hacer clic sobre el botón «Eliminar Usuario» de un paciente con vídeos		Aparece el mensaje de confirmación	Aparece el mensaje de confirmación ✓
Dar a aceptar confirmación		Paciente eliminado de la base de datos, junto con todos sus vídeos y registros	Paciente eliminado de la base de datos, junto con todos sus vídeos y registros ✓

Tabla D.4: CP4: Eliminar paciente con vídeos

---

Identificador: CP5

Prioridad: Alta

Fecha de ejecución: 06/06/2024

Realización: Manual

Objetivo: Probar a borrar un médico con pacientes a su cargo

Precondiciones: Estar logueado como administrador

Postcondiciones: Aviso al administrador del error al eliminar el médico

Acción	Entrada	Salida Esperada	Salida Real
Acceder al panel de gestión de usuarios		Estar en el panel de gestión de usuarios	Estar en el panel de gestión de usuarios ✓
Hacer clic sobre el botón «Médicos»		Aparece la lista de todos los médicos de la aplicación	Aparece la lista de todos los médicos de la aplicación ✓
Hacer clic sobre el botón «Eliminar Usuario» de un médico con pacientes a su cargo		Aparece el mensaje de confirmación	Aparece el mensaje de confirmación ✓
Dar a aceptar confirmación		No se puede eliminar el médico porque tiene pacientes a su cargo.	No se puede eliminar el médico porque tiene pacientes a su cargo. ✓

Tabla D.5: CP5: Eliminar médico con pacientes

---

Identificador: CP6

Prioridad: Baja

Fecha de ejecución: 05/06/2024

Realización: Manual

Objetivo: Probar a darle a guardar cambios sin haber realizado cambios

Precondiciones: Estar logueado como administrador

Postcondiciones: No se envía el formulario y aviso al usuario

Acción	Entrada	Salida Esperada	Salida Real
Acceder al panel de gestión de usuarios		Estar en el panel de gestión de usuarios	Estar en el panel de gestión de usuarios ✓
Hacer clic sobre el botón «Médicos»		Aparece la lista de todos los médicos de la aplicación	Aparece la lista de todos los médicos de la aplicación ✓
Hacer clic sobre el botón «Editar Usuario» de un médico		Aparece el formulario con los datos del médico, que permite editarlos	Aparece el formulario con los datos del médico, que permite editarlos ✓
Dar a «Guardar cambios» sin haber realizado cambios		Mensaje de que no hay cambios para guardar.	Mensaje de que no hay cambios para guardar. ✓

Tabla D.6: CP6: Modificar cero cambios

---

Identificador: CP7

Prioridad: Media

Fecha de ejecución: 06/06/2024

Realización: Manual

Objetivo: Poner el mismo correo que ya tiene otro paciente

Precondiciones: Estar logueado como administrador

Postcondiciones: Aviso al administrador del error al editar usuario

Acción	Entrada	Salida Esperada	Salida Real
Acceder al panel de gestión de usuarios		Estar en el panel de gestión de usuarios	Estar en el panel de gestión de usuarios ✓
Hacer clic sobre el botón «Médicos»		Aparece la lista de todos los médicos de la aplicación	Aparece la lista de todos los médicos de la aplicación ✓
Hacer clic sobre el botón «Editar Usuario» de un médico		Aparece el formulario con los datos del médico, que permite editarlos	Aparece el formulario con los datos del médico, que permite editarlos ✓
Escribir en el campo del correo electrónico existente	Correo electrónico del correo de otro médico y darle a «Guardar cambios»	Error al editar el usuario	Error al editar el usuario ✓

Tabla D.7: CP7: Editar correo electrónico

---

Identificador: CP8

Prioridad: Media

Fecha de ejecución: 06/06/2024

Realización: Manual

Objetivo: Predicción con un vídeo con diferente periodicidad

Precondiciones: Estar logueado como administrador

Postcondiciones: Predicción correcta

Acción	Entrada	Salida Esperada	Salida Real	
Acceder al panel de gestión de usuarios		Estar en el panel de gestión de usuarios	Estar en el panel de gestión de usuarios	✓
Hacer clic sobre el botón «Pacientes»		Aparece la lista de todos los pacientes de la aplicación	Aparece la lista de todos los pacientes de la aplicación	✓
Hacer clic sobre el botón «Subir vídeo» de un paciente		Aparece el formulario para subir vídeos	Aparece el formulario para subir vídeos	✓
Introducir el vídeo y su información	Archivo de vídeo: prueba.mp4; Fecha del vídeo: fecha con diferente periodicidad al resto; Mano; Amplitud; Lentitud	Formulario relleno	Formulario relleno	✓
Hacer clic sobre el botón «Subir vídeo»	Datos formulario	Vídeo subido a la base de datos	Vídeo subido a la base de datos	✓
Hacer clic sobre el botón «Mostrar vídeos» de ese paciente		Redirección a ventana con la lista de vídeos y gráficas con sus características	Redirección a ventana con la lista de vídeos y gráficas con sus características	✓
Hacer clic sobre el botón «Predecir»		Aparecen los gráficos con la predicción	Aparecen los gráficos con la predicción porque el modelo se recalibra con el nuevo vídeo	✓

Tabla D.8: CP8: Predicción con diferente periodicidad

---

Identificador: CP9

Prioridad: Media

Fecha de ejecución: 06/06/2024

Realización: Manual

Objetivo: Predicción al eliminar un vídeo

Precondiciones: Estar en la página de mostrar vídeos

Postcondiciones: Predicción correcta

Acción	Entrada	Salida Esperada	Salida Real
Hacer clic sobre el botón «Eliminar vídeo» de un vídeo de un paciente		Aparece el mensaje de confirmación	Aparece el mensaje de confirmación ✓
Dar a aceptar confirmación		Vídeo eliminado de la base de datos	Vídeo eliminado de la base de datos ✓
Hacer clic sobre el botón «Predecir»		Aparecen los gráficos con la predicción	Aparecen los gráficos con la predicción porque el modelo se recalibra con el cambio de datos ✓

Tabla D.9: CP9: Predicción al eliminar vídeo

---

Identificador: CP10

Prioridad: Alta

Fecha de ejecución: 06/06/2024

Realización: Manual

Objetivo: Seleccionar fechas en las que no hay registros

Precondiciones: Estar en la página de mostrar datos sensor

Postcondiciones: Fechas no seleccionadas

Acción	Entrada	Salida Esperada	Salida Real
Seleccionar gráfica a mostrar del desplegable	Elección	Tipo de gráfica seleccionado	Tipo de gráfica seleccionado ✓
Pulsar en seleccionar fechas a mostrar		Se muestra el calendario para elegir fechas	Se muestra el calendario para elegir fechas ✓
Tratar de seleccionar una fecha sin registros		No deja seleccionarla	No deja seleccionarla ✓

Tabla D.10: CP10: Seleccionar fechas sin registros

---

Identificador: CP11

Prioridad: Media

Fecha de ejecución: 06/06/2024

Realización: Manual

Objetivo: Subir dos vídeos con el mismo nombre a la base de datos

Precondiciones: Estar logueado como administrador

Postcondiciones: Ambos vídeos en la base de datos

Acción	Entrada	Salida Esperada	Salida Real	
Acceder al panel de gestión de usuarios		Estar en el panel de gestión de usuarios	Estar en el panel de gestión de usuarios	✓
Hacer clic sobre el botón «Pacientes»		Aparece la lista de todos los pacientes de la aplicación	Aparece la lista de todos los pacientes de la aplicación	✓
Hacer clic sobre el botón «Subir vídeo» de un paciente		Aparece el formulario para subir vídeos	Aparece el formulario para subir vídeos	✓
Introducir el vídeo y su información	Archivo de vídeo: prueba.mp4; Fecha del vídeo; Mano; Amplitud; Lentitud	Formulario relleno	Formulario relleno	✓
Hacer clic sobre el botón «Subir vídeo»	Datos formulario	Vídeo subido a la base de datos	Vídeo subido a la base de datos con un nombre que identifica su fecha, mano, paciente y el instante en el que fue subido a la base de datos	✓
Introducir el segundo vídeo y su información	Archivo de vídeo: prueba.mp4; Fecha del vídeo; Mano; Amplitud; Lentitud	Formulario relleno	Formulario relleno	✓
Hacer clic sobre el botón «Subir vídeo»	Datos formulario	Vídeo subido a la base de datos	Vídeo subido a la base de datos con un nombre que identifica su fecha, mano, paciente y el instante en el que fue subido a la base de datos	✓

---

Tabla D.11: CP11: Subir vídeos con mismo nombre

---

Identificador: CP12

Prioridad: Baja

Fecha de ejecución: 06/06/2024

Realización: Manual

Objetivo: Indicar que un vídeo fue grabado más tarde de la fecha actual

Precondiciones: Estar logueado como administrador

Postcondiciones: Aviso al administrador del error

Acción	Entrada	Salida Esperada	Salida Real
Acceder al panel de gestión de usuarios		Estar en el panel de gestión de usuarios	Estar en el panel de gestión de usuarios ✓
Hacer clic sobre el botón «Pacientes»		Aparece la lista de todos los pacientes de la aplicación	Aparece la lista de todos los pacientes de la aplicación ✓
Hacer clic sobre el botón «Subir vídeo» de un paciente		Aparece el formulario para subir vídeos	Aparece el formulario para subir vídeos ✓
Introducir el vídeo y su información	Archivo de vídeo; Fecha del vídeo: posterior a la fecha actual; Mano; Amplitud; Lentitud	Formulario rellenado	Formulario rellenado ✓
Hacer clic sobre el botón «Subir vídeo»	Datos formulario	Se avisa de que la fecha del vídeo no puede ser posterior a la fecha actual y no se sube el vídeo	Se avisa de que la fecha del vídeo no puede ser posterior a la fecha actual y no se sube el vídeo ✓

Tabla D.12: CP12: Fecha del vídeo incoherente

## *Apéndice E*

---

# **Documentación de usuario**

---

## **E.1. Introducción**

En este apartado se describen 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. Además se describen sus requisitos necesarios e instrucciones de instalación.

Tiene como objetivo principal servir como guía a los usuarios, para poder aprovechar al máximo todas las características de la aplicación.

## **E.2. Requisitos de usuarios**

Disponer de conexión a Internet y una versión reciente de cualquier navegador: Google Chrome, Microsoft Edge, Mozilla Firefox o Safari. Durante el desarrollo y las pruebas se ha hecho uso de los navegadores Google Chrome y Microsoft Edge, las versiones 125.0.6422.142 (Build oficial) (64 bits) y 125.0.2535.92 (Compilación oficial) (64 bits) respectivamente.

Dado que la aplicación está desplegada únicamente en un servidor de la Universidad de Burgos, es necesario estar conectado a la intranet de la UBU para acceder a ella. Esto se debe a que la aplicación es accesible a través de direcciones IP privadas, pero no está disponible mediante direcciones IP públicas. Por lo tanto, es imprescindible encontrarse físicamente en los recintos de la universidad y conectarse a su red o utilizar una conexión VPN que emule estar dentro de la intranet de la UBU.

En un futuro, si el prototipo de aplicación realizado resulta útil para los miembros del HUBU y deciden utilizarlo, se desplegará en un servidor externo y podrá ser accedido desde cualquier lugar a través de Internet.

### **E.3. Instalación**

Se entiende como instalación el proceso por el cual se transfiere un nuevo programa o aplicación a un ordenador para ser configurado y preparado para ser utilizado.

Durante el proyecto se ha desarrollado una aplicación web, alojada en un servidor, por lo que no es necesario descargar e instalar la aplicación en el equipo, sino que simplemente se ha de acceder a ella con el navegador, mediante la dirección <http://10.168.168.34:8000/>.

### **E.4. Manual del usuario**

Una vez que el usuario accede al enlace de la aplicación, tiene acceso a la página principal. Existen una serie de funcionalidades básicas accesibles para cualquier usuario, pero el resto son únicamente accesibles para ciertos tipos de usuarios una vez hayan iniciado sesión.

En este apartado se mostrarán las funcionalidades básicas y las funcionalidades de cada tipo de usuario: paciente, médico y administrador. Cabe destacar que existe un videotutorial disponible en YouTube, para todas aquellas personas que prefieren aprender con vídeos en vez de con manuales, a través del siguiente enlace: [https://youtu.be/tYYHd\\_bVTjg](https://youtu.be/tYYHd_bVTjg). El tiempo de visualización del tutorial es de unos 15 minutos.

#### **Funcionalidades básicas**

##### **Página principal**

Al acceder al enlace, la primera toma de contacto con la aplicación es la página principal (Figura E.1). En ella, además del logo y cierta información sobre sus funcionalidades, encontramos una barra de navegación, donde se puede acceder a las ventanas de «contacto» o «iniciar sesión», así como cambiar el idioma.

Por otra parte, en la parte inferior de la página aparecen el logo de la Universidad de Burgos y de la Asociación de Parkinson de Burgos, permitiendo al usuario acceder a sus páginas oficiales si pincha en ellos.



Figura E.1: Página principal

## Contacto

Al pulsar sobre las letras de «Contacto» en la barra de navegación, el usuario podrá ver una página con información sobre los creadores de la aplicación (el tutor y la alumna) y el objetivo de esta (Figura E.2). Además aparece el apartado de soporte para los usuarios de la aplicación, mostrando los correos electrónicos de los creadores de la *app*, así como la ubicación en la que se les puede encontrar (Figura E.3).

Figura E.2: Página principal - Contacto 1

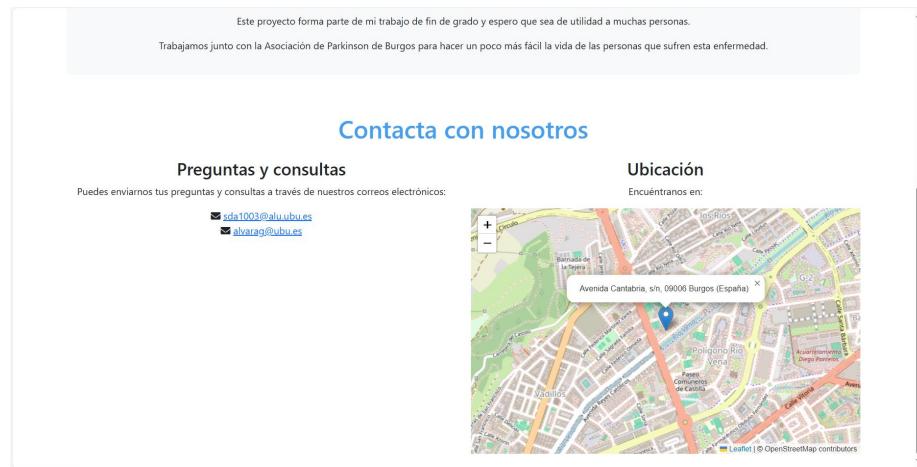


Figura E.3: Página principal - Contacto 2

## Idioma

Si se pulsa sobre «Idioma», se desplegará un menú en el que el usuario puede elegir el idioma en el que quiere que se muestre la aplicación, pudiendo elegir entre Inglés, Francés o Español (Figura E.4). Esta opción de la barra de navegación está siempre visible por si se necesita cambiar el idioma en algún otro momento.



Figura E.4: Página principal - Idioma

## Inicio de sesión

El inicio de sesión (Figura E.5) consiste en un simple formulario en el que se debe introducir el nombre de usuario y la contraseña de la aplicación, para posteriormente darle a enviar.

La imagen muestra la interfaz de usuario de la aplicación 'Tremor'. En la parte superior, hay una barra azul con iconos para 'Contacto' y 'Idioma', así como enlaces para 'Página principal' y 'Iniciar sesión'. El logo 'Tremor' es una firma en negro. La sección central se titula 'Inicio sesión' y contiene un cuadro para los datos de inicio de sesión. Dentro del cuadro, se ven los campos 'Nombre de usuario' y 'Contraseña', ambos con placeholder 'Ingresa tu nombre de usuario' y 'Ingresa tu contraseña'. A continuación, hay un botón 'Enviar' en azul. Los campos están rodeados por un efecto de sombra.

Figura E.5: Página principal - Inicio de Sesión

Dependiendo del tipo de usuario que sea, se redirige al usuario a su página de bienvenida.

## Funcionalidades para pacientes

### Página bienvenida paciente

El primer cambio que se observa en la Figura E.6 es que, en la parte derecha de la barra de navegación, aparece el nombre de usuario y la foto del usuario que ha iniciado sesión. Si se pulsa en el desplegable, permite volver a esta página o cerrar la sesión. Para posteriormente volver a la página principal del paciente se podrá hacer también pulsando sobre el logo de la aplicación, en el lateral izquierdo de la barra de navegación.

La sesión también se puede cerrar al darle al botón rojo de «Cerrar sesión», lo que redirigirá al usuario de nuevo a la página principal de la aplicación.



Figura E.6: Página principal - Paciente

Esta página contiene la información del perfil del paciente que ha iniciado sesión. Este puede editar algunos de sus datos personales si así lo desea, mediante un formulario que se despliega al darle al botón de «Editar datos personales» (Figura E.7). Para hacer efectivos los cambios en la base de datos se debe dar al botón de «Guardar cambios». Si por el contrario, desea cerrar el formulario, solo debe pulsar de nuevo en el botón de «Editar datos personales» para ocultarlo.



Figura E.7: Página principal - Paciente (editar datos personales)

Existen otros dos botones, uno que redirige al paciente a la ventana donde se muestran las gráficas de los datos del sensor y otro que le lleva a la

ventana donde se pueden reproducir sus vídeos, mostrar y predecir sus características.

Si pulsa en «Revisar mi evolución» se le redirigirá a la siguiente ventana:

### Mostrar datos sensor

En la Figura E.8 se muestra la ventana para mostrar los datos del sensor. En la parte superior se encuentran las migas de pan, que nos indican la ruta seguida hasta llegar a la página actual y nos permiten volver a páginas anteriores clicando en ellas. El resto de la página la ocupa un formulario, donde se debe indicar las gráficas a mostrar y en qué fechas.

Bienvenido / Mostrar Datos Sensor

**Paciente: Pepe Aguilar**

Selecciona la gráfica a mostrar:  
Seleccionar

Selecciona las fechas a mostrar:  
Selecciona fechas...

Fechas disponibles:  
2019-07-23 a 2019-07-30  
2019-09-05 a 2019-09-12  
2023-02-14 a 2023-03-01

Mostrar datos

Figura E.8: Mostrar Datos Paciente

El objetivo de este apartado es poder ver gráficas que muestren los datos recogidos por el sensor que ha debido de llevar el usuario. Se puede seleccionar entre cuatro temas de gráficas (Figura E.9):

- Parámetros de Bradicinesia: calcula una gráfica con la marcha media filtrada, otra con la desviación estándar media de la marcha y otra con el número de pasos considerados. Si se pulsa sobre los puntos de la gráfica se puede ver el valor exacto de ese momento.
- Parámetros de FoG y Discinesia: muestra los episodios de FoG, la probabilidad de discinesia y la confianza en la discinesia del paciente.
- Información de los pasos: muestra la longitud, el número, la velocidad y la cadencia de los pasos dados por el paciente a lo largo del tiempo.

- Parámetros de Estado Motor, Discinesia y Bradicinesia a 10 minutos: como su nombre indica muestra el estado motor, discinesia y bradicinesia del paciente en intervalos de 10 minutos.

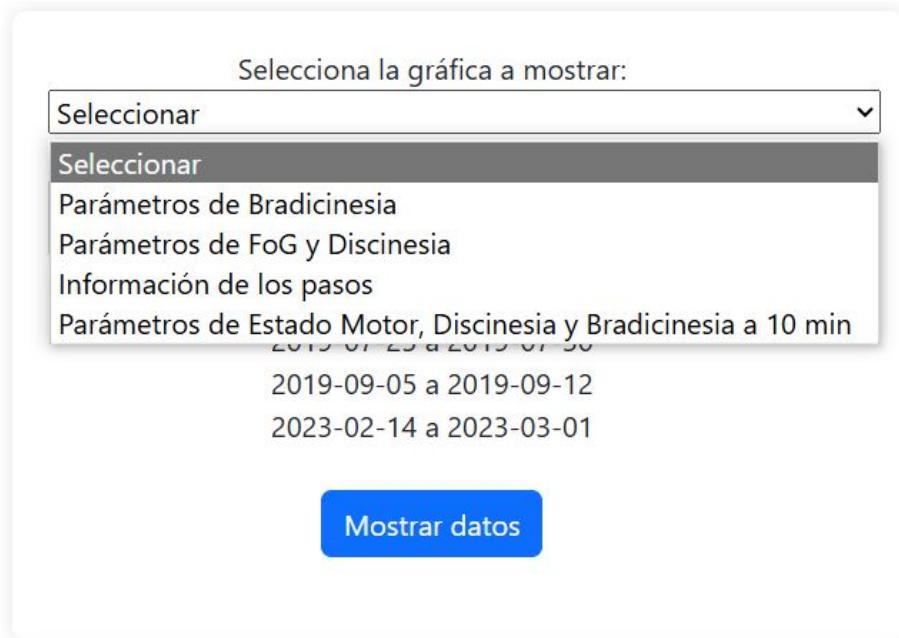


Figura E.9: Mostrar Datos Paciente - Seleccionar

Se debe indicar de qué fechas se quiere que se muestren los gráficos mediante un calendario (Figura E.10). Se puede elegir un día en específico o un rango de fechas entre los disponibles. Las fechas no disponibles no son seleccionables.

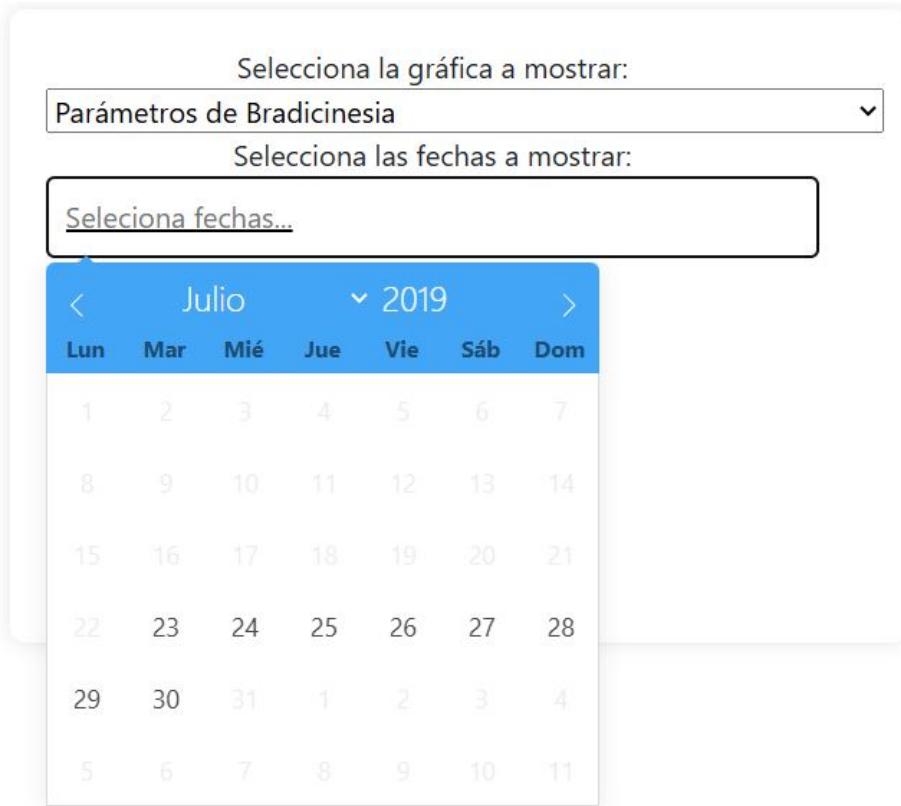


Figura E.10: Mostrar Datos Paciente - Fechas

Al elegir estos dos aspectos y dar al botón de «Mostrar datos», aparecerán tantas miniaturas como gráficas correspondan en esa selección (Figura E.11).

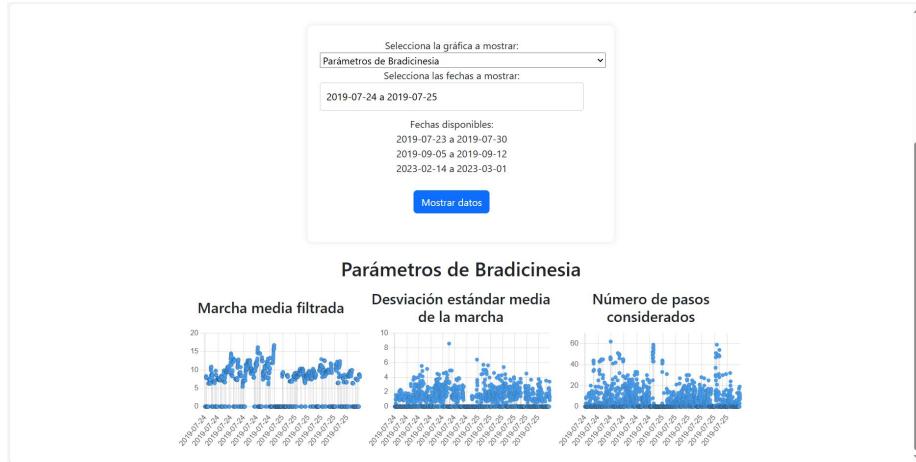


Figura E.11: Mostrar Datos Paciente - Miniaturas

Si se pulsa sobre una de ellas, se mostrará más abajo dicha miniatura pero en tamaño completo (Figura E.12). Como ejemplo se ha pulsado sobre la gráfica de «Marcha media filtrada», mostrando una gráfica con la evolución de la marcha del paciente en metros por segundo al cuadrado durante las fechas seleccionadas.

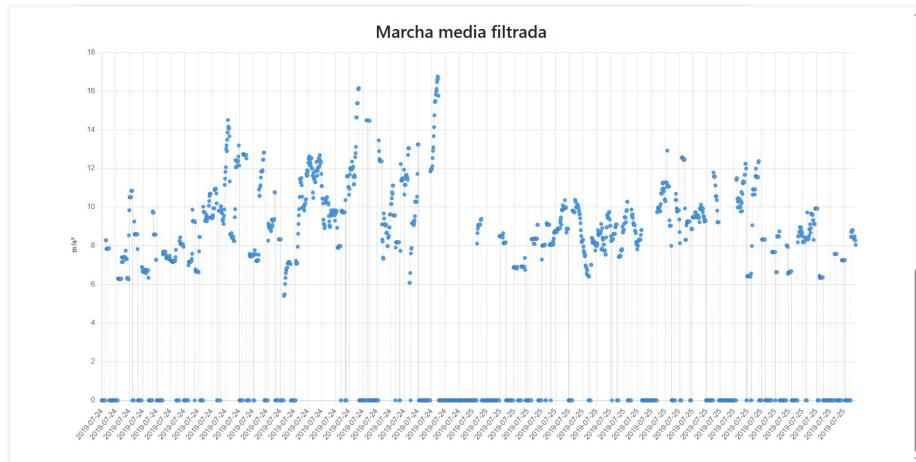


Figura E.12: Mostrar Datos Paciente - Gráfico

Si pulsa en «Mostrar vídeos» se le redirigirá a la siguiente ventana:

### Mostrar vídeos

En la Figura E.13 se observa un listado con los vídeos grabados por el paciente Pepe Aguilar. De cada vídeo se indica el nombre de archivo, la

mano que aparece y la fecha en la que fue grabado. Además permite visualizar los vídeos si se pulsa en «Reproducir vídeo» o eliminarlos si se pulsa en «Eliminar vídeo». Si el paciente no tiene vídeos, se comunicará mediante un mensaje en pantalla.

Título	Mano del video	Fecha	Reproducir	Eliminar
ID1_2018-02-07_derecha_2024-05-13_17-53-13.mp4	Derecha	2018-02-07	Reproducir video	Eliminar video
ID1_2018-02-07_izquierda_2024-05-13_17-54-04.mp4	Izquierda	2018-02-07	Reproducir video	Eliminar video
ID1_2018-08-25_derecha_2024-05-13_17-50-25.mp4	Derecha	2018-08-25	Reproducir video	Eliminar video
ID1_2018-08-25_izquierda_2024-05-13_17-51-05.mp4	Izquierda	2018-08-25	Reproducir video	Eliminar video
ID1_2019-03-13_derecha_2024-05-08_21-39-12.mp4	Derecha	2019-03-13	Reproducir video	Eliminar video

Figura E.13: Mostrar Vídeos

En la parte inferior de la ventana (Figura E.14) se muestran dos gráficos que muestran la evolución de ciertos movimientos de los vídeos a lo largo del tiempo para ambas manos.

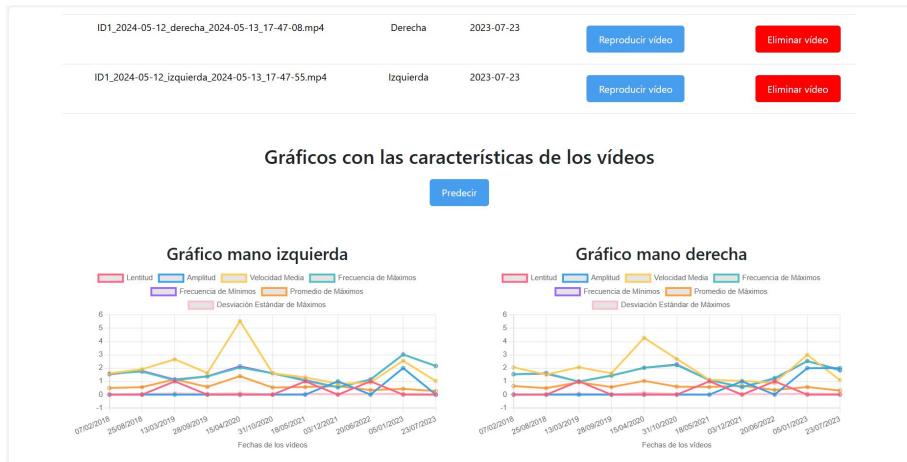


Figura E.14: Mostrar Vídeos - Gráficas

Si se pulsa sobre el botón «Predecir» se generarán dos gráficos (Figura E.15), uno por cada mano, mediante inteligencia artificial, donde

las cuatro últimas fechas son calculadas mediante un modelo de *machine learning* a partir de los datos disponibles.

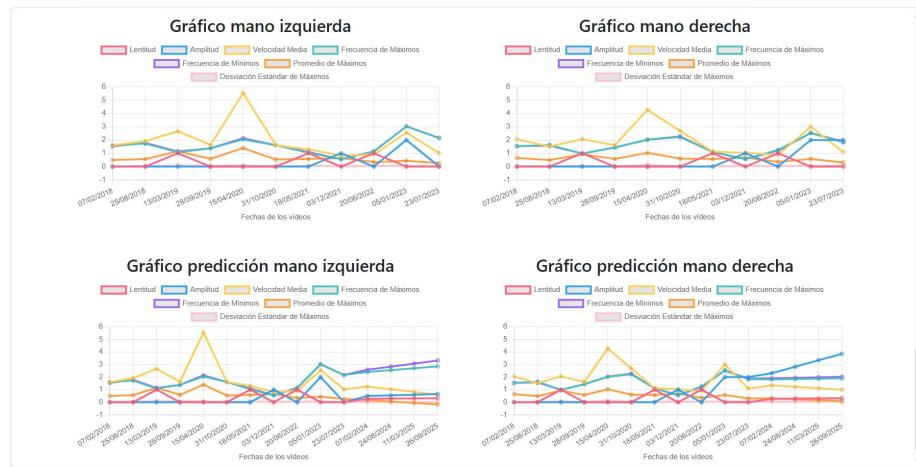


Figura E.15: Mostrar Vídeos - Gráficas con predicción

## Funcionalidades para médicos

### Página bienvenida médico

Si el usuario que inicia sesión es un médico aparecerá la ventana de bienvenida de la Figura E.16.

Al igual que cuando inician sesión otros usuarios, en la parte derecha de la barra de navegación, aparece el nombre de usuario y la foto del médico. Si se pulsa en el desplegable, permite volver a esta página o cerrar la sesión. Para posteriormente volver a la página principal del médico se podrá hacer también pulsando sobre el logo de la aplicación, en el lateral izquierdo de la barra de navegación.

La sesión también se puede cerrar al darle al botón rojo de «Cerrar sesión», lo que redirigirá al usuario de nuevo a la página principal de la aplicación.



Figura E.16: Página principal - Médico

La última funcionalidad de la página principal de los médicos es un botón, «Mostrar listado pacientes», el cual redirige al usuario a la siguiente pantalla.

### Listado de pacientes

La funcionalidad principal de la pantalla de la Figura E.17 es mostrar al médico una tabla con todos los pacientes que tiene a su cargo. Por cada paciente aparece su id, nombre y foto, junto con una serie de botones para realizar diferentes acciones.

ID	Nombre	Foto	Acciones
1	Pepe		<a href="#">Información personal</a> <a href="#">Subir video</a> <a href="#">Mostrar videos</a> <a href="#">Subir datos sensor</a> <a href="#">Mostrar datos sensor</a>
2	Trixí		<a href="#">Información personal</a> <a href="#">Subir video</a> <a href="#">Mostrar videos</a> <a href="#">Subir datos sensor</a> <a href="#">Mostrar datos sensor</a>
3	Doll		<a href="#">Información personal</a> <a href="#">Subir video</a> <a href="#">Mostrar videos</a> <a href="#">Subir datos sensor</a> <a href="#">Mostrar datos sensor</a>
4	Marta		<a href="#">Información personal</a>

Figura E.17: Listado pacientes Médico

Si se pulsa sobre «Información personal» se despliegan la lateralidad, si tiene sensor o no, la fecha de nacimiento, la dirección y el teléfono de dicho

paciente (Figura E.18), existiendo la posibilidad de editar dichos datos y guardar los cambios en la base de datos.

**Información personal**

Lateralidad:	Sensor:
<input type="button" value="Diestro ▾"/>	<input type="button" value="Sí ▾"/>
Fecha de Nacimiento:	
<input type="text" value="11/02/1988"/> <input type="button" value=""/>	
Dirección:	
<input type="text" value="C.Vitoria 43, 3C, Burgos"/>	
Teléfono:	
<input type="text" value="600042767"/>	
<input type="button" value="Guardar cambios"/> <input type="button" value="Cancelar"/>	

Figura E.18: Listado pacientes Médico - Información personal

Si se pulsa sobre «Subir vídeo», aparece un formulario (Figura E.19) en el que se ha de subir: vídeo (al darle a «Elegir archivo» se abre el explorador de archivos y permite escoger el archivo de vídeo que se desee subir), fecha de grabación de dicho vídeo, mano con la que se realiza el movimiento de pinza y datos de amplitud y velocidad del movimiento de pinza (siendo 0 un movimiento bien realizado y 4 una discapacidad severa).

**Subir vídeo**

Archivo de vídeo:

Elegir archivoNo se ha seleccionado ningún archivo

Fecha del vídeo:

CALENDARIO

Mano del vídeo: Seleccionar ▾ Lentitud: Seleccionar ▾ Amplitud: Seleccionar ▾

**Subir vídeo**

Figura E.19: Listado pacientes Médico - Subir vídeo

Si se pulsa sobre «Subir datos sensor», aparece un formulario (Figura E.20) en el que se ha de subir el archivo .csv que contenga los datos obtenidos del sensor de dicho paciente.

**Subir vídeo**    **Mostrar vídeos**    **Subir datos sensor**

Archivo CSV:

Elegir archivoNo se ha seleccionado ningún archivo

**Subir archivo CSV**

Figura E.20: Listado pacientes Médico - Subir datos sensor

Los botones de «Mostrar vídeos» y «Mostrar datos sensor» presentes en cada paciente de la lista redirigen al usuario a las páginas que muestran los gráficos de los datos del sensor de dicho paciente o los videos y características de esos videos de dicho paciente, explicadas en el apartado anterior de pacientes.

## Funcionalidades para administradores

### Página bienvenida administrador

Si el usuario que inicia sesión es un administrador aparecerá la ventana de bienvenida de la Figura E.21.

Al igual que cuando inician sesión otros usuarios, en la parte derecha de la barra de navegación, aparece el nombre de usuario y la foto del administrador. Si se pulsa en el desplegable, permite volver a esta página o cerrar la sesión. Para posteriormente volver a la página principal del administrador se podrá hacer también pulsando sobre el logo de la aplicación, en el lateral izquierdo de la barra de navegación.

La sesión también se puede cerrar al darle al botón rojo de «Cerrar sesión», lo que redirigirá al usuario de nuevo a la página principal de la aplicación.

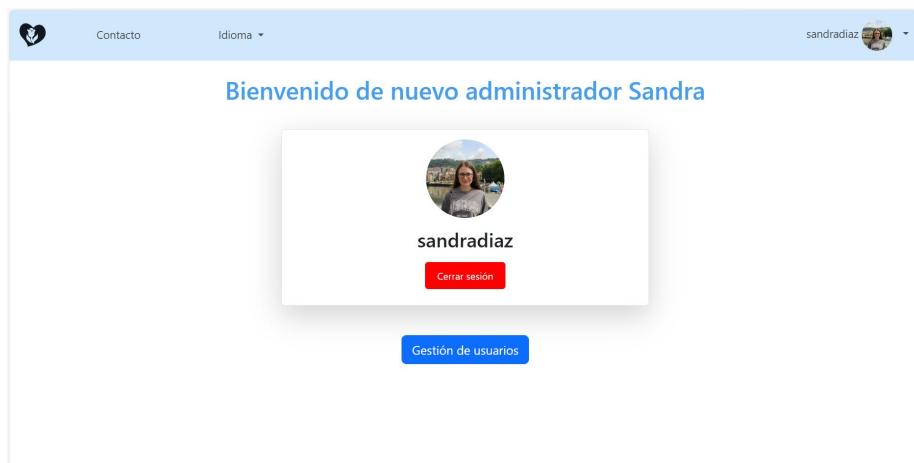


Figura E.21: Página principal - Administrador

La última funcionalidad de la página principal de los administradores es un botón, «Gestión de usuarios», el cual redirige al usuario a la siguiente pantalla.

## Gestión de usuarios

La funcionalidad principal de la ventana de la Figura E.22 es poder gestionar todos los usuarios de la aplicación. Existen tres tipos de usuarios: administradores, médicos y pacientes.



Figura E.22: Gestión de usuarios

Los administradores pueden acceder a los datos de cualquiera de ellos, existiendo 3 tablas, una para cada tipo de usuario, que se explican a continuación:

- Tabla de administradores Al pulsar sobre «Administradores» se muestra una tabla que contiene a todos los administradores de la aplicación web (Figura E.23). De cada uno de ellos se muestra su nombre y su foto, y se pueden realizar las acciones de editar todos sus campos o eliminar dicho usuario de la base de datos.

The screenshot shows a web-based user management system. At the top, there's a breadcrumb navigation: 'Bienvenido / Gestión de Usuarios'. Below it, a title 'Gestión de usuarios' and a subtitle 'Elige la tabla a gestionar:' followed by three buttons: 'Administradores' (selected), 'Médicos', and 'Pacientes'. A green button 'Añadir Administrador' is visible. The main area is titled 'Tabla de Administradores' and displays two rows of data:

ID	Nombre	Foto	Acciones
1	Sandra		<button>Editar usuario</button> <button>Eliminar Usuario</button>
2	Álvar		<button>Editar usuario</button> <button>Eliminar Usuario</button>

Figura E.23: Gestión de usuarios - Administradores

Además los administradores pueden crear administradores nuevos introduciendo sus datos mediante un formulario que se despliega al pulsar sobre «Añadir Administrador» y se muestra en la Figura E.24.

Nombre:

Apellido:

Nombre de Usuario:

Contraseña:

Correo Electrónico:

Foto:

No se ha seleccionado ningún archivo

Figura E.24: Gestión de usuarios - Añadir Usuario

- Tabla de médicos Al pulsar sobre «Médicos» se muestra una tabla que contiene a todos los médicos de la aplicación web (Figura E.25). De cada uno de ellos se muestra su nombre y su foto, y se pueden realizar las acciones de editar todos sus campos o eliminar dicho usuario de la base de datos.

ID	Nombre	Foto	Acciones
1	José Félix		<button>Editar usuario</button> <button>Eliminar Usuario</button>
2	Zabrina		<button>Editar usuario</button> <button>Eliminar Usuario</button>
3	Fernanda		<button>Editar usuario</button> <button>Eliminar Usuario</button>

Figura E.25: Gestión de usuarios - Médicos

Además los administradores pueden crear médicos nuevos introduciendo sus datos mediante un formulario que se despliega al pulsar sobre «Añadir Médico», y que es igual que el usado para los administradores en la Figura E.24.

- Tabla de pacientes Al pulsar sobre «Pacientes» se muestra una tabla que contiene a todos los pacientes de la aplicación web (Figura E.26). Dicha tabla es más completa que las anteriores. De cada uno de ellos se muestra su nombre y su foto, y se pueden realizar las acciones de editar todos sus campos o eliminar dicho usuario de la base de datos. También se pueden realizar las acciones que pueden realizar los médicos sobre los pacientes, explicadas en el apartado de médicos (subir vídeo, subir datos sensor, mostrar vídeos y mostrar datos sensor).

The screenshot shows a web-based user management system. At the top, there's a breadcrumb navigation: 'Bienvenido / Gestión de Usuarios'. Below it, a title 'Gestión de usuarios' and a subtitle 'Elige la tabla a gestionar:' followed by three buttons: 'Administradores', 'Médicos', and 'Pacientes'. The 'Pacientes' button is highlighted. The main area is titled 'Tabla de Pacientes' and contains a table with two rows of patient data. The columns are labeled 'ID', 'Nombre', 'Foto', and 'Acciones'. Row 1 (Pepe) has actions: 'Subir video', 'Subir datos sensor', 'Mostrar videos', 'Mostrar datos sensor', 'Editar usuario' (in orange), and 'Eliminar Usuario' (in red). Row 2 (Trixí) has actions: 'Subir video' and 'Subir datos sensor'. A green 'Añadir Paciente' button is located above the table.

ID	Nombre	Foto	Acciones
1	Pepe		<button>Subir video</button> <button>Subir datos sensor</button> <button>Mostrar videos</button> <button>Mostrar datos sensor</button> <button>Editar usuario</button> <button>Eliminar Usuario</button>
2	Trixí		<button>Subir video</button> <button>Subir datos sensor</button>

Figura E.26: Gestión de usuarios - Pacientes

Además los administradores pueden crear pacientes nuevos introduciendo sus datos mediante un formulario que se despliega al pulsar sobre «Añadir Paciente» y se muestra en la Figura E.27.

Nombre:	Apellido:
<input type="text"/>	<input type="text"/>
Nombre de Usuario:	Contraseña:
<input type="text"/>	<input type="text"/>
Correo Electrónico:	
<input type="text"/>	
Foto:	
<input type="button" value="Elegir archivo"/> No se ha seleccionado ningún archivo	
Fecha de Nacimiento:	Dirección:
<input type="text" value="dd/mm/aaaa"/> <input type="button" value=""/>	<input type="text"/>
Teléfono:	¿Tiene Sensor?
<input type="text"/>	<input type="checkbox"/> Sí <input checked="" type="checkbox"/>
Médico Asignado:	Lateralidad:
<input type="button" value="José Felix Martínez ▾"/>	<input type="button" value="Diestro ▾"/>
<input type="button" value="Agregar Usuario"/> <input type="button" value="Cancelar"/>	

Figura E.27: Gestión de usuarios - Añadir paciente

## *Apéndice F*

---

# **Anexo de sostenibilización curricular**

---

## **F.1. Introducción**

En este último anexo se abordan los aspectos de sostenibilidad aplicados al Trabajo de Fin de Grado. Integrar la sostenibilidad en proyectos médicos o universitarios, además de en la vida diaria, es esencial para mitigar su impacto ambiental y mejorar la calidad de vida global.

Se ha utilizado como base de conocimientos la guía de la CRUE<sup>1</sup>, que promueve una investigación y desarrollo responsables y sostenibles, ya que durante el alumnado las competencias adquiridas en términos de sostenibilidad fueron escasas.

Se van a desarrollar tres secciones: realizar una aplicación que ayuda al envejecimiento saludable, utilizar materiales reciclables y biodegradables en los sensores y emplear modelos de *machine learning* y equipos *hardware* eficientes.

## **F.2. Envejecimiento saludable**

El envejecimiento es una gran preocupación en términos de sostenibilidad social y sanitaria. Desarrollar aplicaciones como la actualmente presentada puede tener un impacto significativo en la calidad de vida de las personas mayores (en este caso de los que sufren la enfermedad de Parkinson).

<sup>1</sup>[https://www.crue.org/wp-content/uploads/2020/02/Directrices\\_Sostenibilidad\\_Crue2012.pdf](https://www.crue.org/wp-content/uploads/2020/02/Directrices_Sostenibilidad_Crue2012.pdf)

La utilización de sensores y la progresiva toma de vídeos de los pacientes permite realizar una monitorización continua, facilitando una detección temprana de avances en la enfermedad. Esto permite a los médicos llevar una gestión más efectiva de los tratamientos, ajustando las dosis en función de lo observado en las gráficas que proporciona la aplicación web.

Además la aplicación, al incluir un apartado en el que los pacientes pueden ver sus propios avances, proporciona independencia en la gestión de la enfermedad. Sentirse autosuficiente aporta un beneficio psicológico y emocional a las personas mayores, aumentando su autoestima y su sensación de control sobre sus propias vidas.

Por último, la monitorización efectiva puede reducir las hospitalizaciones y sus costes asociados, beneficiando tanto a los individuos y sus familias como al sistema sanitario. Además reduce los desplazamientos al centro de salud, reduciendo la huella de carbono de los combustibles y los problemas asociados al desplazamiento (pacientes con problemas severos de movilidad o que vivan muy alejados del hospital y no dispongan de medios de transporte para poder acudir a él).

### **F.3. Materiales de los sensores**

Se desconocen los materiales utilizados actualmente para fabricar el dispositivo médico STAT-ON, que dispone de sensores para monitorizar los síntomas motores de los pacientes, desarrollado por el equipo de SENSE4CARE [9], y su posterior reciclado, pero se van a dar ideas para tratar de que sean lo más sostenibles posibles.

Para reducir su impacto ambiental, en su fabricación se debería fomentar el uso de materiales ecológicos y reciclables. Se pueden reemplazar los plásticos convencionales por polímeros biodegradables (como el ácido poliláctico, derivado del maíz o la caña de azúcar, o los polihidroxialcanoatos, producidos por microorganismos) que tardan menos en descomponerse. Utilizar metales como el acero inoxidable o el aluminio, con altas tasas de reciclabilidad, disminuiría la huella de carbono, además de ser duraderos y resistentes a la corrosión. Además, se debería tratar de evitar sustancias tóxicas y contaminantes como el plomo y el mercurio en los circuitos electrónicos o las baterías del dispositivo.

Para ayudar con su reciclado tras su vida útil se deberían diseñar los sensores para ser fácilmente desmontables, facilitando la separación de sus materiales para su reciclaje individual.

## F.4. Software

Durante el proyecto, es esencial el uso de algoritmos de *machine learning* para analizar y posteriormente predecir datos de salud. Sin embargo, tanto el servidor como los modelos de *machine learning* utilizados implican un consumo de energía significativo, que debe ser considerado en términos de sostenibilidad.

### Consumo de energía del servidor

El proyecto se aloja en un servidor proporcionado por la UBU. El funcionamiento continuo de dicho servidor, necesario para soportar la infraestructura del proyecto actual y del resto de proyectos que aloje, conlleva un consumo energético constante. Para tratar de minimizar este consumo, sería interesante analizar el funcionamiento del equipo. Es crucial seleccionar *hardware* eficiente, que ofrezca un alto rendimiento con bajo consumo. Alguna recomendación sería utilizar discos sólidos SSD en lugar de los tradicionales discos duros HDD, módulos de memoria DDR4 o DDR5, procesadores específicos para servidores o sistemas de enfriamiento eficientes como la refrigeración líquida.

### Consumo de energía de los modelos de *machine learning*

Los modelos grandes y complejos, como los utilizados para entrenar a sistemas como ChatGPT, consumen tanta energía como una ciudad mediana. Esto no solo implica un alto coste económico, sino también un gran impacto ambiental debido a la cantidad de energía que requieren y sus emisiones asociadas.

Emplear modelos más simples y con algoritmos optimizados reduce el consumo de energía, ayudando a mitigar el impacto ambiental del proyecto. Es importante considerar un equilibrio entre la precisión de los modelos y su eficiencia energética, promoviendo la sostenibilidad en el contexto actual del preocupante aumento del calentamiento global.



---

## Bibliografía

---

- [1] Grupo Atico34. El consentimiento en el rgpd y lopd, 2020. [Internet; descargado 14-enero-2024].
- [2] Catalin Andrei Cacuci. Identificación de Parkinson por visión artificial, 2023.
- [3] Gabinete de Comunicación. La Universidad de Burgos y la Asociación Parkinson Burgos intensifican su colaboración, 2023. [Internet; descargado 27-enero-2024].
- [4] EUR-Lex. Reglamento general de protección de datos (RGPD), 2022. [Internet; descargado 14-enero-2024].
- [5] Exact. ¿Cómo calcular la rentabilidad de un proyecto?, 2022. [Internet; descargado 27-noviembre-2023].
- [6] Gestron. Cuánto paga una empresa por un trabajador, 2023. [Internet; descargado 27-noviembre-2023].
- [7] Movement Disorder Society Task Force on Rating Scales for Parkinson's Disease. The Unified Parkinson's Disease Rating Scale (UPDRS): Status and recommendations. *Movement Disorders*, 18(7):738–750, 2003.
- [8] Asociación Parkinson Salamanca. El símbolo de la enfermedad de Parkinson, 2023. [Internet; descargado 16-enero-2024].
- [9] SENSE4CARE. Experts in the detection and analysis of human movement patterns applied to healthcare. [Internet; descargado 22-mayo-2024].

- [10] Eva Zamora. *¿Cómo se calcula la amortización de equipos informáticos?*", year = "2022. [Internet; descargado 25-noviembre-2023].