



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 — 8 de mayo  
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>v</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 . . . . .	20
<b>Apéndice B Especificación de Requisitos</b>	<b>27</b>
B.1. Introducción . . . . .	27
B.2. Objetivos generales . . . . .	28
B.3. Catálogo de requisitos . . . . .	28
B.4. Especificación de requisitos . . . . .	31
<b>Apéndice C Especificación de diseño</b>	<b>47</b>
C.1. Introducción . . . . .	47
C.2. Diseño de datos . . . . .	48
C.3. Diseño de interfaces . . . . .	49
C.4. Diseño procedimental . . . . .	63
C.5. Diseño arquitectónico . . . . .	63
<b>Apéndice D Documentación técnica de programación</b>	<b>65</b>
D.1. Introducción . . . . .	65
D.2. Estructura de directorios . . . . .	65

D.3. Manual del programador . . . . .	67
D.4. Compilación, instalación y ejecución del proyecto . . . . .	68
D.5. Pruebas del sistema . . . . .	68
<b>Apéndice E Documentación de usuario</b>	<b>69</b>
E.1. Introducción . . . . .	69
E.2. Requisitos de usuarios . . . . .	69
E.3. Instalación . . . . .	69
E.4. Manual del usuario . . . . .	70
<b>Apéndice F Anexo de sostenibilización curricular</b>	<b>77</b>
F.1. Introducción . . . . .	77
<b>Bibliografía</b>	<b>79</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>	9
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>	11
A.13.	Gráfico <i>burndown</i> - <i>Sprint 7</i>	12
A.14.	Informe de estado - <i>Sprint 7</i>	12
A.15.	Gráfico <i>burndown</i> - <i>Sprint 8</i>	14
A.16.	Informe de estado - <i>Sprint 8</i>	14
A.17.	Gráfico <i>burndown</i> - <i>Sprint 9</i>	15
A.18.	Informe de estado - <i>Sprint 9</i>	15
A.19.	Gráfico <i>burndown</i> - <i>Sprint 10</i>	17
A.20.	Informe de estado - <i>Sprint 10</i>	17
A.21.	Gráfico <i>burndown</i> - <i>Sprint 11</i>	19
A.22.	Informe de estado - <i>Sprint 11</i>	19
B.1.	Diagrama de casos de uso	32
C.1.	Diagrama entidad-relación	48
C.2.	Diagrama relacional	48
C.3.	<i>Mockup</i> - Página principal	49

C.4. <i>Mockup</i> - Página principal (info) . . . . .	50
C.5. <i>Mockup</i> - Página principal (contacto) . . . . .	51
C.6. <i>Mockup</i> - Página principal (idioma) . . . . .	51
C.7. <i>Mockup</i> - Inicio sesión . . . . .	52
C.8. <i>Mockup</i> - Bienvenida paciente . . . . .	53
C.9. <i>Mockup</i> - Evolución paciente . . . . .	54
C.10. <i>Mockup</i> - Predicción paciente . . . . .	55
C.11. <i>Mockup</i> - Bienvenida médico . . . . .	55
C.12. <i>Mockup</i> - Listado pacientes . . . . .	56
C.13. <i>Mockup</i> - Información personal paciente . . . . .	57
C.14. <i>Mockup</i> - Subir vídeo paciente . . . . .	58
C.15. <i>Mockup</i> - Subir datos paciente . . . . .	58
C.16. <i>Mockup</i> - Evolución médico . . . . .	59
C.17. <i>Mockup</i> - Predicción médico . . . . .	60
C.18. <i>Mockup</i> - Bienvenida admin . . . . .	60
C.19. <i>Mockup</i> - Gestión usuarios . . . . .	61
C.20. <i>Mockup</i> - Modificar usuario . . . . .	62
C.21. <i>Mockup</i> - Eliminar usuario . . . . .	62
C.22. <i>Mockup</i> - Crear usuario . . . . .	63

---

# Índice de tablas

---

A.1. Tabla de contribuciones que las empresas pagan por trabajador	22
A.2. Tabla de costes del proyecto . . . . .	23
B.1. CU-1 Visualizar evolución. . . . .	33
B.2. CU-1.2 Realizar predicción. . . . .	34
B.3. CU-2 Administrar pacientes. . . . .	35
B.4. CU-2.1 Cambiar datos paciente. . . . .	36
B.5. CU-2.2 Subir datos sensor. . . . .	37
B.6. CU-2.3 Administrar vídeos. . . . .	38
B.7. CU-2.3.1 Subir vídeo. . . . .	39
B.8. CU-2.3.2 Eliminar vídeo. . . . .	40
B.9. CU-3 Gestionar sesión. . . . .	41
B.10.CU-4 Administrar usuarios. . . . .	42
B.11.CU-4.1 Crear usuario. . . . .	43
B.12.CU-4.2 Modificar usuario. . . . .	44
B.13.CU-4.3 Eliminar usuario. . . . .	45



## *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 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 párkinson, la enfermedad que aborda el proyecto. Además, se comenzó a familiarizar con las herramientas que se utilizarían posteriormente durante todo el proyecto (GitHub, Jira, Python, entornos virtuales...).

Se comenzó la documentación del proceso en Jira tarde, por lo que el *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					
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					
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 curiosa 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)
TFG-16	Familiarizarse con el código existente	Historia	= Medium	FINALIZADA	2
TFG-24	Crear aplicación web con Flask	Historia	↗ High	FINALIZADA	16

Incidencias Sin Completar					
Clave	Resumen	Tipo de Incidencia	Prioridad	Estado	Story Points (16)
TFG-22	Comenzar documentación memoria	Historia	= Medium	EN CURSO	8
TFG-23	Comenzar documentación anexos	Historia	= Medium	EN CURSO	8

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 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 – Corregir 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.

## A.2. Planificación temporal

9



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

### Informe de estado

Incidencias terminadas						<a href="#">Ver en el navegador de incidencias</a>
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						<a href="#">Ver en el navegador de incidencias</a>
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 L<sup>A</sup>T<sub>E</sub>X, 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.

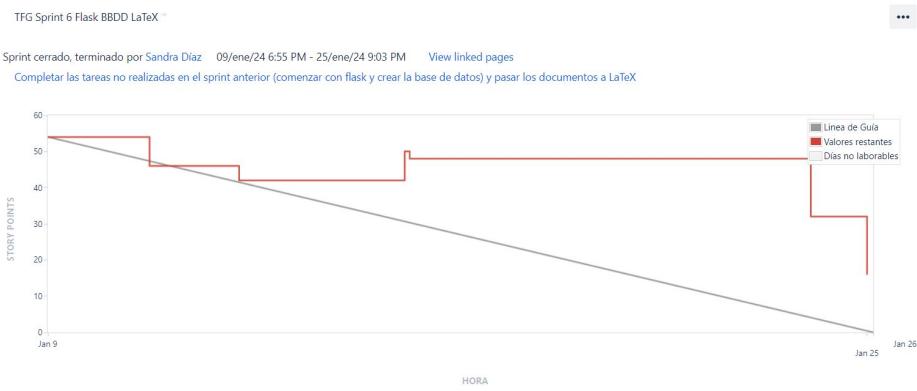


Figura A.11: Gráfico *burndown* - *Sprint 6*

Informe de estado					
Incidencias terminadas					
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					
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 - A tope 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)	Ver en el navegador de incidencias
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)	Ver en el navegador de incidencias
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 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						<a href="#">Ver en el navegador de incidencias</a>
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						<a href="#">Ver en el navegador de incidencias</a>
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
Incidencias terminadas						Ver en el navegador de incidencias
Clave	Resumen	Tipo de Incidencia	Prioridad	Estado	Story Points (62 → 66)	
TFG-68	Mejorar subida de videos 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 vídeos 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 implementar parte del TFG realizado por mi compañero Catalin.

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, centrando las tablas, mostrando los logos de la UBU y de la asociación de parkinson de 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 Catalin, 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 de *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						<a href="#">Ver en el navegador de incidencias</a>
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 - x

Blablá 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.

**Sprint 13 - x**

**Sprint 14 - x**

**Gráfico global final**

X

### A.3. Estudio de viabilidad

El estudio de 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 10 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

---

<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>

equipo tiene una antigüedad de 4 años, por lo que habría que calcular su amortización en el proyecto [7]:

$$\frac{1700 \text{ €}}{4 \text{ años} \times 12 \text{ meses/año}} = 35,42 \text{ €/mes}$$

$$35,42 \text{ €/mes} \times 8 \text{ meses} = 283,36 \text{ €}$$

- 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:

$$\frac{259 \text{ €}}{4 \text{ años} \times 12 \text{ meses/año}} = 5,4 \text{ €/mes}$$

$$5,4 \text{ €/mes} \times 8 \text{ meses} = 43,17 \text{ €}$$

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 One Drive 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.
- 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.

---

<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>

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

- 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.
  - Visual Studio Code: 0 euros.
- 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 siguiente tabla [5]:

<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

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

$$\frac{734 \text{ €/mes}}{1 - (0,236 + 0,015 + 0,006 + 0,067 + 0,002)} = 1089,02 \text{ €/mes}$$

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

<sup>6</sup>[https://www.glassdoor.es/Sueldos/espa%C3%B1a-programador-j%C3%BAnior-sueldo-SRCH\\_IL.0,6\\_IN219\\_K07,25.htm](https://www.glassdoor.es/Sueldos/espa%C3%B1a-programador-j%C3%BAnior-sueldo-SRCH_IL.0,6_IN219_K07,25.htm)

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

$$1089,02 \text{ €/mes} \times 8 \text{ meses} = 8712,17 \text{ €}$$

- 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 siguiente tabla:

<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

## 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

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

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

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 [2]. 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.

Se baraja la idea de incluir publicidad como una fuente potencial de ingresos pasivos, sin afectar negativamente la experiencia de los usuarios.

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 altos, 13951.39 €, y los beneficios han sido nulos. Típicamente la rentabilidad [4] 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.

## **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 [3]): 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) hay que abordar el tema de los derechos de autor.

Además, hay que tener en cuenta las licencias de las herramientas utilizadas durante el proyecto..... Para el proyecto se escoge la más restrictiva



## *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 usará 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 como, por ejemplo, Scikit-learn.
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 ser capaz de **predecir** datos futuros a partir de los datos históricos medidos por el sensor.
  - **RF2.1:** Se debe poder elegir el **rango de tiempo** que se quiere predecir en la gráfica de datos.
- **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 en todo momento el **logo** de la Universidad de Burgos y el logo de la Asociación de Parkinson de Burgos, redireccionando a sus páginas webs.
  - **RF10.1:** El sistema debe mostrar el logo de la propia aplicación, redireccionando 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**.
- **RF14:** El sistema debe poder mostrar la **política de privacidad** y los términos de uso de la aplicación

## 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 5 segundos en mostrar las gráficas o en subir 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 y predecir sus evoluciones futuras. 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 y eliminar vídeos de cada paciente. Al igual que los pacientes, pueden acceder a la visualización y predicción de la evaluación 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. Los casos de uso se representan con elipses conteniendo el nombre y los actores se representan con «monigotes» con el nombre del actor:

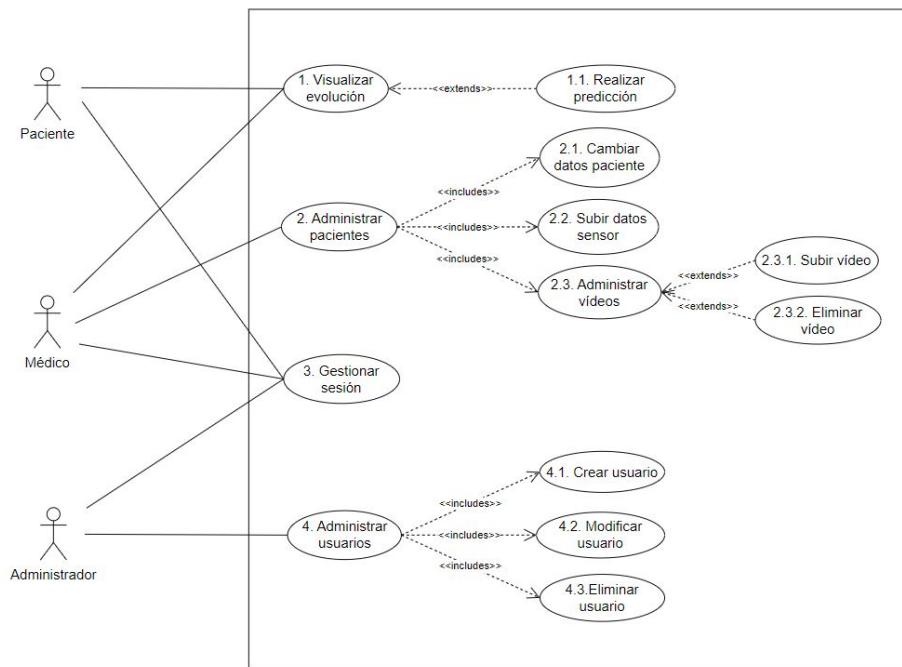


Figura B.1: Diagrama de casos de uso

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 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 o médico. Si es médico 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 indique una fecha previa a la primera fecha con registros, saltará un aviso y dicho intervalo de tiempo aparecerá en blanco en la gráfica. Por el contrario, si se indica una fecha posterior al último registro, se procederá a la predicción de los datos (CU-2).
<b>Importancia</b>	Alta

Tabla B.1: CU-1 Visualizar evolución.

<b>CU-1.2</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>	Relizar gráficas que predigan, utilizando <i>machine learning</i> , la evolución de los pacientes.
<b>Precondición</b>	El usuario, una vez ha iniciado sesión y está en la pantalla de visualizar datos, debe de haber indicado una fecha posterior al último registro.
<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, seleccionando fechas posteriores a los registros existentes.</li> <li>3. El usuario pulsa el botón que hace que se muestren las predicciones.</li> </ol>
<b>Postcondición</b>	Se habrá visualizado una gráfica con la predicción de la evolución del cliente.
<b>Excepciones</b>	No hay excepciones
<b>Importancia</b>	Alta

Tabla B.2: CU-1.2 Realizar predicción.

<b>CU-2</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, añadir/borrar vídeos, subir datos y ver/predecir su evolución.
<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.3: CU-2 Administrar pacientes.

<b>CU-2.1</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 pulsa en el ícono que permite editar los datos.</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.4: CU-2.1 Cambiar datos paciente.

CU-2.2	Subir datos sensor
<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 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. 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.5: CU-2.2 Subir datos sensor.

<b>CU-2.3</b>	<b>Administrar vídeos</b>
<b>Versión</b>	1.0
<b>Autor</b>	Sandra Díaz Aguilar
<b>Requisitos asociados</b>	RF4
<b>Descripción</b>	Permite a los médicos gestionar los vídeos de los pacientes en la aplicación web.
<b>Precondición</b>	El usuario debe haber iniciado sesión y ser médico. Debe existir el perfil del paciente sobre el cual se quieren modificar los vídeos.
<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 le lleva a la gestión de vídeos.</li> </ol>
<b>Postcondición</b>	Se abrirá una sección en la que subir/eliminar vídeos del paciente.
<b>Excepciones</b>	No hay excepciones
<b>Importancia</b>	Media

Tabla B.6: CU-2.3 Administrar vídeos.

CU-2.3.1	Subir vídeo
<b>Versión</b>	1.0
<b>Autor</b>	Sandra Díaz Aguilar
<b>Requisitos asociados</b>	RF4
<b>Descripción</b>	Permite a un médico añadir uno o más videos a la lista de videos asociados a un paciente.
<b>Precondición</b>	El usuario debe haber iniciado sesión y ser un médico. Además, debe situarse en el perfil del paciente al que se quiere asociar dicho video. Los pacientes deben haber grabado previamente esos videos, que deben estar a disposición del médico.
<b>Acciones</b>	<ol style="list-style-type: none"> <li>1. Junto a los videos asociados al paciente aparecerá un símbolo de añadir video. Se debe pulsar en él.</li> <li>2. El médico es redirigido a sus archivos, donde debe escoger el video/videos que desea subir.</li> <li>3. Aparecen los videos escogidos y se debe confirmar la subida.</li> </ol>
<b>Postcondición</b>	El video 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 video adecuado, el sistema no dejará subir el video, mostrando un mensaje de error.
<b>Importancia</b>	Media

Tabla B.7: CU-2.3.1 Subir vídeo.

<b>CU-2.3.2</b>	<b>Eliminar vídeo</b>
<b>Versión</b>	1.0
<b>Autor</b>	Sandra Díaz Aguilar
<b>Requisitos asociados</b>	RF4
<b>Descripción</b>	Permite a un médico eliminar un vídeo de los asociados a un paciente de su lista de pacientes.
<b>Precondición</b>	El usuario debe haber iniciado sesión y ser un médico. Además, debe situarse en el perfil del paciente al que se quiere borrar un vídeo. Para ello, el paciente en cuestión debe tener vídeos asociados.
<b>Acciones</b>	<ol style="list-style-type: none"> <li>1. Se debe pulsar con el botón derecho sobre un vídeo de los relacionados al paciente.</li> <li>2. El médico debe pulsar en el botón que lo elimine.</li> <li>3. Se debe aceptar la confirmación de que el vídeo desea ser borrado.</li> </ol>
<b>Postcondición</b>	El vídeo debe haberse borrado de los registros de la base de datos del paciente.
<b>Excepciones</b>	En caso de no tener vídeos no habrá nada que eliminar.
<b>Importancia</b>	Media

Tabla B.8: CU-2.3.2 Eliminar vídeo.

<b>CU-3</b>	<b>Gestionar sesión</b>
<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>	Para iniciar sesión: <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> Para cerrar sesión: <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.9: CU-3 Gestionar sesión.

<b>CU-4</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, mostrándose una lista con todos los usuarios existentes.</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.10: CU-4 Administrar usuarios.

<b>CU-4.1</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 y rol.</li> <li>3. Pulsar en el botón que lo añade y realizar la confirmación.</li> </ol>
<b>Postcondición</b>	El usuario creado se verá reflejados en la base de datos de usuarios aceptados.
<b>Excepciones</b>	Cada campo del formulario tiene sus propias excepciones: nombre de usuario nuevo y con longitud adecuada, contraseña suficientemente segura y rol existente.
<b>Importancia</b>	Alta

Tabla B.11: CU-4.1 Crear usuario.

<b>CU-4.2</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: nombre de usuario, contraseña y rol.
<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.12: CU-4.2 Modificar usuario.

CU-4.3	Eliminar usuario
<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 ya no aparecerá en la base de datos de usuarios aceptados, 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.13: CU-4.3 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 las 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 y el diagrama relacional realizados al crear la base de datos (y el diccionario de datos).
- Diseño de interfaces: se muestra el *mockup* de la aplicación, cada una de las ventanas diseñadas en Pencil.
- Diseño procedural: 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 para exponer las relaciones entre el cliente, el servidor, la base de datos... así como el diagrama de clases para mostrar las relaciones entre las clases que controlan la funcionalidad de la web en Python.

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

## C.2. Diseño de datos

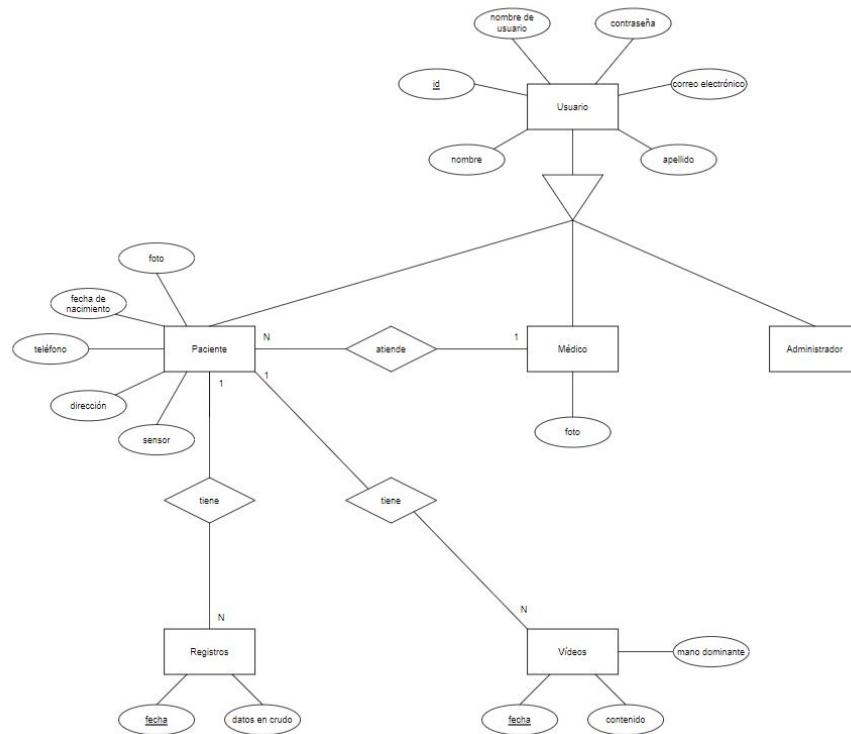


Figura C.1: Diagrama entidad-relación.

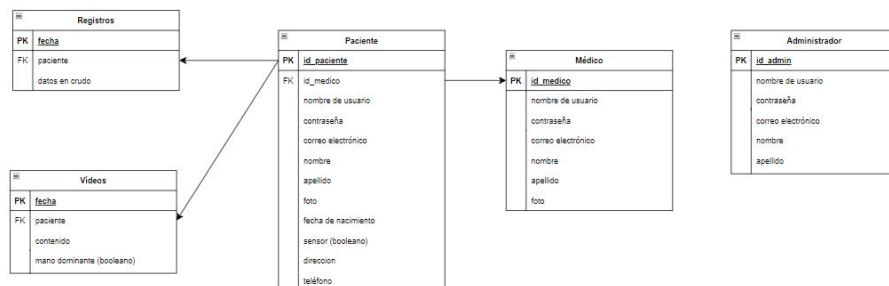


Figura C.2: Diagrama relacional.

## 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 [6]. 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.

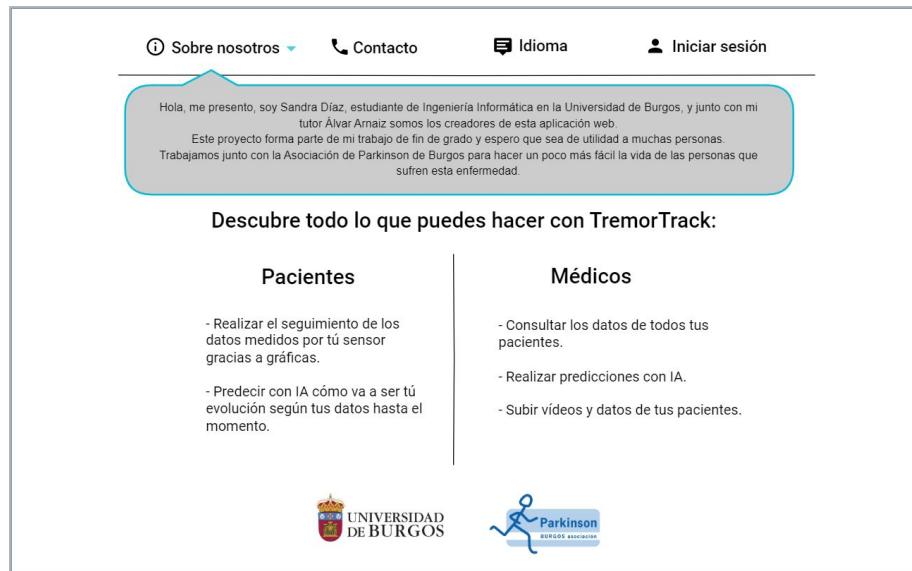


Figura C.4: *Mockup - Página principal (info)*

- 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 español ya que el proyecto del sensor se realizó en España, 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.

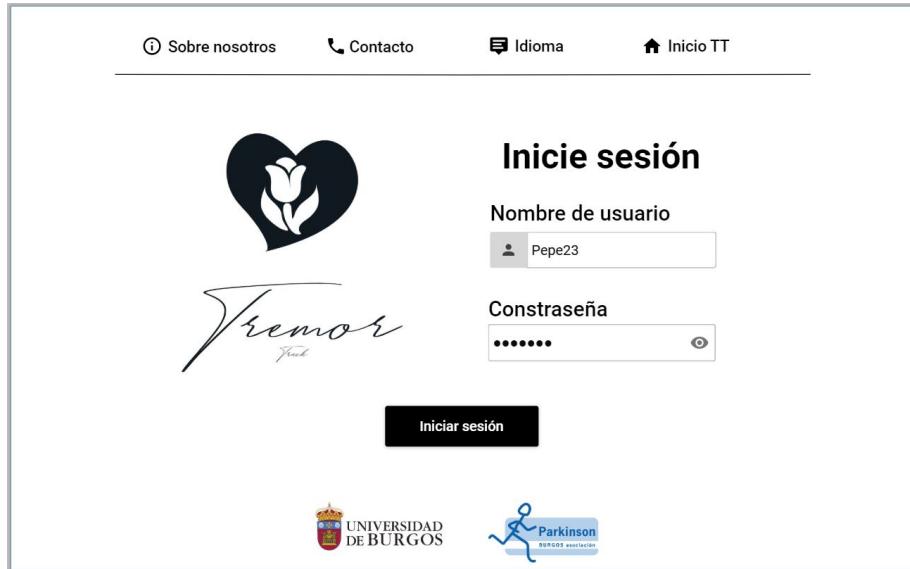


Figura C.7: *Mockup - Inicio sesión*

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.

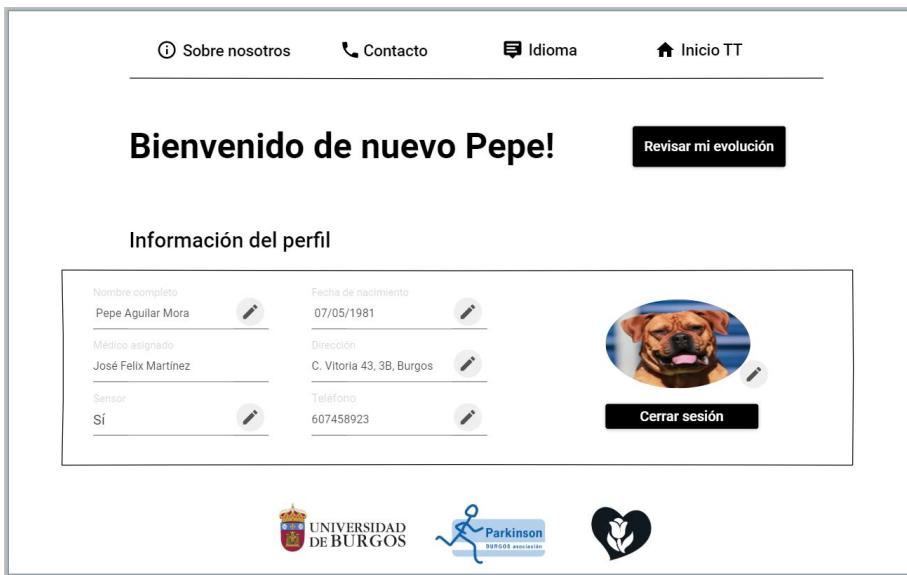


Figura C.8: *Mockup - Bienvenida paciente*

- 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.

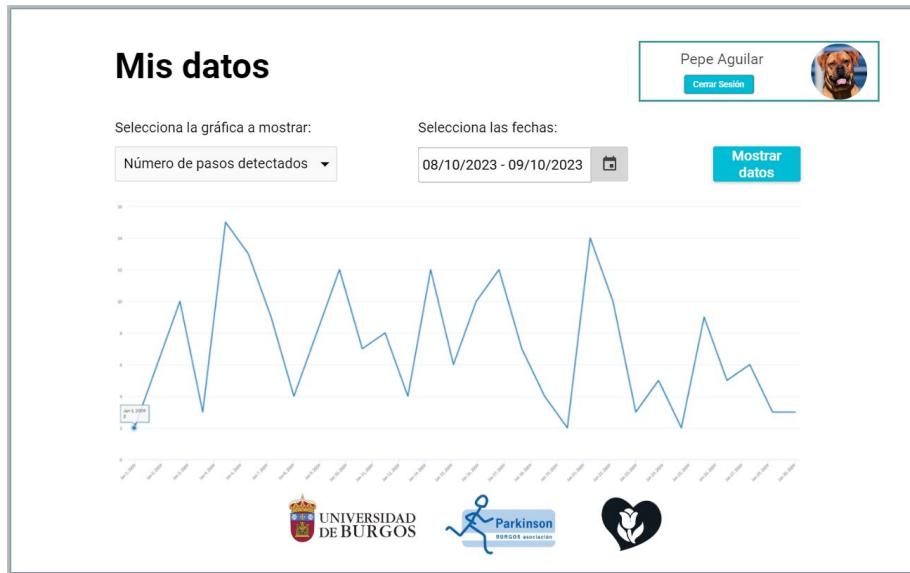


Figura C.9: *Mockup - Evolución paciente*

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:

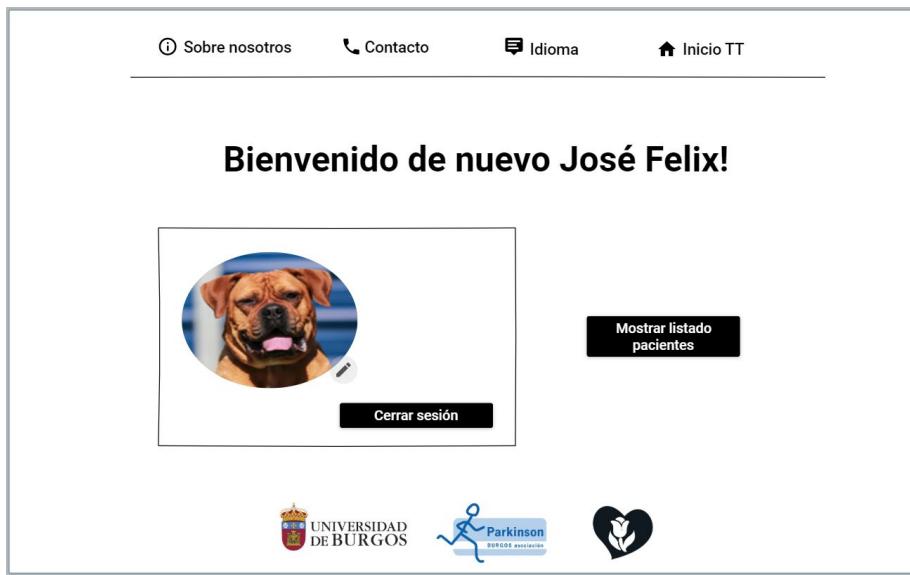


Figura C.11: *Mockup* - Bienvenida médico

- 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 tenga dicho médico asociados. 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.

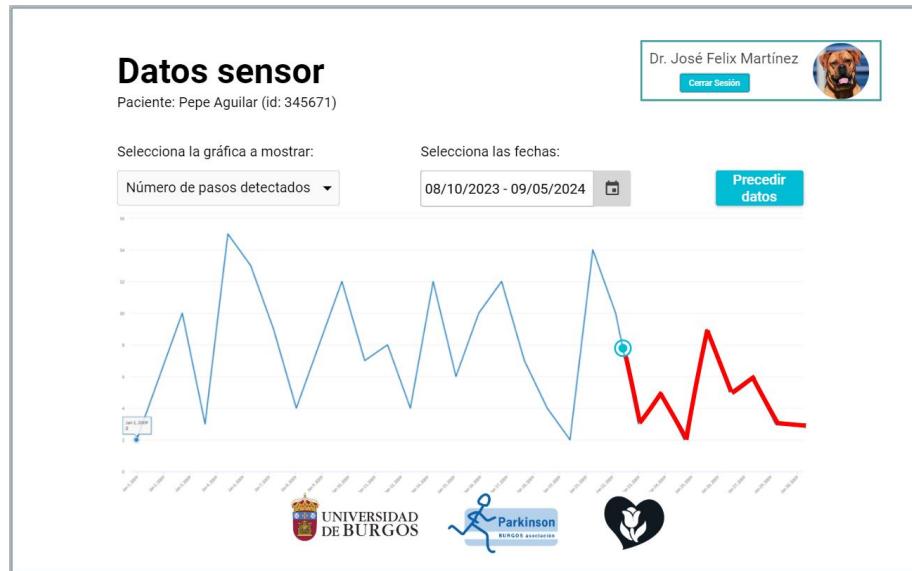


Figura C.17: Mockup - Predicción médica

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.

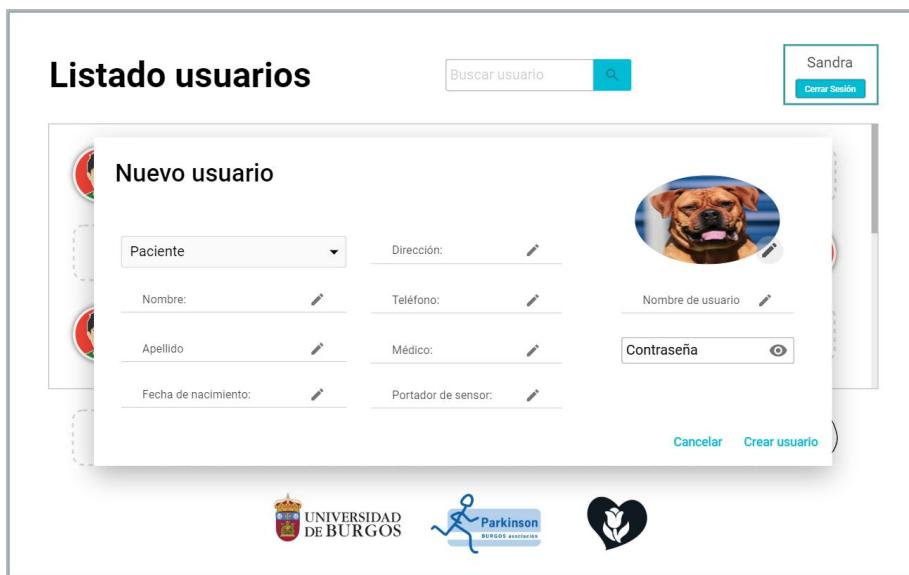


Figura C.22: *Mockup - Crear usuario*

## C.4. Diseño procedimental

## C.5. Diseño arquitectónico



## *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: (TODO: SE JUNTARÁN LAS RAMAS Y SE BORRARÁN LAS COSAS INNECESARIAS EN ALGÚN MOMENTO)

- doc: contiene la documentación del proyecto, formada por la memoria y anexos, tanto los primeros documentos realizados en Word como los documentos finales realizados en L<sup>A</sup>T<sub>E</sub>X y el resultado final del *mockup* de la aplicación.(TODO: BORRAR INNECESARIOS)
- App:(TODO: BORRAR INNECESARIOS)

---

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

- **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. Por ejemplo: **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 y **modelosbdd.py** contiene la estructura de la base de datos.
- **messages.pot** y **babel.cfg**: ficheros generados automáticamente necesarios para las traducciones. En concreto, **messages.pot** contiene todos los textos de la aplicación.
- Documento **.gitignore**: necesario para que GitHub ignore toda la carpeta *Static*, que contiene las fotos de los usuarios o logos y los vídeos y registros de los pacientes. De lo contrario, se superaría el espacio de almacenamiento del repositorio.
- *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, la barra de navegación y los logos. El resto de documentos heredan de él.
- *translations*: formado por carpetas utilizadas para la traducción de la aplicación a los diferentes idiomas disponibles (español, inglés y francés).
- **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. (TODO: BORRAR INNECESARIOS)

Además se ha utilizado una base de datos HeidiSQL con MariaDB (usuario root, contraseña maria) llamada parkinson. Está formada por cinco tablas:

- **administrador**: contiene los datos de los administradores de la aplicación (por ahora solo hay dos), 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 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 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.

## D.3. Manual del programador

Se describen a continuación las herramientas 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 una versión x. 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.

---

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

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

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

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

COM CREAR UN ENTORNO VIRTUAL Y CREAR DOCUMENTO CON LAS BIBLIOTECAS Y CON SUS VERSIONES PARA QUE EL PROGRAMADOR TENGA LA MISMA VERSION QUE YO EN TODO Y NADA FALLE.

Para ejecutar el proyecto, la desarrolladora utilizaba Visual Studio Code. El primer paso es activar el entorno virtual, ubicado en App/env/Scripts y que se activa mediante . activate.

Una vez activado (sale env al principio), se debe volver a la carpeta de app y ejecutar python app.py

Si al realizar cambios en el proyecto se crean nuevos textos que vayan a ser mostrados en la aplicación se debe actualizar los archivos para las traducciones. El texto que necesite internacionalizarse se debe escribir entre trans y endtrans de la siguiente manera: Posteriormente se debe ejecutar el siguiente comando: pybabel extract -F babel.cfg -o messages.pot . Que va a llenar el fichero messages.pot con el texto de la aplicación.

## D.5. Pruebas del sistema

En diferentes navegadores: (TABLAS CON PRECONDICIONES, RESULTADOS, ...) - probar inicio de sesión - probar cerrar sesión - probar restricciones de acceso según usuarios - probar creación de un usuario (administrador, médico o paciente) - probar edición de un usuario (administrador, médico o paciente) - probar eliminación de un usuario (administrador, médico o paciente) - probar subida de registros y vídeos a la base de datos - probar generación de gráficas - probar cambio de idioma

## *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(ENTRE COMILLAS, acceso).

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, Safari o Internet Explorer.

## **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 mediante el navegador.

La dirección que se ha de introducir en el navegador elegido para tener acceso a la aplicación web del proyecto es la siguiente: <https://tremortrack.com>

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

Una vez que el usuario accede al enlace del apartado anterior, tiene acceso a la página principal de la aplicación. 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://youtube....> El tiempo de visualización del tutorial para todos los tipos de usuario es de unos x minutos.

### **Funcionalidades básicas**

#### **Página principal**

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

Por otra parte, en la parte inferior de la página (así como en la parte inferior del resto de páginas que se explicarán a continuación) 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.

#### **Sobre nosotros**

Al pulsar sobre las letras de «Sobre nosotros» 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.

## Contacto

Al pulsar sobre las letras de «Contacto» en la barra de navegación, aparece el apartado de soporte para los usuarios de la aplicación. Se muestran los correos electrónicos de los creadores de la *app*, así como la ubicación en la que se les puede encontrar.

## 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. Esta opción de la barra de navegación está siempre visible por si se necesita cambiar el idioma en algún otro momento.

## Inicio de sesión

El inicio de sesión 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. 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 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.

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.

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 ".<sup>E</sup>ditar datos personales". 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 ".<sup>E</sup>ditar datos personales" para ocultarlo.

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

## Mostrar datos sensor

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 4 temas de gráficas:

- Parámetros de Bradicinesia: muestra una gráfica con la marcha media filtrada, la desviación estándar media de la marcha y el número de pasos considerados. Si se pulsa sobre la parte superior de la gráfica, donde aparecen estos apartados, se podrán mostrar y ocultar cada una de estas tablas. Además, 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 min: como su nombre indica muestra el estado motor, discinesia y bradicinesia del paciente en intervalos de 10 minutos.

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

Al elegir estos dos aspectos y dar al botón de «Mostrar datos», aparecerá una gráfica mostrando la evolución de los datos pedidos durante las fechas seleccionadas.

## Funcionalidades para médicos

### Página bienvenida médico

Si el usuario que inicia sesión es un médico aparecerá la siguiente ventana de bienvenida.

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.

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.

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 esta pantalla 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.

Si se pulsa sobre «Información personal» se despliegan la fecha de nacimiento, dirección y teléfono de dicho paciente, existiendo la posibilidad de editar dichos datos y guardar los cambios en la base de datos.

Si se pulsa sobre «Subir vídeo», aparece un formulario 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 mal estado del paciente y 4 un movimiento bien realizado).

Si se pulsa sobre «Subir datos sensor», aparece un formulario en el que se ha de subir el archivo csv que contenta los datos obtenidos del sensor de dicho paciente.

Los botones de «Mostrar vídeos» y «Mostrar datos sensor» redirigen al usuario a otras páginas que se describen a continuación.

### Mostrar vídeos

En esta pantalla aparecerán todos los vídeos que tenga el paciente seleccionado. Aparecen su título, de qué mano es y su fecha. Si ese paciente no tiene vídeos, se comunicará mediante un mensaje en pantalla. Si se pulsa sobre «Reproducir vídeo» aparecerá el vídeo, pudiéndose reproducir, parar o volver a ocultar. Si se pulsa sobre «Ver estadísticas» aparecerán ciertos datos obtenidos tras filtrar el vídeo mediante funciones creadas por el estudiante de Ingeniería Informática Catalin, en su TFG presentado el curso pasado. Entre estos datos se encuentran: x.

### Mostrar datos sensor

Si se pulsa sobre «Mostrar datos sensor», el médico tendrá acceso a la misma ventana en la que los pacientes pueden ver las gráficas que muestran

los datos recogidos por su sensor. Funcionalidad explicada en el apartado anterior de pacientes.

## **Funcionalidades para administradores**

### **Página bienvenida administrador**

Si el usuario que inicia sesión es un administrador aparecerá la siguiente ventana de bienvenida.

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.

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.

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 esta ventana es poder gestionar todos los usuarios de la aplicación. Existen tres tipos de usuarios: administradores, médicos y pacientes. 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. 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. Además los administradores pueden crear administradores nuevos introduciendo sus datos mediante un formulario que se despliega al pulsar sobre «Añadir Administrador».

#### **Tabla de médicos**

Al pulsar sobre «Médicos» se muestra una tabla que contiene a todos los médicos de la aplicación web. 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. Además los administradores pueden crear médicos nuevos introduciendo sus datos mediante un formulario que se despliega al pulsar sobre «Añadir Médico».

**Tabla de pacientes**

Al pulsar sobre «Pacientes» se muestra una tabla que contiene a todos los pacientes de la aplicación web. 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). Además los administradores pueden crear pacientes nuevos introduciendo sus datos mediante un formulario que se despliega al pulsar sobre «Añadir Paciente».



## *Apéndice F*

---

# **Anexo de sostenibilización curricular**

---

## **F.1. Introducción**

Este anexo incluirá una reflexión personal del alumnado sobre los aspectos de la sostenibilidad que se abordan en el trabajo. Se pueden incluir tantas subsecciones como sean necesarias con la intención de explicar las competencias de sostenibilidad adquiridas durante el alumnado y aplicadas al Trabajo de Fin de Grado.

Más información en el documento de la CRUE [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).

Este anexo tendrá una extensión comprendida entre 600 y 800 palabras.



---

## Bibliografía

---

- [1] Grupo Atico34. El consentimiento en el rgpd y lopd, 2020. [Internet; descargado 14-enero-2024].
- [2] Gabinete de Comunicación. La universidad de burgos y la asociación parkinson burgos intensifican su colaboración, 2023. [Internet; descargado 27-enero-2024].
- [3] EUR-Lex. Reglamento general de protección de datos (rgpd), 2022. [Internet; descargado 14-enero-2024].
- [4] Exact. ¿cómo calcular la rentabilidad de un proyecto?, 2022. [Internet; descargado 27-noviembre-2023].
- [5] Gestron. Cuánto paga una empresa por un trabajador, 2023. [Internet; descargado 27-noviembre-2023].
- [6] Asociación Parkinson Salamanca. El símbolo de la enfermedad de parkinson, 2023. [Internet; descargado 16-enero-2024].
- [7] Eva Zamora. ¿cómo se calcula la amortización de equipos informáticos?, 2022. [Internet; descargado 25-noviembre-2023].