



UNIVERSIDAD LOYOLA ANDALUCÍA
ESCUELA SUPERIOR DE INGENIERÍA
GRADO EN INGENIERÍA INFORMÁTICA Y
TECNOLOGÍAS VIRTUALES

TRABAJO FIN DE GRADO

Aplicación Web para el procesamiento de datos
de pulseras Fitbit

MANUAL TÉCNICO

Autor:

- Borja Sanz de Bremond O'Neill

Tutor:

- Antonio Manuel Durán Rosal

Junio, 2023

AGRADECIMIENTOS

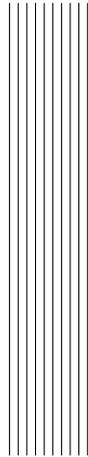
En primer lugar, quiero agradecer a mis padres María y Juan, por su apoyo incondicional y por haber estado ahí cuando los he necesitado.

Quiero agradecer también a mi director del Trabajo Fin de Grado Dr. Antonio Manuel Durán Rosal, por haber confiado en mí y haberme ayudado y guiado en la realización del presente trabajo.

Gracias además a los profesores que me han impartido clase a lo largo de estos cuatro años en la Universidad, proporcionándome y enriqueciéndome en los conocimientos adquiridos a nivel curricular y personal.

Por último, quiero agradecer a mis amigos por esta época que han compartido conmigo en momentos buenos y momentos difíciles.

Gracias.



ÍNDICE GENERAL

Índice General	v
Índice de Figuras	XIII
Índice de Tablas	xv
I Introducción	1
1. Introducción	3
2. Definición del Problema	7
2.1. Identificación del Problema Real	7
2.2. Identificación del Problema Técnico	8
2.2.1. Funcionamiento	9
2.2.2. Entorno	10
2.2.3. Vida esperada	10
2.2.4. Ciclo de Mantenimiento	10
2.2.5. Competencia	11
2.2.6. Aspecto Externo	11
2.2.7. Estandarización	11

2.2.8. Calidad y Fiabilidad	12
2.2.9. Programa de Tareas	13
2.2.10. Pruebas	14
2.2.11. Seguridad	15
3. Objetivos	17
3.1. Objetivos Generales	17
3.2. Objetivos Específicos	18
3.3. Objetivos Formativos	19
4. Antecedentes	21
4.1. Aplicaciones Similares	21
4.1.1. Fitbit App	21
4.1.2. Otras aplicaciones	22
4.2. Grupo de Investigación	22
4.3. Departamento de Psicología	22
4.4. Justificación del Proyecto	23
5. Restricciones	25
5.1. Restricciones Dato	25
5.2. Restricciones Estratégicas	26
6. Recursos	27
6.1. Recursos Humanos	27
6.2. Recursos Hardware	28
6.3. Recursos Software	28
II Análisis y Especificación de Requisitos	31
7. Descripción General del Problema	33
7.1. Datos Recopilados	33
7.2. Bloques de Funcionalidades	35
7.2.1. Gestión de la Web	35

ÍNDICE GENERAL

7.2.2. Gestión de Usuarios	36
7.2.3. Gestión de Datos	37
8. Especificación de Requisitos	41
8.1. Requisitos	41
8.1.1. Requisitos Generales	41
8.1.2. Requisitos de Usuario	42
8.1.3. Requisitos Funcionales	44
8.1.4. Requisitos No Funcionales	48
8.1.5. Requisitos de Negocio	48
8.1.6. Requisitos de Información	49
8.2. Validación de Requisitos	51
8.3. Matriz de Trazabilidad	53
8.4. Características de la Interfaz	56
9. Análisis Funcional	57
9.1. Casos de Uso: Gestión de la Web	59
9.1.1. Caso de Uso 01: Usuario accede a la Web	59
9.1.2. Caso de Uso 02: Usuario Registrado accede a la Web	60
9.1.3. Caso de Uso 03: Administrador accede al Panel de Control	61
9.2. Casos de Uso: Gestión de los Usuarios	62
9.2.1. Caso de Uso 04: Crear usuario registrado	62
9.2.2. Caso de Uso 05: Crear grupo	63
9.2.3. Caso de Uso 06: Buscar usuario registrado	64
9.2.4. Caso de Uso 07: Buscar grupo	65
9.2.5. Caso de Uso 08: Consultar usuario registrado	66
9.2.6. Caso de Uso 09: Consultar grupo	67
9.2.7. Caso de Uso 10: Modificar datos de un usuario registrado	68
9.2.8. Caso de Uso 11: Modificar permisos de un usuario registrado	69

9.2.9. Caso de Uso 12: Asignar grupos de un usuario registrado	70
9.2.10. Caso de Uso 13: Modificar grupos de un usuario registrado	71
9.2.11. Caso de Uso 14: Eliminar grupos de un usuario registrado	72
9.2.12. Caso de Uso 15: Modificar permisos de un grupo	73
9.2.13. Caso de Uso 16: Eliminar permisos de un usuario registrado	74
9.2.14. Caso de Uso 17: Eliminar usuario registrado . .	75
9.2.15. Caso de Uso 18: Eliminar permisos de un grupo	76
9.2.16. Caso de Uso 19: Eliminar grupo	77
9.2.17. Caso de Uso 20: Iniciar sesión	78
9.2.18. Caso de Uso 21: Cerrar sesión	79
9.3. Casos de Uso: Gestión de los Datos	80
9.3.1. Caso de Uso 22: Crear nuevo participante . . .	80
9.3.2. Caso de Uso 23: Buscar participante	81
9.3.3. Caso de Uso 24: Modificar datos de un participante	82
9.3.4. Caso de Uso 25: Visualizar datos de un participante	83
9.3.5. Caso de Uso 26: Exportar datos de un participante	84
9.3.6. Caso de Uso 27: Eliminar participante	85
9.4. Diagrama de Casos de Uso	86
9.4.1. Gestión de la Web	86
9.4.2. Gestión de los Usuarios	87
9.4.3. Gestión de los Datos	89
10. Análisis Dinámico del Sistema	91
10.1. Diagramas de Secuencia: Gestión de los Usuarios . . .	92
10.1.1. Diagrama de Secuencia CU-20: Iniciar sesión . .	92
10.1.2. Diagrama de Secuencia CU-21: Cerrar sesión . .	93
10.2. Diagramas de Secuencia: Gestión de los Datos	94

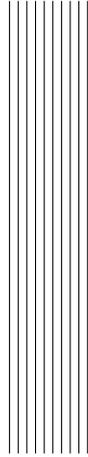
ÍNDICE GENERAL

10.2.1. Diagrama de Secuencia CU-22: Crear nuevo participante	94
10.2.2. Diagrama de Secuencia CU-23: Buscar participante	95
10.2.3. Diagrama de Secuencia CU-24: Modificar datos de un participante	96
10.2.4. Diagrama de Secuencia CU-25: Visualizar datos de un participante	97
10.2.5. Diagrama de Secuencia CU-26: Exportar datos de los participantes	98
10.2.6. Diagrama de Secuencia CU-27: Eliminar participante	99
11.Herramientas Utilizadas	101
11.1. Lenguaje de Programación Python	101
11.1.1. Base de la Web	102
11.1.2. Lectura de Datos	102
11.1.3. Manejo de Datos	102
11.1.4. Base de datos	103
11.1.5. Exportación de Datos	103
11.1.6. Versiones	104
11.2. Web Framework Django	104
11.3. Base de Datos MongoDB	105
11.3.1. MongoDB Compass	105
III Diseño del Sistema	107
12.Diseño de los Datos	109
12.1. Análisis de la Información	109
12.2. Modelo No Relacional	115
13.Diseño Arquitectónico	117
13.1. Patrón Modelo-Vista-Controlador	117

14. Diseño de Clases	119
14.1. Clase VariablesSum	120
14.2. Clase Variable	122
15. Diseño de la Interfaz	125
15.1. Vistas	125
15.1.1. Vista <i>Home</i>	126
15.1.2. Vista <i>Login</i>	127
15.1.3. Vista <i>Logout</i>	127
15.1.4. Vista <i>UploadData</i>	128
15.1.5. Vista <i>LoadData</i>	130
15.1.6. Vista <i>ModifyData</i>	132
15.1.7. Vista <i>ExportData</i>	134
15.1.8. Vista <i>DeleteData</i>	135
15.2. Mapa de Navegación	137
15.2.1. Usuario	137
15.2.2. Usuario Registrado	138
15.2.3. Administrador	139
IV Pruebas	141
16. Pruebas	143
16.1. Pruebas del Sistema	144
16.1.1. Instalación e Inicialización	144
16.1.2. Creación de la base de datos	144
16.1.3. Conexión con la base de datos	145
16.2. Pruebas Funcionales	146
16.2.1. Procesamiento de los datos	146
16.2.2. Creación de nuevos participantes	147
16.2.3. Modificación de los datos de los participantes	147
16.2.4. Visualización de los datos	148
16.2.5. Exportación de los datos	148

ÍNDICE GENERAL

16.2.6. Eliminación de un participante	149
16.3. Pruebas de la Interfaz	149
16.3.1. Accesos	150
16.3.2. Navegación	150
16.3.3. Visualización de los Datos	151
V Conclusiones	153
17. Conclusiones	155
17.1. Objetivos Planteados	155
17.1.1. Objetivos específicos	155
17.1.2. Objetivos formativos	157
17.2. Fase de Pruebas	158
17.3. Conclusiones Personales	158
18. Futuras Mejoras	159
Bibliografía	161
VI Apéndices	163
A. Manual de Instalación	165
A.1. Instalación del Software	165
A.1.1. Lenguaje de Programación y Librerías	165
A.1.2. Entorno de Desarrollo	167
A.1.3. Base de Datos	167
A.2. Creación de la Base de Datos	167
A.3. Página Web	168
A.3.1. Administrador	169
A.3.2. Encender y Apagar	169

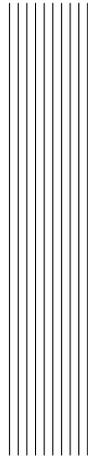


ÍNDICE DE FIGURAS

1.1. Porcentaje de personas con obesidad, por sexo según comunidad autónoma [2]	4
9.1. Diagrama Casos de Uso: Gestión de la Web	86
9.2. Diagrama Casos de Uso: Gestión de los Usuarios	87
9.3. Diagrama Casos de Uso: Gestión de los Grupos	88
9.4. Diagrama Casos de Uso: Gestión de los Usuarios-Grupos	88
9.5. Diagrama Casos de Uso: Gestión de los Datos	89
10.1. Diagrama de Secuencia CU-20	92
10.2. Diagrama de Secuencia CU-21	93
10.3. Diagrama de Secuencia CU-22	94
10.4. Diagrama de Secuencia CU-23	95
10.5. Diagrama de Secuencia CU-24	96
10.6. Diagrama de Secuencia CU-25	97
10.7. Diagrama de Secuencia CU-26	98
10.8. Diagrama de Secuencia CU-27	99
13.1. Patrón <i>Modelo-Vista-Controlador</i>	118
14.1. Diagrama de clases de la clase VariablesSum	121

ÍNDICE DE FIGURAS

14.2. Diagrama de clases de las clases Variable	123
15.1. Vista <i>Home</i>	126
15.2. Vista <i>Login</i>	127
15.3. Vista <i>UploadData</i>	128
15.4. Vista <i>SuccessUpload</i>	129
15.5. Vista <i>LoadData</i>	130
15.6. Vista <i>VisualizeData</i>	131
15.7. Vista <i>ModifyData</i>	132
15.8. Vista <i>SuccessModify</i>	133
15.9. Vista <i>ExportData</i>	134
15.10. Vista <i>DeleteData</i>	135
15.11. Vista <i>Confirmation</i>	136
15.12. Mapa navegación: Usuario	137
15.13. Mapa navegación: Usuario Registrado	138
15.14. Mapa navegación: Administrador	139



ÍNDICE DE TABLAS

4.1. Comparación de los antecedentes	24
8.1. RU/RF: Gestión de la Web	51
8.2. RU/RF: Gestión de los Usuarios	52
8.3. RU/RF: Gestión de los Datos	53
8.4. Matriz de trazabilidad: Requisitos generales-funcionales 1	53
8.5. Matriz de trazabilidad: Requisitos generales-funcionales 2	54
8.6. Matriz de trazabilidad: Requisitos generales-funcionales 3	55
8.7. Matriz de trazabilidad: Requisitos generales-otros . . .	55
14.1. Especificación de la clase VariablesSum	120
14.2. Especificación de las clases Variable	122
A.1. Librerías y versiones	166

Parte I

Introducción



1 Introducción

Normalmente, la obesidad está relacionada con la falta de una alimentación saludable, como la ingesta de comida de alto contenido calórico, donde las personas que sufren de esta condición padecen diversos problemas físicos y psicológicos, afectando en sus vidas diarias de forma negativa.

Las causas que provocan esta situación pueden ser, tal como se ha comentado, la falta de una alimentación saludable, el entorno familiar y social en el que se encuentra la persona, su situación financiera, la falta de actividad física o el metabolismo.

Estas situaciones incrementan la posibilidad de contraer enfermedades que afectan por completo al individuo, inutilizando algunos órganos o procesos internos, como la circulación de la sangre por la grasa acumulada. Provocando en el peor de los casos, la muerte.

En España, los casos por obesidad ha ido incrementando en los últimos años, afectando a una gran cantidad de jóvenes y adultos. Llegando a crear problemas tanto de salud física como psicológica, que los afectados no saben cómo manejar ni controlar. Algunos de los problemas psicológicos a los que se enfrentan las personas con sobrepeso u obesidad son la depresión, frustración, ansiedad o estrés, entre otros muchos inconvenientes.

Los problemas físicos a los que se puede enfrentar una persona con obesidad son la diabetes, presión alta, apnea del sueño o cáncer, entre otros. Además, las tareas cotidianas se vuelven complicadas y tediosas de realizar.

Algunas de las formas para luchar contra la obesidad serían la realización de actividades físicas periódicas, reducción de la ingesta de comida y la mejora de la calidad nutricional.

Como se puede observar en la Figura 1.1, el porcentaje de adultos españoles con obesidad está sobre el 16% de la población. Esto es un grave problema, ya que, tal como se ha indicado previamente, los problemas que padecen dichas personas son graves y han de ser solventados.

	18 y más años			*Entre 2 y 17 años		
	Ambos sexos	Mujeres	Hombres	Ambos sexos	Niñas	Niños
España	16	15.5	16.5	10.3	10.2	10.4
Andalucía	19.7	19.8	19.5	12.5	14.4	10.6
Aragón	15.7	15.1	16.4	2.5	3.5	1.6
Asturias, Principado	19.1	17.6	20.8	7.9	6.4	9.3
Baleares, Illes	16.3	14.2	18.4	10	12.6	7.8
Canarias	17.5	17.9	17.1	10.4	9.9	10.9
Cantabria	13.7	11.9	15.5	2.6	4	0.9
Castilla y León	16.8	16.9	16.7	5.6	7.5	4
Castilla-La Mancha	14.4	11.3	17.4	10.1	8.6	11.7
Cataluña	14.2	15.2	13.2	12.7	10.1	14.8
Comunitat Valenciana	16.7	14.2	19.2	12	11.2	12.7
Extremadura	12.8	11.4	14.3	10.4	7.8	12.7
Galicia	16.5	16.2	16.8	6.2	4.6	7.6
Madrid, Comunidad de	13.4	13	13.9	7.5	9.3	5.8
Murcia, Región de	19.1	18.6	19.5	14.2	10.1	18.3
Navarra, C. Foral de	13.2	13	13.5	1.4	1.4	1.5
País Vasco	13.5	13.5	13.5	11.1	9.4	12.8
Rioja, La	15.8	14.3	17.3	8.3	12.1	5.2
Ceuta	13.5	13.9	13.1	18	18.4	17.7
Melilla	6.8	8.5	5.1	11.5	12.8	10.5

Figura 1.1: Porcentaje de personas con obesidad, por sexo según comunidad autónoma [2]

1. Introducción

En el caso de los estudiantes, la obesidad puede provocar que no rindan en los estudios, que abandonen o, incluso, que puedan experimentar rechazo social. Además, pueden sufrir de enfermedades crónicas como cáncer o diabetes en edades más adultas.

Desde el **Departamento de Psicología**, de la Universidad Loyola Andalucía, han solicitado una herramienta para poder monitorizar la actividad física de los alumnos. Para realizar el seguimiento de los mismos se ha hecho uso de las pulseras de actividad Fitbit [4], que recogen diversos datos sobre la actividad física de los participantes. Concretamente, estas pulseras son capaces de registrar una gran variedad de parámetros de sus portadores: cantidad de calorías quemadas durante el día, los pasos y distancia realizados, la frecuencia cardíaca o la actividad del sueño, entre otros. Sin embargo, el principal problema de la aplicación oficial, es que la visualización de los mismos es muy limitada.

Este Trabajo Fin de Grado pretende proporcionar una herramienta que procesará y guardará los datos recogidos por estas pulseras para poder visualizarlos y estudiarlos de una forma más clara y utilizable por los usuarios finales, sirviendo también como una herramienta en estudios futuros. Por ejemplo, se podrán visualizar los cambios de comportamiento que sufren los alumnos al recibir charlas motivacionales y, así, poder centrarse en los aspectos esenciales para mejorar la eficiencia de los métodos utilizados.



2 Definición del Problema

En esta sección se definen el problema real y el problema técnico, identificando las necesidades y requisitos de la herramienta a desarrollar, para de esta forma definir más adelante los objetivos del Trabajo Fin de Grado (TFG).

2.1. Identificación del Problema Real

El problema real consiste en definir la necesidad tal y como lo ve el usuario final, en este caso, sería el Departamento de Psicología de la Universidad Loyola Andalucía.

El Departamento de Psicología necesita una herramienta web para poder manejar los datos recogidos de los estudiantes que han participado en el seguimiento.

Estos datos son recogidos por las pulseras Fitbit. Sin embargo, el formato de los datos generados no es entendible por el usuario final. Para ello, la herramienta recogerá los datos de los diferentes participantes, los procesará y permitirá su descarga y visualización de una forma clara y sencilla.

Estos datos generados podrán tener diferentes formatos, pudiendo ser descargados sobre un participante en específico o descargar los

2.2. Identificación del Problema Técnico

datos de todos participantes en un mismo fichero.

Por otra parte, la herramienta permitirá el uso de sus funciones a aquellos usuarios que hayan sido registrados en el sistema y hayan iniciado sesión. Ya que, al tratar con información delicada, hay que mantener un control sobre quién puede acceder a los datos.

La herramienta a desarrollar debe cumplir ciertas características y necesidades para que se diferencie de otras aplicaciones que manejan datos de las pulseras Fitbit o similares:

- La aplicación debe ser capaz de recoger los ficheros con los datos de los participantes, para su posterior procesamiento.
- Los datos, al ser introducidos en el sistema, deberán ser procesados para recolectar solo aquella información relevante durante el periodo de seguimiento.
- Los datos procesados deben ser almacenados en un servidor desde el cual se podrá acceder realizando consultas y modificaciones.
- Los datos almacenados podrán ser exportados en ficheros para su posterior visualización.
- La interfaz gráfica debe ser intuitiva, visualmente atractiva e interactiva con los usuarios, mostrando mensajes o avisos cuando sea necesario.
- El diseño debe emplear el patrón de arquitectura *Modelo-Vista-Controlador*.

2.2. Identificación del Problema Técnico

El problema técnico consiste en definir la necesidad desde el punto de vista del desarrollador, ofreciendo más información acerca de la herramienta a desarrollar. Para ello, se hará uso de la metodología PDS (*Product Design Specification*), para que el problema quede definido de una forma estructurada y técnica.

2. Definición del Problema

2.2.1. Funcionamiento

La aplicación web constará de diferentes funcionalidades y utilidades que permitirán el manejo de la información recogida de las pulseras Fitbit. Las funcionalidades relacionadas con el tratamiento de los datos recogidos son:

- **Subida de los Datos:** Los datos recogidos de las pulseras Fitbit no poseen un formato de fácil entendimiento, es por ello, que se necesita de un proceso para recoger los datos relevantes para que puedan ser visualizados de una forma sencilla. Estos datos serán guardados en una base de datos.
- **Modificación de los Datos:** Esta funcionalidad permite la modificación de los datos ya subidos con nuevos datos de las pulseras Fitbit.
- **Eliminar Datos:** Se podrán eliminar todos los datos de un usuario en concreto de la base de datos.

Las funcionalidades relacionadas con la visualización de los datos tratados son las siguientes:

- **Visualización de los Datos:** Página donde se pueden visualizar los datos procesados mediante el uso de gráficas.
- **Exportar Datos:** En caso de que se quieran descargar los datos de uno o más usuarios, se podrá exportar los datos ya procesados en los formatos *.xls* o *.csv*.

Además, la aplicación tendrá otro tipo de funcionalidades relacionadas con el manejo de los usuarios, como es la creación, modificación o eliminación de usuarios registrados. Estos usuarios registrados tendrán permisos para acceder a las diferentes funcionalidades de la aplicación una vez hayan iniciado sesión.

2.2. Identificación del Problema Técnico

2.2.2. Entorno

Los aspectos externos que hay que tener en cuenta durante el desarrollo de la aplicación web son los siguientes:

- **Entorno Software:** Para la realización de este proyecto, se requiere de Django [3], un framework de desarrollo web que utiliza como lenguaje de programación Python [9]. Además, se hará uso de la base de datos no relacional MongoDB [7] para el almacenamiento de la información.
- **Entorno Hardware:** La aplicación web podrá ser instalada tanto en ordenadores personales como en un servidor, sin necesidad de que los dispositivos tengan grandes prestaciones.
- **Entorno de Usuarios:** La aplicación estará diseñada para ser manejada sin necesidad de conocimientos de programación ni manejo de datos, siendo intuitiva y de fácil uso.

2.2.3. Vida esperada

Se considera que la vida esperada que tendrá la aplicación será amplia y a determinar por el usuario final, que hará uso de ella. Ya que es un proyecto real, con aplicaciones reales y con un objetivo claro.

Además, de ser necesario, la aplicación podrá ser mejorada y ampliada con nuevas funcionalidades, pudiendo ser la base de proyectos futuros.

2.2.4. Ciclo de Mantenimiento

Dentro del marco en el que se encuentra este proyecto, es complicado establecer un ciclo de mantenimiento.

Aun así, se espera que el encargado de la aplicación web, tras la realización del TFG, realice comprobaciones constantes para el correcto funcionamiento de la misma.

2. Definición del Problema

2.2.5. Competencia

Este proyecto es una solución a un problema real que no se ha intentado solucionar hasta la fecha, por lo que no existe un competidor directo. Aunque existan aplicaciones que recojan datos o realicen ciertos procesamientos sobre ellos, esta aplicación tiene como objetivo satisfacer las necesidades que tiene el Departamento de Psicología de la Universidad Loyola Andalucía.

Además, los datos proporcionados por las pulseras tienen un formato específico, por lo que el procesamiento interno que se realiza es único al problema establecido.

Todo lo anterior hace que la única competencia pueda ser la aplicación oficial Fitbit, que como se ha indicado, tiene una funcionalidad muy limitada.

2.2.6. Aspecto Externo

La herramienta será accesible a través de la página web, donde los usuarios podrán visitarla. Solo los usuarios registrados podrán acceder a las diferentes funcionalidades. La interfaz de la página web será intuitiva, de esta forma el usuario final tendrá una mejor experiencia.

Respecto a la documentación, se proporcionará un manual de usuario y un manual de instalación, donde el usuario final podrá manejar y desplegar la herramienta de forma correcta y sencilla.

2.2.7. Estandarización

Para el desarrollo de la aplicación web se pretende seguir los estándares de programación. La herramienta hace uso de Python, por lo que se seguirán las buenas prácticas de nomenclatura e *indentación*, entre otros.

Como se hace uso de los lenguajes HTML, CSS y JS, se seguirá la estandarización W3C (*World Wide Web Consortium*) [10]. Algunos

2.2. Identificación del Problema Técnico

de los principios que se van a seguir son los siguientes:

- **Accesibilidad:** Es el contenido web que será procesado por el navegador.
- **Interfaz de usuario:** Es la interacción del usuario con la herramienta, permitiendo un control efectivo.
- **Navegación:** Es el proceso de navegar por la página web, siendo intuitiva y que no desoriente a los usuarios.

Las herramientas que se van a utilizar deberán seguir unos estándares para que el código sea legible y, así, sea más fácil mantener:

- **Lenguaje de Programación:** Las funcionalidades que manejan los datos se programarán en Python, un lenguaje de programación que ofrece, junto a librerías instaladas, un manejo fácil de grandes cantidades de datos. Se seguirán los principios de diseño de uniformidad, consistencia y legibilidad.
- **Almacenamiento:** Todos los datos procesados de los participantes así como la información necesaria para el correcto funcionamiento de la web, se almacenarán en una base de datos no relacional de MongoDB.

2.2.8. Calidad y Fiabilidad

Ambos aspectos son muy importantes a la hora de entregar la herramienta. Se desea crear una aplicación web donde los datos procesados no son modificados por error y la experiencia ofrecida al usuario final es satisfactoria.

Además, se realizarán unas pruebas de calidad para comprobar que el producto final no tiene ningún tipo de fallo y de esta forma, asegurar la integridad de la página web.

2. Definición del Problema

La aplicación manejará los errores y avisos que puedan ocurrir durante los diferentes procesos que existen, evitando que el usuario que la está utilizando, sufra de desconexiones u otros errores graves.

2.2.9. Programa de Tareas

A la hora de desarrollar la herramienta, se van a realizar diferentes fases descomponiendo el proyecto en partes más pequeñas. Las fases no serán desarrolladas de forma aislada y secuencial, sino que se seguirá la filosofía de las metodologías ágiles incrementales, por lo que habrá revisiones periódicas de objetivos y trabajo realizado hasta el momento.

Fase de Estudio y Análisis de Datos: Esta primera fase consiste en realizar un estudio previo de las necesidades del proyecto, recopilando toda la información necesaria para su desarrollo. Este estudio consistirá en los siguientes conceptos y métodos:

- Estudio de la estructura y del contenido de los ficheros con los datos de las pulseras Fitbit, determinando cuáles serán los archivos necesarios a tener en cuenta.
- Estudio del manejo del framework de desarrollo web Django.
- Estudio de las diferentes librerías de Python para el manejo y procesamiento de grandes cantidades de datos.
- Estudio del gestor de base de datos MongoDB para el almacenamiento y gestión de los datos procesados por la herramienta.
- Estudio del comportamiento de la página web, las funcionalidades que esta tendrá y cómo realizará las tareas.

Fase de Diseño: En esta fase habrá que realizar un estudio de la estructura que tendrá la página web, como estarán almacenados los datos y cómo se cumplirán los requisitos planteados. Además, se

2.2. Identificación del Problema Técnico

realizará el diseño de la interfaz para conocer cómo los usuarios interactuarán con el sistema.

Fase de Implementación: Se implementará los diferentes componentes y módulos planteados durante las fases anteriores. Se desarrollará un producto optimizado y con la menor cantidad de errores, con el objetivo de controlar las diferentes casuísticas que puedan ocurrir y ofreciendo la mejor experiencia a los usuarios.

Fase de Pruebas: Mientras se vaya desarrollando la herramienta y cuando se vayan implementando los diferentes módulos y funcionalidades, se realizarán pruebas para comprobar la integridad de la página web y verificar el cumplimiento de los objetivos y requisitos planteados.

Fase de Documentación: Registro de las tareas realizadas y el proceso seguido para el desarrollo del proyecto. Esta documentación realizada estará compuesta por un Manual Técnico y un Manual de Usuario. Se realizará de forma transversal al desarrollo del proyecto.

2.2.10. Pruebas

Para asegurar la integridad y el correcto funcionamiento de la página web, se realizarán una serie de pruebas, de esta forma se verificarán y validarán los objetivos planteados, y los requisitos extraídos en fases previas. Estas pruebas consistirán en buscar todos los errores que puedan existir para su corrección:

- **Pruebas Funcionales:** Sirve para probar las características críticas de las funcionalidades y así garantizar que no hay problemas, centrándose en los requisitos planteados.
- **Pruebas Unitarias:** Son las pruebas a más bajo nivel, centrándose en las funciones y métodos planteados.
- **Pruebas de Integración:** Se prueban diferentes módulos de la aplicación y de esta forma comprobar el funcionamiento de un conjunto de funciones que trabajan a la vez.

2. Definición del Problema

2.2.11. Seguridad

Este apartado es uno de los más importantes del TFG. Como la aplicación tendrá un sistema de autenticación de usuarios, las credenciales estarán encriptadas. Además, sólo las personas autorizadas tendrán acceso a los datos de los participantes. Todo esto será posible mediante la asignación de roles a los visitantes de la aplicación web.



3 Objetivos

En este capítulo se describen los objetivos finales de este Trabajo Fin de Grado (TFG) y se definen las diferentes funcionalidades y características que ofrece la aplicación web.

3.1. Objetivos Generales

El objetivo principal de este proyecto consiste en la recogida, procesamiento y visualización de los datos obtenidos de las pulseras de actividad Fitbit, entregadas a alumnos de la Universidad Loyola Andalucía durante un periodo determinado.

Los datos recogidos de los participantes serán procesados para eliminar información irrelevante, y de esta forma, los cambios que tienen en su comportamiento podrán ser visualizados y estudiados de forma sencilla tras recibir unos seminarios de motivación.

Con el estudio de estos datos, se podrá realizar un seguimiento más personalizado para cada participante y enfocar los seminarios en los puntos claves para causar efecto. Los datos recogidos se centran en la actividad física del participante a lo largo de los días.

3.2. Objetivos Específicos

Dependiendo del tipo de usuario que hace uso de la aplicación web, las funcionalidades a las que pueden acceder y las acciones que pueden realizar, los objetivos específicos varían.

Los **usuarios** que accedan a la página web podrán navegar por aquellas funcionalidades que no estén relacionadas con los datos recogidos. En este caso, solo podrán visitar la página principal donde aparecerá información sobre el proyecto y la funcionalidad de iniciar sesión.

Los **usuarios registrados** tendrán acceso a las siguientes funcionalidades:

- Subida de los datos.
- Modificación de los datos.
- Eliminar datos.
- Visualización de los datos.
- Exportar datos.
- Iniciar y cerrar sesión.

Los **administradores** tendrán acceso a las mismas funcionalidades que los usuarios registrados, pero además, tendrán la capacidad de manejar la página web, los permisos que tendrán los usuarios registrados y sus datos. Algunos de las acciones que podrán realizar son:

- Crear usuarios registrados.
- Eliminar usuarios registrados.
- Consultar la información de los usuarios registrados.
- Modificar los datos de los usuarios registrados.
- Modificar los permisos de los usuarios registrados.

3. Objetivos

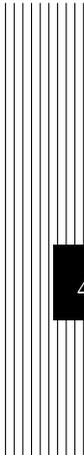
Además, los administradores podrán agrupar a los usuarios registrados en grupos, que otorgarán a los usuarios registrados los permisos que tengan estos grupos. El administrador podrá manejar los diferentes grupos creados:

- Crear grupos.
- Eliminar grupos.
- Modificar los permisos e información de los grupos.
- Agrupar a los usuarios registrados en grupos con diferentes permisos.

3.3. Objetivos Formativos

Estos objetivos son aquellos relacionados con el aprendizaje que se van a obtener con la realización de este TFG:

- Estudiar el framework de desarrollo web Django.
- Entender y comprender el patrón de arquitectura *Modelo-Vista-Controlador*.
- Estudiar conceptos relacionados con la actividad física y los datos recogidos de pulseras de actividad Fitbit.
- Analizar y estudiar métodos de procesamiento de grandes cantidades de datos.
- Reforzar y ampliar mis habilidades y conocimientos sobre la creación de herramientas web.
- Aprender del sistema de creación de textos \LaTeX .



4 Antecedentes

A lo largo de este capítulo se hará un repaso por todos aquellos estudios y conocimientos previos que asientan las bases de este Trabajo Fin de Grado (TFG).

4.1. Aplicaciones Similares

Para la realización del proyecto, se ha realizado un estudio sobre posibles aplicaciones similares que realizan las tareas que necesita este trabajo.

4.1.1. Fitbit App

Fitbit cuenta con una aplicación móvil [5] que permite enlazar una pulsera con un dispositivo móvil, permitiendo la visualización de los datos recopilados, mostrando resúmenes y estadísticas de la actividad física y mental del usuario, y ofreciendo *feedback* sobre la actividad realizada.

Aunque recopile información parecida a la que se necesita, esta aplicación no permite descargar las estadísticas y no satisface las necesidades del cliente final de este TFG.

4.2. Grupo de Investigación

Además, se necesita de una herramienta que permita acceder a cualquiera de los datos recopilados de los participantes, no solo con la pulsera que esté enlazada a la aplicación.

4.1.2. Otras aplicaciones

Existen otras aplicaciones similares a la ya comentada que realizan tareas y procesamiento sobre los datos. Sin embargo, estas aplicaciones no manejan datos con la estructura que se presenta en este TFG, y cuando presentan una estructura similar (pulseras Xiaomi o Huawei, entre otras) tienen los mismos inconvenientes que los comentados previamente.

4.2. Grupo de Investigación

Como antecedente, cabe destacar la experiencia de los integrantes del grupo de investigación **Métodos de Investigación Cuantitativa y Aplicaciones** (MICA) en el desarrollo de sistemas software. Además, este TFG está encuadrado en sus líneas de investigación:

- Desarrollo de metodologías en Deep y Machine Learning: el proyecto sirve de base para futuros trabajos de predicción, clasificación y regresión haciendo uso de los datos preprocesados.
- Aplicaciones de nuevas metodologías cuantitativas a la resolución de problemas del entorno social y empresarial: el proyecto va a tener un impacto social en colaboración con el Departamento de Psicología, que se explica a continuación.

4.3. Departamento de Psicología

Tal y como se ha comentado anteriormente, este TFG se va a desarrollar para satisfacer una necesidad del Departamento de Psicología

4. Antecedentes

de la Universidad Loyola Andalucía. Este Departamento tiene experiencia en distintos ámbitos de esta ciencia. No obstante, el relacionado con este trabajo es el estudio de procesos conductuales y cerebrales que podrían ayudar a promover estos estilos de vida saludables.

4.4. Justificación del Proyecto

La realización de este TFG está justificada por la necesidad de crear una aplicación diferente a las ya existentes y por las funcionalidades requeridas.

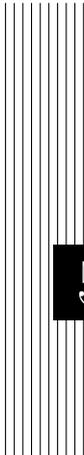
Al no existir una aplicación o herramienta que cumpla los requisitos de este proyecto, se creará una aplicación web que cumpla y satisfaga las necesidades del usuario final.

A continuación, la Tabla 4.1 compara los requisitos principales del TFG con la aplicación ya mencionada. De esta forma, se puede entender la necesidad que existe de realizar este proyecto teniendo en cuenta la escasez de aplicaciones relacionadas, y la experiencia de las personas implicadas.

4.4. Justificación del Proyecto

	Fitbit App	TFG
Subida de datos al sistema	×	✓
Modificación de los datos de los participantes en el sistema	×	✓
Eliminar datos de los participantes en el sistema	×	✓
Almacenamiento selectivo de la información en el sistema	×	✓
Visualizar los datos de los participantes	✓	✓
Visualizar los datos de los participantes de forma detallada	×	✓
Exportar los datos procesados de los participantes	×	✓
Permitir acceso a los datos a diferentes usuarios	×	✓
Procesamiento de la información recopilada de los participantes	✓	✓
Procesamiento de la información recopilada de los participantes de forma selectiva	×	✓

Tabla 4.1: Comparación de los antecedentes



5 Restricciones

A continuación, se expondrán todas aquellas restricciones existentes en el ámbito del diseño y que condicionan la elección de una u otra alternativa a la hora de elegir entre distintas opciones en fases posteriores del Trabajo Fin de Grado (TFG).

5.1. Restricciones Dato

Este tipo de restricciones no pueden modificarse y vienen impuestas por la naturaleza del proyecto. Son las normas o limitaciones del sistema que vienen dadas y son de debido cumplimiento:

- **Restricciones Humanas:** El presente TFG será llevado a cabo por el autor Borja Sanz de Bremond O'Neill y supervisado por el tutor del proyecto Dr. Antonio Manuel Durán Rosal. El experto en el dominio de aplicación y enlace con el Departamento de Psicología será el Dr. Cristian Manuel Olarte Sánchez.
- **Restricciones Económicas:** La herramienta a desarrollar hará uso de recursos de software libre y gratuitos.
- **Restricciones Temporales:** La entrega del TFG, tal como se había acordado, será para la convocatoria de Junio de 2023.

- **Restricciones Hardware:** Los recursos hardware utilizados para el desarrollo del proyecto serán aquellos que tenga a disposición el autor.

5.2. Restricciones Estratégicas

Este tipo de restricciones consisten en limitaciones que pueden ser modificables y que influyen en el desarrollo del proyecto:

- **Sistema Operativo:** Se hará uso de Windows por ser el sistema operativo disponible para el autor del TFG. Si bien es cierto que Windows no es un sistema operativo gratuito, el autor tiene licencia y no supone un coste extra en el desarrollo del mismo.
- **Lenguaje de Programación:** Se utilizará Django junto a su lenguaje de programación Python para programar las diferentes funcionalidades que tendrá el proyecto, por ser un lenguaje de programación de alto nivel, multiparadigma, de propósito general, con extensa documentación y una gran comunidad de desarrollo.
- **Entorno de Desarrollo:** Se utilizará el entorno de desarrollo Visual Studio Code [6], un editor de código desarrollado por Microsoft desde donde se puede depurar y compilar código, así como instalar extensiones y funcionalidades que facilitan el trabajo del desarrollador.
- **Base de Datos:** Se utilizará la base de datos no relacional MongoDB, por la necesidad de guardar los datos con una estructura determinada, donde los documentos almacenados no estarán relacionados con otros documentos.



6 Recursos

A continuación se presentan los distintos recursos humanos, hardware y software con los que se ha realizado el presente Trabajo Fin de Grado (TFG).

6.1. Recursos Humanos

El equipo involucrado en el desarrollo de este TFG está constituido por las siguientes personas:

- **Borja Sanz de Bremond O'Neill:** Alumno del Grado en *Ingeniería Informática y Tecnologías Virtuales* de la Escuela Superior de Ingeniería en la Universidad Loyola Andalucía. Encargado del estudio, diseño, implementación y documentación del TFG.
- **Antonio M. Durán Rosal:** Profesor Adjunto del Departamento de Métodos Cuantitativos de la Universidad Loyola Andalucía e investigador del Grupo de Investigación MICA de la propia Universidad. Encargado de dar soporte técnico y teórico para las tareas de análisis, diseño y programación, así como la dirección y supervisión de la consecución de los objetivos del TFG.

- **Cristian Manuel Olarte Sánchez:** Investigador del Departamento de Psicología de la Universidad Loyola Andalucía. Es el usuario final de la aplicación que ha solicitado la realización de este proyecto, y el encargado de dar soporte en las decisiones relacionadas con el dominio de aplicación del proyecto.

6.2. Recursos Hardware

Para la realización de este proyecto se ha utilizado el portátil personal del alumno, donde se ha instalado una base de datos local y las herramientas necesarias para el proyecto. Además, desde este mismo dispositivo se han desarrollado las pruebas. Las especificaciones del ordenador son las siguientes:

- **AMD Ryzen 7 4800HS 2.90 GHz** (64 bits y 8 núcleos de proceso).
- **16 GB DDR4 3200Mhz** de memoria RAM.
- Disco duro **SDD** de 512 GB.
- Tarjeta gráfica **NVIDIA GeForce GTX 1660 Ti with Max-Q Design**.

6.3. Recursos Software

El sistema operativo del portátil es **Microsoft Windows 10 Home**. Todas las aplicaciones y herramientas utilizadas durante el desarrollo han sido instaladas en este sistema operativo.

En cuanto a los lenguajes y herramientas utilizados durante la fase de programación y desarrollo del proyecto han sido las siguientes:

- **Python 3.9.5:** Lenguaje de programación utilizado en el lado del servidor de la aplicación.

6. Recursos

- **HTML:** Lenguaje de marcado para la estructuración de la página web.
- **CSS:** Lenguaje de diseño gráfico para la presentación de la página web.
- **JavaScript:** Lenguaje de programación encargado de hacer dinámica la aplicación ejecutado en el lado del cliente.
- **Django 4.1.3:** Framework de desarrollo web.
- **MongoDB Compass 1.36.4:** Herramienta para el manejo de la base de datos.
- **Visual Studio Code 1.78.4:** Entorno de texto para la implementación del código.

Además de los lenguajes y herramientas mencionados, se han descargado diferentes librerías en Python para manejar grandes cantidades de datos o conectar la base de datos con Django, entre otras cosas.

Para la documentación se han utilizado las siguientes herramientas:

- **Overleaf:** Entorno de desarrollo para la elaboración de una documentación en $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$.

Parte II

Análisis y Especificación de Requisitos



7 Descripción General del Problema

La idea general del Trabajo Fin de Grado (TFG) es la creación de una aplicación web, que permita al Departamento de Psicología de la Universidad Loyola Andalucía manejar y visualizar los datos recopilados por las pulseras Fitbit de los participantes.

A continuación, se va a proceder a describir los datos que son recopilados y las partes o bloques más importantes de la aplicación web, que ayudan al correcto funcionamiento de esta.

7.1. Datos Recopilados

Hay diferentes tipos de datos, que dependiendo su naturaleza, hay que realizar un procesamiento diferente a la hora de recogerlos, medirlos o manejarlos. Los datos recopilados son los siguientes:

- **Calorías:** Se mide en calorías (C) en intervalos de 1 minuto. Al tratarlos, se realizará la media por horas.
- **Distancia:** Se mide en centímetros (cm) y se recogen en intervalos de 1 minuto. Al tratarlos, se realizará la suma por horas.
- **Pasos:** Se mide en pasos y los datos se recogen en intervalos de 1 minuto. Al tratar con estos, se realizará la suma por horas.

7.1. Datos Recopilados

- **Minutos Activos:** Se mide en minutos (*Min*) y son datos diarios. Consisten en la cantidad de minutos que pasa una persona en diferentes estados de actividad, estos son: ligeramente, moderadamente, muy activo, y sedentario. Se recogerán directamente y se agruparán por días.
- **Frecuencia Cardíaca:** Se mide en pulsaciones por minuto (*bpm*) y se recogen cada x segundos. Primero, se realizarán interpolaciones para tener los datos por minuto y después, se realizará la media por horas.
- **Frecuencia Cardíaca en Reposo:** Se mide en *bpm* y se recogen los datos en intervalos de 1 día. Consiste en la media de Frecuencia Cardíaca cuando el participante está en reposo. Además, se recoge el posible error de los cálculos. Al tratar con estos datos, se recogerán directamente y se agruparán por días.
- **Tiempo en las Zonas de Frecuencia Cardíaca:** Se mide en minutos y se recogen en intervalos de 1 día. Consiste en el número de minutos que pasa el participante en diferentes fases de la Frecuencia Cardíaca: Zona 1, 2, 3 y por debajo de la Zona 1. Se recogerán directamente y se agruparán por días.
- **Sueño:** Los datos recogidos tienen una estructura más compleja, con mucha más información. Los intervalos en los cuales se recogen son cada vez que el participante duerme. De todos estos datos, solo se consideran aquellos que pertenecen al sueño principal. Algunas de las características recopiladas han sido: Duración, hora de inicio y final del sueño, y minutos de sueño profundo, ligero o despierto. Estos datos se recogen directamente y se agrupan por días.

Cuando los datos son tratados, se agruparán en tablas. Estas tablas se construirán de diferentes formas dependiendo de cómo se haya recogido la información:

7. Descripción General del Problema

- Los datos recogidos en intervalos de segundos o minutos, tendrán filas por cada día y columnas por las 24 horas del día.
- Los datos recogidos en intervalos de días, tendrán filas por cada día y columnas con diferentes tipos de datos que se hayan recogido.

7.2. Bloques de Funcionalidades

La aplicación web posee tres bloques característicos, donde cada uno de ellos maneja un tipo de funcionalidad específica y diferenciada de las demás.

Cada uno de los bloques tendrá clases y funciones auxiliares. Siempre se intentará tener en cuentas los diferentes casos que puedan afectar a la integridad de la página, como puede ser al subir datos o modificarlos.

7.2.1. Gestión de la Web

En este bloque se encuentran los archivos base para el correcto funcionamiento de la página web. Con las configuraciones de la base de datos, plantillas, archivos multimedia y estilos:

- **Configuración:** Dentro de los archivos de configuración del sitio se establecen las direcciones de la página, la conexión con la base de datos, el idioma o dónde se encuentran los archivos multimedia.
- **Plantillas:** Son archivos HTML para dar un formato común a todas las páginas.
- **Archivos Multimedia:** Son todos aquellos archivos auxiliares como pueden ser imágenes, fuentes o funciones auxiliares.
- **Estilos:** Se establece el estilo del sitio, que será utilizado por las diferentes páginas para tener una apariencia común.

7.2.2. Gestión de Usuarios

Para poder controlar el acceso de los usuarios a las distintas funcionalidades, hay que comprobar que están registrados y han iniciado sesión dentro del sistema. Estos usuarios registrados son usuarios que han obtenido permisos de los administradores. Algunas de las funcionalidades que están relacionadas con la gestión de usuarios son:

- **Creación de una Cuenta:** Los administradores tienen la capacidad de crear cuentas para que los usuarios puedan acceder a las diferentes funcionalidades que ofrece la herramienta, como modificar o visualizar datos.
- **Eliminación de una Cuenta:** Los administradores podrán eliminar una cuenta de un usuario en caso de que así lo deseen.
- **Modificación de los Datos de una Cuenta:** Los datos de los usuarios que hayan sido registrados en el sistema serán accesibles y modificables por los administradores. Estos podrán modificar tanto la información del usuario como los permisos que tendrá sobre la aplicación.
- **Iniciar Sesión:** Los usuarios que hayan sido registrados tendrán la posibilidad de iniciar sesión con sus credenciales. Una vez hayan iniciado sesión, podrán utilizar las funcionalidades que ofrece la herramienta web.
- **Cerrar Sesión:** En el caso de que los usuarios que hayan iniciado sesión, quieran cerrarla, podrán hacerlo de forma sencilla, volviendo a ser usuarios sin los permisos para poder hacer uso de las funcionalidades.

Dependiendo de los permisos que tenga el usuario registrado, este podrá acceder a las funcionalidades de la página web como son el manejo de datos o la visualización de estos y la administración de los usuarios desde un panel de administración.

7. Descripción General del Problema

7.2.3. Gestión de Datos

En este bloque reside gran parte de las funcionalidades ofrecidas por la aplicación web. Las acciones están relacionadas con el procesamiento de los datos, su visualización y exportación.

Procesamiento de los Datos: Cuando un usuario registrado desee subir o modificar los datos de un participante, la herramienta realizará un procesamiento sobre estos, para que de esta forma, se elimine el ruido y se obtenga la información relevante. Una vez procesados, los datos son estructurados tal como se ha definido anteriormente y quedarán guardados en la base de datos. El procesamiento que se realiza es el siguiente:

- Una vez los ficheros con la información del participante es subida, se tomarán en cuanto aquellos ficheros que tengan información dentro del periodo de seguimiento.
- Tras obtener los ficheros relevantes, se itera por cada tipo de dato y se obtiene su información.
- Por último, se recoge la información relevante y se generan tablas que serán guardadas en la base de datos.

Los datos que serán procesados, ya sean porque se está subiendo la información de un nuevo participante o se esté modificando uno ya existente, seguirán los pasos ya comentados. Para el correcto funcionamiento de la herramienta y para brindar una mejor experiencia al usuario, se han tenido en cuenta las diferentes casuísticas que pueden existir al manejar los ficheros con los datos:

- En el caso de al subir los datos de un nuevo participante, no se determinen las fechas o se suba el archivo con los datos de inicio y fin del seguimiento, se cancelará la acción y se avisará al usuario registrados con el fallo ocurrido.
- En el caso de que al modificar los datos de un participante, no se suba la información correspondiente a una de las características

7.2. Bloques de Funcionalidades

seleccionadas, la herramienta avisará al usuario registrado para que le quede constancia de ello.

- En los casos de subir datos de un nuevo participante o modificar los datos de uno existente, si la información proporcionada de los ficheros no corresponde con las fechas seleccionadas, se avisará al usuario registrado de que dichos datos no se han introducido o modificado.

Visualización de los Datos: A la hora de visualizar los datos, los usuarios registrados tendrán la capacidad de acceder y estudiar los datos de los diferentes participantes mediante gráficas. Los datos del participante que se va a mostrar necesitan un procesamiento para recoger la información estrictamente necesaria y configurar la página web, de tal forma que se muestre dicha información correctamente. A la hora de visualizarlos, se tienen en cuenta diferentes situaciones:

- En caso de que se desee visualizar los datos de un participante que no han sido recogidos, el usuario registrado será avisado del problema.
- Si un usuario registrado desea visualizar los datos dependiendo de si es de día o de noche, y no están subidos los datos del sueño del participante, no se le permitirá y se le avisará del problema.

Exportación de los Datos: A la hora de exportar los datos de uno o todos los participantes, el usuario registrado podrá seleccionar el tipo de formato a descargar entre *.xls* o *.csv*. Además, dependiendo de si se quiere exportar los datos de un solo participante o de todos, el proceso varía:

- **Un participante:** Al exportar los datos de un único participante, el proceso se compone de dos partes. Recuperar los datos de la base de datos y calcular la media de las horas de los datos recopilados. Esta media se añadirá al final de las tablas.

7. Descripción General del Problema

- **Todos los participante:** En este caso, se tiene que recoger todos los datos de los participantes, calcular las medias de los datos recopilados y agrupar estas medias. No es necesario exportar la información por horas o días de los datos.

En el caso de los formatos a descargar la información de los participantes, la forma en que se recogen y procesan es diferente:

- **xls:** Cuando se quiere descargar los datos en formato Excel, el proceso consiste en obtener los datos ya procesados dependiendo de cuantos participantes se exporten y descargar el archivo que se genera directamente.
- **csv:** Cuando se quiere descargar los datos en formato CSV, el proceso es más tedioso. Se generarán diferentes archivos para cada tipo de dato y se recogerán todos en un encapsulado que será descargado.



8 Especificación de Requisitos

8.1. Requisitos

En este apartado se recogen los diferentes tipos de requisitos que tendrá el proyecto, para así, poder desarrollar más adelante el prototipo de interfaz y el mapa de navegación.

8.1.1. Requisitos Generales

Los requisitos generales sirven para jerarquizar el resto de los requisitos, de una forma clara y sencilla. En este proyecto se pueden diferenciar tres requisitos generales muy claros:

- **RG-01:** Gestión de los Usuarios: Son aquellas acciones que tienen que ver con la gestión de información de los usuarios.
- **RG-02:** Gestión de los Datos: Son todas las funcionalidades y acciones que los usuarios registrados pueden realizar y que están relacionadas con los datos recopilados de los participantes.
- **RG-03:** Gestión de la Web: Son aquellas funcionalidades que llevan el manejo y el buen funcionamiento de la página web.

8.1.2. Requisitos de Usuario

Los requisitos de usuario describen aquello que se espera de la herramienta, son las necesidades, las características principales, las interacciones y funciones que debe realizar.

Los requisitos relacionados con la **Gestión de la Web** son los siguientes:

- **RU-01:** Los usuarios tendrán la capacidad de acceder a la página web, sin poder acceder a las funcionalidades de esta.
- **RU-02:** Los usuarios registrados tendrán la capacidad de acceder a las diferentes funcionalidades que la página web ofrece.
- **RU-03:** Los administradores tendrán la capacidad de acceder al panel de control de la página web.

Los requisitos relacionados con la **Gestión de los Usuarios** son los siguientes:

- **RU-04:** El administrador tendrá la capacidad de dar de alta a un usuario en el sistema.
- **RU-05:** El administrador tendrá la capacidad de crear grupos en el sistema para agrupar a diferentes tipos de usuarios registrados.
- **RU-06:** El administrador tendrá la capacidad de buscar a los usuarios registrados en el sistema.
- **RU-07:** El administrador tendrá la capacidad de buscar grupos creados en el sistema.
- **RU-08:** El administrador tendrá la capacidad de consultar los datos de los usuarios registrados.
- **RU-09:** El administrador tendrá la capacidad de consultar los datos de los grupos ya creados.

8. Especificación de Requisitos

- **RU-10:** El administrador tendrá la capacidad de modificar los datos personales de los usuarios registrados.
- **RU-11:** El administrador tendrá la capacidad de modificar los permisos de los usuarios registrados.
- **RU-12:** El administrador tendrá la capacidad de agregar a un usuario registrado a un grupo creado.
- **RU-13:** El administrador tendrá la capacidad de modificar los grupos a lo que pertenece un usuario registrado.
- **RU-14:** El administrador tendrá la capacidad de eliminar los grupos a lo que pertenece un usuario registrado.
- **RU-15:** El administrador tendrá la capacidad de modificar los permisos que tiene un grupo.
- **RU-16:** El administrador tendrá la capacidad de eliminar los permisos de un usuario registrado.
- **RU-17:** El administrador tendrá la capacidad de dar de baja a un usuario registrado del sistema.
- **RU-18:** El administrador tendrá la capacidad de eliminar los permisos de un grupo.
- **RU-19:** El administrador tendrá la capacidad de eliminar un grupo creado del sistema.
- **RU-20:** El usuario registrado tendrá la capacidad de iniciar sesión en el sistema.
- **RU-21:** El usuario registrado tendrá la capacidad de cerrar sesión en el sistema.

Los requisitos relacionados con la **Gestión de los Datos** son los siguientes:

- **RU-22:** El usuario registrado tendrá la capacidad de subir la información sobre un nuevo participante.
- **RU-23:** El usuario registrado tendrá la capacidad de buscar a un participante existente.
- **RU-24:** El usuario registrado tendrá la capacidad de modificar los datos sobre los participantes existentes.
- **RU-25:** El usuario registrado tendrá la capacidad de visualizar los datos de los participantes existentes.
- **RU-26:** El usuario registrado tendrá la capacidad de exportar los datos de uno o más participantes.
- **RU-27:** El usuario registrado tendrá la capacidad de eliminar toda la información de un participante existente.

8.1.3. Requisitos Funcionales

Los requisitos funcionales describen las necesidades y funcionalidades que la página web debe satisfacer, es decir, indican el comportamiento del sistema.

Los requisitos relacionados con la **Gestión de la Web** son los siguientes:

- **RF-01:** El sistema permitirá acceder a la página principal de la web a los usuarios que la visiten.
 - **RF-01.1:** En caso de que el usuario intente acceder a un funcionalidad que no tenga permisos, será redirigido a la página principal.
- **RF-02:** El sistema permitirá a los usuarios registrados acceder a las diferentes funcionalidades que ofrece la página web.
- **RF-03:** El sistema permitirá a los administradores acceder al panel de control de la página web.

8. Especificación de Requisitos

Los requisitos relacionados con la **Gestión de los Usuarios** son lo siguientes:

- **RF-04:** El sistema permitirá a los administrador dar de alta a nuevos usuarios registrados en el sistema.
 - **RF-04.1:** Cada nuevo usuario registrado tendrá asociado un nombre de usuario único.
 - **RF-04.1.1:** En caso de introducir un usuario ya escogido en el sistema, se notificará de que el usuario no es válido.
 - **RF-04.2:** Cada nuevo usuario registrado podrá tener asociado unos permisos.
 - **RF-04.3:** Cada nuevo usuario registrado podrá tener asociado unos grupos.
- **RF-05:** El sistema permitirá a los administradores crear nuevos grupos en el sistema para agrupar a diferentes tipos de usuarios registrados.
 - **RF-05.1** Cada nuevo grupo creado tendrá asociado un nombre único.
 - **RF-05.1.1:** En caso de introducir un nombre de grupo ya escogido en el sistema, se notificará de que el nombre no es válido.
 - **RF-05.2** Cada nuevo grupo creado tendrá unos permisos asociados.
- **RF-06:** El sistema permitirá a los administradores buscar a un usuario registrado de los ya existentes.
- **RF-07:** El sistema permitirá a los administradores buscar un grupo de los ya existentes.
- **RF-08:** El sistema permitirá a los administradores consultar los datos de los usuarios registrados.

- **RF-09:** El sistema permitirá a los administradores consultar los datos de los grupos creados.
- **RF-10:** El sistema permitirá a los administradores modificar los datos personales de los usuarios registrados.
- **RF-11:** El sistema permitirá a los administradores modificar los permisos de los usuarios registrados.
- **RF-12:** El sistema permitirá a los administradores asignar grupos a los usuarios registrados.
 - **RF-12.1:** Debe existir al menos un grupo en el sistema.
- **RF-13:** El sistema permitirá a los administradores modificar los grupos a los que pertenece un usuario registrado.
 - **RF-13.1:** El usuario registrado debe tener al menos un grupo asociado para poder realizar las modificaciones.
- **RF-14:** El sistema permitirá a los administradores eliminar los grupos a los que pertenezca un usuario registrado.
- **RF-15:** El sistema permitirá a los administradores modificar los permisos de los grupos existentes.
- **RF-16:** El sistema permitirá a los administradores eliminar los permisos asociados a un usuario registrado.
 - **RF-16.1:** El usuario registrado debe tener al menos un permiso asociado para poder realizar las modificaciones.
- **RF-17:** El sistema permitirá a los administradores dar de baja a los usuarios registrados del sistema.
- **RF-18:** El sistema permitirá a los administradores eliminar los permisos asociados a un grupo.
 - **RF-18.1:** El grupo debe tener al menos un permiso asociado para poder realizar las modificaciones.

8. Especificación de Requisitos

- **RF-19:** El sistema permitirá a los administradores eliminar un grupo del sistema.
- **RF-20:** El sistema permitirá a los usuarios registrado iniciar sesión en el sistema.
 - **RF-20.1:** En caso de introducir las credenciales de forma incorrecta, se notificará de lo sucedido.
- **RF-21:** El sistema permitirá a los usuarios registrado cerrar sesión en el sistema.

Los requisitos relacionados con la **Gestión de los Datos** son los siguientes:

- **RF-22:** El sistema permitirá a los usuarios registrados subir información sobre un nuevo participante.
 - **RD-22.1** Cada nuevo participante generado, tendrá un identificador interno único.
- **RF-23:** El sistema permitirá a los usuarios registrados buscar a los participantes existentes.
- **RF-24:** El sistema permitirá a los usuarios registrados modificar los datos de los participantes existentes.
- **RF-25:** El sistema permitirá a los usuarios registrados visualizar los datos de los participantes existentes.
 - **RF-25.1:** Si el participante no tiene datos sobre el *Sueño*, no se permitirá visualizar los datos en las franjas horarias de sueño o despierto.
 - **RF-25.2:** Si el participante no tiene datos sobre otro tipo de información, no se permitirá visualizar dichos datos.
- **RF-26:** El sistema permitirá a los usuarios registrados exportar los datos de uno o más participantes.

- **RF-27:** El sistema permitirá a los usuarios registrados eliminar toda la información de los participantes existentes.

8.1.4. Requisitos No Funcionales

Los requisitos no funcionales son restricciones que afectan al sistema y describen la calidad y rendimiento que debe tener este:

- **RNF-01:** La aplicación web estará disponible en inglés.
- **RNF-02:** La aplicación web debe ser intuitiva y fácil de usar.
- **RNF-03:** El tiempo de respuesta debe de ser el mínimo posible.
- **RNF-04:** Los datos guardados en el sistema deben estar seguros.
- **RNF-05:** El sistema manejará los errores en su debida forma para asegurar la buena experiencia de los usuarios.

8.1.5. Requisitos de Negocio

Estos requisitos son las reglas y necesidades definidas por el usuario final que deben ser respetadas durante el desarrollo del proyecto:

- **RN-01:** El sistema comprobará que los usuarios que no estén registrados, no tengan acceso a la información recopilado de los participantes.
- **RN-02:** El sistema comprobará que no existen dos usuarios con el mismo nombre.
- **RN-03:** El sistema comprobará que no existen dos grupos con el mismo nombre.

8. Especificación de Requisitos

8.1.6. Requisitos de Información

Estos requisitos describen la información relevante que debe ser almacenada en la base de datos, para que pueda ser almacenada y gestionada:

- **RI-01:** El sistema almacenará el Nombre y los Permisos para cada grupo.
- **RI-02:** El sistema almacenará de los usuarios registrados: Nombre, contraseña, permisos asociados y grupos asignados.
- **RI-03:** El sistema almacenará de los participantes: Identificador, nombre, fecha de inicio y fin del periodo de seguimiento y datos recopilados.

Como se ha explicado anteriormente, los datos recopilados son de diferente naturaleza, por lo que, la información que se guarda sobre estos y la estructura que tienen, varía:

- **Calorías:** Se guardan por cada día el número de sesión, la fecha, el día de la semana, el tratamiento y la suma de calorías quemadas por horas.
- **Distancia:** Se guardan por cada día el número de sesión, la fecha, el día de la semana, el tratamiento y la distancia recorrida por horas.
- **Pasos:** Se guardan por cada día el número de sesión, la fecha, el día de la semana, el tratamiento y los pasos realizados por horas.
- **Minutos Activos:** Se guardan por cada día el número de sesión, la fecha, el día de la semana, los minutos activos de actividad ligera, moderada, muy activa, sedentaria y su suma.
- **Frecuencia Cardíaca:** Se guardan por cada día el número de sesión, la fecha, el día de la semana, el tratamiento y la media de pulsaciones por minuto (*bpm*) por horas.

8.1. Requisitos

- **Frecuencia Cardíaca en Reposo:** Se guardan por cada día el número de sesión, la fecha, el día de la semana, las *bpm* y el error en ese día determinado.
- **Tiempo en las Zonas de Frecuencia Cardíaca:** Se guardan por cada día el número de sesión, la fecha, el día de la semana, el número de minutos en las zonas 3, 2, 1 y su suma.
- **Sueño:** Se guardan por cada día el número de sesión, la fecha, las horas de inicio y fin de la actividad, la duración (*milisegundos*), los minutos dormido, despierto y en la cama, las veces que lo ha estado, los minutos y la media de tiempo que ha estado en sueño profundo, despierto, ligero y REM (*Rapid Eye Movements*).

8.2. Validación de Requisitos

A continuación, se va a comprobar que las funcionalidades del proyecto se corresponden con las necesidades del usuario final. De esta forma, se pretenden validar los requisitos especificados y se evitarán problemas futuros de implementación. Para realizar la validación, se tendrán en cuenta los siguientes parámetros:

- **Validez:** Se ha de tener en cuenta a todos los usuarios que vayan a hacer uso de la aplicación.
- **Consistencia:** No deben existir contradicciones entre los requisitos definidos.
- **Completitud:** Deben estar definidos todos los requisitos.
- **Realismo:** Los requisitos deben ser capaces de desarrollarse con la tecnología actual.
- **Verificabilidad:** Se ha de comprobar que todos los requisitos definidos cumplen su función.

La relación de los diferentes bloques de funcionalidades de los requisitos de usuario y los requisitos funcionales se puede observar en las Tablas 8.1, 8.2, y 8.3:

Requisitos de Usuario	Requisitos Funcionales
RU-01	RF-01, RF-01.1
RU-02	RF-02
RU-03	RF-03

Tabla 8.1: RU/RF: Gestión de la Web

8.2. Validación de Requisitos

Requisitos de Usuario	Requisitos Funcionales
RU-04	RF-04, RF-04.1, RF-04.1.1, RF-04.2, RF-04.3
RU-05	RF-05, RF-05.1, RF-05.1.1, RF-05.2
RU-06	RF-06
RU-07	RF-07
RU-08	RF-08
RU-09	RF-09
RU-10	RF-10
RU-11	RF-11
RU-12	RF-12, RF-12.1
RU-13	RF-13, RF-13.1
RU-14	RF-14
RU-15	RF-15
RU-16	RF-16, RF-16.1
RU-17	RF-17
RU-18	RF-18, RF-18.1
RU-19	RF-19
RU-20	RF-20, RF-20.1
RU-21	RF-21

Tabla 8.2: RU/RF: Gestión de los Usuarios

8. Especificación de Requisitos

Requisitos de Usuario	Requisitos Funcionales
RU-22	RF-22, RF-22.1
RU-23	RF-23
RU-24	RF-24
RU-25	RF-25, RF-25.1, RF-25.2
RU-26	RF-26
RU-27	RF-27

Tabla 8.3: RU/RF: Gestión de los Datos

8.3. Matriz de Trazabilidad

Una vez descritos los diferentes requisitos funcionales, no funcionales, de negocio y de información, se presenta las siguientes matrices de trazabilidad (Tablas 8.4, 8.5, 8.6 y 8.7) para tener una visión global y resumida de cómo se relacionan entre ellos.

	RG-01	RG-02	RG-03
RF-01			×
RF-02			×
RF-03			×

Tabla 8.4: Matriz de trazabilidad: Requisitos generales-funcionales 1

8.3. Matriz de Trazabilidad

	RG-01	RG-02	RG-03
RF-04	×		
RF-05	×		
RF-06	×		
RF-07	×		
RF-08	×		
RF-09	×		
RF-10	×		
RF-11	×		
RF-12	×		
RF-13	×		
RF-14	×		
RF-15	×		
RF-16	×		
RF-17	×		
RF-18	×		
RF-19	×		
RF-20	×		
RF-21	×		

Tabla 8.5: Matriz de trazabilidad: Requisitos generales-funcionales 2

8. Especificación de Requisitos

	RG-01	RG-02	RG-03
RF-22		×	
RF-23		×	
RF-24		×	
RF-25		×	
RF-26		×	
RF-27		×	

Tabla 8.6: Matriz de trazabilidad: Requisitos generales-funcionales 3

	RG-01	RG-02	RG-03
RNF-01			×
RNF-02			×
RNF-03			×
RNF-04	×	×	×
RNF-05			×
RN-01	×		
RN-02	×		
RN-03	×		
RI-01	×		
RI-02	×		
RI-03		×	

Tabla 8.7: Matriz de trazabilidad: Requisitos generales-otros

8.4. Características de la Interfaz

La interfaz seguirá unas pautas y características específicas para cumplir los objetivos descritos. Estos requisitos son:

- La información mostrada se presentará de forma clara y concisa, para transmitir de manera precisa lo mostrado y así el usuario tendrá una mejor experiencia.
- Los elementos, como botones, iconos u otros elementos, mantendrán una coherencia entre ellos.
- Se utilizará una tipografía sencilla, clara y lo suficientemente grande para su lectura. De esta forma, la información mostrada será más fácil de entender.
- La interfaz será interactiva, mostrando mensajes sobre lo que va ocurriendo durante el uso de la aplicación.
- Al cambiar el tamaño de las ventanas, la información que aparezca se adaptará a las nuevas dimensiones.
- Los colores utilizados en los textos, botones y demás elementos, mantendrán un contraste para su fácil visualización.
- Existirá una consistencia en el significado de los diferentes botones, para que el usuario sepa siempre cómo actúa cada uno.
- Los títulos y nombres utilizados para describir las características y funcionalidades deben ser autodescriptivos.



9 Análisis Funcional

En esta sección se van a describir y definir los diferentes casos de uso que tiene este Trabajo Fin de Grado (TFG). Los casos de uso son secuencias de acciones que realiza un sistema para completar una acción, ya sea una interacción de uno o más actores. Es una forma de visualizar el cómo responde el sistema antes diferentes situaciones.

La estructura que se va a seguir a la hora de definir los diferentes casos de uso es la siguiente:

- **Identificador.**
- **Actores:** Puede ser uno o varios y se comunica con el sistema.
- **Descripción:** Breve descripción sobre el caso de uso.
- **Precondiciones:** Acciones que se deben llevar a cabo para realizar la acción que se describe.
- **Flujo Principal:** Pasos que se realizan para completar el caso de uso de forma exitosa.
- **Postcondiciones:** Acción final al realizar el flujo principal.
- **Flujo Alternativo:** Pasos que se realizan por situaciones que no llevan a completar el caso de uso de forma correcta.

Dentro de la página web se diferencian tres tipos de usuarios/actores, que dependiendo del rol que tengan, las acciones que pueden realizar dentro de la página web varían. Estos actores son:

- **Usuario:** Es la persona que accede a la página web pero no tiene permisos para utilizar las funcionalidades que esta ofrece, como el manejo y visualización de los datos.
- **Usuario Registrado:** Es la persona que accede a la página web y tiene permisos para utilizar las funcionalidades que esta ofrece, como son el manejo y visualización de los datos.
- **Administrador:** Es la persona, que además de los permisos que tienen los usuarios registrados, pueden crear nuevos usuarios y controlar los permisos de cada uno de estos.

Los casos de uso que se van a describir están relacionados directamente con los requisitos funcionales descritos anteriormente. Por lo que habrá 27 casos de uso, uno por cada requisito.

9. Análisis Funcional

9.1. Casos de Uso: Gestión de la Web

9.1.1. Caso de Uso 01: Usuario accede a la Web

Identificador: CU-01.

Actores: Usuario.

Descripción: El sistema deberá comportarse como se describe en el siguiente caso de uso cuando el usuario desee acceder a la página web.

Precondiciones: Ninguna.

Flujo Principal:

1. El usuario solicita acceder a la página principal de la web.
2. El sistema envía al usuario a la página web.

Postcondiciones: El usuario accede a la página principal de la página web.

Flujo Alternativo:

- 1.1. El usuario solicita acceder a una funcionalidad sin tener los permisos correspondientes.
- 1.2. El sistema comprueba que no tiene permisos y redirige al usuario a la página principal de la web.

9.1.2. Caso de Uso 02: Usuario Registrado accede a la Web

Identificador: CU-02.

Actores: Usuario Registrado.

Descripción: El sistema deberá comportarse como se describe en el siguiente caso de uso cuando el usuario registrado desee acceder a la página web y a sus funcionalidades.

Precondiciones: El usuario debe de haber iniciado sesión.

Flujo Principal:

1. El usuario registrado solicita acceder a la página web y sus funcionalidades.
2. El sistema comprueba los permisos del usuario registrado y envía al usuario registrado a la página que ha solicitado.

Postcondiciones: El usuario registrado accede a la página que ha solicitado.

Flujo Alternativo:

- 2.1. El sistema comprueba que el usuario registrado no tiene permisos para acceder a la página solicitada y lo redirige a la página principal.

9. Análisis Funcional

9.1.3. Caso de Uso 03: Administrador accede al Panel de Control

Identificador: CU-03.

Actores: Administrador.

Descripción: El sistema deberá comportarse como se describe en el siguiente caso de uso cuando el administrador desee acceder al panel de control de la página web.

Precondiciones: Ninguna.

Flujo Principal:

1. El administrador solicita acceder al panel de control.
2. El sistema solicita las credenciales del administrador.
3. El administrador introduce sus credenciales (nombre de usuario y contraseña).
4. El sistema verifica la identidad del administrador e informa que todo está correcto.

Postcondiciones: El administrador accede al panel de control de la página web.

Flujo Alternativo:

- 2.1. Si el administrador ya ha iniciado sesión desde el CU-20, el sistema verifica los permisos de este e informa que todo está correcto.
- 4.1. El sistema comprueba que las credenciales no coinciden, informa del error y vuelve al paso 2.

9.2. Casos de Uso: Gestión de los Usuarios

9.2.1. Caso de Uso 04: Crear usuario registrado

Identificador: CU-04.

Actores: Administrador.

Descripción: El sistema deberá comportarse como se describe en el siguiente caso de uso cuando el administrador desee dar de alta a un nuevo usuario.

Precondiciones: El administrador debe de haber accedido al panel de control de la página web.

Flujo Principal:

1. El administrador solicita registrar a un nuevo usuario registrado en el sistema.
2. El sistema solicita al administrador introducir el usuario y contraseña que tendrá el usuario registrado.
3. El administrador introduce las credenciales del usuario registrado.
4. El sistema comprueba que el nombre del usuario es único e informa que todo está correcto.

Postcondiciones: El administrador da de alta a un nuevo usuario.

Flujo Alternativo:

- 4.1. El sistema comprueba que el nombre del usuario ya está registrado, informa del error y vuelve al paso 2.

9. Análisis Funcional

9.2.2. Caso de Uso 05: Crear grupo

Identificador: CU-05.

Actores: Administrador.

Descripción: El sistema deberá comportarse como se describe en el siguiente caso de uso cuando el administrador desee crear un nuevo grupo.

Precondiciones: El administrador debe de haber accedido al panel de control de la página web.

Flujo Principal:

1. El administrador solicita crear un nuevo grupo.
2. El sistema solicita al administrador introducir el nombre del grupo y los permisos de este.
3. El administrador introduce los datos del grupo.
4. El sistema comprueba que nombre del grupo es único e informa que todo está correcto.

Postcondiciones: El administrador crea un nuevo grupo.

Flujo Alternativo:

- 4.1. El sistema comprueba que el nombre del grupo está en uso, informa del error y vuelve al paso 2.

9.2.3. Caso de Uso 06: Buscar usuario registrado

Identificador: CU-06.

Actores: Administrador.

Descripción: El sistema deberá comportarse como se describe en el siguiente caso de uso cuando el administrador desee buscar a un usuario registrado existente.

Precondiciones: El administrador debe de haber accedido al panel de control de la página web.

Flujo Principal:

1. El administrador solicita buscar a un usuario registrado.
2. El sistema solicita el nombre del usuario.
3. El administrador introduce el nombre del usuario registrado.
4. El sistema busca a todos los usuarios registrados que contengan el nombre introducido y se lo muestra al administrador.

Postcondiciones: El administrador encuentra al usuario registrado.

Flujo Alternativo:

- 4.1. El sistema no encuentra a ningún usuario con el nombre introducido, informa de lo sucedido y vuelve al paso 2.

9. Análisis Funcional

9.2.4. Caso de Uso 07: Buscar grupo

Identificador: CU-07.

Actores: Administrador.

Descripción: El sistema deberá comportarse como se describe en el siguiente caso de uso cuando el administrador desee buscar un grupo existente.

Precondiciones: El administrador debe de haber accedido al panel de control de la página web.

Flujo Principal:

1. El administrador solicita buscar a un grupo.
2. El sistema solicita el nombre del grupo.
3. El administrador introduce el nombre del grupo.
4. El sistema busca todos los grupos que contenga el nombre introducido y se lo muestra al administrador.

Postcondiciones: El administrador encuentra el grupo.

Flujo Alternativo:

- 4.1. El sistema no encuentra a ningún grupo con el nombre introducido, informa de lo sucedido y vuelve al paso 2.

9.2.5. Caso de Uso 08: Consultar usuario registrado

Identificador: CU-08.

Actores: Administrador.

Descripción: El sistema deberá comportarse como se describe en el siguiente caso de uso cuando el administrador desee consultar la información de un usuario registrado existente.

Precondiciones: El administrador debe de haber accedido al panel de control de la página web.

Flujo Principal:

1. El administrador solicita consultar la información de un usuario registrado.
2. El sistema muestra toda la información sobre el usuario registrado.

Postcondiciones: El administrador consulta la información del usuario registrado.

Flujo Alternativo: Ninguno.

9. Análisis Funcional

9.2.6. Caso de Uso 09: Consultar grupo

Identificador: CU-09.

Actores: Administrador.

Descripción: El sistema deberá comportarse como se describe en el siguiente caso de uso cuando el administrador desee consultar los datos de un grupo existente.

Precondiciones: El administrador debe de haber accedido al panel de control de la página web.

Flujo Principal:

1. El administrador ejecuta el caso de uso 7.
2. El administrador solicita consultar la información de un grupo.
3. El sistema muestra toda la información sobre el grupo.

Postcondiciones: El administrador consulta los permisos del grupo.

Flujo Alternativo: Ninguno.

9.2.7. Caso de Uso 10: Modificar datos de un usuario registrado

Identificador: CU-10.

Actores: Administrador.

Descripción: El sistema deberá comportarse como se describe en el siguiente caso de uso cuando el administrador desee modificar los datos personales de un usuario registrado.

Precondiciones: El administrador debe de haber accedido al panel de control de la página web.

Flujo Principal:

1. El administrador realiza el caso de uso 8.
2. El administrador solicita modificar los datos personales del usuario registrado.
3. El sistema solicita al administrador los datos a modificar.
4. El administrador introduce los datos a modificar.
5. El sistema comprueba los datos introducidos, informa que todo está correcto y solicita confirmación.
6. El administrador confirma la acción de modificar los datos.
7. El sistema guarda los cambios realizados e informa que todo está correcto.

Postcondiciones: El administrador modifica los datos personales del usuario registrado.

Flujo Alternativo:

- 5.1. El sistema comprueba que el nombre modificado del usuario registrado está en uso, informa del error y vuelve al paso 3.
- 6.1. El administrador cancela la acción de modificar los datos y se termina el caso de uso.

9. Análisis Funcional

9.2.8. Caso de Uso 11: Modificar permisos de un usuario registrado

Identificador: CU-11.

Actores: Administrador.

Descripción: El sistema deberá comportarse como se describe en el siguiente caso de uso cuando el administrador desee modificar los permisos de un usuario registrado.

Precondiciones: El administrador debe de haber accedido al panel de control de la página web.

Flujo Principal:

1. El administrador realiza el caso de uso 8.
2. El administrador solicita modificar los permisos que tiene el usuario registrado.
3. El sistema solicita al administrador seleccionar los permisos que se quieren modificar.
4. El administrador selecciona los permisos a modificar.
5. El sistema solicita confirmación de las modificaciones.
6. El administrador confirma la acción de modificar los permisos.
7. El sistema guarda los cambios realizados e informa que todo está correcto.

Postcondiciones: El administrador modifica los permisos del usuario registrado.

Flujo Alternativo:

- 6.1. El administrador cancela la acción de modificar los permisos y se termina el caso de uso.

9.2.9. Caso de Uso 12: Asignar grupos de un usuario registrado

Identificador: CU-12.

Actores: Administrador.

Descripción: El sistema deberá comportarse como se describe en el siguiente caso de uso cuando el administrador desee asignar uno o más grupos a un usuario registrado.

Precondiciones: El administrador debe de haber accedido al panel de control de la página web.

Flujo Principal:

1. El administrador realiza el caso de uso 8.
2. El administrador solicita asignar uno o más grupos a un usuario registrado.
3. El sistema solicita al administrador seleccionar los grupos a asignar.
4. El administrador selecciona los grupos a asignar.
5. El sistema solicita confirmación de las asignaciones.
6. El administrador confirma la acción de asignar los grupos.
7. El sistema guarda los cambios realizados e informa que todo está correcto.

Postcondiciones: El administrador asigna uno o más grupos al usuario registrado.

Flujo Alternativo:

- 6.1. El administrador cancela la acción de asignar los grupos a un usuario registrado y se termina el caso de uso.

9. Análisis Funcional

9.2.10. Caso de Uso 13: Modificar grupos de un usuario registrado

Identificador: CU-13.

Actores: Administrador.

Descripción: El sistema deberá comportarse como se describe en el siguiente caso de uso cuando el administrador desee modificar los grupos a los que pertenece un usuario registrado.

Precondiciones: El administrador debe de haber accedido al panel de control de la página web.

Flujo Principal:

1. El administrador realiza el caso de uso 8.
2. El administrador solicita modificar los grupos a los que pertenece el usuario registrado.
3. El sistema solicita al administrador seleccionar los grupos a modificar.
4. El administrador selecciona los grupos a modificar.
5. El sistema solicita confirmación de las modificaciones.
6. El administrador confirma la acción de modificar los grupos.
7. El sistema guarda los cambios realizados e informa que todo está correcto.

Postcondiciones: El administrador modifica los grupos a los que pertenece el usuario registrado.

Flujo Alternativo:

- 6.1. El administrador cancela la acción de modificar los grupos a los que pertenece un usuario registrado y se termina el caso de uso.

9.2.11. Caso de Uso 14: Eliminar grupos de un usuario registrado

Identificador: CU-14.

Actores: Administrador.

Descripción: El sistema deberá comportarse como se describe en el siguiente caso de uso cuando el administrador desee eliminar los grupos a los que pertenece un usuario registrado.

Precondiciones: El administrador debe de haber accedido al panel de control de la página web.

Flujo Principal:

1. El administrador realiza el caso de uso 8.
2. El administrador solicita eliminar los grupos a los que pertenece el usuario registrado.
3. El sistema solicita confirmación de las modificaciones.
4. El administrador confirma la acción de eliminar todos los grupos a los que pertenece el usuario registrado.
5. El sistema guarda los cambios realizados e informa que todo está correcto.

Postcondiciones: El administrador elimina los grupos a los que pertenece el usuario registrado.

Flujo Alternativo:

- 3.1. El administrador cancela la acción de eliminar los grupos a los que pertenece un usuario registrado y se termina el caso de uso.

9. Análisis Funcional

9.2.12. Caso de Uso 15: Modificar permisos de un grupo

Identificador: CU-15.

Actores: Administrador.

Descripción: El sistema deberá comportarse como se describe en el siguiente caso de uso cuando el administrador desee modificar los permisos que tiene un grupo.

Precondiciones: El administrador debe de haber accedido al panel de control de la página web.

Flujo Principal:

1. El administrador realiza el caso de uso 9.
2. El administrador solicita modificar los permisos del usuario registrado.
3. El sistema solicita al administrador seleccionar los permisos a modificar.
4. El administrador selecciona los permisos a modificar.
5. El sistema solicita confirmación de las modificaciones.
6. El administrador confirma la acción de modificar los permisos.
7. El sistema guarda los cambios realizados e informa que todo está correcto.

Postcondiciones: El administrador modifica los permisos del grupo.

Flujo Alternativo:

- 6.1. El administrador cancela la acción de modificar los permisos que tiene un usuario registrado y se termina el caso de uso.

9.2.13. Caso de Uso 16: Eliminar permisos de un usuario registrado

Identificador: CU-16.

Actores: Administrador.

Descripción: El sistema deberá comportarse como se describe en el siguiente caso de uso cuando el administrador desee eliminar los permisos que tiene un usuario registrado.

Precondiciones: El administrador debe de haber accedido al panel de control de la página web.

Flujo Principal:

1. El administrador realiza el caso de uso 8.
2. El administrador solicita eliminar los permisos que tiene el usuario registrado.
3. El sistema solicita confirmación de las modificaciones.
4. El administrador confirma la acción de eliminar todos los permisos que tiene el usuario registrado.
5. El sistema guarda los cambios realizados e informa que todo está correcto.

Postcondiciones: El administrador elimina los permisos que tiene el usuario registrado.

Flujo Alternativo:

- 3.1. El administrador cancela la acción de eliminar los permisos que tiene un usuario registrado y se termina el caso de uso.

9. Análisis Funcional

9.2.14. Caso de Uso 17: Eliminar usuario registrado

Identificador: CU-17.

Actores: Administrador.

Descripción: El sistema deberá comportarse como se describe en el siguiente caso de uso cuando el administrador desee eliminar a un usuario registrado.

Precondiciones: El administrador debe de haber accedido al panel de control de la página web.

Flujo Principal:

1. El administrador realiza el caso de uso 8.
2. El administrador solicita eliminar al usuario registrado del sistema.
3. El sistema solicita confirmación de la eliminación.
4. El administrador confirma la acción de eliminar al usuario registrado.
5. El sistema elimina al usuario registrado del sistema e informa que todo está correcto.

Postcondiciones: El administrador elimina al usuario registrado.

Flujo Alternativo:

- 4.1. El administrador cancela la acción de eliminar al usuario registrado y se termina el caso de uso.

9.2.15. Caso de Uso 18: Eliminar permisos de un grupo

Identificador: CU-18.

Actores: Administrador.

Descripción: El sistema deberá comportarse como se describe en el siguiente caso de uso cuando el administrador desee eliminar los permisos que tiene un grupo.

Precondiciones: El administrador debe de haber accedido al panel de control de la página web.

Flujo Principal:

1. El administrador realiza el caso de uso 9.
2. El administrador solicita eliminar los permisos que tiene el grupo.
3. El sistema solicita confirmación de las modificaciones.
4. El administrador confirma la acción de eliminar todos los permisos que tiene el grupo.
5. El sistema guarda los cambios realizados e informa que todo está correcto.

Postcondiciones: El administrador elimina los permisos que tiene el grupo.

Flujo Alternativo:

- 3.1. El administrador cancela la acción de eliminar los permisos que tiene un grupo y se termina el caso de uso.

9. Análisis Funcional

9.2.16. Caso de Uso 19: Eliminar grupo

Identificador: CU-19.

Actores: Administrador.

Descripción: El sistema deberá comportarse como se describe en el siguiente caso de uso cuando el administrador desee eliminar un grupo.

Precondiciones: El administrador debe de haber accedido al panel de control de la página web.

Flujo Principal:

1. El administrador realiza el caso de uso 9.
2. El administrador solicita eliminar al usuario registrado del sistema.
3. El sistema solicita confirmación de la eliminación.
4. El administrador confirma la acción de eliminar al usuario registrado.
5. El sistema elimina al usuario registrado del sistema e informa que todo está correcto.

Postcondiciones: El administrador elimina el grupo.

Flujo Alternativo:

- 4.1. El administrador cancela la acción de eliminar el grupo y se termina el caso de uso.

9.2.17. Caso de Uso 20: Iniciar sesión

Identificador: CU-20.

Actores: Usuario Registrado.

Descripción: El sistema deberá comportarse como se describe en el siguiente caso de uso cuando el usuario registrado desee iniciar sesión en el sistema.

Precondiciones: Ninguna

Flujo Principal:

1. El usuario registrado solicita iniciar sesión en el sistema.
2. El sistema solicita al usuario registrado que introduzca sus credenciales.
3. El usuario registrado introduce sus credenciales (nombre de usuario y contraseña).
4. El sistema verifica la identidad del usuario registrado e informa que todo está correcto.

Postcondiciones: El usuario registrado ha iniciado sesión.

Flujo Alternativo:

- 4.1. El sistema comprueba que las credenciales no coinciden, informa del error y vuelve al paso 2.

9. Análisis Funcional

9.2.18. Caso de Uso 21: Cerrar sesión

Identificador: CU-21.

Actores: Usuario Registrado.

Descripción: El sistema deberá comportarse como se describe en el siguiente caso de uso cuando el usuario registrado desee cerrar sesión el sistema.

Precondiciones: El usuario registrado debe de haber iniciado sesión.

Flujo Principal:

1. El usuario registrado solicita cerrar sesión en el sistema.
2. El sistema cierra la sesión e informa que todo está correcto.

Postcondiciones: El usuario registrado ha cerrado sesión.

Flujo Alternativo: Ninguno.

9.3. Casos de Uso: Gestión de los Datos

9.3.1. Caso de Uso 22: Crear nuevo participante

Identificador: CU-22.

Actores: Usuario registrado.

Descripción: El sistema deberá comportarse como se describe en el siguiente caso de uso cuando el usuario registrado desee cargar datos sobre un nuevo participante.

Precondiciones: El usuario debe de haber iniciado sesión.

Flujo Principal:

1. El usuario registrado solicita al sistema cargar datos de un nuevo participante.
2. El sistema solicita al usuario registrado que introduzca los datos.
3. El usuario registrado rellena los campos solicitados y sube los ficheros con los datos de las pulseras.
4. El sistema procesa los datos y los guarda en la base de datos.
5. Los datos procesados son guardados en la base de datos y el sistema informa al usuario registrado que todo está correcto, con un informa de los datos subidos.

Postcondiciones: El usuario registrado sube archivos de un nuevo participante al sistema y este los guarda.

Flujo Alternativo:

- 4.1. El usuario registrado no ha subido los ficheros con los datos, se informa del error y se vuelve al paso 2.

9. Análisis Funcional

9.3.2. Caso de Uso 23: Buscar participante

Identificador: CU-23.

Actores: Usuario registrado.

Descripción: El sistema deberá comportarse como se describe en el siguiente caso de uso cuando el usuario registrado desee buscar a un participante.

Precondiciones: El usuario debe de haber iniciado sesión.

Flujo Principal:

1. El usuario registrado solicita al sistema buscar a un participante.
2. El sistema recoge los identificadores y nombres de los participantes y se los muestra al usuario registrado en forma de lista.
3. El usuario registrado selecciona al participante que desee.

Postcondiciones: El usuario registrado encuentra al participante.

Flujo Alternativo:

- 2.1. Si no hay participantes en el sistema, se devolverá una lista vacía y se termina el caso de uso.

9.3.3. Caso de Uso 24: Modificar datos de un participante

Identificador: CU-24.

Actores: Usuario registrado.

Descripción: El sistema deberá comportarse como se describe en el siguiente caso de uso cuando el usuario registrado desee modificar los datos de un participante.

Precondiciones: El usuario debe de haber iniciado sesión.

Flujo Principal:

1. El usuario registrado realiza el caso de uso 23.
2. El usuario registrado solicita al sistema modificar los datos del participante.
3. El sistema solicita al usuario registrado seleccionar los tipos de datos a modificar y los ficheros con los datos.
4. El usuario registrado selecciona los tipos de datos a modificar y sube los ficheros con los datos.
5. El sistema procesa los datos y los guarda en la base de datos.
6. Los datos procesados son guardados y modificados en la base de datos y el sistema informa que todo está correcto.

Postcondiciones: El usuario registrado modifica los daos de un participante.

Flujo Alternativo:

- 2.1. Si no hay participantes en el sistema se termina el caso de uso.
- 5.1. El usuario registrado no selecciona ningún tipo de dato a modificar, se informa del error y se vuelve al paso 2.
- 5.2. El usuario registrado no ha subido los ficheros con los datos, se informa del error y se vuelve al paso 2.

9. Análisis Funcional

9.3.4. Caso de Uso 25: Visualizar datos de un participante

Identificador: CU-25.

Actores: Usuario registrado.

Descripción: El sistema deberá comportarse como se describe en el siguiente caso de uso cuando el usuario registrado desee visualizar los datos de un participante.

Precondiciones: El usuario debe de haber iniciado sesión.

Flujo Principal:

1. El usuario registrado realiza el caso de uso 23.
2. El usuario registrado solicita visualizar los datos del participante.
3. El sistema solicita al usuario registrado que seleccione los datos a visualizar, la franja horaria (día y noche, día o noche) y la medida de los datos (horas o días).
4. El usuario registrado selecciona los datos a visualizar.
5. El sistema solicita a la base de datos los datos solicitados.
6. La base de datos envía al sistema los datos del participante.
7. El sistema procesa los datos del participante para visualizarlos y los muestra al usuario registrado.

Postcondiciones:

Flujo Alternativo:

- 2.1. Si no hay participantes en el sistema se termina el caso de uso.
- 5.1. El usuario registrado no selecciona los datos a visualizar, se informa del error y se vuelve al paso 2.
- 6.1. La base de datos no encuentra los datos solicitados, informa al sistema del error y vuelve al paso 2.
- 7.1. Si el usuario selecciona franjas horarias (día o noche) y no hay datos subidos del sueño, se informa del error y vuelve al paso 2.

9.3.5. Caso de Uso 26: Exportar datos de un participante

Identificador: CU-26.

Actores: Usuario registrado.

Descripción: El sistema deberá comportarse como se describe en el siguiente caso de uso cuando el usuario registrado desee exportar los datos de uno o más participantes.

Precondiciones: El usuario debe de haber iniciado sesión.

Flujo Principal:

1. El usuario registrado realiza el caso de uso 23.
2. El usuario registrado solicita exportar los datos del participante o de todos ellos.
3. El sistema solicita al usuario registrado que seleccione el formato a descargar los datos.
4. El usuario registrado selecciona el formato (Excel o CSV).
5. El sistema solicita a la base de datos los datos del o de los participantes.
6. La base de datos recoge y envía los datos al sistema.
7. El sistema genera los archivos y solicita al usuario registrado guardarlos en su disco local.
8. El usuario registrado confirma la acción y descarga el archivo.

Postcondiciones:

Flujo Alternativo:

- 2.1. Si no hay participantes en el sistema se termina el caso de uso.
- 7.1. Si el usuario registrado ha seleccionado a todos los participantes, se procesan todos los datos y se generan nuevos datos.
- 8.1. El usuario registrado cancela la acción y se termina el caso de uso.

9. Análisis Funcional

9.3.6. Caso de Uso 27: Eliminar participante

Identificador: CU-27.

Actores: Usuario registrado.

Descripción: El sistema deberá comportarse como se describe en el siguiente caso de uso cuando el usuario registrado desee eliminar a un participante de la base de datos.

Precondiciones: El usuario debe de haber iniciado sesión.

Flujo Principal:

1. El usuario registrado realiza el caso de uso 23.
2. El usuario registrado solicita eliminar al participante.
3. El sistema solicita al usuario registrado confirmación.
4. El usuario registrado confirma la acción.
5. El sistema elimina al participante de la base de datos e informa que todo está correcto.

Postcondiciones:

Flujo Alternativo:

- 2.1. Si no hay participantes en el sistema se termina el caso de uso.
- 4.1. El usuario registrado cancela la acción y el sistema cancela el caso de uso.

9.4. Diagrama de Casos de Uso

En los siguientes diagramas se puede observar cómo se relacionan los casos de uso entre ellos y que actores pueden hacer uso de qué funcionalidad.

9.4.1. Gestión de la Web

El diagrama de los casos de uso relacionados con la Gestión de la Web se muestra en la Figura 9.1, donde los usuarios registrados pueden realizar las tareas de los usuarios y los administradores pueden realizar las tareas de los todos los tipos de usuarios.

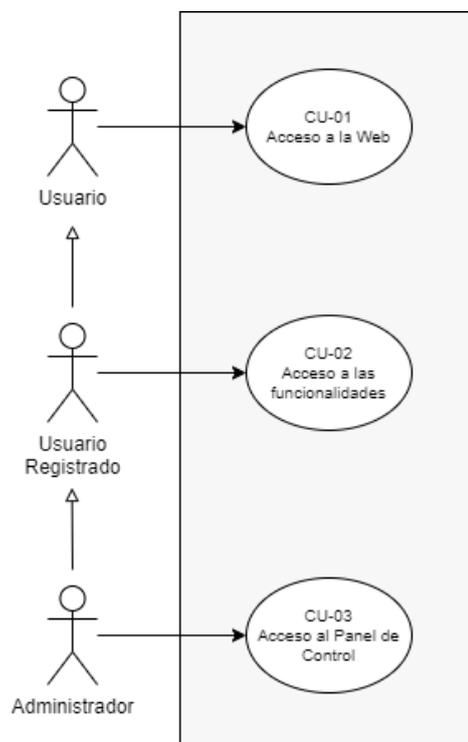


Figura 9.1: Diagrama Casos de Uso: Gestión de la Web

9. Análisis Funcional

9.4.2. Gestión de los Usuarios

A continuación, se puede observar los diagramas de los casos de uso de la Gestión de los Usuarios. Se diferencia entre los relacionados con los usuarios registrados, con los grupos y los casos de uso que controlan los grupos dentro de los usuarios registrados.

En la Figura 9.2 se observan las acciones que pueden realizar los usuarios registrados de iniciar y cerrar sesión. Los administradores pueden crear, consultar, modificar y eliminar los datos de los usuarios registrados. Algunos de los casos de uso dependen de otros para poder ser realizados.

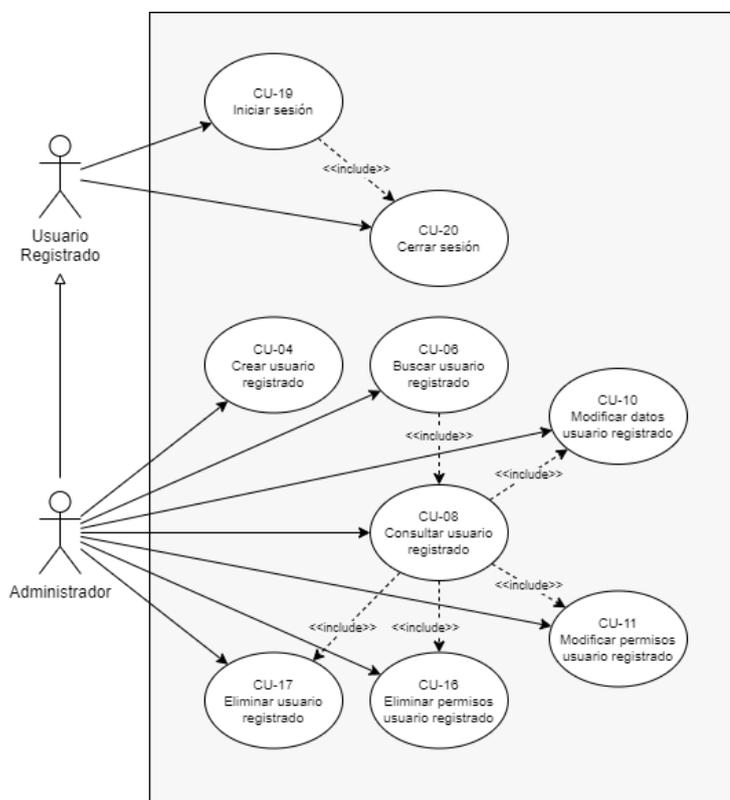


Figura 9.2: Diagrama Casos de Uso: Gestión de los Usuarios

En cuanto a la gestión que realizan los administradores sobre los grupos, es similar a la realizada con los usuarios registrados (Figura 9.3).

9.4. Diagrama de Casos de Uso

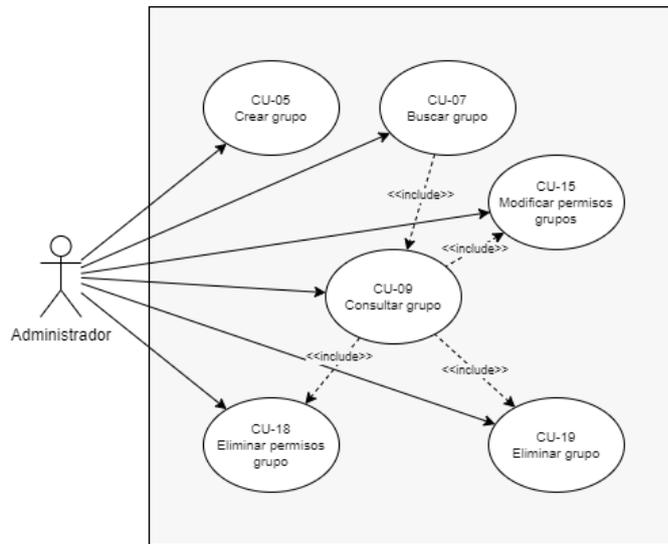


Figura 9.3: Diagrama Casos de Uso: Gestión de los Grupos

En la gestión de los usuarios, en la Figura 9.4 se observan las acciones que se pueden realizar sobre los usuarios registrados con relación a los grupos. Para estos casos de uso, el administrador debe haber realizado la consulta de los datos de un usuario registrado.

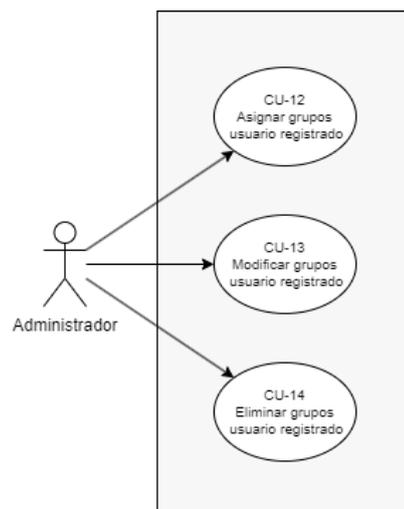


Figura 9.4: Diagrama Casos de Uso: Gestión de los Usuarios-Grupos

9. Análisis Funcional

9.4.3. Gestión de los Datos

Por último, la Figura 9.5 muestra el diagrama de los casos de uso de la Gestión de los Datos. Donde los usuarios registrados deben de haber iniciado sesión antes de poder realizar las diferentes acciones.

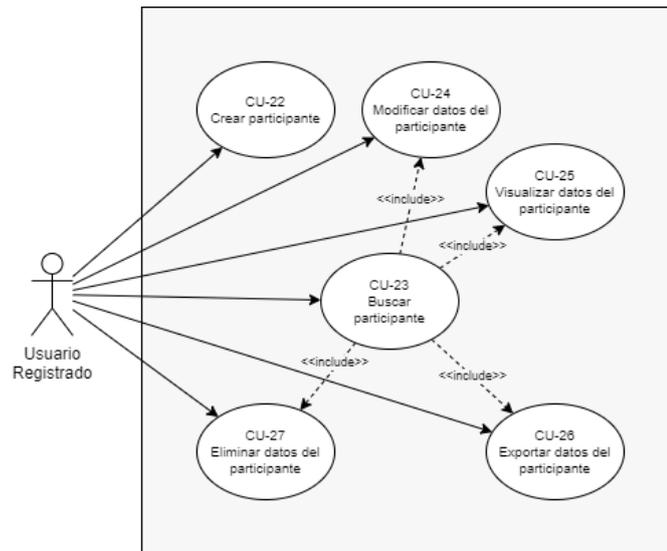
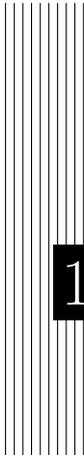


Figura 9.5: Diagrama Casos de Uso: Gestión de los Datos



10 Análisis Dinámico del Sistema

Una vez realizados los diagramas de los casos de uso, se van a realizar los diagramas de secuencia de los casos de uso más relevantes. Estos diagramas de secuencia son unos esquemas conceptuales que permiten representar de forma gráfica el comportamiento del sistema.

Los diagramas de secuencia que se han realizado no muestran con exactitud los mensajes que se mostrarían o cómo aparecería la información por pantalla. Lo que si muestra este tipo de diagramas, es el comportamiento que tiene el usuario con el sistema.

Al utilizar la arquitectura *Modelo-Vista-Controlador*, en los diagramas de secuencia aparecerán estos tres componentes y el tipo de usuario que interactúa con el sistema. El usuario interactuará con la vista del sistema, que dependiendo de las acciones que se realicen, se comunicará con el controlador y este con la base de datos (Modelo). De forma resumida, así funcionan las capas de la arquitectura:

- **Modelo:** Componente encargado de gestionar los datos del sistema. Se encarga de administrar los accesos a la información. Solo se comunica con el controlador y la base de datos.
- **Vista:** Muestra la información de la página con un formato. Es el componente con el que el usuario interactúa. Solo se comunica con el controlador.

10.1. Diagramas de Secuencia: Gestión de los Usuarios

- **Controlador:** Gestiona las acciones que realiza el usuario. Es el que realiza las peticiones de datos al modelo y proporciona la información a mostrar en la vista. Se comunica con el modelo y la vista.

10.1. Diagramas de Secuencia: Gestión de los Usuarios

A continuación se pueden observar los diagramas de secuencia de los casos de uso más importantes en la gestión de usuarios.

10.1.1. Diagrama de Secuencia CU-20: Iniciar sesión

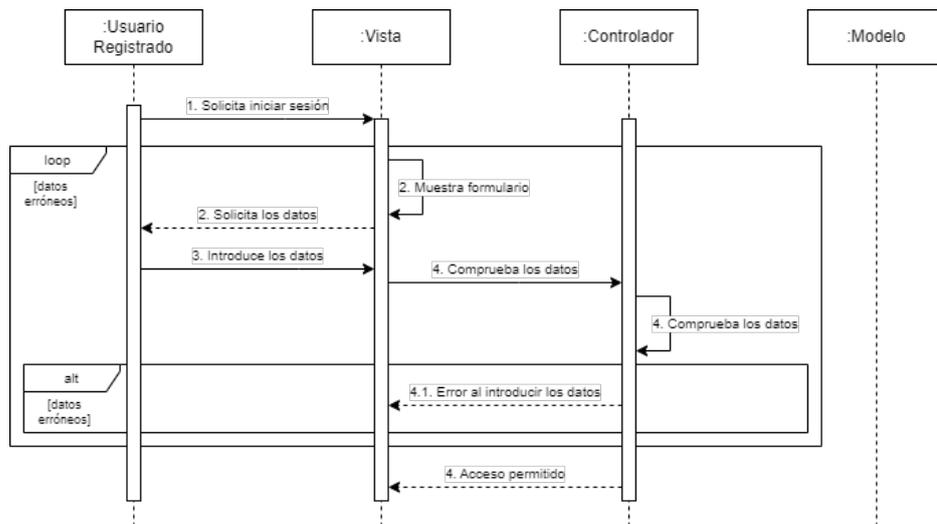


Figura 10.1: Diagrama de Secuencia CU-20

10. Análisis Dinámico del Sistema

10.1.2. Diagrama de Secuencia CU-21: Cerrar sesión

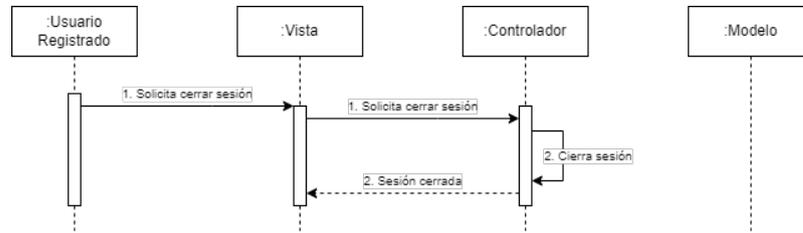


Figura 10.2: Diagrama de Secuencia CU-21

10.2. Diagramas de Secuencia: Gestión de los Datos

En esta sección se pueden observar los diagramas de secuencia de los casos de uso de la gestión de los datos.

10.2.1. Diagrama de Secuencia CU-22: Crear nuevo participante

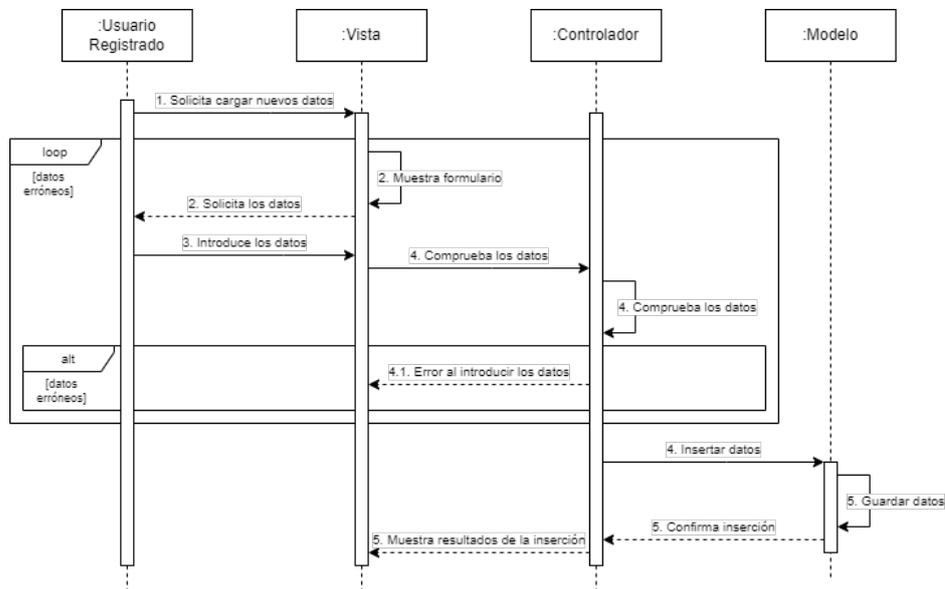


Figura 10.3: Diagrama de Secuencia CU-22

10. Análisis Dinámico del Sistema

10.2.2. Diagrama de Secuencia CU-23: Buscar participante

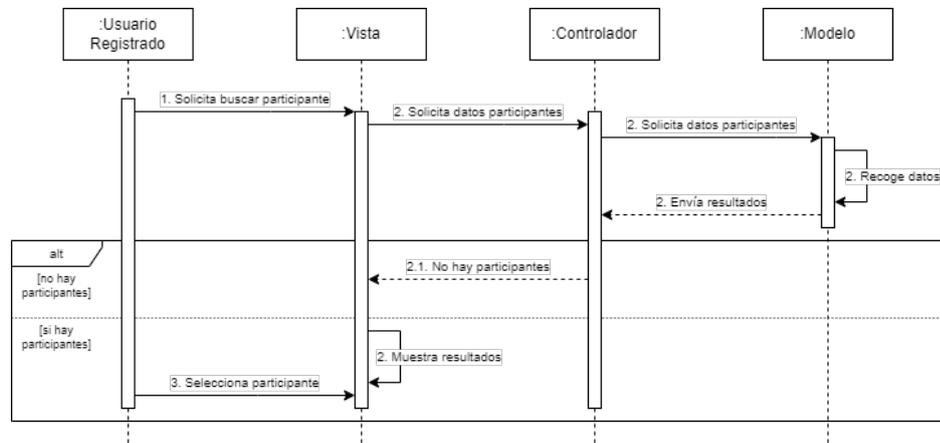


Figura 10.4: Diagrama de Secuencia CU-23

10.2. Diagramas de Secuencia: Gestión de los Datos

10.2.3. Diagrama de Secuencia CU-24: Modificar datos de un participante

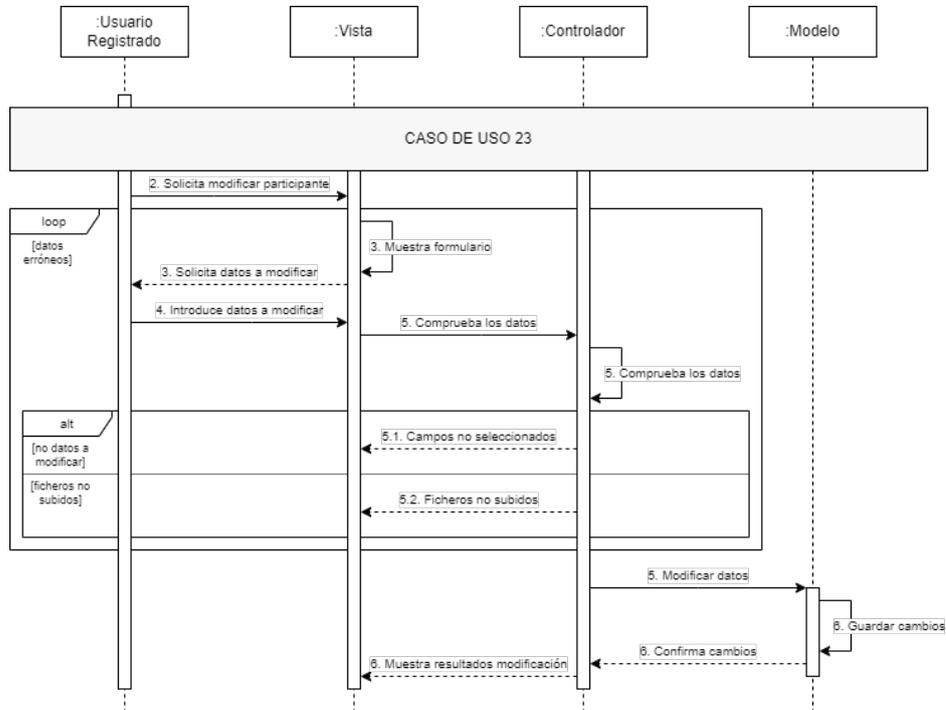


Figura 10.5: Diagrama de Secuencia CU-24

10. Análisis Dinámico del Sistema

10.2.4. Diagrama de Secuencia CU-25: Visualizar datos de un participante

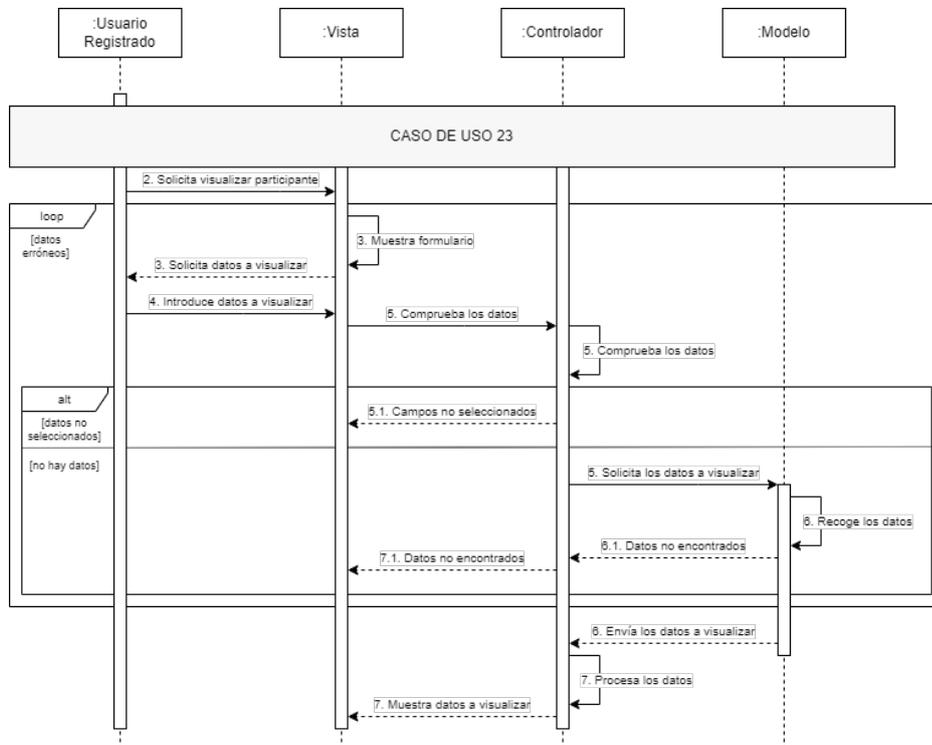


Figura 10.6: Diagrama de Secuencia CU-25

10.2. Diagramas de Secuencia: Gestión de los Datos

10.2.5. Diagrama de Secuencia CU-26: Exportar datos de los participantes

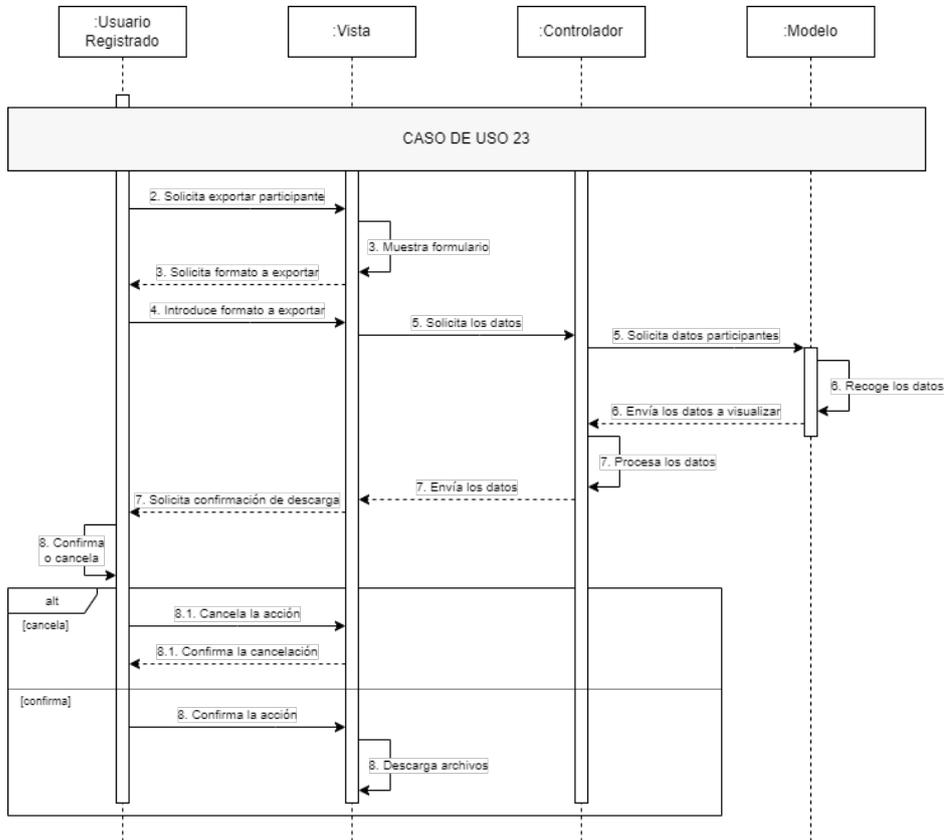


Figura 10.7: Diagrama de Secuencia CU-26

10. Análisis Dinámico del Sistema

10.2.6. Diagrama de Secuencia CU-27: Eliminar participante

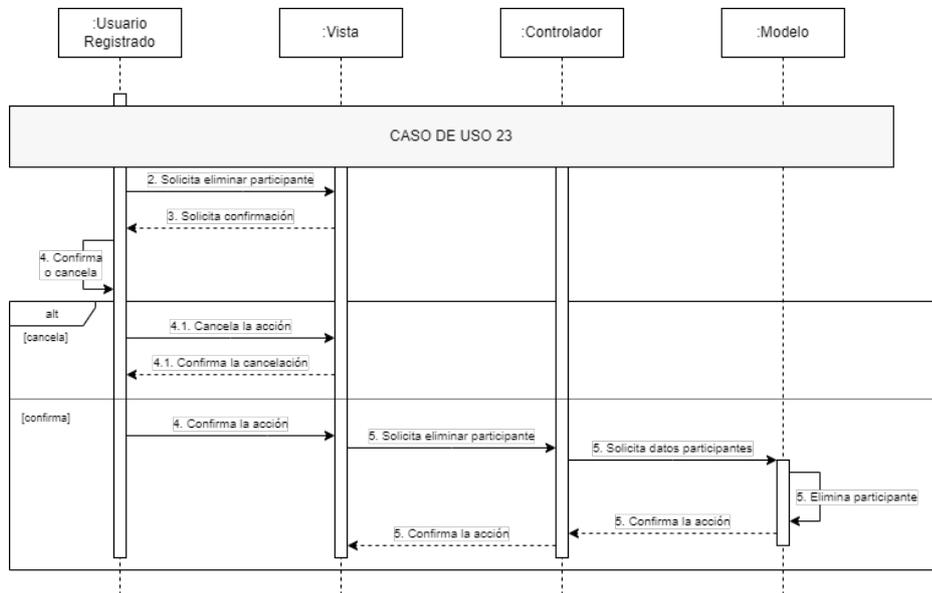
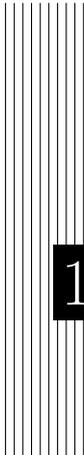


Figura 10.8: Diagrama de Secuencia CU-27



11 Herramientas Utilizadas

En este apartado se van a describir las tecnologías utilizadas para la realización y desarrollo de este Trabajo Fin de Grado (TFG), con el fin de conocer con detalle cómo funciona cada una de estas tecnologías.

11.1. Lenguaje de Programación Python

Python es un lenguaje de programación de alto nivel e interpretado. Fue creado a principios de los 90 y es un lenguaje multiparadigma, lo que permite adoptar diferentes estilos de programación, como pueden ser la programación orientada a objetos (POO), la programación interpretada o la funcional.

Es un lenguaje que se enfoca en la legibilidad del código, haciendo que sea sencillo de aprender y de utilizar. Además, es un lenguaje interpretado, lo que permite ser compilado por un intérprete, siendo más flexible y dinámico. Es también multiplataforma, permitiendo ejecutar cualquier código en diferentes sistemas operativos.

Una de las principales ventajas de utilizar Python, son las librerías y módulos que se pueden descargar. Las librerías utilizadas para el desarrollo de este proyecto se dividen en las siguientes categorías.

11.1.1. Base de la Web

La primera y más importante librería utilizada para este proyecto ha sido Django. Esta librería permite descargar el framework de desarrollo web Django, lo que favorece la creación e inicialización de una página web de forma sencilla.

También permite utilizar determinados comandos para encender la página web, conectarte a la base de datos, crear módulos dentro de la web o migrar los cambios realizados, entre otras muchas funcionalidades.

11.1.2. Lectura de Datos

Para leer los ficheros con la información sobre los participantes, se necesita la librería Json5, que, como su propio nombre indica, permite la lecturas de archivos Json (*JavaScript Object Notation*).

11.1.3. Manejo de Datos

Con la librería de Json5 se han obtenido los datos, pero a la hora de procesarlos y realizar los cálculos necesarios se hacen uso de dos librerías:

- **Numpy:** Es un librería utilizada para los cálculos numéricos y científicos, permitiendo la realización de cálculos multidimensionales y de alto nivel.
- **Pandas:** Es una librería para el análisis de los datos, así como su manejo. Con esta librería se pueden generar *datasets* que consisten en conjuntos de datos de diferentes tipos de variables.

Con estas dos librerías, los datos recogidos de los ficheros de los participantes pueden ser manejados con facilidad, realizando cálculos y cambios sobre los diferentes datos de las tablas.

11. Herramientas Utilizadas

Además, para cierto tipo de datos también se tienen que utilizar librerías internas de Python como *datetime*, para el manejo de fechas, ya que, las mismas necesitan ser manejadas con precisión.

11.1.4. Base de datos

Para almacenar los diferentes datos recopilados de los participantes, así como la información de la web, se ha optado por utilizar MongoDB. Para poder conectar este tipo de base de datos con Django se necesitan diferentes librerías:

- **Djongo:** Es la librería encargada de conectar la web con la base de datos. Permite el manejo de los datos de la base de datos, como su modificación o actualización.
- **Mongoengine:** Librería para trabajar con los datos de la base de datos en Python.
- **Pymongo:** Librería con herramientas para trabajar con MongoDB desde Python.

11.1.5. Exportación de Datos

Una vez ya están los datos recogidos, procesados y almacenados en la base de datos, tal vez, los usuarios registrados deseen exportar estos mismos datos, para ello se necesitan de diferentes librerías para crear los archivos a descargar.

Gran parte de la generación de los archivos se puede realizar con la librería de Pandas, ya que permite transformar los datasets en el formato deseado (Excel y CSV). En el caso de querer descargar un archivo Excel, no hará falta descargar nuevas librerías.

Para los archivos CSV, el proceso es más complejo, se necesita generar por cada tipo de característica un archivo separado que serán agrupados en un comprimido. Este comprimido se realiza gracias a la librería Zipp, que crea compresiones en formato *.zip*.

11.1.6. Versiones

La versión utilizada de Python ha sido la **3.9.5**. Y las versiones instaladas y utilizadas de las diferentes librerías durante el proyecto han sido las siguientes:

- **Django:** 4.1.3.
- **Json5:** 0.9.10.
- **Numpy:** 1.23.3.
- **Pandas:** 1.5.0.
- **Djongo:** 1.3.6.
- **Mongoengine:** 0.26.0.
- **Pymongo:** 3.12.1.
- **Zipp:** 3.8.1.

11.2. Web Framework Django

Django es un framework de desarrollo web basado en Python que permite realizar un desarrollo de forma rápida, limpia y siguiendo los paradigmas de diseño.

Es una herramienta que permite crear aplicaciones web desde los conceptos de forma rápida. Se encarga de diferentes tareas como la autenticación de usuarios, administración o permisos. Ofrece una seguridad ante diferentes tipos de ataques, como pueden ser a la base de datos o a información sensible. Además, es una aplicación flexible y versátil que permite escalar la web.

11.3. Base de Datos MongoDB

MongoDB es un sistema de base de datos NoSQL (*Not Only Structured Query Language*), es decir, no relacional, orientado a objetos. Los datos son almacenados en archivos BSON [11], un tipo de dato parecido al JSON pero en binario. Se ha utilizado este sistema por la naturaleza de los datos almacenados de los participantes.

Se pueden realizar todo tipo de consultas para crear datos, acceder a ellos, modificarlos o eliminarlos. Algunas de las características de este sistema son las siguientes:

- **Flexibilidad:** La herramienta permite realizar consultas de forma sencilla al estar basado en consultas de documentos. Además, posee funciones que permite realizar diferentes acciones sobre la base de datos.
- **Escalabilidad:** Permite almacenar grandes cantidades de datos, cuyos servidores se pueden ampliar de forma horizontal.
- **Indexación:** Se pueden crear índices de forma sencilla, además de que se pueden administrar. De esta forma, se mejora el rendimiento de las consultas.
- **Almacenamiento:** Permite trabajar como un sistema de ficheros, lo que facilita la forma de almacenar nuevos documentos.

Durante la realización del TFG se ha hecho uso de la versión **6.0.5**.

11.3.1. MongoDB Compass

MongoDB Compass [8] es una de las varias herramientas gratuitas que ofrece MongoDB para manejar, optimizar y analizar los datos almacenados. Esta herramienta ofrece las siguientes posibilidades:

- **Patrones:** La herramienta analiza los documentos y ofrece unas estadísticas y unos esquemas para entender los datos almacenados de una forma más profunda.

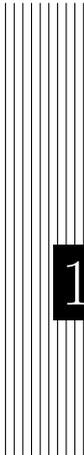
11.3. Base de Datos MongoDB

- **Consultas:** El buen rendimiento de la herramienta permite realizar consultas de forma rápida y sin un gran coste.
- **Ejecución:** Permite añadir índices y eliminar aquellos que sean irrelevantes, monitorizando el servidor y las bases de datos, para poder encontrar problemas de rendimiento.
- **Almacenar:** Se pueden agregar nuevos datos de forma sencilla gracias a cómo está construido. Pudiendo analizar y previsualizar los datos a medida que van siendo introducidos.

Durante la realización del TFG se ha hecho uso de la versión **1.36.4**.

Parte III

Diseño del Sistema



12 Diseño de los Datos

Tras definir los objetivos y requisitos del proyecto, así como el funcionamiento de este, se realizará el diseño del sistema, empezando por el diseño de los datos. Lo primero que se realizará, será una descripción de los datos almacenados en la base de datos.

Como se trabajará con MongoDB, una base de datos no relacional, la información se almacenará en formato BSON. Este formato consiste en una lista ordenada de elementos, donde cada elemento tiene un nombre, un tipo y un valor, ya sea numérica, literal, *boolean* u otra lista ordenada dentro del mismo, entre otros tipos.

12.1. Análisis de la Información

A continuación, se va a definir la estructura de la información almacenada de los participantes. Esta información ha sido previamente procesada para obtener exclusivamente la información útil durante el periodo de seguimiento. Esto se debe a que muchos de los datos recogidos por las pulseras Fitbit no contienen información relevante.

Antes de definir la estructura de los datos, se van a definir los tipos de variables especiales que son utilizados para recoger la información.

12.1. Análisis de la Información

- ***ObjectId***: Este tipo de variable utilizada en MongoDB es única y se compone de 12 bytes de información (4 bytes de la hora de creación, 5 bytes de valores random y 3 bytes de contador). Este valor será generado para identificar los elementos de una colección.

El participante contendrá los siguientes datos que conformarán los documentos con todos los datos de los participantes.

- ***_id***: Identificador interno de la base de datos de tipo *ObjectId*. Este elemento se genera automáticamente cada vez que se agrega un nuevo participante al sistema.
- ***iden***: Identificador asociado por la herramienta al crear un nuevo participante de tipo *String*. Este identificador no tiene por qué ser único.
- ***name***: Nombre otorgado por el usuario registrado al crear un nuevo participante y de tipo *String*.
- ***startDate***: Fecha de inicio del seguimiento de tipo *Timestamp*. Este valor puede ser introducido por el usuario registrado al crear un nuevo participante o al subir un archivo con la información acerca del seguimiento.
- ***endDate***: Fecha del final del seguimiento de tipo *Timestamp*. Este valor puede ser introducido por el usuario registrado al crear un nuevo participante o al subir un archivo con la información acerca del seguimiento.
- ***dataSaved***: Lista de objetos con la información de la actividad física del participante.
 - ***calories***: Contendrá una lista de elementos con los siguientes valores:
 - ***Session***: Valor numérico entero que representa el día del seguimiento.

12. Diseño de los Datos

- **Date:** *String* que especifica el día en el que se han recogido los datos.
- **Day:** *String* que especifica el día de la semana en el que se han recogido los datos.
- **Treatment:** *String* para marcar el tipo de tratamiento aplicado al participante.
- **Hours:** Lista de valores decimales que representan el valor de las calorías quemadas por el participante a cada hora del día (*Hour 1 - Hour 2 - ... - Hour 24*).
- **distance:** Contendrá una lista de elementos con los siguientes valores:
 - **Session:** Valor numérico entero que representa el día del seguimiento.
 - **Date:** *String* que especifica el día en el que se han recogido los datos.
 - **Day:** *String* que especifica el día de la semana en el que se han recogido los datos.
 - **Treatment:** *String* para marcar el tipo de tratamiento aplicado al participante.
 - **Hours:** Lista de valores decimales que representan el valor de la distancia recorrida por el participante a cada hora del día (*Hour 1 - Hour 2 - ... - Hour 24*).
- **steps:** Contendrá una lista de elementos con los siguientes valores:
 - **Session:** Valor numérico entero que representa el día del seguimiento.
 - **Date:** *String* que especifica el día en el que se han recogido los datos.
 - **Day:** *String* que especifica el día de la semana en el que se han recogido los datos.
 - **Treatment:** *String* para marcar el tipo de tratamiento aplicado al participante.

12.1. Análisis de la Información

- **Hours:** Lista de valores decimales que representan el valor de los pasos realizados por el participante a cada hora del día (*Hour 1 - Hour 2 - ... - Hour 24*).
- **heart_rate:** Contendrá una lista de elementos con los siguientes valores:
 - **Session:** Valor numérico entero que representa el día del seguimiento.
 - **Date:** *String* que especifica el día en el que se han recogido los datos.
 - **Day:** *String* que especifica el día de la semana en el que se han recogido los datos.
 - **Treatment:** *String* para marcar el tipo de tratamiento aplicado al participante.
 - **Hours:** Lista de valores decimales que representan el valor de la frecuencia cardíaca del participante a cada hora del día (*Hour 1 - Hour 2 - ... - Hour 24*).
- **resting_heart_rate:** Contendrá una lista de elementos con los siguientes valores:
 - **Session:** Valor numérico entero que representa el día del seguimiento.
 - **Date:** *String* que especifica el día en el que se han recogido los datos.
 - **Day:** *String* que especifica el día de la semana en el que se han recogido los datos.
 - **BPM:** Valor decimal que muestra la media de la frecuencia cardíaca durante ese día.
 - **Error:** Valor decimal que muestra el posible error recogido de las frecuencias cardíacas durante ese día.
- **time_heart_rate_zones:** Contendrá una lista de elementos con los siguientes valores:
 - **Session:** Valor numérico entero que representa el día del seguimiento.

12. Diseño de los Datos

- **Date:** *String* que especifica el día en el que se han recogido los datos.
- **Day:** *String* que especifica el día de la semana en el que se han recogido los datos.
- **Zones:** Valores numéricos enteros que muestran el número de minutos que el participante ha pasado en las zonas de las frecuencias cardíacas durante ese día (*Zone 3, Zone 2, Zone 1 y Below Zone 1*).
- **Sume:** Valor numérico entero que representa la suma de los minutos de cada zona de ese día.
- **active_minutes:** Contendrá una lista de elementos con los siguientes valores:
 - **Session:** Valor numérico entero que representa el día del seguimiento.
 - **Date:** *String* que especifica el día en el que se han recogido los datos.
 - **Day:** *String* que especifica el día de la semana en el que se han recogido los datos.
 - **Minutes:** Valores numéricos enteros que muestran el número de minutos que el participante ha pasado en las diferentes zonas de actividad (*Lightly Active Minutes, Moderately Active Minutes, Very Active Minutes y Sedentary Minutes*).
 - **Sume:** Valor numérico entero que representa la suma de los minutos de cada zona de ese día.
- **sleep:** Contendrá una lista de elementos con los siguientes valores:
 - **Session:** Valor numérico entero que representa el día del seguimiento.
 - **Date:** *Timestamp* que especifica el día en el que se han recogido los datos.
 - **Start Time:** *String* que especifica el día y la hora de

12.1. Análisis de la Información

inicio de la actividad de sueño del participante en ese día.

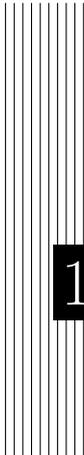
- **End Time:** *String* que especifica el día y la hora final de la actividad de sueño del participante en ese día.
- **Duration:** Valor numérico entero que representa los milisegundos que el participante ha estado durmiendo.
- **Minutes Asleep:** Valor numérico entero que representa los minutos que el participante ha estado durmiendo en la actividad del sueño de ese día.
- **Minutes Awake:** Valor numérico entero que representa los minutos que el participante ha estado despierto en la actividad del sueño de ese día.
- **Time in Bed:** Valor numérico entero que representa los minutos que el participante ha estado durmiendo en ese día.
- **Counts:** Valor numérico entero que representa el número de veces que ha pasado en la fase del sueño en específico (*Deep, Wake, Light* y *REM Count*).
- **Minutes:** Valor numérico entero que representa los minutos que ha pasado en la fase del sueño en específico (*Deep, Wake, Light* y *REM Minutes*).
- **Thirty day average minutes:** Valor numérico entero que representa la media de los minutos que ha pasado en la fase del sueño durante los últimos 30 días (*Deep, Wake, Light* y *REM Thirty day average minutes*).

12.2. Modelo No Relacional

Las bases de datos no relacionales poseen un sistema de almacenamiento que no utiliza SQL (*Structured Query Language*). Las ventajas de utilizar un modelo así son las siguientes:

- Fácil gestión de información no estructurada.
- Tiene escalabilidad que permite ampliar los volúmenes de datos almacenados, añadiendo nuevos servidores de forma sencilla.
- Es un modelo de datos flexible, permitiendo una variedad de tipos de variables.
- Está diseñado para el manejo de grandes volúmenes de datos.

Al hacer uso de un modelo no relacional, los diferentes documentos creados no guardan relación con otros documentos de esta o de otra colección.



13 Diseño Arquitectónico

El diseño de la arquitectura del sistema tiene como objetivo planificar el desarrollo del proyecto y es la base del diseño de las diferentes funcionalidades.

Para ello, se va a utilizar un patrón de diseño. Los patrones consisten en soluciones generales, que son efectivos al resolver problemas similares y reutilizables al aplicarse en diferentes problemas de diseño. Las ventajas de los patrones son la reutilización de soluciones, el beneficio del conocimiento, el establecer un vocabulario común y no reinventar.

13.1. Patrón Modelo-Vista-Controlador

Este patrón [1] se compone de tres componentes que trabajan conjuntamente, separando la lógica de negocio, el manejo de datos y la interacción del usuario con el sistema (Figura 13.1).

- **Modelo:** Componente que gestiona los datos almacenados en la base de datos, realizando las consultas y modificaciones de estos. Se comunica únicamente con el controlador.
- **Vista:** Componente que gestiona la presentación y las inter-

13.1. Patrón Modelo-Vista-Controlador

acciones del usuario con el sistema, manejando las vistas, sus formatos y la representación de la información. Se suscribe a los cambios que determina el controlador.

- **Controlador:** Es el intermediario entre la vista y el modelo. Es el componente que define los cambios del modelo al recibir las acciones de la vista. Además, define los cambios y el comportamiento de la vista.

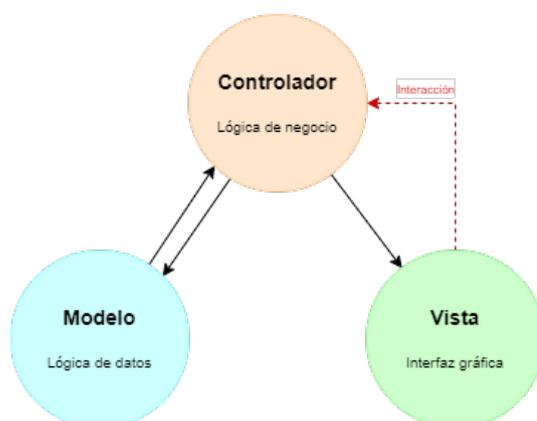
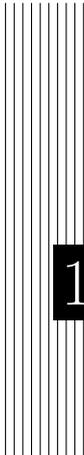


Figura 13.1: Patrón *Modelo-Vista-Controlador*

El patrón funciona de forma unidireccional, donde las acciones siguen un flujo. Una vez el usuario ha interactuado con la vista, el controlador es notificado del acto y realiza las funciones que son solicitadas por el usuario. En caso de que sea necesario, se comunicará con el modelo para la obtención o modificación de los datos. Finalmente, el controlador realizará los cambios pertinentes sobre la vista.



14 Diseño de Clases

Este capítulo tiene por objetivo especificar y describir las clases a implementar en el sistema, así como las relaciones entre ellas. Al trabajar con el patrón *Modelo-Vista-Controlador*, la lógica de datos y la lógica de negocio estarán separadas.

Al trabajar con Django, muchas de las clases y funciones para el manejo de la información de los usuarios y administradores del panel de control, son controladas por el propio framework. En cambio, el manejo de los datos de los participantes y de las vistas creadas sí que requiere de desarrollo.

Se necesitan diferentes clases para el procesado de los diferentes tipos de variables recopiladas de los participantes. Pero todas estas clases tienen una estructura común de funciones, diferenciándose en cómo procesa cada una la información internamente, sin depender de otras clases. Por ello, se van a describir las dos clases que si se diferencian.

14.1. Clase VariablesSum

Existen tres variables cuyos procesamientos son idénticos, ya que la estructura de los datos recogidos es la misma y el método para procesarlos también. Esta clase es utilizada para procesar los datos de las variables *Calories*, *Distance* y *Steps*.

Clase: VariablesSum		
Clase que procesará la información recopilada de las variables <i>Calories</i> , <i>Distance</i> y <i>Steps</i> .		
Datos		
- valid_files	array	Guarda los archivos válidos de las diferentes variables.
- valid_files_type	array	Guarda de los <i>valid_files</i> los archivos de la variable en específico que se va a procesar.
- startDate	datetime	Guarda la fecha inicial del seguimiento del participante.
- endDate	datetime	Guarda la fecha final del seguimiento del participante.
- data_json	json	Guarda los valores procesados en formato json.
- data_matrix	pandas.DataFrame	Guarda los valores procesados en formato DataFrame.
Métodos		
+ __init__		Método constructor de la clase.
+ set_type	void	Función que determina el tipo de datos a procesar.
+ get_data	void	Función que procesa los datos introducidos.
+ get_data_json	array	Función que devuelve un json de los datos procesados.

Tabla 14.1: Especificación de la clase VariablesSum

14. Diseño de Clases



Figura 14.1: Diagrama de clases de la clase VariablesSum

14.2. Clase Variable

En este apartado se van a describir las clases que tienen una misma estructura, pero que se diferencian por la forma en que procesa la información. Existe un tipo de clase para cada variable cuyo procesamiento no consista en sumar valores. Estas clases son: *Sleep*, *Heart_Rate*, *Resting_Heart_Rate*, *Time_Heart_Rate_Zones* y *Active_Minutes*.

Clase: Variable		
Clase que procesará la información recopilada de las variables <i>Sleep</i> , <i>Heart_Rate</i> , <i>Resting_Heart_Rate</i> , <i>Time_Heart_Rate_Zones</i> y <i>Active_Minutes</i> .		
Datos		
- valid_files	array	Guarda los archivos válidos de la variable a procesar.
- startDate	datetime	Guarda la fecha inicial del seguimiento del participante.
- endDate	datetime	Guarda la fecha final del seguimiento del participante.
- data_json	json	Guarda los valores procesados en formato json.
- data_matrix	pandas.DataFrame	Guarda los valores procesados en formato DataFrame.
Métodos		
+ __init__		Método constructor de cada clase.
+ get_data	void	Función que procesa los datos introducidos.
+ get_data_json	array	Función que devuelve un json de los datos procesados.

Tabla 14.2: Especificación de las clases Variable

14. Diseño de Clases

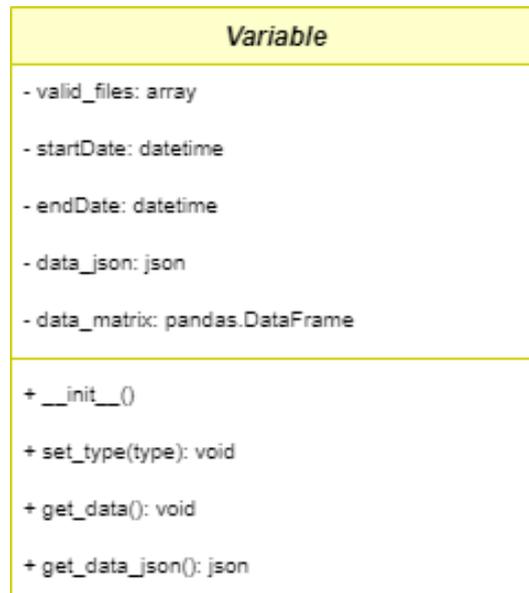
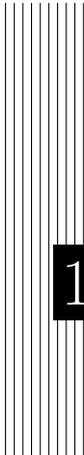


Figura 14.2: Diagrama de clases de las clases Variable



15 Diseño de la Interfaz

En este capítulo se presenta la interfaz gráfica de la aplicación web. Esta interfaz es un entorno visual mediante el cual, los usuarios interactúan con el sistema. Para la realización de esta, se seguirán los requisitos definidos sobre cómo se presentará y qué características tendrá.

Los usuarios que visiten la página solo podrán acceder a la página principal y a la vista de inicio de sesión. Mientras que los usuarios registrados tendrán la capacidad de acceder a las diferentes funcionalidades que ofrece la web. Además, los administradores podrán hacer uso del panel de control.

15.1. Vistas

A continuación, se presentarán las vistas con la que los usuarios registrados podrán interactuar y se indicarán los diferentes elementos que las componen. Tras analizar estas vistas, se mostrará un mapa de navegación, para entender de una forma más clara desde qué vistas se pueden acceder a qué otras vistas.

15.1.1. Vista *Home*

Vista principal de la página web a la cual podrán acceder todos los tipos de usuarios y se mostrará la información relevante acerca del proyecto.

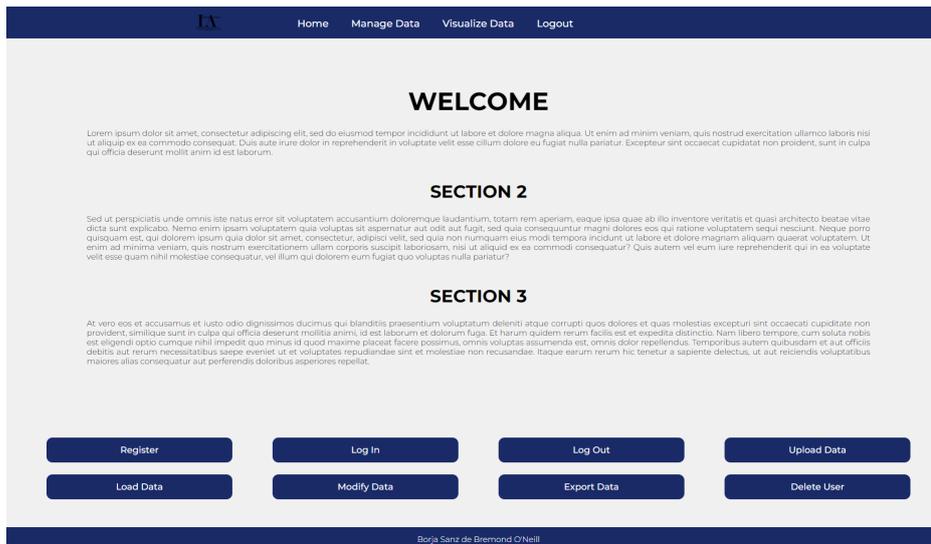


Figura 15.1: Vista *Home*

- **Cabecera:** Acceso a las diferentes funcionalidades a las que pueden acceder los usuarios registrados.
- **Información:** Información relevante acerca del proyecto.
- **Botones:** Permite la navegación a través de la aplicación web.

15. Diseño de la Interfaz

15.1.2. Vista *Login*

Vista donde el usuario registrado podrá iniciar sesión al introducir sus credenciales.

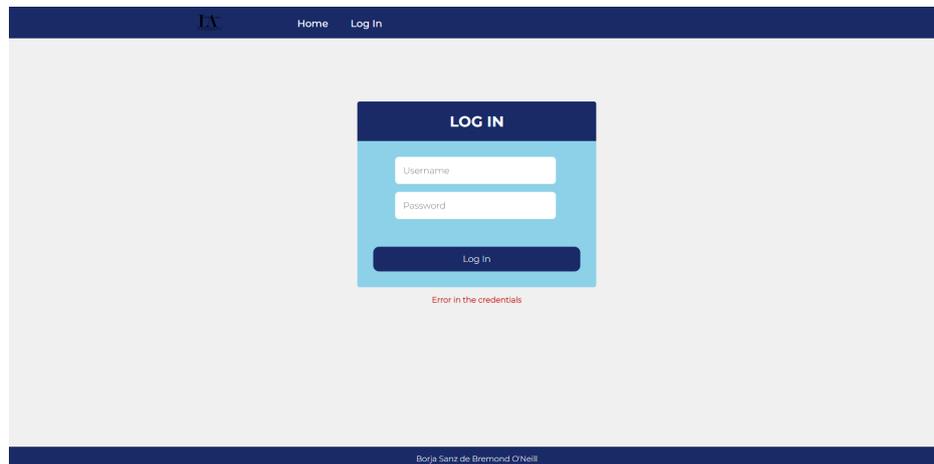


Figura 15.2: Vista *Login*

- **Cabecera:** Acceso a las diferentes funcionalidades a las que pueden acceder los usuarios registrados.
- **Formulario:** El usuario registrado deberá introducir sus credenciales para iniciar sesión: Nombre y contraseña.
- **Login:** Acción de iniciar sesión.
- **Error:** Mensaje de error que aparecerá al introducir mal las credenciales.

15.1.3. Vista *Logout*

Vista donde el usuario registrado podrá cerrar sesión al pulsar el botón. Este botón aparecerá en la cabecera de todas las vistas cuando el usuario registrado haya iniciado sesión.

15.1.4. Vista *UploadData*

Desde esta ventana los usuarios registrados serán capaces de subir los datos de un nuevo participante, que será creado al rellenar el formulario.

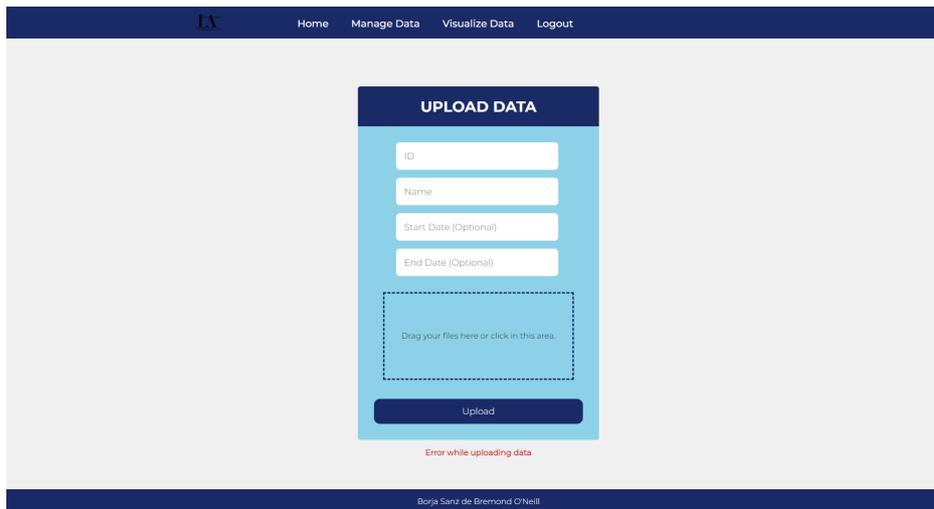


Figura 15.3: Vista *UploadData*

- **Cabecera:** Acceso a las diferentes funcionalidades a las que pueden acceder los usuarios registrados.
- **Identificador y Nombre:** Campo obligatorio del formulario para identificar de forma sencilla a los participantes.
- **Fechas:** Campo opcional del formulario para especificar las fechas de inicio y final del seguimiento del participante.
- **Archivos:** Elemento para cargar los archivos con la información del participante.
- **Cargar:** Acción de cargar los datos al sistema y crear un nuevo participante.
- **Error:** Mensaje de error que aparecerá al introducir mal los datos o haya ocurrido algún error.

15. Diseño de la Interfaz

Vista *SuccessUpload*

En esta vista se observan diferentes mensajes mostrando que el nuevo participante ha sido creado correctamente, con los tipos de datos que han sido subidos al sistema.

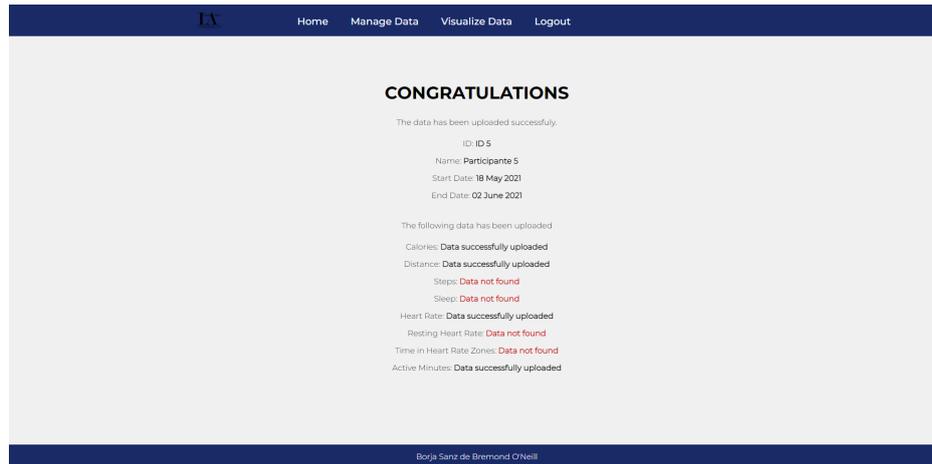


Figura 15.4: Vista *SuccessUpload*

- **Cabecera:** Acceso a las diferentes funcionalidades a las que pueden acceder los usuarios registrados.
- **Participante:** Se muestra la información relevante del participante: Identificador, nombre y fechas de inicio y fin del seguimiento.
- **Datos:** Se muestra los diferentes tipos de datos con un mensaje especificando si el participante contiene ese tipo de datos.

15.1.5. Vista *LoadData*

Vista para la visualización de los datos procesados de los participantes. Desde aquí se puede seleccionar al participante y el tipo de dato a visualizar.

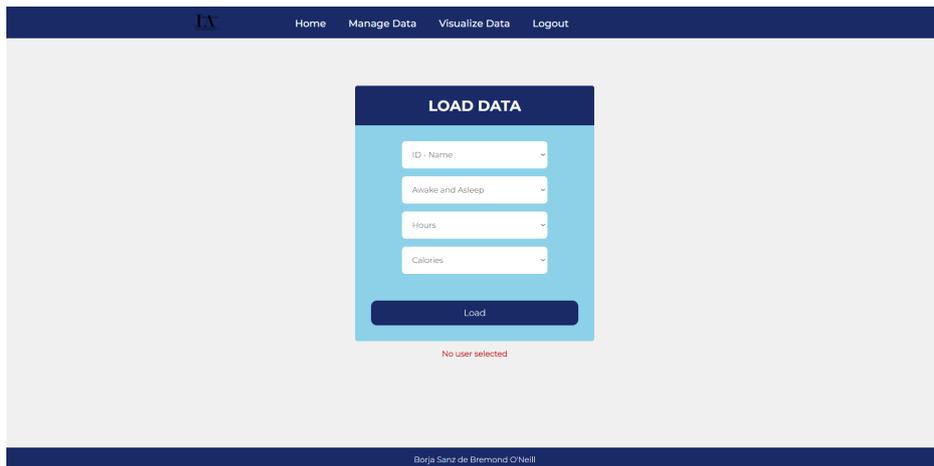


Figura 15.5: Vista *LoadData*

- **Cabecera:** Acceso a las diferentes funcionalidades a las que pueden acceder los usuarios registrados.
- **Formulario:** Se especificará el participante a visualizar, la franja horaria, la medida de tiempo para mostrar la información y el tipo de variable.
- **Cargar:** Acción de visualizar los datos de un participante.
- **Error:** Mensaje de error que aparecerá al no seleccionar un participante o al intentar visualizar un tipo de dato que el participante no contiene.

15. Diseño de la Interfaz

Vista *VisualizeData*

Se muestra la información del participante seleccionado. El usuario registrado será capaz de seleccionar dentro de esta ventana el tipo de información o el cómo quiere visualizarla sin la necesidad de ir a la vista *LoadData*.

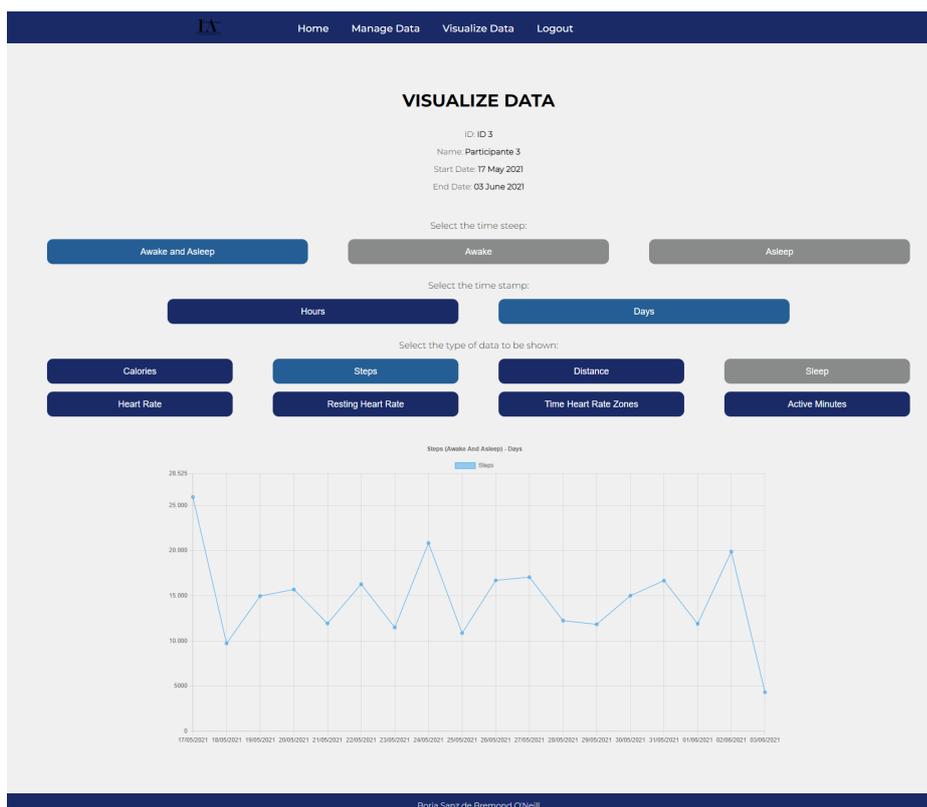


Figura 15.6: Vista *VisualizeData*

- **Cabecera:** Acceso a las diferentes funcionalidades a las que pueden acceder los usuarios registrados.
- **Participante:** Muestra la información relevante del participante: Identificador, nombre y fechas de inicio y fin del seguimiento.
- **Variables:** Accesos a las diferentes variaciones de la gráfica, como son el tipo de variable, franja horaria o medida de tiempo. Los botones en grises son aquellas variables que no se pueden

combinar o que no están disponibles por la falta de datos del participante.

- **Gráfica:** Muestra los datos de la variable seleccionada en forma de gráfica, depende también de los parámetros marcados. Cuando existen más de fecha mostrada, se pueden seleccionar aquellas que se quieren mostrar o no.

15.1.6. Vista *ModifyData*

En esta vista el usuario registrado podrá modificar los datos de los participantes al seleccionar el participante, los datos a modificar y subir los archivos correspondientes.

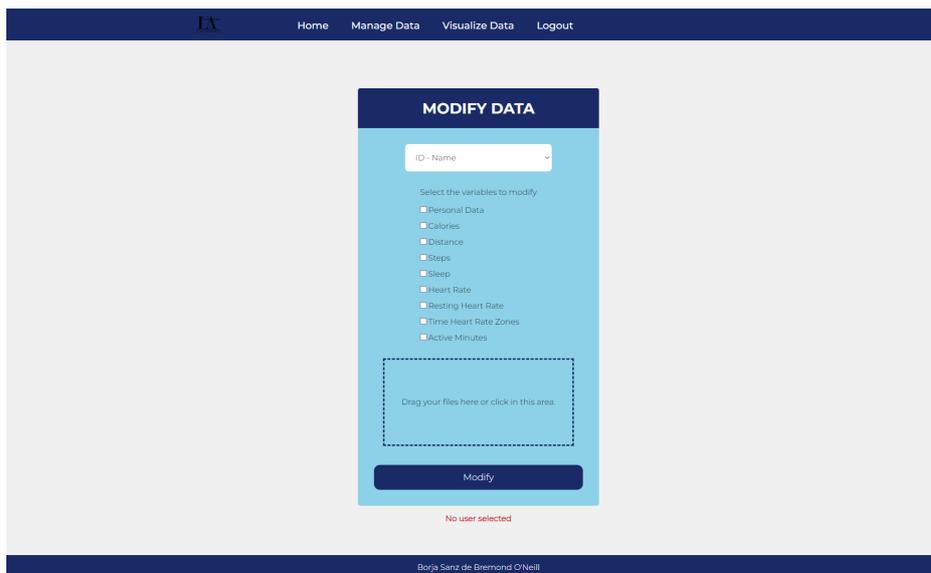


Figura 15.7: Vista *ModifyData*

- **Cabecera:** Acceso a las diferentes funcionalidades a las que pueden acceder los usuarios registrados.
- **Participante:** Se selecciona el participante al que se le modificará los datos.
- **Formulario:** Se seleccionan los tipos de datos a modificar.

15. Diseño de la Interfaz

- **Archivos:** Elemento para cargar los archivos con la información del participante.
- **Modificar:** Acción de modificar los datos del participante.
- **Error:** Mensaje de error que aparecerá al no seleccionar un participante, no seleccionar los datos a modificar o algún error al tratar con los archivos subidos.

Vista *SuccessModify*

En esta vista se observan diferentes mensajes mostrando que el participante ha sido modificado correctamente, con los tipos de datos que han sido subidos al sistema.

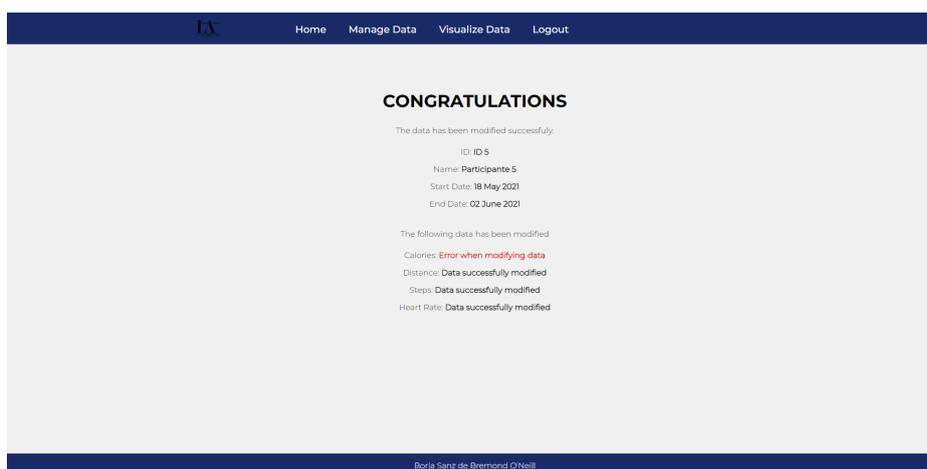


Figura 15.8: Vista *SuccessModify*

- **Cabecera:** Acceso a las diferentes funcionalidades a las que pueden acceder los usuarios registrados.
- **Participante:** Se muestra la información relevante del participante: Identificador, nombre y fechas de inicio y fin del seguimiento.
- **Datos:** Se muestra una lista con los tipos de datos que han sido seleccionados en el formulario anterior, especificando si los datos han sido modificados o no.

15.1.7. Vista *ExportData*

Desde esta vista los usuarios registrados serán capaces de descargar los datos de uno o de todos los participantes de una forma sencilla.

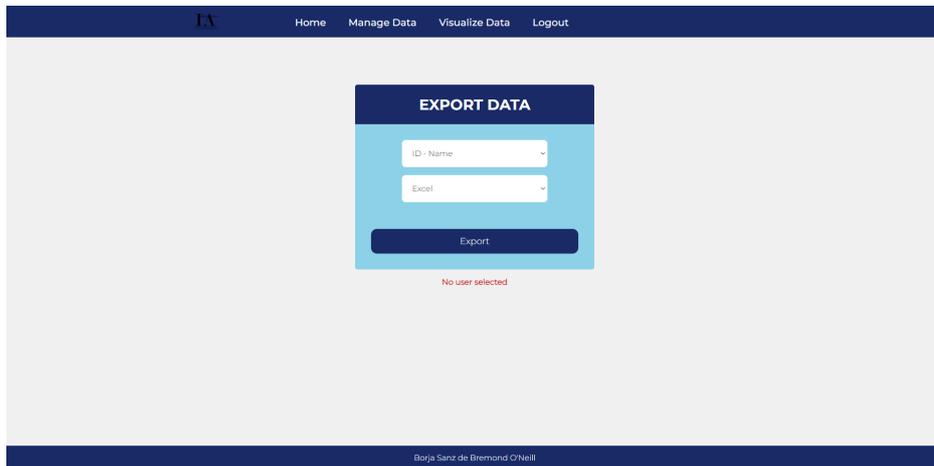


Figura 15.9: Vista *ExportData*

- **Cabecera:** Acceso a las diferentes funcionalidades a las que pueden acceder los usuarios registrados.
- **Participante:** Se selecciona el participante o todos los participantes cuyos datos serán exportados.
- **Formato:** Se selecciona el formato a descargar los archivos (*.xls* o *.csv*).
- **Exportar:** Acción de exportar los datos del participante.
- **Error:** Mensaje de error que aparecerá al no seleccionar un participante.

15. Diseño de la Interfaz

15.1.8. Vista *DeleteData*

Vista donde el usuario registrado puede eliminar todos los datos de un participante.

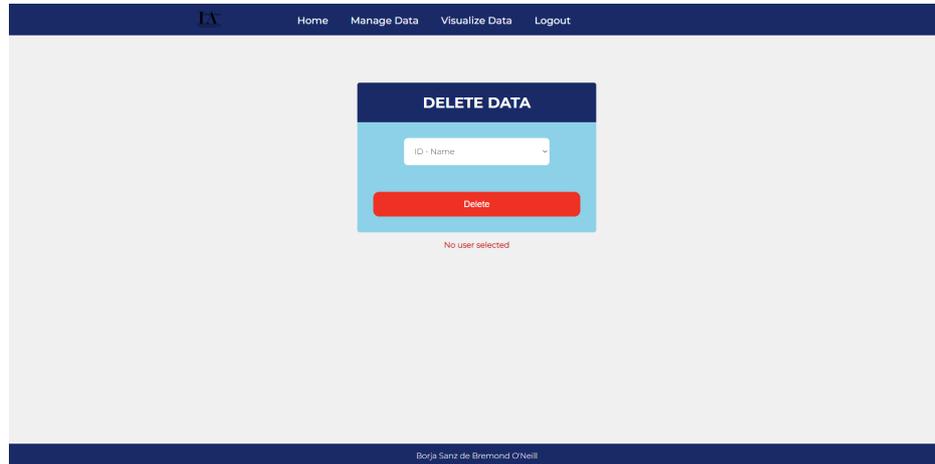


Figura 15.10: Vista *DeleteData*

- **Cabecera:** Acceso a las diferentes funcionalidades a las que pueden acceder los usuarios registrados.
- **Participante:** Se selecciona el participante que va a ser eliminado.
- **Eliminar:** Acción de eliminar al participante. Al activarla, aparecerá una ventana emergente pidiendo una confirmación sobre la acción.
- **Error:** Mensaje de error que aparecerá al no seleccionar a un participante.

Confirmation

Ventana emergente que aparece sobre la vista *DeleteData*, solicitando confirmación sobre la acción de eliminar toda la información de un participante.

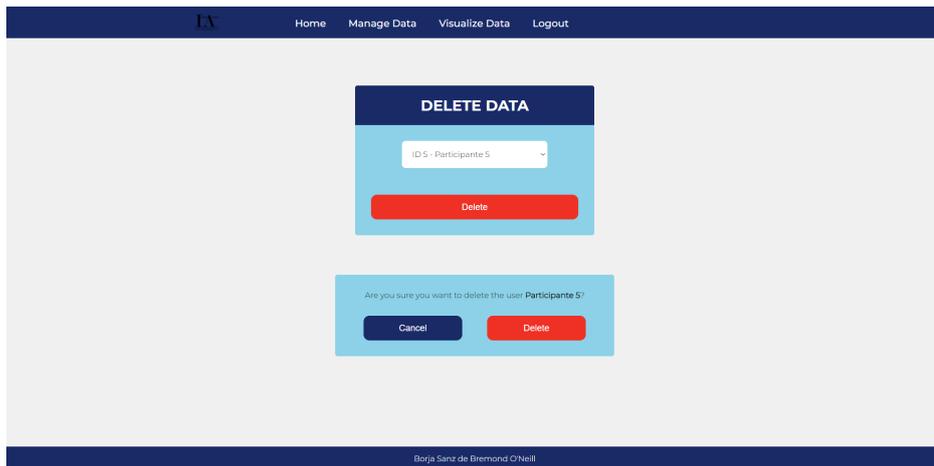


Figura 15.11: Vista *Confirmation*

- **Cabecera:** Acceso a las diferentes funcionalidades a las que pueden acceder los usuarios registrados.
- **Mensaje:** Se muestra un texto pidiendo una confirmación por parte del usuario registrado.
- **Cancelar:** Acción de cancelar el eliminar al participante. Una vez cancelado, desaparecerá la ventana emergente.
- **Confirmar:** Acción de confirmar el eliminar al participante. Una vez eliminado el participante, desaparecerá la ventana emergente y aparecerá un mensaje confirmando la acción en la misma posición que el mensaje de error.

15.2. Mapa de Navegación

Los mapas de navegación son representaciones gráficas de las vistas de una página web, representando la jerarquía de las páginas para que los usuarios accedan de forma intuitiva.

15.2.1. Usuario

En la Figura 15.12 se puede observar el mapa de navegación de los usuarios que visitan la página web. Al no tener permisos, este tipo de usuarios pueden visitar la vista inicial (*Home*) y el *Login*, en el caso de que quiera iniciar sesión ya que realmente es un usuario registrado.



Figura 15.12: Mapa navegación: Usuario

15.2.2. Usuario Registrado

En el caso de que sea un usuario registrado el que visita la página, el mapa de navegación es más complejo (Figura 15.13). Lo primero que tendrá que hacer el usuario registrado es iniciar sesión. Hasta este momento el mapa de navegación es igual al del usuario, pero una vez ha iniciado sesión, el mapa de navegación se amplía.

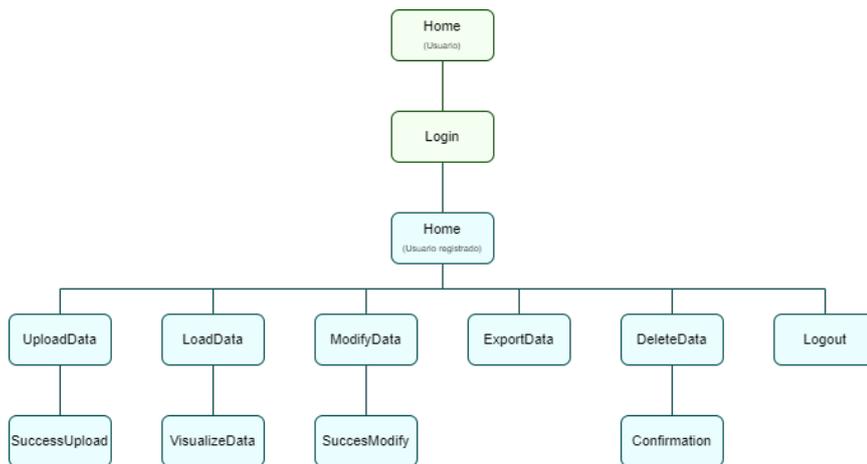


Figura 15.13: Mapa navegación: Usuario Registrado

Una vez el usuario registrado haya iniciado sesión, volverá a la vista *Home* que tendrá ciertas diferencias con la vista *Home* de los usuarios no registrados, ya que tendrá los accesos directos a las diferentes funcionalidades en el menú de navegación.

En todo momento, sin importar la vista en la que se encuentre el usuario registrado, tendrá la posibilidad de acceder a cualquier funcionalidad desde el menú de navegación, excepto por aquellas vistas que se encuentran en la última capa, ya que estas vistas solo pueden ser accedidas por su padre, jerárquicamente hablando.

15. Diseño de la Interfaz

15.2.3. Administrador

Los administradores podrán navegar por la web tal como lo hacen los usuarios registrados y además podrán acceder al panel de control, desde el cuál tendrán acceso a nuevas funcionalidades para la gestión de los usuarios (Figura 15.14).

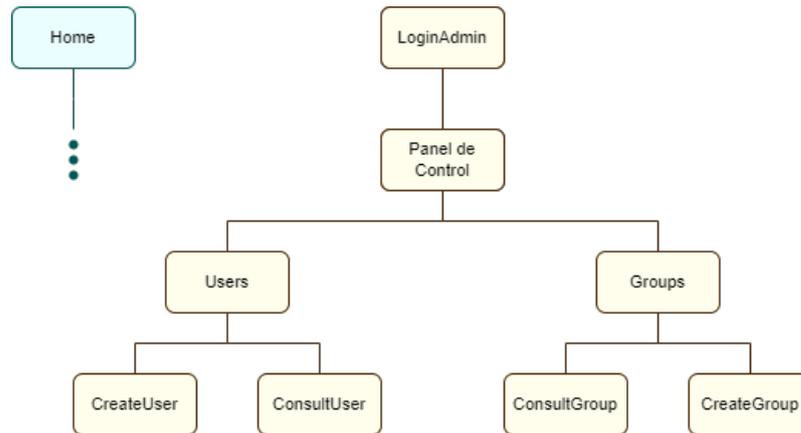
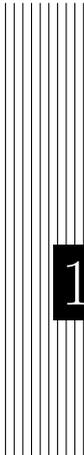


Figura 15.14: Mapa navegación: Administrador

Parte IV

Pruebas



16 Pruebas

Para comprobar el correcto funcionamiento de la herramienta web, se han realizado una serie de pruebas y validaciones. Estas pruebas tienen como objetivo encontrar errores para poder solucionarlos y de esta forma, garantizar la calidad del Trabajo Fin de Grado (TFG). Las pruebas realizadas se han basado en los siguientes tipos:

- **Pruebas Funcionales:** Se han comprobado las diferentes elementos de la herramienta, validando los componentes del sistema para asegurar que sus procesos son los adecuados.
- **Pruebas Unitarias:** Se han comprobado las funciones y clases implementadas a bajo nivel.
- **Pruebas de Integración:** Con las diferentes partes del sistema implementadas, se ha comprobado que funcionan correctamente al trabajar a la vez.

A la hora de describir las pruebas realizadas, se estarán definiendo de forma que se agrupan los tres tipos de pruebas descritos anteriormente, sin necesidad de especificar cada uno por separado. De esta forma se tendrá una visión global de las pruebas.

A continuación, se van a describir las pruebas realizadas sobre el sistema, indicando en qué consiste la prueba, los problemas encontrados y la solución que se ha aplicado. Se han dividido las pruebas en tres bloques, las relacionadas con el sistema, las funciones del sistema respecto a los usuarios registrados y la interacción con la interfaz.

16.1. Pruebas del Sistema

Estas pruebas están relacionadas con la instalación y puesta en marcha de la página web. Durante la inicialización del sistema y de la conexión de este con la base de datos ocurrieron ciertas complicaciones que fueron resueltas.

16.1.1. Instalación e Inicialización

Descripción: Durante la instalación de Django, se ha necesitado instalar unas librerías de Python para poder crear un nuevo proyecto e iniciar la página web. Además, se han realizado ciertas pruebas para aprender a utilizar Django.

Pruebas:

- Crear un proyecto vacío e iniciar la página web.
- Realizar migraciones a la base de datos por defecto SQL (*Structured Query Language*).
- Crear nuevos módulos y vistas.

Problemas: No se han encontrado problemas.

Soluciones: No ha sido necesario.

16.1.2. Creación de la base de datos

Descripción: Como se iba a utilizar MongoDB como base de datos, lo primero fue descargar las herramientas necesarias para su manejo

16. Pruebas

y crear la base de datos donde se conectaría el sistema.

Pruebas:

- Creación de una base de datos y conexión de esta con la herramienta MongoDB Compass.
- Inserción, consulta, modificación y eliminación de documentos.
- Inserción, consulta, modificación y eliminación de datos dentro de los documentos.

Problemas: No se han encontrado problemas.

Soluciones: No ha sido necesario.

16.1.3. Conexión con la base de datos

Descripción: Al utilizar MongoDB como base de datos, había que modificar la base de datos que crea por defecto el sistema. Para ello, se tenían que realizar migraciones y especificar las credenciales de la conexión con la nueva base de datos.

Pruebas:

- Instalar las librerías necesarias de Python para conectar Django con MongoDB.
- Realizar conexión entre el sistema y la base de datos al introducir las credenciales.

Problemas:

- Al realizar las migraciones para conectar Django con MongoDB, aparecía un error por la consola.
- Al realizar las conexiones con la base de datos, no se conectaba a ninguna base de datos o no se modifican las ya creadas.

Soluciones:

- Para el primer problema encontrado, la solución era descargar una versión más antigua de Django.
- En el caso de la conexión con la base datos, hacía falta determinar los parámetros para la conexión correctamente.

16.2. Pruebas Funcionales

Estas pruebas están relacionadas con las diferentes funciones y métodos creados para el correcto funcionamiento que pueden realizar los usuarios registrados. Durante el desarrollo de las diferentes funcionalidades, las diferentes funciones y clases implementadas fueron comprobadas por separado y conjuntamente.

16.2.1. Procesamiento de los datos

Descripción: Se han creado diferentes funciones y métodos para el procesamiento de los datos de los participantes. Estas funciones deben tener en cuenta muchas casuísticas y deben manejar los errores.

Pruebas:

- Especificar las fechas del seguimiento y recoger los datos de los ficheros con información relevante.
- Procesar los datos sin perder información.
- Obtener los datos en diferentes formatos, ya sea *json* o *pandas.DataFrame*.

Problemas: A la hora de seleccionar los ficheros con información relevante o que se encontrasen dentro de las franjas especificadas, no se escogían todos aquellos ficheros que si cumplían las condiciones.

Soluciones: Se modificó la función que recogía los archivos con datos relevantes o que estaban dentro de las fechas especificadas.

16. Pruebas

16.2.2. Creación de nuevos participantes

Descripción: Métodos y funciones implementados para comprobar el correcto funcionamiento del procesamiento de los datos de un participante y su almacenamiento en la base de datos, así como las comprobaciones al rellenar el formulario para la subida de datos.

Pruebas:

- Crear un nuevo participante rellenando correctamente el formulario.
- No rellenar todos los campos del formulario e intentar crear un nuevo participante.
- No introducir ficheros al subir los datos.
- Introducir ficheros que no contuviesen información relevante.

Problemas: Al subir los archivos de un nuevo participante, no se conseguía recoger los ficheros, lo que provocaba que no se pudiese obtener los datos a procesar.

Soluciones: Para solucionar el problema, se tenía que modificar las funciones de recogida de los ficheros y de lectura de estos.

16.2.3. Modificación de los datos de los participantes

Descripción: Métodos y funciones implementados para comprobar el correcto funcionamiento del procesamiento de los datos de un participante y su modificación en la base de datos. Así como las comprobaciones al rellenar el formulario para la subida de datos.

Pruebas:

- Modificar los datos de un participante rellenando correctamente el formulario.

- No rellenar correctamente el formulario e intentar modificar al participante.
- No introducir ficheros al modificar los datos.
- Introducir ficheros que no contuviesen información relevante.

Problemas: No se han encontrado problemas.

Soluciones: No ha sido necesario.

16.2.4. Visualización de los datos

Descripción: Métodos y funciones implementados para la obtención de los datos del participante que se desea visualizar. Para ello, el usuario registrado deberá rellenar un formulario.

Pruebas:

- El usuario registrado rellena el formulario correctamente.
- No seleccionar a un participante al que visualizar.
- Intentar visualizar una variable o forma de medir dicha variable, que un participante no contiene.

Problemas: No se han encontrado problemas.

Soluciones: No ha sido necesario.

16.2.5. Exportación de los datos

Descripción: Métodos y funciones implementados para la exportación y descarga de los datos de los participantes. El usuario registrado deberá rellenar un formulario y los datos a exportar serán procesados para poder ser descargados.

Pruebas:

- Descargar los datos de uno o de todos los participantes.

16. Pruebas

- Descargar los datos procesados en diferentes formatos.

Problemas:

- Una vez procesados los datos, el fichero a descargar no contenía información de los participantes.
- No se generaban los archivos que iban a ser descargados por el sistema.

Soluciones: Para solucionar ambos problemas se tuvieron que modificar las funciones utilizadas para generar los archivos, así como los métodos utilizados para su descarga.

16.2.6. Eliminación de un participante

Descripción: Métodos y funciones implementados para la eliminación de un participante de la base de datos. Para ello, el usuario deberá rellenar seleccionar a un participante y confirmar la acción.

Pruebas:

- El usuario registrado selecciona un participante y confirma la acción de eliminarlo.
- El usuario registrado selecciona un participante y cancela la acción de eliminarlo.
- El usuario registrado no selecciona un participante.

Problemas: No se han encontrado problemas.

Soluciones: No ha sido necesario.

16.3. Pruebas de la Interfaz

Estas pruebas están relacionadas con las interacciones de los diferentes usuarios con las vistas de las páginas, así como sus permisos a la hora de navegar o cómo se debe mostrar cierta información.

16.3.1. Accesos

Descripción: El sistema debe comprobar los permisos de los usuarios cuando intenten acceder a alguna de las funcionalidades.

Pruebas:

- Los usuarios intentan acceder a las funcionalidades que necesitan permisos o al panel de control.
- Los usuarios registrados intentan acceder a las funcionalidades que necesitan permisos o al panel de control.
- Los administradores intentan acceder a las funcionalidades que necesitan permisos o al panel de control.

Problemas: No se han encontrado problemas.

Soluciones: No ha sido necesario.

16.3.2. Navegación

Descripción: El sistema deberá permitir a los diferentes usuarios navegar por la página web de forma intuitiva. Dependiendo del tipo de usuario, la página mostrará cierta información para ayudar al usuario a navegar.

Pruebas:

- Los usuarios registrados y administradores inician sesión y deben aparecer todas las funcionalidades.
- Los usuarios registrados y administradores al cerrar sesión, deberán ver la misma información que los usuarios sin registrar.

Problemas: No se han encontrado problemas.

Soluciones: No ha sido necesario.

16. Pruebas

16.3.3. Visualización de los Datos

Descripción: A la hora de mostrar la información en la vista *VisualizeData*, el sistema debe permitir al usuario registrado seleccionar los datos a visualizar. La información del participante aparecerá en forma de gráfica, con la que el usuario registrado podrá interactuar.

Pruebas:

- Seleccionar los diferentes tipos de datos de los participantes.
- Seleccionar las diferentes formas de mostrar los datos de los participantes.

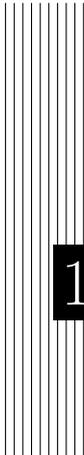
Problemas:

- Al recoger los datos para que se genere la gráfica, los datos mostrados no eran los correctos.
- Al seleccionar los diferentes tipo y formas de mostrar los datos, no se mostraban los datos correctamente.

Soluciones:

- Al generar las gráficas, los datos recogidos deben ser procesados para recoger lo mínimo necesario, sino, la gráfica mostrará información sin sentido.
- Al seleccionar los tipos y formas de mostrar los datos, se deben crear unas restricciones y condiciones para que el sistema conozca la información a recoger y procesar.

Parte V
Conclusiones



17 Conclusiones

Tras el desarrollo y la implementación de las diferentes funcionalidades del presente Trabajo Fin de Grado (TFG) y habiendo alcanzado los objetivos planteados, se realizará un repaso del proyecto en forma de conclusiones.

17.1. Objetivos Planteados

Durante la definición del proyecto se describieron diferentes objetivos a satisfacer y que finalmente han sido cumplidos exitosamente.

17.1.1. Objetivos específicos

De los objetivos específicos planteados para este proyecto, se han alcanzado los siguientes:

- Se ha desarrollado una herramienta web que permite diferenciar entre diferentes tipos de usuarios, para permitir qué tipo de funcionalidad pueden utilizar.
- Los usuarios tienen la capacidad de visitar las vistas *Home* y *Login*.

17.1. Objetivos Planteados

- Se ha desarrollado una herramienta web que permite a los usuarios registrados:
 - Subir archivos al sistema que serán procesados y almacenados en la base de datos, creando un nuevo participante.
 - Modificar los datos de los participantes al subir nuevos datos que serán procesados y modificados en la base de datos.
 - Eliminar los datos de los participantes de la base de datos.
 - Visualizar los datos de los participantes, siendo mostrados en forma de gráfica y procesados para que se adapten a esta.
 - Exportar los datos de los participantes, así como unos resúmenes adicionales en diferentes formatos de descarga.
 - Iniciar y cerrar sesión, modificando las vistas y las funcionalidades a las que podrá acceder.

- Se ha desarrollado una herramienta web que permite a los administradores con relación a los usuarios registrados:
 - Crear y registrar a nuevos usuarios en el sistema, para que los usuarios tengan la capacidad de acceder a nuevas funcionalidades.
 - Eliminar del sistema a los usuarios registrados.
 - Consultar la información sobre los usuarios registrados, para su posterior modificación.
 - Modificar los datos de los usuarios registrados, permitiendo modificar las credenciales y añadir nuevos datos sobre ellos.
 - Modificar los permisos que tienen los usuarios registrados para que tengan la capacidad de acceder a nuevas funcionalidades de la página web o del panel de control.

- Se ha desarrollado una herramienta web que permite a los administradores con relación a los grupos:

17. Conclusiones

- Crear grupos que agruparán a los usuarios registrados y asignar por igual los permisos que tendrán.
- Eliminar grupos del sistema.
- Modificar los permisos e información de los grupos.
- Asignar y agrupar a los usuarios registrados en los grupos creados y de esta forma, tengan asignados los mismos permisos.

17.1.2. Objetivos formativos

De los objetivos de formación planteados para este proyecto, se han alcanzado los siguientes:

- Se han adquirido conocimientos del manejo y de la creación de aplicaciones web utilizando Django.
- Se han adquirido conocimientos y habilidades sobre el patrón de arquitectura *Modelo-Vista-Controlador*.
- Se ha estudiado acerca de las pulseras de actividad Fitbit, así como los tipos de datos que recogen y sus formatos.
- Se ha estudiado acerca de métodos de procesamiento de grandes cantidades de datos, desarrollando diferentes métodos y funciones para su manejo.
- Se han reforzado y ampliado los conocimientos y habilidades en el ámbito de la creación de herramientas web, desarrollando diferentes funcionalidades que han permitido tener un mejor manejo de este tipo de herramientas.
- Se han adquirido conocimientos sobre el uso del sistema de creación de textos \LaTeX .

17.2. Fase de Pruebas

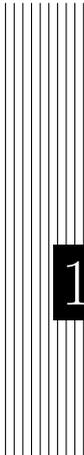
Durante el desarrollo del proyecto, todas las funcionalidades que se han ido implementando, así como las clases y métodos creados, han estado sometidos a todas las pruebas, verificando constantemente que la herramienta funcionase correctamente.

Aun habiendo realizado todas las pruebas, queda la posibilidad de que aparezcan nuevos errores inesperados que no se han encontrado antes.

17.3. Conclusiones Personales

El desarrollo de este TFG ha traído consigo muchas ventajas, he podido aprender y crear páginas web con nuevas herramientas, realizar un proyecto con un uso real y relacionado con un ámbito que me atrae.

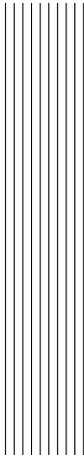
Y pese a los inconvenientes que han podido ir surgiendo en el desarrollo de este proyecto, estoy altamente satisfecho con la experiencia obtenida y el trabajo implementado.



18 Futuras Mejoras

Aun habiendo completado los objetivos planteados para el proyecto, han surgido ideas que podrían mejorar el sistema y añadir nuevas funcionalidades a la aplicación web, creando una herramienta más amplia y con mayores capacidades. Algunas de estas ideas son:

- Mejora del procesamiento y manejo de los datos recogidos por las pulseras Fitbit al crear o modificar un participante.
- Mejora de la interfaz gráfica de las diferentes vistas de la página web.
- Mejora de la subida de los ficheros con los datos de los participantes, permitiendo cargar carpetas completas y no solo seleccionando los ficheros.
- Implementación de modelos de *Machine Learning* para realizar un seguimiento más personalizado a las actividades físicas de cada participante.

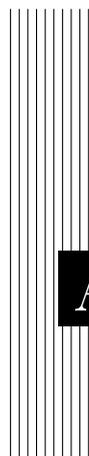


BIBLIOGRAFÍA

- [1] J. M. Aguilar. ¿qué es el patrón mvc en programación y por qué es útil?, 2023. URL <https://www.campusmvp.es/recursos/post/que-es-el-patron-mvc-en-programacion-y-por-que-es-util.aspx>.
- [2] M. de Sanidad. Porcentaje de personas con obesidad, por sexo según comunidad autónoma, 2020. URL <https://www.sanidad.gob.es/estadEstudios/sanidadDatos/tablas/tabla10.htm>.
- [3] Django. Django, 2023. URL <https://www.djangoproject.com>.
- [4] Fitbit. Fitbit, 2023. URL <https://www.fitbit.com/global/es/home>.
- [5] F. LLC. Fitbit app, 2023. URL <https://play.google.com/store/apps/details?id=com.fitbit.FitbitMobile&hl=es&gl=US>.
- [6] Microsoft. Visual studio code, 2023. URL <https://code.visualstudio.com>.
- [7] MongoDB. Mongoddb, 2023. URL <https://www.mongodb.com>.
- [8] MongoDB. Mongoddb compass, 2023. URL <https://www.mongodb.com/products/compass>.
- [9] Python. Python, 2023. URL <https://www.python.org>.
- [10] W3C. W3c, 2023. URL <https://www.w3.org>.

- [11] Wikipedia. BSON — Wikipedia, the free encyclopedia. <https://es.wikipedia.org/wiki/BSON>, 2023. [Online; accessed 15-January-2023].

Parte VI
Apéndices



A Manual de Instalación

Este manual es una explicación de cómo la aplicación web debe ser instalada y puesta en marcha. Para ello, se van a describir los diferentes pasos a seguir por el usuario que desee montar la misma en un dispositivo personal.

Como el autor del presente Trabajo Fin de Grado (TFG) ha estado desarrollando la aplicación web en el sistema operativo Windows, el manual se centrará en describir la instalación en este sistema.

A.1. Instalación del Software

Para empezar el proceso de montar la página web, el usuario deberá instalar diferentes herramientas y librerías para el correcto funcionamiento de esta.

A.1.1. Lenguaje de Programación y Librerías

Lo primero que se necesita instalar es el lenguaje de programación Python, ya que es el lenguaje utilizado para el manejo de la página web y de los datos recopilados.

La versión de Python a descargar es la **3.9.5**. Para poder instalarla

A.1. Instalación del Software

solo hará falta seguir las instrucciones del siguiente [enlace](#), y descargar y ejecutar el archivo más adecuado al sistema utilizado.

Una vez se tiene instalado el lenguaje de programación, hay que instalar un administrador de paquetes, para poder agregar las diferentes librerías. Para ello, se utilizará *pip* y se deberán seguir los pasos descritos en el siguiente [enlace](#).

Con el gestor de paquetes instalado ya se puede agregar las librerías mencionadas en el apartado 11 acerca de las herramientas utilizadas con mayor facilidad. Para ello solo hará falta acceder al símbolo del sistema (CMD) de Windows e instalar uno a uno escribiendo el siguiente comando:

```
pip install <package_name>==<version>
```

Donde el *package_name* es el nombre de la librería a instalar y *version* es la versión de esta librería. En la Tabla A.1 se pueden observar los nombres de las librerías con las versiones necesarias que deben ser instaladas.

Librería	<i>package_name</i>	<i>version</i>
Django	django	4.1.3
Json5	json5	0.9.10
Numpy	numpy	1.23.3
Pandas	pandas	1.5.0
Djongo	djongo	1.3.6
Mongo Engine	mongoengine	0.26.0
Pymongo	pymongo	3.12.1
Zipp	zipp	3.8.1

Tabla A.1: Librerías y versiones

A. Manual de Instalación

A.1.2. Entorno de Desarrollo

Existen diferentes herramientas para editar código e implementar las funcionalidades. Para este proyecto se ha utilizado Visual Studio Code, ya que permite compilar y depurar código desde terminales integradas, lo que facilita el proceso de creación de la aplicación web.

Para instalar esta herramienta es necesario descargar y ejecutar el archivo en el siguiente [enlace](#). Desde esta herramienta se podrá tanto editar código como encender y apagar la página web. En el caso de que no se requiera de manejar código, no es necesaria la instalación de esta herramienta.

A.1.3. Base de Datos

La base de datos utilizada es MongoDB y para poder manejar y visualizar los datos que se vayan a crear, hará falta la herramienta MongoDB Compass. Para instalar esta herramienta hará falta seguir las instrucciones del siguiente [enlace](#).

A.2. Creación de la Base de Datos

Para crear la base de datos hay que registrarse e iniciar sesión dentro de la página web de [MongoDB](#). Desde aquí se podrán crear diferentes bases de datos (*clusters*) con características diferentes.

Para este trabajo, se ha creado una base de datos de tipo *Shared* que permite crear y utilizar los servicios de MongoDB de forma gratuita.

Una vez se crea la base de datos hay que conectarla con MongoDB Compass, para ello, desde la aplicación descargada, habrá que introducir la dirección del *cluster* con las credenciales. En caso de utilizar una base de datos en local, habrá que introducir la dirección con el puerto del *localhost*.

A.3. Página Web

Ya se tienen todas las herramientas y librerías necesarias para el correcto funcionamiento de la aplicación. Ahora, el usuario deberá descargar todos los archivos de la página web en su disco local. Una vez esté todo descargado, el usuario tendrá que conectarse con la base de datos.

Dentro del proyecto, hay que acceder al archivo de configuración *settings.py* desde donde se podrá especificar la base de datos para poder conectarse. Para llegar al archivo se tendrá que seguir el siguiente flujo:

FitBitAnalysis → *FitBitAnalysis* → *settings.py*

El primer *FitBitAnalysis* es la carpeta que abarca todo la página web y el segundo es el bloque principal del sitio donde se encuentran las configuraciones más importantes.

Una vez dentro del archivo *settings.py*, el usuario deberá configurar el apartado de *Database*, modificando las variables *DB_NAME* y *HOST*:

- ***DB_NAME***: Nombre definido al crear el *cluster*.
- ***HOST***: Dirección de la base de datos.

Una vez establecidos los nuevos valores de las variables, el usuario dentro de la terminal y habiendo entrado en el primer *FitBitAnalysis*, escribirá los siguientes comandos para enlazar la base de datos y comprobar que todo funciona correctamente:

python manage.py makemigrations

python manage.py migrate

El primer comando genera los archivos para realizar los cambios establecidos sobre la página web. El segundo comando sirve para aplicar estos cambios registrados.

A. Manual de Instalación

A.3.1. Administrador

En este punto de la instalación de la página web, ya se tiene el sitio funcional, pero aún hace falta realizar unas acciones. En el caso de que no exista un administrador, se deberá crear uno, para la creación y manejo de los usuarios registrados y, de esta forma, acceder a las funcionalidades. Para ello habrá que escribir el siguiente comando tal como se realizó anteriormente:

```
python manage.py createsuperuser
```

Al ejecutar el comando, se deberá introducir un nombre de usuario (*Username*), un correo electrónico (*Email address*) y una contraseña (*Password*).

A.3.2. Encender y Apagar

Lo único que hace falta ahora es encender la página web y disfrutar de las funcionalidades implementadas. Para ello, habrá que ejecutar el siguiente comando:

```
python manage.py runserver
```

Al ejecutarlo, la página web estará en funcionamiento hasta que se cierre el servidor. Para cerrarlo solo habrá que pulsar las siguientes teclas dentro de la terminal:

```
CTRL-BREAK (CTRL-C)
```

