



**UNIVERSIDAD DE CASTILLA-LA MANCHA
ESCUELA SUPERIOR DE INFORMÁTICA**

GRADO EN INGENIERÍA EN INFORMÁTICA

TRABAJO FIN DE GRADO

**Sistema web para la gestión de un programa de
salud orientado a caminatas saludables**

Noelia María Granados Carrasco

Julio, 2021

**SISTEMA WEB PARA LA GESTIÓN DE UN PROGRAMA DE SALUD
ORIENTADO A CAMINATAS SALUDABLES**



**UNIVERSIDAD DE CASTILLA-LA MANCHA
ESCUELA SUPERIOR DE INFORMÁTICA**

Departamento de Tecnologías y Sistemas de la Información

**GRADO EN INGENIERÍA EN INFORMÁTICA
TECNOLOGÍAS DE LA INFORMACIÓN**

**Sistema web para la gestión de un programa de
salud orientado a caminatas saludables**

Autor: Noelia María Granados Carrasco

Tutor académico: David Vallejo Fernández

Cotutor académico: Santiago Sánchez Sobrino

Julio, 2021

Noelia María Granados Carrasco

Ciudad Real – España

© 2021 Noelia María Granados Carrasco

Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.3 or any later version published by the Free Software Foundation; with no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts. A copy of the license is included in the section entitled "GNU Free Documentation License".

Se permite la copia, distribución y/o modificación de este documento bajo los términos de la Licencia de Documentación Libre GNU, versión 1.3 o cualquier versión posterior publicada por la *Free Software Foundation*; sin secciones invariantes. Una copia de esta licencia esta incluida en el apéndice titulado «GNU Free Documentation License».

Muchos de los nombres usados por las compañías para diferenciar sus productos y servicios son reclamados como marcas registradas. Allí donde estos nombres aparezcan en este documento, y cuando el autor haya sido informado de esas marcas registradas, los nombres estarán escritos en mayúsculas o como nombres propios.

TRIBUNAL:

Presidente:

Vocal:

Secretario:

FECHA DE DEFENSA:

CALIFICACIÓN:

PRESIDENTE

VOCAL

SECRETARIO

Fdo.:

Fdo.:

Fdo.:

Resumen

Cada vez son más las personas que se preocupan por su salud, tanto física como mental hoy en día. Además, buscan nuevas formas de poder ponerse en forma mientras escapan de la rutina, compartiendo a la vez experiencias con personas con objetivos similares.

Un ejemplo de la preocupación por la salud en la sociedad fue llevado a cabo mediante un estudio con la población española. Según datos del estudio *WIN World Survey 2019* que el Instituto DYM realizó en colaboración con *WIN International*, el 65 % de las personas en España afirman preocuparse frecuentemente por llevar una vida saludable.

El sedentarismo es uno de los problemas que más preocupa a la Organización Mundial de la Salud (OMS)¹. Al menos un 60 % de la población no realiza la actividad física necesaria para obtener beneficios para la salud. Esto se debe en parte a la insuficiente participación en la actividad física durante el tiempo de ocio y a un aumento de los comportamientos sedentarios durante las actividades laborales y domésticas.

Debido a la pandemia causada por el *COVID-19* que nos ha tocado atravesar, se ha necesitado un poco de ingenio para poder adaptarse a la nueva situación. Con ello, se ha visto incrementado el interés de la población por la salud, surgiendo también un mayor interés por las caminatas o paseos saludables.

El objetivo de este proyecto es crear un servicio web que permite gestionar caminatas saludables en el condado de Suffolk (Reino Unido). Esto posibilita un entorno en el que compartir experiencias e historias entre personas, al mismo tiempo que se disfruta de la naturaleza y de los beneficios que ofrece caminar.

Cabe destacar, además, la complejidad que supone integrar el nuevo proyecto dentro de un sistema real en producción.

Esto es debido a que existe una arquitectura y un sistema ya establecido. Por lo que se deben crear nuevas funcionalidades a partir de las ya existentes. Además, es un proyecto que tiene otros servicios que son utilizados de manera diaria por personas en Reino Unido. Por lo que es necesario que la introducción de nuevas mejoras y creaciones del nuevo servicio *Walking for Health* no interfieran con lo que ya hay establecido para no poder en riesgo la operabilidad del sistema. También, la forma de trabajar es diferente ya que existen distintos entornos de desarrollo.

Gracias a la utilización de las numerosas tecnologías de las que se hace uso en este proyecto ha sido posible fomentar aún más la actividad física de las personas. Creándoles así la necesidad de mantenerse activos por los beneficios que supone en la salud las caminatas saludables.

¹<https://www.who.int>

Abstract

More and more people today are concerned about their health, both physical and mental. In addition, they are looking for new ways to get fit while escaping from routine and sharing experiences with people with similar goals.

An example of the concern for health in society was carried out through a study with the Spanish population. According to data from the WIN World Survey 2019 study that the DYM Institute conducted in collaboration with WIN International, 65 % of people in Spain claim to be frequently concerned about leading a healthy life.

Sedentary lifestyles are one of the problems that most concerns the World Health Organization (WHO). At least 60 % of the population does not get the physical activity needed for health benefits. This is due in part to insufficient participation in leisure-time physical activity and an increase in sedentary behaviors during work and household activities.

Due to the pandemic caused by the COVID-19 that we have been through, it has taken some ingenuity to be able to adapt to the new situation. With this, there has been an increase in the population's interest in health, and a greater interest in healthy walks or strolls.

The aim of this project is to create a web service to manage healthy walks in the county of Suffolk (UK). This provides an environment in which people can share experiences and stories with each other, while enjoying nature and the benefits of walking.

It is also worth noting the complexity of integrating the new project into a real system in production.

This is because there is an already established architecture and system. Therefore, new functionalities must be created from the existing ones. In addition, it is a project that has other services that are used on a daily basis by people in the UK. Therefore, it is necessary that the introduction of new improvements and creations of the new service, Walking for Health, do not interfere with what is already established so as not to jeopardize the operability of the system. Also, the way of working is different as there are different development environments.

Thanks to the use of the many technologies used in this project, it has been possible to further encourage people to be physically active. Thus creating the need to stay active for the health benefits of healthy walking.

Agradecimientos

En primer lugar, me gustaría agradecer el esfuerzo que realizan día a día todos los profesores de la Escuela Superior de Informática de Ciudad Real. Sobre todo, agradecer a mis tutores y mentores de proyecto, David Vallejo Fernández y Santiago Sánchez Sobrino. Gracias por darme la oportunidad de hacer este Trabajo Fin de Grado, por vuestra dedicación y constancia.

A dos de las personas más importantes en mi vida, mis padres, Maribel y Julián, gracias por haberme dado la oportunidad de poder estudiar esta bonita carrera que desde siempre me ha gustado. Por haber confiado en mí, por haberme dado una buena educación y por estar siempre que lo he necesitado.

A mis abuelos, que me cuidan desde el cielo. Gracias por todo lo que hicisteis por mí.

A mi hermana Irene, por ser la mejor hermana y amiga del mundo, gracias por estar a mi lado, motivarme y animarme a ser mejor persona. Estoy muy orgullosa de todo lo que estás consiguiendo.

A Raúl, por ser mi alma gemela, por permanecer todos los días a mi lado, por ayudarme a superar las pequeñas dificultades que a veces nos pone la vida, por avanzar y progresar juntos en la misma dirección y por haber pasado juntos esta bonita etapa en la residencia y la Universidad.

A mis tíos, Julio y Ana, por todo el cariño, apoyo y comprensión recibidos, gracias por el tiempo de calidad que pasamos cuando nos vemos.

Dar las gracias a mis compañeros de clase, con los que he trabajado y reído durante todos estos años de carrera y de los que me llevo un bonito recuerdo.

Agradecer también a mis mejores amigas que siempre han estado ahí, su apoyo, consejos y risas. Por todos los buenos momentos que hemos pasado y los que nos quedan por vivir.

Noelia María Granados Carrasco

Índice general

Resumen	V
Abstract	VII
Agradecimientos	IX
Índice general	XI
Índice de figuras	XV
Índice de cuadros	XVII
Índice de listados	XIX
Acrónimos	XXI
Glosario	XXIII
1. Introducción	1
1.1. Entorno de realización del TFG	2
1.2. Estructura del documento	2
2. Objetivos	5
3. Estado del Arte	7
3.1. Desarrollo tecnológico en salud	7
3.1.1. Telemedicina	7
3.1.2. Por qué es importante invertir en salud	9
3.1.3. Visión general de los sistemas de desarrollo tecnológico en salud . .	10
3.1.4. Sistemas de fomento de la actividad física	11
3.2. Caminatas saludables	12
3.2.1. SUP Yoga Camp	12

3.2.2.	New York Road Runners	12
3.2.3.	Wikiloc	13
3.2.4.	Runkeeper	14
3.2.5.	Strava	15
3.2.6.	Google Fit	16
3.2.7.	Comparativa de las páginas web analizadas	17
3.3.	Desarrollo web	19
3.3.1.	Visión general	19
3.3.2.	Arquitecturas web	19
3.3.3.	Tecnologías web	21
3.4.	Computación en la nube	23
3.4.1.	Conceptos fundamentales	23
3.4.2.	Cómo funciona Cloud Computing	23
3.4.3.	Ventajas de Cloud Computing	24
3.4.4.	Modelos de negocio	24
3.4.5.	Plataformas de computación en la nube	25
4.	Metodología	27
4.1.	Metodologías ágiles	27
4.1.1.	Scrum	28
4.2.	Plan de trabajo	31
4.3.	Pila tecnológica utilizada en el proyecto	32
5.	Arquitectura	37
5.1.	Organización del proyecto	37
5.1.1.	Diagrama de general del sistema web	37
5.1.2.	Arquitectura frontend y backend del sistema	39
5.1.3.	Estructura general del proyecto	41
5.2.	Diseño de la base de datos	43
5.2.1.	Elección de la Base de Datos	43
5.2.2.	Anonimización de la Base de Datos	44
5.2.3.	Entidades	45
5.2.4.	Diagrama de base de datos Walking for Health	45
5.3.	Roles y permisos	45
5.3.1.	Permisos a nivel de servidor	45
5.3.2.	Permisos a nivel de cliente	47

5.3.3.	Directivas personalizadas para proteger el frontend	47
5.4.	Filtrado de datos	48
5.5.	Validación de formularios	49
5.5.1.	Ejemplos de validación en formularios	49
5.6.	Módulos	50
5.6.1.	Pre-caminata	50
5.6.2.	Post-caminata	51
5.6.3.	Walkers	51
5.6.4.	Groups	54
5.6.5.	Volunteers	54
5.6.6.	Manage Walks	54
5.6.7.	Contact	55
5.6.8.	Otras funcionalidades	56
5.7.	Testing	56
6.	Resultados	59
6.1.	Sprint 0	59
6.2.	Sprint 1	60
6.2.1.	Sección de caminantes o walkers	60
6.2.2.	Crear y listar grupos de caminatas	61
6.2.3.	Asignar caminante a un grupo de caminatas	61
6.3.	Sprint 2	61
6.3.1.	Crear y listar lugares de encuentro o venues	64
6.3.2.	Crear y listar voluntarios	64
6.3.3.	Crear y listar caminatas saludables	67
6.4.	Sprint 3	67
6.4.1.	Crear y listar reservas de caminatas	67
6.4.2.	Visualizar rutas programadas en el calendario del sistema	70
6.4.3.	Envío de SMS con la información de la ruta programada	71
6.5.	Sprint 4	71
6.5.1.	Registrar caminata realizada	71
6.5.2.	Eventos de calendario para crear sesiones de formación	73
6.5.3.	Envío de SMS avisando de la cancelación de una ruta	73
6.6.	Costes y recursos del proyecto	73
6.7.	Estadísticas del proyecto	74

7. Conclusiones	75
7.1. Análisis de objetivos alcanzados	75
7.2. Dificultades	77
7.3. Competencias académicas trabajadas	77
7.4. Trabajos futuros	78
Bibliografía	81
A. Historias de usuario	85
B. Funcionalidades de Walking for Health	91
B.1. Sección de caminantes o walkers	91
B.1.1. Caminantes creados por el usuario admin (My walkers)	91
B.1.2. Caminantes recientes (Recent walkers)	92
B.1.3. Búsqueda de caminantes (Search)	93
B.1.4. Lista de espera (Waiting List)	94
B.2. Crear y listar lugares de encuentro o venues	95
B.3. Sesiones de formación para los voluntarios	97
B.4. Envío de SMS de una ruta programada	98
B.5. Envío de SMS con la cancelación de una ruta	99
C. Diagrama de base de datos Walking for Health	101

Índice de figuras

1.1. Health Walks (OneLife Suffolk)	2
3.1. Formulario para reservar caminatas saludables en Sup Yoga Camp	13
3.2. Calendario de eventos en NYRR	13
3.3. Interfaz de la aplicación Wikiloc	14
3.4. Interfaz gráfica de Runkeeper	15
3.5. Interfaz gráfica de Strava	16
3.6. Interfaz de Google Fit	17
3.7. Tabla comparativa de las aplicaciones analizadas	18
3.8. Arquitectura web	20
3.9. Servicios ofrecidos por Amazon Web Services	26
4.1. Visión general de la metodología ágil Scrum	28
4.2. Sprints realizados en el desarrollo de Walking for Health	31
4.3. Cronograma de desarrollo del servicio web	32
4.4. Ejemplo de tablero Kanban	35
4.5. Pila tecnológica del proyecto	35
5.1. Diagrama general del sistema	38
5.2. Estructura del proyecto desde el lado del cliente	41
5.3. Estructura del proyecto desde el punto de vista del backend	41
5.4. Estructura general del proyecto	42
5.5. Diagrama de secuencia pre-caminata	52
5.6. Diagrama de secuencia post-caminata	53
6.1. Creación y listado de grupos de caminatas saludables	62
6.2. Añadir caminante a grupo de caminata	63
6.3. Crear y listar voluntarios	65
6.4. Cuestionario de salud de la Evaluación de Riesgos COVID	66
6.5. Creación y listado de una caminata saludable	68

0. ÍNDICE DE FIGURAS

6.6. Crear y listar reservas de caminatas	69
6.7. Visualización de las rutas programadas en el calendario	70
6.8. Registro de caminata en el sistema	72
6.9. Costes y recursos del proyecto	74
6.10. Estadísticas del repositorio generadas por Bitbucket	74
B.1. Subsección para visualizar los caminantes creados por el administrador . .	91
B.2. Caminantes añadidos recientemente al sistema	92
B.3. Búsqueda de caminantes y clientes del resto de servicios web	93
B.4. Lista de espera de caminantes en la sección de Waiting list	94
B.5. Creación de una venue	95
B.6. Listar venues	96
B.7. Sesión de formación programada para un advisor en el sistema	97
B.8. SMS enviado a un caminante y lista de SMS enviados en el sistema	98
B.9. SMS de la cancelación de una caminata	99
C.1. Diagrama general de la base de datos	101

Índice de cuadros

7.1. Justificación de competencias	77
A.1. HU1 Introducción a los servicios web	85
A.2. HU2 Listado y búsqueda de caminantes	85
A.3. HU3 Crear y listar grupos de caminatas	86
A.4. HU4 Asignar un caminante a un grupo de caminatas	86
A.5. HU5 Crear y listar voluntarios	86
A.6. HU6 Evaluación de Riesgos COVID	87
A.7. HU7 Crear y listar venues	87
A.8. HU8 Crear y listar caminatas saludables	87
A.9. HU9 Reserva de caminatas saludables	88
A.10.HU10 Visualización de rutas programadas en el calendario	88
A.11.HU11 Envío de SMS con la información de la ruta programada	88
A.12.HU12 Registro de caminata realizada	89
A.13.HU13 Sesiones de formación para voluntarios	89
A.14.HU14 Cancelación de una caminata programada	89

Índice de listados

5.1. Permisos a nivel de servidor	46
5.2. Proteger el acceso mediante un método específico	47
5.3. Directiva personalizada Book Walk	47
5.4. Uso de la directiva PermissionsCheckDirective	48
5.5. Uso del filtrado de datos en voluntarios	48
5.6. Validación del máximo número de miembros por grupo de caminata	50
5.7. Validación de la zona rural y pública en la sección de reservar una caminata	50
5.8. Concatenación de fecha y hora en una reserva de caminata	57

Acrónimos

- ACID** Atomicidad, Consistencia, Aislamiento y Durabilidad. 43, 44
- API** Interfaz de Programación de Aplicaciones. 32, 39, 40, 44, 45, 77
- AWS** Amazon Web Services. 25, 26, 33, 37
- CaaS** Communication as a Service. 24
- CPU** Central Processing Unit. 37, 43
- CRUD** Create, read, update, delete. 44
- CSS** Cascading Style Sheets. 19, 40
- DDL** Lenguaje de definición de datos. 43
- DOM** Document Object Model. 21
- DTO** Data Transfer Object. 32, 39
- EC2** Elastic Compute Cloud. 37, 73
- FK** Furious Koalas Interactive. 2, 11, 31, 74
- GPS** Sistema de Posicionamiento Global. 14, 15
- HTML** HyperText Markup Language. 19, 47, 56, 74
- HTTP** Hypertext Transfer Protocol. 39
- IaaS** Infrastructure as a Service. 24
- IoT** Internet of Things. 26
- JS** JavaScript. 19, 22, 57
- MaaS** Monitoring as a Service. 25
- MCG** My Client Genius. 43
- MVC** Modelo Vista Controlador. 21
- OMS** Organización Mundial de la Salud. 7, 16
- PaaS** Platform as a Service. 25, 37
- PHP** Hypertext Preprocessor. 19

0. ACRÓNIMOS

POJO Plain Old Java Object. 22

RAM Random Access Memory. 22

RDS Relational Database Service. 37, 73

RRSS Redes Sociales. 17

S3 Simple Storage Service. 37, 73

SaaS Software as a Service. 25

SES Simple Email Service. 37, 73

SGBD Sistema Gestor de Base de Datos. 43

SMS Servicio de mensajes cortos. 4, 17, 18, 37, 55, 56, 67, 71, 73, 76, 77, 79

SNS Simple Notification Service. 37

SPA Single-Page Applications. 21

TFG Trabajo Fin de Grado. 1–5, 7, 11, 19, 25, 27, 30, 31, 34, 57, 59, 74–79

TI Tecnologías de la Información. 10, 77

TIC Tecnologías de la información y la comunicación electrónicas. 7, 9, 10

UCLM Universidad de Castilla-La Mancha. 2, 5

UML Lenguaje Unificado de Modelado. 45

URI Uniform Resource Identifier. 39

VPN Virtual Private Network. 26

Glosario

- backend** Parte de un sistema o aplicación informática que no es visible para el usuario de manera directa y que se encarga de almacenar y manipular los datos. 19, 30, 32, 37, 40, 44, 45, 78
- bash** Intérprete de comandos que ejecuta, una por una, las instrucciones introducidas por el usuario o contenidas en un script y devuelve los resultados. 33
- data binding** Mecanismo que enlaza los elementos de la interfaz con la parte en la que se construye la información a mostrar. Es el enlace entre el front y el back de una página web o aplicación. 21
- e-salud** Según la OMS, es el apoyo que la utilización costo-eficaz y segura de las tecnologías de la información y las comunicaciones ofrece a la salud y a los ámbitos relacionados con ella. Con inclusión de los servicios de atención de salud, la vigilancia y la documentación sanitarias, así como la educación, los conocimientos y las investigaciones en materia de salud. 10
- endpoint** Puerto de comunicaciones empleado para realizar llamadas a procedimientos remotos plural. 32, 39, 44–46
- fat client** Programa cliente de una arquitectura cliente-servidor cuando la mayor carga de cómputo está desplazada hacia la computadora que ejecuta dicho programa. También se conoce como cliente pesado. 20
- framework** Conjunto estandarizado de conceptos, prácticas y criterios para enfocar una problemática particular que sirve como referencia para resolver nuevos problemas similares. 28, 32, 33, 49, 57, 59
- frontend** Conversión de datos en una interfaz gráfica para que el usuario pueda interactuar con la información de forma digital usando HTML, CSS y JavaScript. 19, 21, 30, 32, 40, 47, 78
- full-stack developer** Desarrollador encargado de manejar cada uno de los aspectos relacionados con la creación y el mantenimiento de una aplicación web. Es fundamental que tenga conocimientos en desarrollo frontend y backend. 19
- gateway** Dispositivo que actúa de interfaz de conexión entre aparatos o dispositivos y posibilita la compartición de recursos entre dos o más ordenadores. 26
- JavaScript** Lenguaje de programación encargado de dotar de mayor interactividad y dinamismo a las páginas web. Se le reconoce como uno de los tres lenguajes nativos de la web junto a HTML y a CSS. 32, 33

0. GLOSARIO

product Backlog Lista de características que han sido priorizadas y que contiene descripciones breves sobre todo lo que se desea para el producto que se va a desarrollar. 29, 30

script Término informal que se usa para designar a un programa relativamente simple, también llamado secuencia de comandos guión plural. 33, 41

Scrum Marco de trabajo para desarrollo ágil de software. Es un proceso en el que se aplican de manera regular un conjunto de buenas prácticas para trabajar colaborativamente, en equipo y obtener el mejor resultado posible en proyectos. 4, 27–31, 59, 78

sprint Nombre que van a recibir cada uno de los ciclos o iteraciones en un proyecto Scrum. 3, 27–31, 34, 59–61, 67, 71

stakeholders Parte interesada o interesados. Hace referencia a una persona, organización o empresa que tiene interés en una organización dada. 4, 30, 31

TypeScript Superconjunto de JavaScript, que esencialmente añade tipos estáticos y objetos basados en clases. 21, 32, 33, 40, 41

Capítulo 1

Introducción

ESTE Trabajo Fin de Grado (TFG) se enmarca en el contexto de un sistema global responsable de la gestión de servicios y programas de salud. De esta manera, el sistema integrará el rol de administrador, que es el rol principal.

Cabe mencionar que solamente existe un rol encargado de llevar toda la lógica del sistema, el cual es el administrador. Además, la publicidad y la difusión de las caminatas saludables se realiza de forma externa al sistema y queda fuera de su alcance.

Cabe destacar que este proyecto surge de la idea de poder mejorar e integrar el sistema actual existente llamado *Health Walks*¹ creado por la empresa OneLife Suffolk (Reino Unido)².

En la actualidad no existe un sistema informático integral que dé soporte a este proceso a nivel de operaciones. Tampoco está integrado con otros servicios de OneLife Suffolk. Por este motivo, surge *Walking for Health* el cual será integrado en el sistema real y abordará la problemática que se plantea en este TFG.

En primer lugar, lo que se pretende es digitalizar un proceso que actualmente no lo está. Para posteriormente facilitar el proceso de gestión e integración de las caminatas saludables en un sistema real.

El objetivo final estará orientado a dar soporte a *Health Walks*, creando un servicio que permita automatizar todas las actividades que actualmente soporta el sistema y añadiendo algunas funcionalidades relevantes para mejorar este servicio ofrecido en el condado de Suffolk a todos sus ciudadanos.

Actualmente existen características que no están soportadas en *Health Walks*, pero que gracias a *Walking for Health* podrá ser posible. Algunas de ellas son, por ejemplo, la ausencia de gestión de caminatas que permita reservarlas y programarlas en el calendario del sistema, añadir caminantes a grupos de caminatas, registrar paseos que ya han sido realizados o visualizar mediante un mapa el punto de partida de una ruta, entre otros.

¹<https://onelifesuffolk.co.uk/services/health-walks/>

²<https://onelifesuffolk.co.uk/>

1. INTRODUCCIÓN

En la Figura 1.1 se muestra la interfaz del sistema de *Health Walks* creado por la empresa OneLife Suffolk.

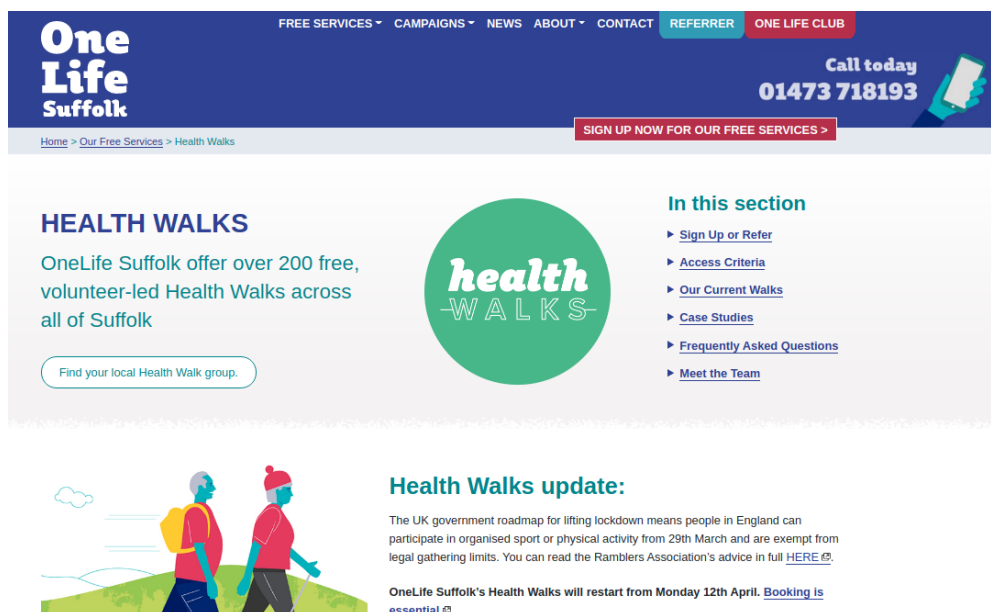


Figura 1.1: Health Walks (OneLife Suffolk)

1.1 Entorno de realización del TFG

La realización de este TFG ha sido posible gracias al convenio FORTE que tiene la Escuela Superior de Informática de la Universidad de Castilla-La Mancha (UCLM) con la empresa Furious Koalas Interactive (FK) la cual es una *spinoff* de la UCLM.

Esta empresa tiene como objetivo proporcionar servicios de alta calidad en áreas como la Gamificación, Juegos Serios, Simuladores y Visión por Computador. Algunos de sus *partners* más conocidos son: Indra, UCLM, Telefónica y Tecnobit, entre otros.

Por otro lado, cabe destacar que FK es *partner* tecnológico de la empresa británica *More-Life* desde finales de 2019.

De esta manera, la estudiante ha podido llevar a cabo la realización del proyecto dentro de la misma empresa. Donde la integración se ha realizado progresivamente, ya que ha tenido lugar dentro de un equipo de trabajo con un proyecto ya en curso. Además, se ha hecho uso de una compleja pila tecnológica que ha sido estudiada en detalle antes de realizar este proyecto, consiguiendo como resultado una solución que resuelve un problema real.

1.2 Estructura del documento

A continuación, se expone la estructura que tendrá este TFG, el cual se ha estructurado en 7 capítulos y 3 anexos.

- **Capítulo 1. Introducción.** En este capítulo se realiza una breve introducción del problema que tiene como objetivo resolver este TFG, así como el entorno y la estructura del proyecto realizado. Se pretende ofrecer al lector de este documento una visión general del trabajo realizado.
- **Capítulo 2. Objetivos.** En este otro capítulo del presente documento se detallan el objetivo general y los objetivos secundarios que se pretenden alcanzar para garantizar una adecuada finalización de este TFG.
- **Capítulo 3. Estado del Arte.** En este capítulo del Estado del Arte se realiza un análisis comparativo de otros servicios y páginas web que utilizan tecnologías y propósitos similares a la de este TFG. Además, se profundiza en el término de desarrollo tecnológico en salud y el concepto de computación en la nube. También se comentarán las arquitecturas existentes en la actualidad y las tecnologías web utilizadas en este proyecto.

En definitiva, este capítulo pretende posicionar al lector en el contexto de sistemas similares al propuesto en este TFG.

- **Capítulo 4. Metodología.** En este capítulo se explica la metodología que se ha llevado a cabo para el desarrollo del proyecto, profundizando en los elementos imprescindibles que intervienen en ella. Además, se mostrará y explicará el plan de trabajo seguido. Y finalmente, la pila tecnológica donde se incluyen las tecnologías y herramientas seguidas.
- **Capítulo 5. Arquitectura.** En este otro capítulo se explica de manera detallada la arquitectura general que tiene el proyecto. Además, se detalla el diseño y la implementación elegida para la solución propuesta.

Por otro lado, se incluyen diagramas que permiten visualizar los resultados obtenidos, favoreciendo asimismo la comprensión del proceso explicado.

- **Capítulo 6. Resultados.** En este capítulo se explican los distintos *sprints* que se han seguido durante todo el desarrollo del proyecto, en los que se explica el proceso de *pre-caminata* y *post-caminata* en detalle. También se expondrán los costes, recursos y algunas estadísticas del proyecto.
- **Capítulo 7. Conclusiones.** En este último capítulo se incluye una conclusión final del proyecto, las mejoras futuras que podría tener este sistema web desarrollado, dificultades encontradas durante el desarrollo y las competencias académicas trabajadas.

1. INTRODUCCIÓN

En cuanto a los anexos, se ha incluido 3, los cuales se explican a continuación.

- **Anexo A. Historias de usuario.** En este anexo se encuentran las historias de usuario que han sido utilizadas en el desarrollo de este proyecto. Estas historias de usuario se han incluido gracias a las enormes ventajas que aportan en un proyecto que hace uso de una metodología ágil como es *Scrum*. Hacer uso de historias de usuario durante todo el proceso de desarrollo permite tener una interacción directa con el cliente para comprobar que lo que se está desarrollando es lo que busca y está a la altura de sus expectativas.

Las historias de usuario permiten tener una mejor comunicación y contacto directo con el cliente. Además de fomentar la colaboración entre los *stakeholders* y el equipo ágil. También facilitan en gran medida la planificación e implementación ya que permiten dividir el proyecto en pequeñas entregas.

Finalmente, estas historias de usuario permiten estimar el esfuerzo requerido en el proceso de desarrollo y necesitan poco mantenimiento.

- **Anexo B. Otras funcionalidades de Walking for Health.** En este anexo se han incluido otras funcionalidades implementadas para el servicio web desarrollado en este TFG. Estas funcionalidades son: la creación y listado de caminantes, crear y listar lugares de encuentro, envío de notificaciones utilizando Servicio de mensajes cortos (SMS) y la creación de sesiones de formación para los voluntarios.
- **Anexo C. Diagrama de base de datos.** En este anexo se puede visualizar el diagrama de la base de datos realizado para el servicio web *Walking for Health*. En él se pueden observar las distintas relaciones de entidades o tablas utilizadas para el desarrollo de este proyecto.

Capítulo 2

Objetivos

ESTE TFG se enmarca en el ámbito del servicio de salud denominado *Health Walks*, desarrollado por *FK Interactive* (*spin-off* de la UCLM) para la empresa británica *More Life Ltd*. El **objetivo principal** de este TFG consiste en diseñar, desarrollar y validar un componente software que permita la gestión de un programa de salud orientado a proporcionar caminatas saludables para los ciudadanos del condado de Suffolk (Reino Unido). La finalidad del servicio es evitar el sedentarismo e incrementar el nivel de actividad física de las personas inscritas en dicho programa de salud para aumentar progresivamente su calidad de vida.

A continuación, se introducen los **sub-objetivos** en los que se divide el objetivo general previamente expuesto.

1. Estudio de las **tecnologías y herramientas que soporten el desarrollo y despliegue del servicio**, incluyendo los sistemas web que tratan de solucionar una problemática similar a la que se enfrenta este proyecto.
 - Investigación en libros académicos, artículos y sistemas informáticos que aborden una problemática similar a la de este TFG, para poder tomar conciencia de los avances existentes en el ámbito de la salud. Esta información será recopilada y comentada en el capítulo 3.
 - Análisis de las tecnologías existentes en la actualidad más adecuadas para abordar la problemática de este TFG. Para ello, se realiza un estudio previo al desarrollo de este servicio.
 - Integrar el servicio *Walking for Health* en el sistema actual.
2. Creación, desarrollo e integración del **proceso de caminatas saludables**.
 - Implementar el proceso de pre-caminata y post-caminata. En el proceso de pre-caminata se gestionará todo lo referente a la creación, reserva y visualización en el calendario de los paseos saludables. En el proceso de post-caminata se gestionará la información pertinente recogida una vez que la caminata haya tenido lugar.

2. OBJETIVOS

- Creación de formularios que permitan registrar y reservar caminatas saludables en el sistema, en un horario y fecha programadas.
3. Desarrollo de **formularios de salud** que permitan registrar información relevante para su posterior estudio y análisis.
- Incluir preguntas que permitan evaluar si el caminante es vulnerable o vive con una persona vulnerable. De ser afirmativa alguna de las preguntas anteriores se deberá cumplimentar la Evaluación de Riesgos *COVID*.
 - Creación del formulario de la Evaluación de Riesgos *COVID*. Este formulario contendrá una serie de preguntas útiles, para que el administrador del sistema almacene datos relevantes sobre el *COVID* previos a la realización de la caminata.
4. Creación de **nuevas funcionalidades escalables y consistentes** que permitan gestionar caminatas saludables y dar soporte para su correcta visualización.
- Incorporar un mecanismo que permita avisar a los caminantes de las rutas programadas.
 - Creación de eventos en el calendario del sistema que permitan al administrador visualizar las caminatas programadas en una fecha y hora determinadas.
 - Incorporar nuevos eventos en el calendario del sistema que permitan gestionar las sesiones de formación de los voluntarios.
 - Creación de formularios que recojan y almacenen la información necesaria después de que el caminante haya asistido al paseo saludable.
5. Implementación de una **arquitectura modular**.
- La modularidad es la posibilidad de subdividir un sistema en piezas más pequeñas llamadas módulos, las cuales deben ser tan independientes como sea posible de los restantes módulos. Cada módulo debe asegurar que un cambio en su interior no afecta al exterior. Para alcanzar esto se requiere de una alta cohesión y un bajo acoplamiento.
- La implementación de la modularidad a través del diseño de una arquitectura permite lograr una coherencia visual entre unidades de contenido de distinto tipo, ya sea elementos o componentes.
- Una de las principales ventajas de usar este enfoque es poder reutilizar los módulos en distintos contextos. La reusabilidad que resulta de integrar este tipo de arquitectura permite optimizar el flujo de trabajo.
- Al seguir una arquitectura modular, se divide el sistema en módulos independientes, los cuales son un grupo de acciones denominadas funciones o submódulos que comparten un conjunto de datos comunes llamados atributos.

Capítulo 3

Estado del Arte

EN este capítulo se explican y analizan las distintas herramientas y tecnologías que han sido utilizadas para la creación y diseño del servicio desarrollado en este TFG. Indicando además sus ventajas y desventajas.

Finalmente, se procederá a realizar un estudio comparativo de algunas de las aplicaciones y sistemas web existentes en la actualidad que fomentan la actividad física y los paseos saludables.

3.1 Desarrollo tecnológico en salud

La tecnología está cambiando de manera constante el mundo. En consecuencia, el ámbito de la salud también se ha beneficiado de este cambio.

Un ejemplo de ello fue la aparición de la pulsera de actividad *Power Balance*. Esta pulsera mide las calorías quemadas, el sueño, entre otros. Actualmente existen dispositivos o aplicaciones más actualizadas. Por ejemplo: entrenadores virtuales, programas de adelgazamiento, aplicaciones para medir el oxígeno en sangre, etc. Éstos son ejemplos de cómo la tecnología ha hecho posible que exista una mayor preocupación por cuidar la salud.¹

A continuación, se muestra un ejemplo claro de lo que ha supuesto el desarrollo tecnológico en la medicina y en la evolución de la salud en general.

3.1.1 Telemedicina

La telemedicina es definida según la Organización Mundial de la Salud (OMS) como la prestación de servicios de salud por parte de profesionales sanitarios, a través de la utilización de las Tecnologías de la información y la comunicación electrónicas (TIC). Esto permite el intercambio de información válida para el diagnóstico, tratamiento, prevención de enfermedades, investigación, evaluación y formación de profesionales sanitarios. Todo ello con el objetivo final de mejorar la salud de la población y de las comunidades².

¹<https://dirigentesdigital.com/tecnologia/la-tecnologia-esta-cambiando-el-mundo>

²<http://www.atryshealth.com>

3. ESTADO DEL ARTE

A continuación, se explican los 3 tipos de telemedicina que existen en la actualidad³.

- **Monitorización de pacientes a distancia.** Permite el control domiciliario de pacientes con enfermedades crónicas utilizando dispositivos que recopilan datos sobre niveles de azúcar en sangre, presión arterial y otros signos vitales.
- **Tecnologías de almacenamiento y envío.** Permite el almacenamiento de datos clínicos para poder enviarlos a otros centros médicos como la obtención de imágenes de rayos X en el ambulatorio y enviarlas al momento a un centro especializado para su interpretación.
- **Telemedicina interactiva.** Este tipo de telemedicina permite a médicos y pacientes comunicarse en tiempo real. Suele ser una videoconferencia, que el paciente puede realizar desde su casa o en un centro de salud cercano.

Cabe destacar el progreso en las comunicaciones y tecnologías de la información aplicadas al ámbito de la salud. Este gran avance en la sociedad ha permitido desarrollar en gran medida la telemedicina.

Hoy en día, se presenta como herramienta clave para mejorar la calidad de vida de las personas, lograr un sistema de salud más sostenible, ahorro de costes y mayor eficiencia a nivel asistencial. Además, supone una mejora de la atención clínica en aquellas regiones de difícil acceso o inaccesibles a la asistencia sanitaria. En esos países, la telemedicina puede dar más poder a los profesionales sanitarios locales mediante el apoyo a la autosuficiencia.

La telemedicina acorta distancias entre médico y paciente⁴ y mejora la eficiencia en los servicios sanitarios, tanto públicos como privados, mejora la calidad de atención, la eficiencia y seguridad. La telemedicina fomenta el apoyo inmediato a la población brindando mejor calidad en la atención. Además, mejora la distribución geográfica de los recursos humanos en salud y disminuye el aislamiento en la práctica de salud rural.

Con la telemedicina, la relación médico–paciente pasa a ser más colaborativa y participativa, y el número de profesionales de la salud que recomiendan recursos digitales a sus pacientes como complemento de su visita está aumentando cada vez más.

Sin embargo, aunque las evidencias demuestran que la telemedicina se ha utilizado en prácticamente todos los países del mundo⁵, son pocos en los que esté completamente integrada en la práctica médica. Los médicos no suelen realizar procedimientos médicos de forma remota a través de la telemedicina. En cambio, la telemedicina se utiliza principalmente como una herramienta de diagnóstico para apoyar el sistema de salud local.

³<https://clinic-cloud.com/blog/que-es-telemedicina-definicion-tipos/>

⁴<https://campus.sanofi.es/es/noticias/2020/telemedicina-y-como-se-aplica-en-los-sistemas-de-salud>

⁵<https://nacionfarma.com/origenes-de-la-telemedicina-y-casos-de-exito-en-el-mundo/>

Los usos más habituales de la telemedicina son la teleconsulta, telecardiología, teleradiología, y teledermatología.

La telemedicina puede ser utilizada como una herramienta de enseñanza en la educación médica, con la garantía de que un médico supervise y confirme el diagnóstico.

Por otra parte, algunos problemas con los que se encuentra la telemedicina son la resistencia de algunos pacientes y médicos para adoptar los nuevos avances, ya que prefieren los modelos tradicionales. Otros médicos carecen de los conocimientos necesarios en las TIC para utilizar eficazmente los enfoques de telemedicina.

Otro obstáculo importante son las consideraciones legales, como la falta de políticas que rijan la privacidad de los pacientes y la confidencialidad respecto a la transferencia, almacenamiento e intercambio de datos entre profesionales de la salud⁶.

Otra de las preocupaciones con respecto a la telemedicina es su relación con la tecnología. Por ejemplo, un ancho de banda bajo puede llevar a una mala resolución de la imagen, lo que puede comprometer el diagnóstico médico.

Finalmente, se podría decir que los beneficios que aporta la telemedicina son muchos más que sus desventajas [SPL⁺11]. Por esta razón, en un futuro no muy lejano, la telemedicina cobrará mayor importancia en la sociedad.

3.1.2 Por qué es importante invertir en salud

Hoy en día, la esperanza y calidad de vida de los ciudadanos de los países en desarrollo se está viendo incrementada. Esto es interesante para invertir en empresas que se dedican directa o indirectamente a ello, es decir, creando contenido específico para aumentar la calidad de vida de las personas y con ello, consiguiendo que sean más felices.

Más aún, tras la pandemia provocada por el *COVID-19* pone de manifiesto la importancia que tiene la salud en nuestras vidas, así como realizar una inversión en un entorno de incertidumbre económica y social.

Con los años, aumenta la población mundial y con ello se requiere una mayor atención sanitaria. Un ejemplo de ello es que a medida que la población envejece aumenta la probabilidad de padecer patologías, lo que se traduce en un mayor gasto de sanidad por parte de los gobiernos e individuos. Se estima que para 2050, los habitantes del planeta mayores de 60 años pasarán de 605 millones a 2000 millones⁷. Esto se convierte en una población mundial que está envejeciendo a pasos acelerados.

Un motivo fundamental, por el que es importante invertir en salud son los buenos resultados en este ámbito tras muchos años de investigación. Durante los últimos años las compa-

⁶<https://campus.sanofi.es/es/noticias/2020/telemedicina-y-como-se-aplica-en-los-sistemas-de-salud>

⁷<https://www.who.int/health-topics/ageing>

ñías farmacéuticas han realizado importantes inversiones en investigación en torno a la salud y al bienestar.

Los grandes fondos de capital de riesgo han situado su objetivo en pequeñas empresas que se dedican a la investigación y desarrollo de proyecto relacionados con la salud y el bienestar. Gracias a los fondos que reciben estas empresas se están realizando inversiones muy importantes especializadas en este sector, con proyectos que requieren de un menor tiempo e inversión que los realizados por grandes empresas.

3.1.3 Visión general de los sistemas de desarrollo tecnológico en salud

En la década en la que vivimos, cada vez son más los sanitarios que atienden a personas con enfermedades y problemas relacionados con la poca actividad física y una alimentación inadecuada [Her04]. Por tanto, para adecuar la demanda creciente con la disminución de los recursos, se deberán rediseñar los sistemas haciéndolos más tecnológicos y desarrollando la disciplina de la informática en la salud.

La introducción de nueva tecnología en la prevención, diagnóstico y recuperación del paciente repercute en los indicadores de resultados en salud, así como en el incremento de la esperanza y calidad de vida de las personas.

Los sistemas sanitarios son sistemas socio-técnicos en los que los resultados surgen de la interacción de la población y de las tecnologías. Por tanto, no se podrán diseñar nuevos sistemas de organización eficientes independientes del desarrollo de las Tecnologías de la Información (TI).

De esta manera, las TIC serán indispensables en todos los sistemas sanitarios. La e-salud se define como el uso de las TIC para mejorar la salud de las personas.

Hoy en día, la aparición de las TIC está causando nuevos cambios en los sistemas de salud. Uno de los cambios más importantes es la interacción de los sistemas de salud con el paciente. Por este motivo, el trabajo de los sanitarios se realizará en Internet, lo que implicará nuevas posibilidades en la gestión clínica y la planificación sanitaria que afectará también a la docencia e investigación.

Cabe recordar que, debido al envejecimiento de la población y las enfermedades, el gasto en salud se está viendo incrementado a un mayor ritmo. Actualmente, existe una preocupación a nivel mundial por la sostenibilidad de los sistemas de salud. Más aún, con la pandemia que se está viviendo. Esto obliga a buscar un equilibrio entre la incorporación de las nuevas tecnologías y los resultados en salud⁸.

⁸<https://www.antaes-healthlines.com/nc/articulo/la-innovacion-tecnologica-impacta-en-los-sistemas-de-salud/>

Por lo tanto, el impacto de las nuevas tecnologías en el ámbito de la salud conlleva a plantear 5 aspectos básicos:

- **Evaluación.** En la actualidad, las tecnologías más punteras suponen un alto coste. Por ese motivo, es necesario evaluar las necesidades tecnológicas que se requieren, el momento en el que se deben implantar en el sistema y adecuar las tecnologías a las necesidades de la población.
- **Modelos de compra.** Los sistemas de desarrollo tecnológico basados en la salud, deben analizar las necesidades y materializar el beneficio que se espera de la innovación en este sector.
- **Cambio de enfoque hacia el estilo de vida del paciente.** El cambio de hábitos en la vida de la población implicaría un gran impacto en la salud. Por tanto, cada vez más, es necesario introducir en el mercado aplicaciones y servicios inteligentes enfocados en la prevención y promoción de un estilo de vida saludable.
- **Cambio de enfoque en el modelo de prestación servicios.** La innovación tecnológica está ayudando a cambiar los modelos de prestación de servicios. Un ejemplo de ello es la aplicación de la tecnología al seguimiento de pacientes a través de la telemedicina. Este modelo conlleva innumerables ahorros al sistema sin disminuir la calidad de la asistencia prestada a los pacientes.
- **Priorizar tecnologías.** Es necesario tener un conocimiento anticipado de las tecnologías emergentes para poder llevar acabo una evaluación de las tecnologías en la salud. Y con ello, invertir más recursos en formación para proporcionar las tecnologías más punteras a los sistemas orientados en salud.

3.1.4 Sistemas de fomento de la actividad física

Hoy en día, existen numerosos sistemas y aplicaciones orientados a mejorar la salud en general y fomentar la actividad física en todas las edades.

Walking for Health

El servicio *Walking for Health* desarrollado con la empresa FK para este TFG, es un gran ejemplo de ello.

Caminar es sencillo, gratis y una de las formas más fáciles que existen para empezar a ponerse en forma. Además, ayuda a ser más saludable, perder peso, compartir tiempo con amigos, estar en contacto con la naturaleza, reducir el estrés y los sentimientos de aislamiento.

Con este nuevo servicio web, todas las personas que así lo deseen podrán apuntarse a las caminatas saludables que se ofrecen. Este servicio surge de la idea de fomentar la actividad física de las personas en todas las edades.

3.2 Caminatas saludables

Caminar es uno de los ejercicios más saludables para conseguir equilibrar la mente, el cuerpo y mantenerse en forma. En concreto, la caminata saludable es una actividad que beneficia las funciones orgánicas y psicoemocionales.

En estos paseos, un conjunto de personas se reúnen para socializar mientras caminan, obteniendo al mismo tiempo ambos beneficios que esta actividad física les ofrece. De esta manera, se disminuye en gran medida la ansiedad y se genera un estado de mayor calma y armonía.

Algunos de los beneficios⁹ que trae consigo realizar asiduamente caminatas saludables son: la mejora de atención, control del cuerpo, mejora de la circulación, facilita la conciencia corporal, tonifica los músculos, mejora la comunicación con otras personas, aumenta la vitalidad, la concentración, permite la liberación de tensiones y estrés, etc.

A continuación, se va a realizar un estudio y análisis de algunos de los sistemas existentes en la actualidad que están orientados a promover la salud, los paseos saludables y la actividad física.

3.2.1 SUP Yoga Camp

SUP Yoga Camp¹⁰ es un campamento donde se motiva e impulsa el cuidado de la naturaleza combinado con prácticas deportivas. Las caminatas saludables están incluidas también, entre otras actividades, las cuales permiten caminar en la naturaleza con atención plena en el momento presente. Ofrecen numerosas actividades para ayudar a las personas a desconectar de su rutina diaria, como, por ejemplo: yoga y meditación, SUP Indoor en piscinas, SUP yoga o SUP Pilates.

La reserva de caminatas saludables se realiza a través de un formulario donde se pide a los nuevos usuarios el nombre, *e-mail*, número de teléfono, fecha y comentarios adicionales para la reserva de un paseo.

La Figura 3.1 muestra el procedimiento que hay que llevar a cabo para poder asistir a una caminata saludable.

3.2.2 New York Road Runners

Esta organización sin fines de lucro, cuya sede se encuentra en la ciudad de Nueva York tiene como misión ayudar e inspirar a las personas a través del *running* o carreras al aire libre.

En la Figura 3.2 se pueden visualizar las distintas caminatas y carreras que esta organización programa en distintas épocas del año.

⁹<https://www.supyogacamp.com/caminatas-saludables/>

¹⁰<https://www.supyogacamp.com/caminatas-saludables/>

Agenda Tu Salida de Caminatas Saludables Online

Reserva tu sesión con tiempo, con un mínimo de 24hs de antelación.
Confirmaremos tu cupo y reserva vía email y whatsapp. Muchas Gracias!

Nombre:

Email:

Número de teléfono:

Fecha:

Comentarios:

[Reserva ahora](#)

Figura 3.1: Formulario para reservar caminatas saludables en Sup Yoga Camp

Mes	Todas las distancias	todas las localizac...	Todo tipo de carre...	RESETEAR TODO
marzo				
07	MAR. SOL.	NYRR Washington Heights Mile	08 A.M. 1 Milla, Manhattan, No Miembro: \$ 25 / Miembro: \$ 20	INFORMACIÓN DE CARRERA ¿Tienes un código de invitación?
20	20 DE MARZO - 28 DE MARZO SE SIENTO.	Virtual United Airlines NYC Half Run por amor a correr	20 de marzo - 28 de marzo 13.1 millas, en cualquier lugar, gratis	+ AÑADIR AL CARRITO
20	20 DE MARZO - 28 DE MARZO SE SIENTO.	Virtual United Airlines NYC Half Run por la medalla	20 de marzo - 28 de marzo 13.1 millas, en cualquier lugar, no miembro: \$ 60 / miembro: \$ 50 Virtual 6	+ AÑADIR AL CARRITO

Figura 3.2: Calendario de eventos en NYRR

Cada año 695.000 corredores participan en carreras comunitarias, entrenamientos, programas y caminatas ofrecidas por esta organización. Además, ofrecen eventos y programas gratuitos para niños y adolescentes, sirviendo a más de 240.000 niños en todo el país¹¹.

3.2.3 Wikiloc

Wikiloc¹² es una aplicación web híbrida donde se pueden almacenar y compartir rutas al aire libre georreferenciadas y puntos de interés de todo el mundo. Las rutas pueden dividirse

¹¹<https://www.nyrr.org/es/>

¹²<https://es.wikiloc.com/>

3. ESTADO DEL ARTE

en varias categorías según el desplazamiento; incluyendo senderismo, ciclismo y multitud de actividades similares. Wikiloc está disponible para *iOS*, *Android* y versión web como puede verse en la Figura 3.3.

Esta aplicación permite encontrar la ruta ideal para el caminante con filtros como kilómetro, desnivel acumulado o también acotando la zona que le interesa en el mapa.

Otra característica que ofrece es la grabación de las actividades al aire libre sobre un determinado mapa. Obteniendo estadísticas en tiempo real como velocidad, distancia recorrida, gráficas de elevación, fotos y puntos de interés a lo largo de la ruta, para posteriormente poder almacenarlo en la aplicación.

En la Figura 3.3 se puede observar la interfaz de esta aplicación, así como las distintas rutas realizadas y puntos de interés para el caminante.



Figura 3.3: Interfaz de la aplicación Wikiloc

3.2.4 Runkeeper

Runkeeper¹³ es una aplicación para el seguimiento de actividades físicas haciendo uso del *Sistema de Posicionamiento Global (GPS)* del dispositivo. Está disponible para *iOS* y *Android*. Esta aplicación proporciona estadísticas de cada actividad al finalizar el entrenamiento. Runkeeper ofrece también suscripción premium llamada Runkeeper Go, que proporciona

¹³<https://runkeeper.com/cms/>

estadísticas más detalladas de la actividad, comparación de rendimiento y planes de entrenamiento.

En Runkeeper buscan mantener siempre la interfaz lo más sencilla posible. Para iniciar una carrera o ruta únicamente es necesario escoger el tipo de entrenamiento que se desea realizar, haciendo uso de la distancia recorrida, tiempo, ritmo o incluso por series personalizadas.

Por otra parte, cabe destacar que aunque la aplicación está centrada en el *running*, ofrece también soporte para muchos otros deportes como el ciclismo, ciclismo de montaña, senderismo, esquí, *snowboard*, patinaje, natación, remo o caminatas.

En la Figura 3.4 se muestra la interfaz gráfica de la aplicación.

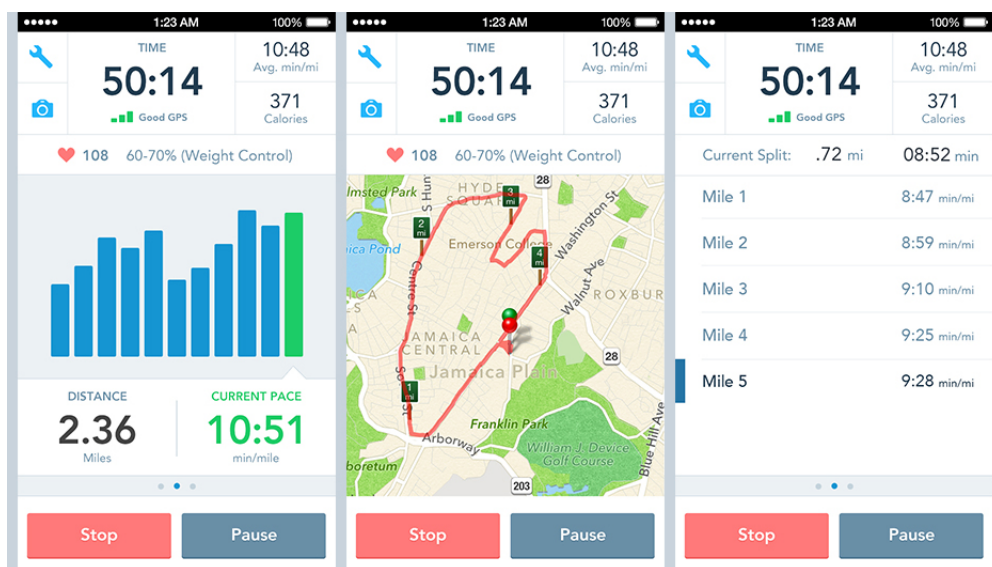


Figura 3.4: Interfaz gráfica de Runkeeper

3.2.5 Strava

Strava¹⁴ es una red social enfocada a deportistas, pero además es una aplicación de seguimiento GPS deportiva.

Los miembros de la red pueden ver la velocidad de la trayectoria, las diferencias de altura y el consumo de energía utilizando un teléfono inteligente o un dispositivo GPS compatible. Además, a través de sensores se pueden guardar más datos como la frecuencia cardíaca, la cadencia y la potencia. Las actividades pueden ser subidas desde dispositivos GPS compatibles a través de la aplicación para móviles *Android* o *iPhone* o subirlas manualmente desde un archivo.

Una vez registrada la actividad deportiva es posible obtener mediante la aplicación móvil o en el sitio web, un análisis detallado de los datos. Además, estos datos pueden ser comparados con los de otros miembros de la red, existiendo la posibilidad de seguir las actividades

¹⁴<https://www.strava.com/?hl=es>

3. ESTADO DEL ARTE

de otras personas y otorgar *Kudos* a sus actividades; parecido al *Me gusta* de Facebook¹⁵.

La lista de actividades incluye: carrera a pie (correr, caminar, senderismo), ciclismo, natación, esquí, patinaje, surf, *crossfit*, escalada, actividades de gimnasio, entre otras. En la Figura 3.5 se muestra la interfaz gráfica de Strava para versión móvil.

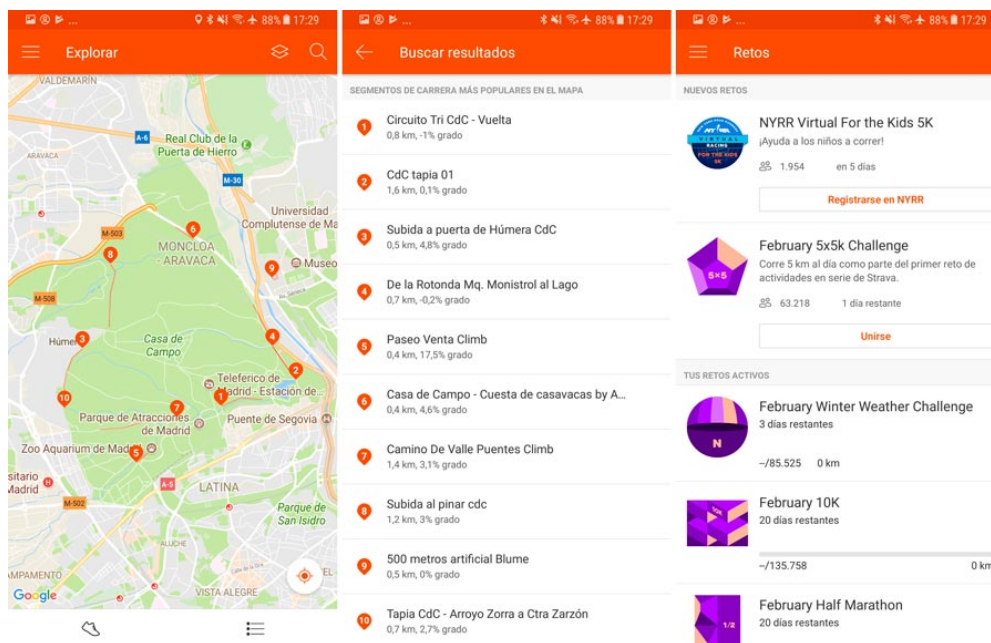


Figura 3.5: Interfaz gráfica de Strava

3.2.6 Google Fit

Google Fit¹⁶ es una aplicación creada por *Google* para medir el ejercicio físico. Utiliza diferentes parámetros para saber si se está haciendo el ejercicio suficiente. El principal objetivo es la de monitorización para poder tener una vida más sana y activa. Esta aplicación también tiene la opción de introducir otras actividades que no hayan sido registradas con la aplicación.

Todos los datos que recibe Google Fit, los utiliza para ofrecer objetivos de actividad con los que, con ayuda de la OMS, puedan mejorar el bienestar y la calidad de vida de las personas.

El primero de los objetivos que tiene esta aplicación son los minutos de actividad. Este parámetro va contando cada vez que la persona hace una actividad física, desde pasear, correr o realizar otras actividades similares. Y el segundo objetivo son los Puntos Cardio. *Google* ofrece este tipo de puntos con actividades que aceleren la frecuencia cardíaca como las caminatas rápidas.

La Figura 3.6 muestra la interfaz gráfica de Google Fit en su versión móvil.

¹⁵<https://support.strava.com/hc/es-mx/articles/216918397>

¹⁶<https://www.google.com/fit/>

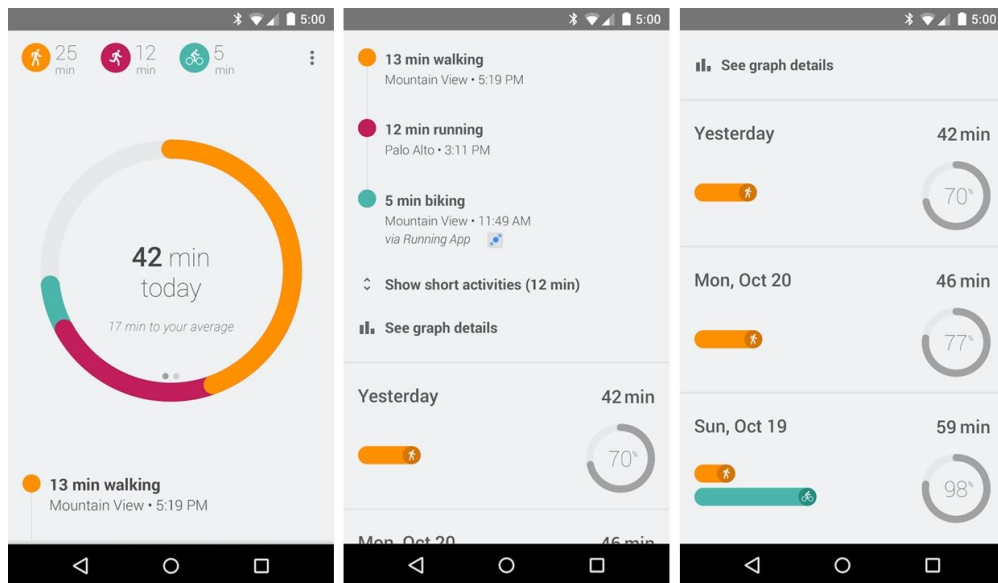


Figura 3.6: Interfaz de Google Fit

3.2.7 Comparativa de las páginas web analizadas

El objetivo de este proyecto es desarrollar un servicio web que soporte paseos saludables orientados a la población del condado de Suffolk (Reino Unido). Este servicio también incluirá la búsqueda de caminatas, creación de grupos, calendario de paseos programados y notificaciones para avisar a los caminantes.

En la Figura 3.7 se realiza una comparativa de las distintas características más significativas de las páginas web que han sido analizadas. Los criterios que se han seguido para ello se muestran a continuación.

1. **Calendario.** La página web dispone de un calendario donde se incluyen los distintos eventos. Además, se puede consultar en él la fecha y la hora.
2. **Disposición de filtros de búsqueda.** La página web permite filtrar por distintos campos de búsqueda, como por ejemplo: fecha, horario, lugar, etc.
3. **Almacenamiento de la información relacionada con las caminatas.** En la página web se puede almacenar la información relacionada con las caminatas. Por ejemplo, cuándo han asistido los caminantes al paseo o el día en el que ha tenido lugar.
4. **Variedad de paseos en el servicio ofrecido.** Se ofrece variedad en los paseos o caminatas que realizan, incluyendo rutas por diferentes lugares a los habituales, multicultural, etc.
5. **Disposición de métodos de contacto.** El servicio dispone de diversos métodos de contacto para que el caminante pueda ponerse en contacto con la persona encargada de llevar a cabo estos paseos. Estos métodos de contacto generalmente son a través de *e-mail*, SMS o Redes Sociales (RRSS).

3. ESTADO DEL ARTE

6. **Método para recibir notificaciones.** Los caminantes pueden saber cuándo tendrán lugar los paseos por medio de un método de notificaciones, normalmente SMS o *e-mail*.
7. **Registro post-caminata.** La información de la caminata realizada es guardada en un sistema gestor de información, como por ejemplo, una base de datos.
8. **Creación de grupos de caminatas.** La página web permite crear grupos de caminatas y asociar a los caminantes a dichos grupos.

Figura 3.7: Tabla comparativa de las aplicaciones analizadas

	Walking for Health	SUP YOGA CAMP	Wikiloc	Runkeeper	NYRR	Strava	Google Fit
Calendario	SI	-	-	SI	SI	SI	-
Filtros de búsqueda	SI	-	SI	SI	SI	SI	SI
Almacenamiento de información de las caminatas	SI	-	SI	SI	-	SI	SI
Variedad de paseos	SI	SI	SI	-	SI	SI	-
Contacto	SI	SI	SI	SI	SI	SI	SI
Notificaciones	SI	-	SI	SI	-	SI	SI
Registro post-caminata	SI	-	SI	SI	-	SI	SI
Creación de grupos de caminatas saludables	SI	SI	SI	-	-	-	-

3.3 Desarrollo web

En esta sección se comentarán los aspectos fundamentales del desarrollo web, así como las arquitecturas existentes en la actualidad y las tecnologías web utilizadas en este TFG.

3.3.1 Visión general

El desarrollo web implica mantener y construir sitios en la web. El trabajo de un desarrollador web es un trabajo en segundo plano. Esto permite a una página web tener un funcionamiento rápido, buena apariencia y gran rendimiento para conseguir una agradable experiencia al usuario.

Existen tres tipos de visiones¹⁷.

- El desarrollador *frontend* es el que se encarga del diseño e interactividad de la página web, utilizando lenguajes como *Cascading Style Sheets (CSS)*, *JavaScript (JS)* y *HyperText Markup Language (HTML)*.
- Mientras que el desarrollador *backend* es el que se encarga del almacenamiento de los datos asegurando un desempeño adecuado, es decir, se encarga de lo que no ve el usuario. Utiliza algunos lenguajes del lado del servidor como *NodeJS*, *Hypertext Preprocessor (PHP)* o *Java*, *Ruby*.
- El desarrollador full-stack developer es aquel que se encarga tanto del *frontend* como del *backend*.

3.3.2 Arquitecturas web

Una aplicación web es proporcionada por un servicio web y utilizada por clientes que se conectan desde cualquier navegador.

La arquitectura web tiene tres componentes fundamentales: un servidor web, una conexión de red y uno o más clientes.

El servidor web distribuye páginas de información formateada a los clientes que las solicitan. Los requerimientos son realizados a través de una conexión de red, y para ello se usa el protocolo *HTTP*. Una vez que se solicita esta petición y la recibe el servidor web, éste localiza la página web en su sistema de archivos y la envía de vuelta al navegador que la solicitó. Las aplicaciones web están basadas en el modelo cliente/servidor que gestionan servidores web y que utilizan como interfaz páginas web.

Las capas o niveles de un sistema web clásico¹⁸ son las siguientes:

1. **Nivel de presentación.** Es el encargado de generar la interfaz de usuario en función de las acciones llevadas a cabo por dicho usuario.

¹⁷<https://economiatit.com/desarrollador-web/>

¹⁸<https://programacionwebisc.wordpress.com/2-1-arquitectura-de-las-aplicaciones-web/>

3. ESTADO DEL ARTE

2. **Nivel de negocio.** Contiene toda la lógica que modela los procesos de negocio y es donde se realiza todo el procesamiento necesario para atender a las peticiones del usuario.
3. **Nivel de administración de datos.** Encargado de hacer persistente toda la información, suministra y almacena información para el nivel de negocio [Jua04].

En la Figura 3.8 se muestra una arquitectura web tradicional.

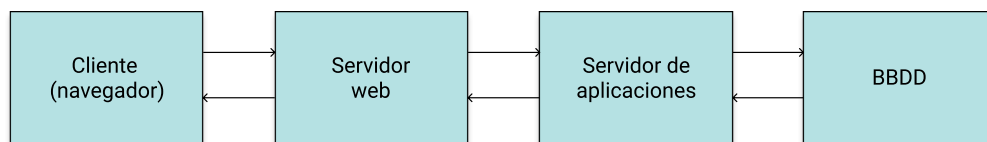


Figura 3.8: Arquitectura web

En cuanto a los componentes de una aplicación web, son los siguientes:

1. **Lógica de negocio.** Es la parte más importante de la aplicación, define los procesos que involucran al sistema y constituye el conjunto de operaciones que se requieren para proveer el servicio.
2. **Administración de datos.** Incluye la manipulación de bases de datos y archivos.
3. **Interfaz.** Los usuarios acceden a través de navegadores, móviles o *PDA*s, entre otros. Además, la funcionalidad es accesible a través del navegador.

Las aplicaciones web se modelan mediante lo que se conoce como modelo de capas. Una capa representa un elemento que procesa o trata una información. Los tipos de modelos se nombran a continuación.

- **Modelo de dos capas.** La información atraviesa 2 capas entre la interfaz y la administración de los datos.
- **Modelo de n-capas.** La información atraviesa varias capas. El más común es el modelo de 3 capas.

Modelo de 2 capas. Gran parte de la aplicación corre en el lado del cliente.

Las capas son:

- **Cliente (*fat client*.)** La lógica de negocio está inmersa dentro de la aplicación que realiza la interfaz de usuario, en el lado del cliente.
- **Servidor.** Administra los datos.

Algunas de las limitaciones de este modelo es que es difícilmente escalable, tiene una alta carga de la red, la funcionalidad está limitada y el número de conexiones es reducido.

Modelo de 3 Capas. Está diseñada para superar las limitaciones de las arquitecturas ajustadas al modelo de 2 capas, introduce una capa intermedia (capa de proceso). Las aplica-

ciones web actuales se ajustan a este modelo. Entre presentación y los datos, los procesos pueden ser manejados de forma separada a la interfaz de usuario y a los datos. Esta capa intermedia centraliza la lógica de negocio haciendo la administración más sencilla. Además, los datos se pueden integrar de varias fuentes.

Las capas de este modelo son:

1. **Capa de presentación.** Se encarga de mandar la información a la capa de proceso para su procesamiento. Recibe los resultados de la capa de proceso y se encarga de visualizar la presentación al usuario.
2. **Capa de proceso (servidor web).** Recibe la entrada de datos de la capa de presentación, interactúa con la capa de datos para realizar operaciones y manda los resultados procesados a la capa de presentación.
3. **Capa de datos (servidor de datos).** Se encarga de almacenar, recuperar, mantener y asegurar la integridad de los datos.

3.3.3 Tecnologías web

Angular

Angular es un framework Modelo Vista Controlador (MVC) desarrollado por *Google* para el desarrollo web *frontend*. Permite crear fácilmente aplicaciones *Single-Page Applications* (SPA). De esta forma se puede separar la lógica del diseño, manteniendo siempre ambas partes conectadas gracias al *data binding*. La capa visual no sabe lo que sucede en la capa lógica pero mantiene el control sobre el *Document Object Model* (DOM) y permite actualizar el contenido siempre que se desee.

A continuación, se explican algunas de las ventajas¹⁹ que supone utilizar *Angular* para el desarrollo de sistemas web:

- *Angular* hace uso de *TypeScript*, lo que permite que las aplicaciones sean más fáciles de mantener, ya que cualquier cambio podrá llevarse a cabo rápidamente y sin errores.
- Utiliza componentes web. Estos son porciones de código que pueden ser reutilizadas en otros proyectos realizados con *Angular* (reusabilidad). Son fáciles de convertir también en componentes web nativos que pueden ser utilizados en otro tipo de aplicaciones web. Esto permite la actualización y mantenimiento eficiente del código.

Angular permite que las pruebas unitarias sean mucho más sencillas de realizar ya que utiliza componentes que son independientes entre sí.

- Tiene una mejor legibilidad ya que la coherencia en la codificación hace que la lectura del código sea una tarea relativamente sencilla para los nuevos desarrolladores. Por tanto, aumenta la productividad.

¹⁹<https://www.esparkinfo.com/advantages-of-angularjs.html>

3. ESTADO DEL ARTE

- Con *Angular* no se necesitan funciones adicionales de *getter* y *setter* dado que cada objeto que utiliza es *Plain Old Java Object (POJO)*. Esto implica que tenemos una definición de clases simples. Se permite la manipulación de objetos y podemos eliminar o agregar propiedades de objetos.

Node.js

Node.js es el entorno con tiempo de ejecución en tiempo real de JS. Fue creado por los desarrolladores originales de JS.

Tanto JS como *Node.js* se ejecutan en el motor de tiempo de ejecución JS V8 (V8 es el nombre del motor de JS creado por *Google Chrome*). Es lo que toma JS y lo ejecuta mientras navega con Chrome. Este motor utiliza el código JS y lo convierte en un código de máquina más rápido. El código de máquina es un código de nivel más bajo que la computadora puede ejecutar sin necesidad de interpretarlo primero, ignorando la compilación y por lo tanto, aumentando su velocidad.

Node.js utiliza un modelo de entrada y salida sin bloqueo controlado por eventos que lo hace ligero y eficiente. Permitiendo además, la creación de aplicaciones de red rápidas, ya que es capaz de manejar una gran cantidad de conexiones simultáneas con un alto nivel de rendimiento, lo que equivale a una alta escalabilidad [TV10].

En comparación con las técnicas tradicionales de servicio web donde cada conexión (que crea una solicitud) genera un nuevo subproceso, ocupando la *Random Access Memory (RAM)* del sistema y regularmente maximizando la cantidad de *RAM* disponible.

Cuando llega una nueva solicitud se genera un tipo de evento. El servidor empieza a procesarlo y, cuando hay una operación de bloqueo de entrada y salida no espera hasta que se complete y en su lugar crea una función de devolución de llamada. El servidor empieza a procesar otro evento y cuando finaliza la operación de entrada y salida, continuará trabajando en la solicitud ejecutando la devolución de llamada tan pronto como sea posible.

Algunas de las ventajas que *Node.js* proporciona a los desarrolladores web son las siguientes:

- Es un lenguaje fácil de aprender que puede ser manejado por programadores *Java*, ya que JS se encuentra incorporado en la plataforma de *Node.js*.
- El modelo de entrada y salida basado en eventos facilita el manejo simultáneo de peticiones.
- Permite enviar archivos de gran peso mediante el uso de esta tecnología innovadora.
- La compilación se realiza en tiempo de ejecución, lo que permite una mayor optimización de las funciones.
- Alta escalabilidad.

- Alto rendimiento en proyectos y sistemas que requieran de ejecución en tiempo real.

Node.js es una de las plataformas software más utilizada hoy en día y se ha convertido en una de las más populares para el desarrollo de aplicaciones web, aplicaciones de escritorio y servicios.

3.4 Computación en la nube

A continuación, se va a hablar de la computación en la nube, también conocida como *Cloud Computing*, así como sobre cuáles son sus principales ventajas y los distintos tipos que existen.

3.4.1 Conceptos fundamentales

La computación en la nube es una tecnología que permite acceder de manera remota desde cualquier lugar del mundo y en cualquier momento a software, almacenamiento de archivos y procesamiento de datos a través de Internet, sin necesidad de tener que estar conectado a un ordenador o servidor local.

Además, *Cloud Computing* utiliza la conectividad e Internet para hospedar recursos y programas, permitiendo al usuario ingresar en ellos a través de cualquier dispositivo.

3.4.2 Cómo funciona Cloud Computing

En la computación en la nube se utiliza un servidor remoto para poder conectar los dispositivos de los usuarios a los recursos centralizados. Un servidor remoto central es el que se encarga de almacenar los datos y programas necesarios. Éste puede estar en el propio país o incluso al otro lado del mundo [Tho11].

Existen 3 tipos de computación en la nube²⁰ o modelos de despliegue:

- **Nube pública.** Los recursos computacionales son provistos por un tercero y están disponibles para cualquier persona que desee hacer uso de ella.

En la nube pública todo está disponible en Internet y se comparte entre varios usuarios de manera simultánea, lo que permite que los recursos se mantengan estandarizados. Este tipo de computación en la nube es la más indicada para los negocios que quieran ahorrar en inversión.

- **Nube privada.** En este tipo de nube, la empresa tiene la infraestructura en su dominio interno y ofrece acceso restringido a los usuarios seleccionados por ella. Como la nube está diseñada exclusivamente para la empresa, todos los procesos se dirigen a la realidad de su negocio.

La nube privada se usa por organizaciones que deben cumplir ciertas normas sobre la seguridad y privacidad de los datos. Como instituciones financieras y gubernamentales.

²⁰<https://openwebinars.net/blog/tipos-de-cloud-computing/>

3. ESTADO DEL ARTE

- **Nube híbrida.** Es la unión de los dos tipos mencionados anteriormente, es decir, esta combinación permite que se compartan datos y aplicaciones entre ellas. Teniendo en cuenta que algunos recursos se utilizan de manera privada y otros de manera pública o bien enlazados mediante otras tecnologías.

3.4.3 Ventajas de Cloud Computing

- **Trabajo remoto.** Se puede acceder a los datos de manera remota a través de cualquier dispositivo que esté conectado a Internet.
- **Centralización de la información.** Se evita que los datos se mantengan en distintos programas, con distintos tipos de acceso y autenticación.
- **Reducción de costos de infraestructura.** Permite reducir o eliminar el gasto que conllevan las compras de hardware, software, instalación y manutención de los sistemas.
- **Economía de espacio.** Los recursos se encuentran almacenados en la nube.
- **Seguridad.** Debido a los controles minuciosos y alta seguridad en todos los datos que se encuentran en la nube.

Otra ventaja significativa de la computación en la nube son los servicios de asistencia técnica y *backups*, usabilidad de plataformas y actualización automática del servidor.

3.4.4 Modelos de negocio

Para poder diseñar una empresa y que ésta tenga éxito depende de numerosos factores. Entre ellos, el diseño del sistema, el plan financiero, los gastos y el modelo de negocio, el cual permite especificar cómo obtiene los beneficios una empresa. El modelo de negocio permite visualizar el funcionamiento que debe tener una empresa a partir de los factores mencionados anteriormente, es decir, describe la lógica de cómo una empresa crea y entrega valor.

Tipos principales de modelo de negocio en el Cloud Computing

- **Communication as a Service (CaaS).** También conocido como **Comunicación como Servicio**, es la solución de comunicaciones de una empresa externa. Los proveedores de este tipo son los responsables de gestionar el hardware y el software requerido para la entrega de *VoIP* y videoconferencias a sus clientes. Algunas empresas que ofrecen este tipo de servicio son *Skype* o *Telco Inc.*
- **Infrastructure as a Service (IaaS).** Este tipo de modelo se centra en un modelo de servicio que entrega una infraestructura estandarizada y optimizada para las aplicaciones del cliente. Se conoce también como **Infraestructura como Servicio**. Las empresas que ofrecen este tipo de servicio son por ejemplo: *Fexiscale* o *GoGrid*.

- **Monitoring as a Service (MaaS).** Este modelo ofrece seguridad de un externo. También es llamado **Monitoreo como Servicio**. El monitoreo de la seguridad requiere proteger a un cliente de posibles amenazas cibernéticas. Un ejemplo de ello es *Panda Soft*.
- **Platform as a Service (PaaS).** Es aquel modelo que permite desarrollar y ejecutar aplicaciones web. Este modelo provee todas las facilidades para soportar el ciclo de vida completo, desarrollo de aplicaciones y servicios web disponibles en Internet. Se conoce también como **Plataforma como Servicio**. Las empresas más conocidas que utilizan este modelo son: *Microsoft Azure*, *RollBase*, *Google App Engine*, *Amazon Web Services (AWS)*.
- **Software as a Service (SaaS).** También es llamado **Software como Servicio**. Es un modelo de distribución de software en el que las aplicaciones se centralizan por un proveedor de servicio y se encuentran disponibles sobre una red, que normalmente es Internet. Algunas empresas que lo utilizan son: *Salesforce*, *MobileMe*, *Google Apps*, entre otros.

Este modelo SaaS ofrece numerosas ventajas entre las que destacan: ausencia de hardware, alta escalabilidad, alta tasa de aceptación en la organización, rápidas actualizaciones, alta recuperabilidad y ahorro de costes.

3.4.5 Plataformas de computación en la nube

Existen numerosas plataformas que permiten desplegar *Cloud Computing*. Algunas de las más conocidas hoy en día son: *Windows Azure*, *Google App Engine*, *Openstack*, *Amazon Elastic Compute Cloud (EC2)*, entre los más destacados.

En el siguiente apartado, se va a explicar de manera detallada la plataforma en la nube AWS, la cual ha sido fundamental para el desarrollo de este TFG.

Amazon Web Services

AWS es la plataforma en la nube más completa. Ofrece más de 200 servicios integrales de centros de datos a nivel global. Millones de clientes, incluso las compañías más grandes y los organismos gubernamentales líderes lo utilizan para reducir costos, aumentar su agilidad e innovar de una manera más rápida y eficaz. AWS es un tipo de PaaS.

AWS es una colección de servicios web ofrecidos a través de Internet en *Amazon*. Proporciona una plataforma de infraestructura escalable de alta fiabilidad y de bajo coste en la nube que impulsa a miles de empresas. Para el almacenamiento de datos, *Amazon* ofrece servicios de almacenamiento para bases de datos relacionales y bases de datos no relacionales, dependiendo de la necesidad del usuario [TCGS14].

3. ESTADO DEL ARTE

Cuenta con una gran cantidad de servicios y características que supera la de cualquier otro proveedor de la nube. Ofreciendo desde tecnologías de infraestructura hasta tecnologías emergentes como, por ejemplo: aprendizaje automático, inteligencia artificial o *Internet of Things (IoT)*. Esto hace que llevar las aplicaciones existentes a la nube sea más rápido, fácil y rentable.

La infraestructura se creó para cumplir con los requisitos de seguridad del ejército, los bancos internacionales y otras organizaciones que deben cumplir requisitos de confidencialidad estrictos. Esto cuenta con el respaldo de un amplio conjunto de herramientas de seguridad en la nube, con 230 servicios y características de seguridad y gobernanza. Es compatible con 90 estándares de seguridad y certificaciones de conformidad y los 117 servicios de AWS que almacenan datos de los clientes ofrecen la función de cifrar esos datos²¹.

AWS ofrece, entre muchas otras cosas, almacenamiento de datos públicos y privados, *hosting* web dinámico, *hosting* de aplicaciones web, *Virtual Private Network (VPN)*, gateway, *streaming* multimedia, auto escalado y equilibrio de carga.

En la Figura 3.9 se muestra de manera general todos los servicios ofrecidos hoy en día por AWS.



Figura 3.9: Servicios ofrecidos por Amazon Web Services

²¹<https://aws.amazon.com/es/what-is-aws/>

Metodología

4.1 Metodologías ágiles

LA metodología elegida para llevar a cabo el desarrollo de este proyecto ha sido una metodología *Agile*. En particular, se ha elegido la metodología *Scrum* [PDB⁺09].

El mundo del desarrollo de software ha cambiado desde la aparición de Internet, el cual permite crear ecosistemas de trabajo mucho más colaborativos en los que el flujo de información es mucho más rápido que con las estructuras tradicionales.

Las metodologías tradicionales de desarrollo de software se están viendo reemplazadas progresivamente por metodologías más actuales e innovadoras. Por este motivo, y por los grandes beneficios que ofrece implementar y seguir una metodología *Agile*, se ha optado por implementarla en el desarrollo de este TFG.

A continuación, se exponen algunas de las ventajas [CCD⁺07] que ofrece el uso de estas metodologías ágiles.

1. **Rápido aprendizaje del equipo de desarrollo.** Permite tener un aprendizaje más rápido que puede utilizarse en los siguientes *sprints* del proyecto, debido a que las interacciones se completan en un relativo corto periodo de tiempo. Concretamente, el desarrollo de este proyecto ha tenido una duración de 4 semanas en cada *sprint*.
2. **Mejor conocimiento de la dimensión del proyecto.** Gracias a las iteraciones o *sprints*, el proyecto se gestiona en bloques más sencillos y manejables. De esta forma, se puede identificar fácilmente los objetivos de cada etapa, así como los contratiempos que puedan surgir.
3. **Fecha de entrega más realista.** Al segmentar el objetivo de entrega en los *sprints* se consigue que la fecha de entrega se ajuste mucho mejor. De esta manera, las fechas de entrega finales se ajustan a lo que se ha planificado previamente.
4. **Responsabilidad y autonomía.** Utilizar una metodología *Agile* requiere que todas las partes del proyecto (clientes, proveedores, equipo, etc.) colaboren. Por tanto, se trata de una metodología que fomenta la responsabilidad del equipo y ofrece un alto nivel de autonomía por parte de los miembros.

4. METODOLOGÍA

5. **Feedback obtenido con mayor rapidez.** En las metodologías como *Scrum* se realizan reuniones diarias llamadas *Scrum daily meetings*, donde se proponen soluciones de manera inmediata a los problemas y contratiempos. De esta manera, el tiempo de reacción ante las incidencias disminuye significativamente.
6. **El producto final se ajusta más a lo que quiere el cliente.** Con las metodologías ágiles está contemplado en feedback del cliente como parte estructural del proceso de desarrollo. Esto favorece que el producto final se ajuste más a lo que el cliente necesita y no a lo que el cliente pidió en un primer momento.

4.1.1 Scrum

Scrum es una metodología de desarrollo *Agile*. Aunque surgió como modelo para el desarrollo de productos tecnológicos, también se emplea en entornos que trabajan con requisitos inestables y que requieren rapidez y flexibilidad; situaciones que se presentan con frecuencia en el desarrollo de sistemas de software [Ser16].

En la siguiente Figura 4.1 se muestra la metodología *Scrum* seguida para este proyecto.

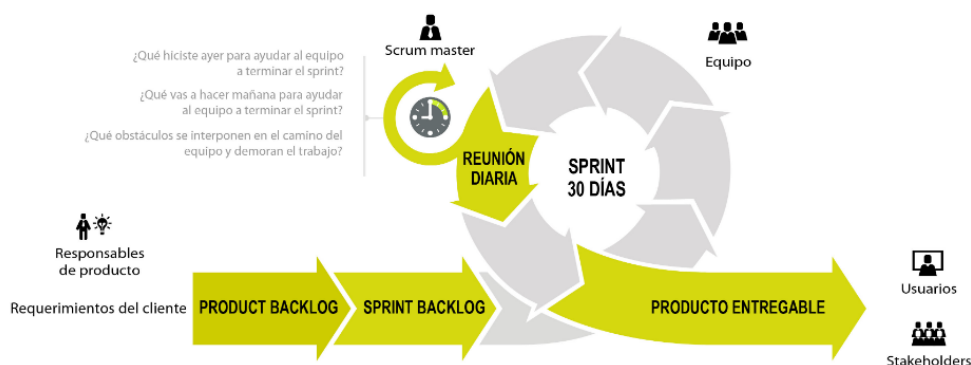


Figura 4.1: Visión general de la metodología ágil Scrum

<https://www.leadersummaries.com/es/libros/resumen/detalle/Metodologia-Scrum>

La metodología *Scrum* surgió de la inspiración de Nonaka y Takeuchi al ver el avance de formaciones de las melés en los partidos de rugby¹. De esta manera, pudieron bautizar esta nueva forma de trabajar, que ya se estaba dando en otras empresas como Honda o Canon.

Por eso, *Scrum* y equipo auto-organizado van siempre de la mano. La metodología *Agile Scrum* es la que presenta el *framework* de trabajo ágil más adecuado para integrar el Diseño Centrado en el Usuario [Roj17].

Sprints

Siguiendo la metodología *Scrum*, el proyecto se ha estructurado en una serie de *sprints*. Éstos son intervalos prefijados de tiempo (no inferior a una semana ni superior a un mes)

¹<https://proyectosagiles.org/historia-de-scrum/>

durante el cual se crea un incremento de producto como: “Hecho o Terminado” utilizable, potencialmente entregable. A lo largo de la fase de desarrollo, existen *sprints* consecutivos de duración constante.

Las ventajas de trabajar en un sistema que hace uso de *sprints* permite mejorar el foco, ya que sólo se trabaja durante un tiempo limitado en unas tareas concretas. El equipo puede enfocarse mejor en ellas y conseguir finalizarlas en el tiempo establecido. De esta forma, se mejora la transparencia y comunicación del proyecto.

Al trabajar con *sprints* se minimizan los riesgos y se puede realizar una traza del conjunto de tareas, conociendo en todo momento qué tareas están en curso, terminadas o pendientes de realización.

Características del sprint

Es fundamental estimar la duración que va a tener el *sprint*. Para este proyecto, la duración de cada *sprint* ha sido de 4 semanas. Esta elección ha sido tomada por las siguientes razones:

Al tener un *sprint* más largo, como el de este proyecto, es más sencillo ajustar las historias de usuario dentro del *sprint* para que constituyan un objetivo con valor real. Las historias de usuario son descripciones cortas y simples de una característica narrada desde la perspectiva del usuario o cliente del sistema. Por lo general, siguen una plantilla del tipo:

Como <Usuario>, Quiero <algún objetivo>, Para que <motivo>

El nuevo servicio se ha ido desarrollando poco a poco, lo que ha implicado que el impacto de las reuniones o eventos relacionados con los *sprints*, como la planificación o las revisiones, ha sido menor.

A diferencia de los *sprints* más cortos, el efecto de las incidencias, problemas y consultas habituales se desvanecen mucho más que en un *sprint* largo. Un *sprint* de larga duración va a resistir mucho mejor las grandes incidencias que puedan ocurrir.

Las fases de un *sprint* son las siguientes:

- **Reunión de planificación del sprint.** En esta reunión se define la funcionalidad en el incremento planeado, cómo se creará ese incremento y el objetivo final del *sprint*.
- **Scrum daily meeting.** El objetivo de esta reunión diaria es que el equipo de desarrollo sincronice las actividades y se evalúe el progreso para finalizar el trabajo en el *sprint product Backlog*. Algunas de las ventajas del *Scrum daily* son: la mejora de la comunicación en el equipo, elimina reuniones y obstáculos innecesarios para el desarrollo y mejoran el nivel de conocimiento del proyecto que se está realizando. En el desarrollo de este proyecto, la duración del *Scrum daily meeting* ha sido de 20 minutos aproximadamente.

4. METODOLOGÍA

- **Trabajo de desarrollo.** Los *sprints* permiten previsibilidad al garantizar la inspección y adaptación de los avances hacia una meta concreta. Además, hay que asegurarse de que no se realizan cambios que afecten al objetivo final del *sprint*.
- **Revisión del sprint.** Tiene lugar al final del *sprint* para adaptar si fuera necesario el *product Backlog*. Todo el equipo y los *stakeholders* participan y colaboran durante la revisión de lo que se realizó en el *sprint*.

El resultado de la revisión del *sprint* es la pila del *sprint*, la duración y fecha de la reunión de revisión y el objetivo del *sprint* [PR09].

- **Retrospectiva del sprint.** El objetivo principal es crear un plan de mejora que va a ser ejecutado durante el siguiente *sprint*. El propósito de retrospectiva es el siguiente:
 1. Revisar cómo fue el último *sprint* en lo que respecta a las personas, relaciones, procesos y herramientas.
 2. Identificar y ordenar los temas principales que salieron bien y las potenciales mejoras.
 3. Crear un plan para la implementación de mejoras con respecto a cómo el equipo *Scrum* hace su trabajo.

Roles de Scrum

Es necesario identificar los distintos roles cuando se trabaja con una metodología *Scrum*, ya que estos roles tendrán un papel fundamental en el desarrollo del sistema.

A continuación, se detallan los distintos roles y personas encargadas de cumplirlos en este proyecto.

- **Product owner.** Es el que se encarga de maximizar el valor del trabajo entregado y de priorizar las historias de usuario colocándolas en el *product Backlog*. Este rol será asumido por el tutor y co-tutor de este proyecto, David Vallejo Fernández y Santiago Sánchez Sobrino.
- **Scrum Máster.** Es el encargado de asegurar que el proyecto se desarrolla adecuadamente y de eliminar los problemas que puedan surgir durante el proceso. Este rol será llevado a cabo por el autor de este TFG, Noelia María Granados Carrasco.
- **Equipo de desarrollo.** En este proyecto, el equipo de desarrollo ha sido de un único desarrollador, que es también el autor de este TFG, Noelia María Granados Carrasco. Asumiendo los roles de desarrollador *backend*, *frontend*, administrador de la base de datos y *tester* del sistema.

Artefactos de Scrum

Existen una serie de elementos en la metodología *Scrum* que sirven de apoyo para que esta metodología *Agile* funcione correctamente. Estos elementos son el *product backlog* y el *sprint backlog*.

- **Product backlog.** Contiene todo el trabajo necesario para el desarrollo del sistema. Es el resultado del trabajo del *Product Owner* con los diferentes stakeholders.
- **Sprint backlog.** Es el artefacto que permite visualizar todo el trabajo incluido en el *sprint* en curso y cuyo objetivo es dar transparencia al estado del desarrollo durante el *sprint*. Una de las maneras de representarlo es mediante una pizarra *Kanban*², que se compone de tareas con diferentes horas de trabajo para ver cómo progresa y evoluciona el proyecto.

4.2 Plan de trabajo

A continuación, se va a mostrar la planificación del desarrollo del proyecto realizado.

La dedicación al TFG ha sido con un horario a tiempo parcial, debido a que se ha compaginado con las prácticas en la misma empresa, FK. La duración del tiempo invertido en días es una estimación, y en cada semana está representado el esfuerzo que se ha realizado a lo largo de todo el proyecto. La metodología ágil utilizada ha sido *Scrum* la cual fomenta el desarrollo del proyecto basado en *sprints*. Éste ha sido el enfoque empleado en el proyecto, ya que aporta los requisitos necesarios para cumplir con los objetivos que se marcaron al principio.

El cronograma que se puede visualizar en la Figura 4.3 muestra el desarrollo del servicio web realizado. Éste representa las principales fases de desarrollo del sistema.

En las Figuras 4.2 y 4.3 se muestran los diferentes *sprints* que se han llevado a cabo a lo largo de desarrollo del servicio web. Con cada *sprint* se alcanzan los objetivos del proyecto.

Nº de Sprint	Fecha de inicio	Fecha de finalización
Sprint 0	15-feb	14-mar
Sprint 1	15-mar	11-abr
Sprint 2	12-abr	09-may
Sprint 3	10-may	06-jun
Sprint 4	07-jun	04-jul

Figura 4.2: Sprints realizados en el desarrollo de Walking for Health

²<https://kanbanize.com/es/recursos-de-kanban/primeros-pasos/que-es-tablero-kanban>

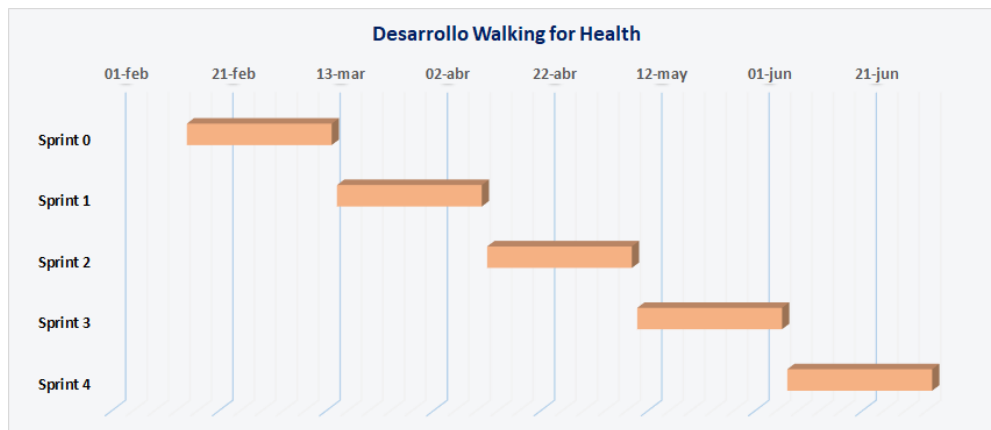


Figura 4.3: Cronograma de desarrollo del servicio web

4.3 Pila tecnológica utilizada en el proyecto

En esta sección, se explica la pila tecnológica que ha sido utilizada durante el desarrollo de este proyecto. La pila tecnológica estará dividida en 4 bloques: *backend*, *frontend*, orquestación y gestión del proyecto.

Backend

- **Node.js.** Es el entorno en tiempo de ejecución multiplataforma de código abierto, para la capa del servidor basado en el lenguaje de programación *JavaScript*. Node.js es asíncrono, con entrada y salida de datos en una arquitectura orientada a eventos y basado en el motor V8 de *Google*. En este proyecto, se ha hecho uso del entorno de ejecución de *JavaScript*, Node.js, en la versión 12.16.1 LTS y la versión *npm* ha sido la 6.13.4. Este *framework* ofrece un mejor rendimiento en comparación con otros como *Spring*, *Django* o *Laravel*.
- **NestJS.** La versión ha sido la 6.6.4. Este otro *framework* ha sido utilizado para el enrutamiento del lado del servidor y definición de *endpoints*. Ofrece una mejor integración con *Angular* que con *Express.js* (por ejemplo, TS por defecto o módulos).
- **AWS RDS PostgreSQL.** Es la base de datos relacional compatible con *PostgreSQL*. La versión utilizada ha sido la 11.6.
 - **TypeORM.** Utilizado para mapear clases *TypeScript* (entidades) a tablas de la base de datos.
 - **class-transformer.** Se utiliza para transformar entidades a *DTOs*, utilizados para la comunicación entre el cliente y el servidor.
- **Postman.** Utilizado para almacenar y compartir un entorno que haga uso de la Interfaz de Programación de Aplicaciones (API) del sistema. Gracias a esta potente herramienta se ha podido hacer *testing* y probar los *endpoints* del sistema.

- **Jest.** Para testear todo el código *JavaScript*. En este proyecto se ha utilizado como *framework* de prueba para realizar las pruebas unitarias y tests *end-to-end (e2e)* necesarios. Utilizar Jest ofrece numerosas ventajas a la hora de realizar tests, ya que proporciona informes de cobertura de código integrados, un entorno rápido y aislado. Jest realiza los tests en paralelo de forma distribuida en diversas instancias para maximizar el rendimiento. Funciona con *TypeScript* y se integra en él a través de *ts-jest*.

Frontend

- **Angular.** El cual es un *framework* para aplicaciones web desarrollado en *TypeScript*, de código abierto y mantenido por *Google*. *Angular* se utiliza para crear y mantener aplicaciones web. Concretamente se ha utilizado *Angular 8*.
- **SCSS.** Hoja de estilo que contiene el lenguaje *Sass*, similar a un archivo *.CSS* pero con capacidades adicionales. SCSS es usado para especificar el estilo de todos los componentes *Angular* del sistema.

Orquestación

- **AWS Elastic Beanstalk.** Facilita el despliegue de aplicaciones altamente escalables en diferentes tecnologías de *AWS* de forma automática (PaaS).
- **Angular Universal.** Facilita la integración entre las partes del lado del servidor y del lado del cliente de la arquitectura, proporcionando el renderizado del lado del servidor de las aplicaciones de *Angular* y una única solución mono-repo.
- **Docker.** Permite crear contenedores ligeros y portables para que las aplicaciones software puedan ejecutarse en cualquier máquina, independientemente del sistema operativo que la máquina tenga por debajo. De esta forma, se facilitan los despliegues.

En este proyecto se utiliza *Docker* para aislar algunos servicios, por ejemplo, la base de datos local.

- **Bitbucket.** Es un servicio de alojamiento basado en web, para los proyectos que utilizan el sistema de control de versiones Mercurial y Git. Todas las actualizaciones de código del servicio web se encuentran alojadas en un repositorio de Bitbucket.

Finalmente, se ha hecho uso de la utilidad de bash *Make* para ejecutar todos los comandos disponibles. Sirve como fachada a los *scripts npm*, comandos *docker* y otros *scripts*.

Herramientas para la gestión y coordinación del proyecto

- **Microsoft Teams.**³ Plataforma unificada de comunicación y colaboración que combina chat persistente en el lugar de trabajo, reuniones de vídeo, almacenamiento de

³<https://www.microsoft.com/es-es/microsoft-365/microsoft-teams/group-chat-software>

archivos e integración de aplicaciones. Esta herramienta ha sido utilizada para la comunicación del alumno con el tutor académico de este TFG.

- **Slack.**⁴ Plataforma de mensajes basada en canales que sirve como herramienta de comunicación en equipo. Ha sido utilizada para la comunicación con el equipo en la empresa donde se ha llevado a cabo el desarrollo de este proyecto.
- **Trello.**⁵ Software de administración de proyectos con interfaz web y cliente para *iOS* y *Android*. Se utiliza principalmente para organizar proyectos y está basado en tarjetas.

- **Tablero Kanban.** Es una herramienta ágil de gestión de proyectos diseñada para visualizar el trabajo, limitar el trabajo en curso y maximizar la eficiencia. La metodología a la que mejor se adapta es a *Scrum* debido a la división en *sprints*. Por este motivo y por las numerosas ventajas que ofrece ha sido utilizado en este proyecto.

En resumen, el tablero *Kanban* de este proyecto se divide en tres secciones básicas que muestran el estado de las tareas: **por hacer, en proceso y finalizado**. En inglés se conoce como *To Do, In Progress y Done*. Para ello, se han utilizado tarjetas para representar los elementos de trabajo y columnas en cada etapa del proceso. Las tarjetas se mueven de izquierda a derecha para mostrar el progreso y ayudar a coordinar mejor el trabajo.

La combinación de *Scrum* y *Kanban* se denomina *Scrumban*⁶, la cual combina las mejores características de ambos métodos. Reúne la naturaleza preceptiva de *Scrum* y la capacidad de mejora del proceso de *Kanban*, permitiendo a los equipos moverse hacia el desarrollo *Agile* y mejorar constantemente sus procesos.

En la Figura 4.4 se muestra el tablero *Kanban* utilizado en este proyecto.

- **Overleaf.**⁷ Editor colaborativo de LaTeX basado en la nube, que es comúnmente utilizado para escribir, editar y publicar documentos científicos. Ha sido utilizado para el desarrollo de la memoria de este TFG.
- **Figma.**⁸ Editor de gráficos vectoriales y herramienta de creación de prototipos que se basa principalmente en la web, con funciones adicionales sin conexión habilitadas por aplicaciones de escritorio para *macOS* y *Windows*. Esta herramienta ha sido utilizada para la realización de la mayoría de los gráficos y tablas presentes en este TFG.
- **Visual Studio Code.**⁹ Editor de código fuente desarrollado por *Microsoft* para *Windows*, *GNU/Linux* y *macOS*. Incluye soporte para la depuración, control integrado de

⁴<https://slack.com/intl/es-es/>

⁵<https://trello.com/es>

⁶<https://kanbantool.com/es/scrumban-scrum-y-kanban>

⁷<https://www.overleaf.com>

⁸<https://www.figma.com>

⁹<https://code.visualstudio.com>

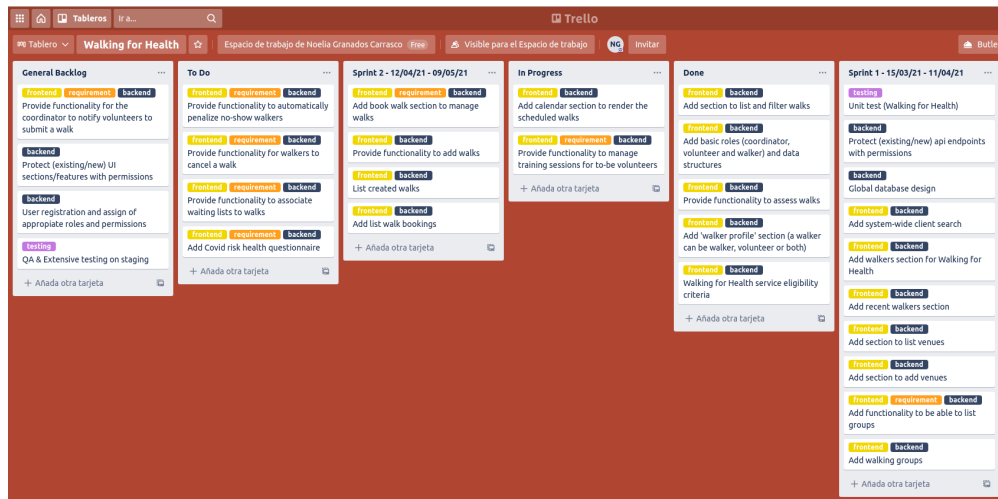


Figura 4.4: Ejemplo de tablero Kanban

Git, resaltado de sintaxis, finalización inteligente de código, fragmentos y refactorización de código. Esta herramienta ha sido utilizada para el desarrollo del servicio web.

- **Visual Paradigm.**¹⁰ Herramienta de soporte al modelado visual que permite generar distintos diagramas. Además del soporte de modelado, proporciona capacidades de generación de informes y código. Se ha utilizado para realizar los 2 diagramas de secuencia de pre-caminata y post-caminata del capítulo 5.

En la siguiente Figura 4.5 se muestra la pila tecnológica utilizada en el desarrollo de este proyecto.




















Backend	Frontend	Orquestación	Gestión/coordinación
    	 	   	       

Figura 4.5: Pila tecnológica del proyecto

¹⁰<https://www.visual-paradigm.com>

Capítulo 5

Arquitectura

EN este capítulo se va a proceder a analizar las distintas partes de las que se compone la arquitectura de este proyecto. También se explicará en detalle los distintos módulos que componen el servicio web.

5.1 Organización del proyecto

Para esta subsección se han diseñado una serie de diagramas que permiten tener una idea general de las distintas partes de la estructura del proyecto.

5.1.1 Diagrama de general del sistema web

El sistema es una aplicación web *Angular* que se comunica con un *backends Node.js* montado en la nube de *Amazon* y gestionado por el servicio de *AWS Elastic Beanstalk*, que es un PaaS. Este servicio permite añadir o eliminar recursos de acuerdo a diferentes criterios como, por ejemplo, la demanda de Central Processing Unit (CPU) o memoria.

Las instancias desplegadas en la nube, que son servidores virtuales montados sobre *AWS Elastic Compute Cloud (EC2)* mantienen su persistencia en el servicio *AWS Relational Database Service (RDS)*, un servicio gestor de bases de datos relacional que está configurado para utilizar *PostgreSQL*. Este servicio mantiene instancias de computación ejecutando un servidor *PostgreSQL* configurado en contextos de seguridad y copias de seguridad periódicas.

También se hace uso de contenedores o *buckets* de *AWS Simple Storage Service (S3)* para almacenar los archivos de la aplicación de carácter más específico al dominio del negocio como, por ejemplo, plantillas de formularios, copias de la base de datos anonimizadas para trabajo en local, etc.

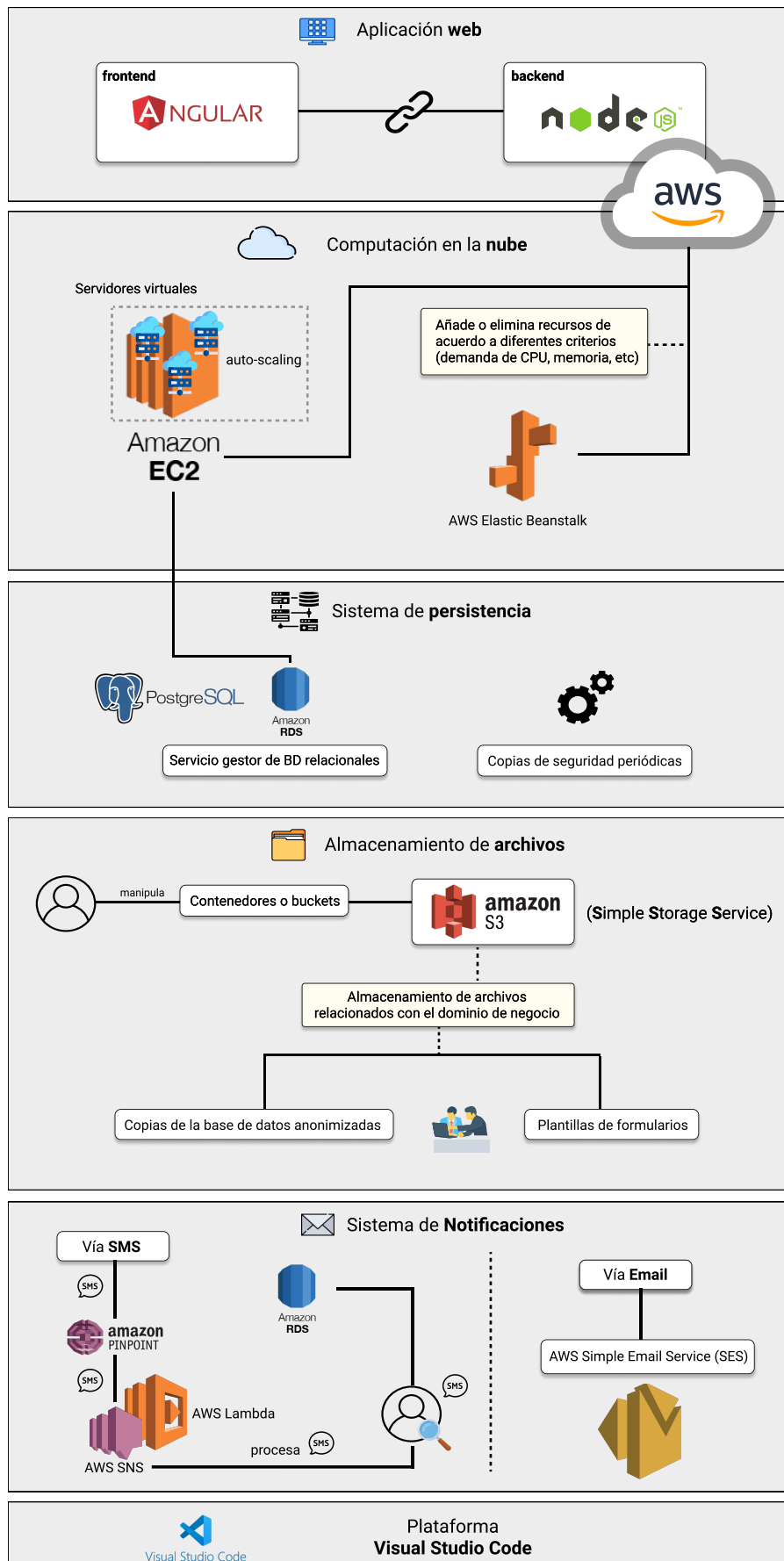
Para el sistema de notificaciones a usuarios vía *e-mail* se hace uso del servicio de *AWS Simple Email Service (SES)*. Y para las que son vía SMS se utiliza *AWS PinPoint*. Esto sirve para configurar un número de teléfono virtual que permita enviar y recibir mensajes. Y finalmente, se utiliza *Simple Notification Service (SNS)* para configurar una cola de recepción que disparará eventos cuando se reciban los mensajes al número de teléfono definido.

En la siguiente Figura 5.1 se representa el diagrama general del sistema.

5. ARQUITECTURA

Figura 5.1: Diagrama general del sistema

Esta imagen se ha diseñado con recursos de Flaticon.com: <https://www.flaticon.com/>



5.1.2 Arquitectura frontend y backend del sistema

Tanto la aplicación servidor como la aplicación del cliente son compiladas y desplegadas en una sola aplicación haciendo uso de *Angular*. Aunque ambas aplicaciones compartan la misma base de código, están aisladas y sus clases/métodos no pueden ser compartidos excepto la parte común, es decir, utilidades, *DTOs*. Esas partes comunes no tienen dependencias del cliente ni del servidor.

Al final, el cliente no puede acceder a la base de datos por lo que tiene que comunicarse con el servidor para pedir los datos necesarios. Esto se hace enviando y recibiendo *DTOs*, que son objetos simples que se deserializan con la información relevante necesaria para que el servidor haga las consultas. De esta forma, el servidor responde al cliente con otros *DTOs* llenos de los datos solicitados. Los *DTOs* deben ser lo más simples y aislados posibles, ya que su uso está orientado a ser utilizados de forma independiente en cada *endpoint* de la aplicación.

Para ello, se ha hecho uso de la API REST, la cual permite que el cliente no necesite conocer los detalles de implementación del servidor y el servidor no tenga que preocuparse por cómo se usan los datos que envía el cliente.

Los métodos principales de una API REST son los siguientes:

- **GET.** Se utiliza para poder recuperar un determinado recurso.
- **POST.** Se usa la mayoría de las veces para crear un nuevo recurso. También se utiliza para enviar datos a un recurso ya existente para su procesamiento.
- **PUT.** Este método se utiliza para crear o editar un recurso.
- **PATCH.** Se encarga de realizar actualizaciones. En el cuerpo de la petición se añaden los cambios que deben hacerse.
- **DELETE.** Este último método se utiliza cuando se quiere eliminar un recurso.

Los recursos en una API REST se identifican por medio de una Uniform Resource Identifier (URI). Esta URI permite acceder al recurso o realizar cualquier operación de modificación sobre éste. Concretamente, en el desarrollo del servicio web se han utilizado 2 tipos de *DTOs*.

- **DTOs de petición.** Son los *DTOs* que se utilizan para ser enviados desde el cliente al servidor. Incluyen datos relacionados con las peticiones *Hypertext Transfer Protocol (HTTP)*. Se encuentran en el subdirectorio *request* y heredan de la clase *RequestDto*, que proporciona algunos campos comunes para ser rellenados en los *DTOs* hijos. Se construyen haciendo uso del método *makeRequest()*.
- **DTOs de respuesta.** Son *DTOs* utilizados para ser recibidos en el cliente desde el servidor. Se encuentran en el subdirectorio *response* del proyecto y heredan de la clase *ResponseDto*. Se construyen haciendo uso del método *makeResponse*.

5. ARQUITECTURA

El servicio web se divide en dos partes, la parte del cliente, *frontend* que utiliza *Angular* y la parte del servidor, *backend* que hace uso de *Node.js* y *NestJS*. Ambas partes están guardadas en el mono-repo. Esto tiene algunas ventajas y desventajas.

En cuanto a las ventajas, el desarrollo es más sencillo de realizar, ya se dispone de un único repositorio para todo el proyecto y el código sin dependencias se puede reutilizar entre cliente y servidor.

Por otro lado, no encontramos demasiadas desventajas, pero hay que tener en cuenta que algunas dependencias no pueden actualizarse porque ambas partes dependen de ellas. Por ejemplo, *TypeScript* no se puede actualizar ya que Angular 8 sólo soporta la versión de *TypeScript* <3.6.

A modo de recordatorio, se sabe que la parte del *frontend* es la parte de un sitio web que interactúa con los usuarios, por eso se dice que está del lado del cliente. Y *backend* es la parte que se conecta con la base de datos y atiende las peticiones recibidas del *frontend*. Por eso, el *backend* se ejecuta en el lado del servidor.

Para hacer más sencilla su explicación y visualización se divide el *frontend* y el *backend* en dos diagramas, mostrados a continuación. Al final de esta sección también se incluirá una vista general de la organización del proyecto.

Organización del frontend

En primer lugar, se parte de un directorio inicial llamado *src*, que contendrá toda la lógica relacionada con el *frontend*. De este directorio *src* salen dos más, que son el *app/* y el *assets/*. El *assets/* se encargará de contener tanto los recursos del proyecto como las imágenes e iconos utilizados en el servicio, como hojas de estilo CSS. Por otro lado, del directorio *app/* sale el directorio *services/* que se encarga de comunicar la parte del *frontend* con el *backend*, es decir, hace de anclaje entre la parte del cliente y del servidor. Y finalmente, la carpeta *ui/* es la que contiene todos los componentes web de *Angular* que serán necesarios para implementar las diferentes secciones que contiene el servicio.

En la Figura 5.2 se puede observar la estructura desde el lado del cliente.

Organización del backend

El *backend* se estructura partiendo de un directorio llamado *server* que contendrá toda la lógica referente a la parte del servidor. En *common/* se encuentra el código que puede ser utilizado en el *frontend*. En el directorio de *database/* se encuentra todo lo relacionado con la base de datos, es decir, las entidades, los parámetros de conexión y las migraciones.

Con el directorio *events/* se tienen definidos los distintos eventos manejados por el servicio. Los directorios *filters/* y *guards/* son utilizados por *NestJS*. También, en la carpeta *modules/* se encontrarán los módulos que definen la API del servicio. En *app.module.ts* se

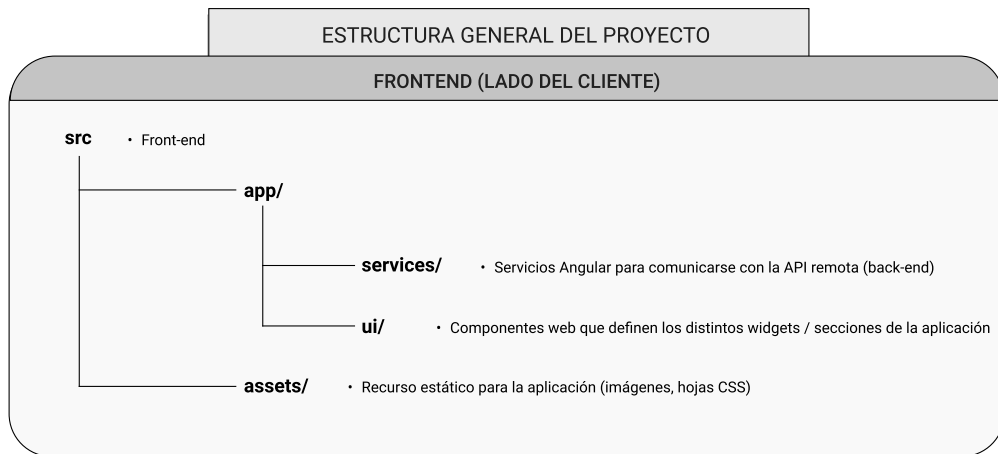


Figura 5.2: Estructura del proyecto desde el lado del cliente

incluyen todos los módulos definidos para *NestJS*. Y finalmente, el archivo de *TypeScript* llamado *main.ts* es el encargado de ejecutar y definir la configuración que ha sido establecida para este servicio. *Main.ts* se encarga de ejecutar las migraciones y asegurarse de que la zona horaria sea UTC.

En la Figura 5.3 se puede observar la estructura desde el lado del servidor.

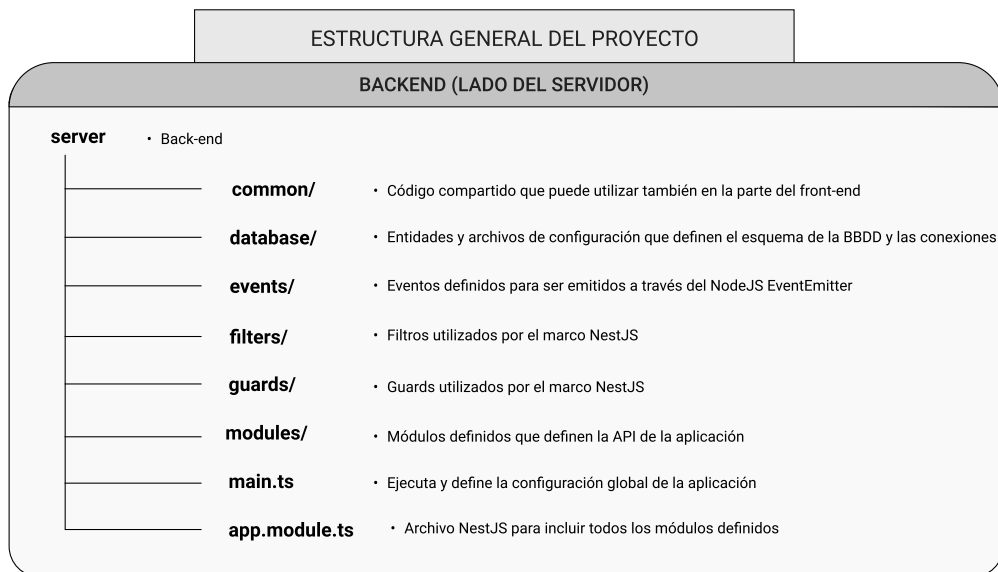


Figura 5.3: Estructura del proyecto desde el punto de vista del backend

5.1.3 Estructura general del proyecto

La Figura 5.4 muestra la estructura u organización del proyecto siguiendo una jerarquía de directorios. Como se puede observar, se han incluido otros directorios como el de la documentación técnica necesaria para desarrollar el proyecto. También se ha incluido el directorio de *e2e/* que contendrá los diferentes tests ejecutados al finalizar la parte de desarrollo para probar que el servicio funciona correctamente. Se han incluido en la carpeta de *scripts* diversos *scripts* necesarios para agilizar algunos procesos recurrentes como, por ejemplo, el

5. ARQUITECTURA

empaquetado y despliegue de la aplicación en la nube de *Amazon*, la ejecución de los tests y el proceso de anonimización de la base de datos. También se han incluido algunos paquetes y *json* necesarios para el despliegue del servicio.

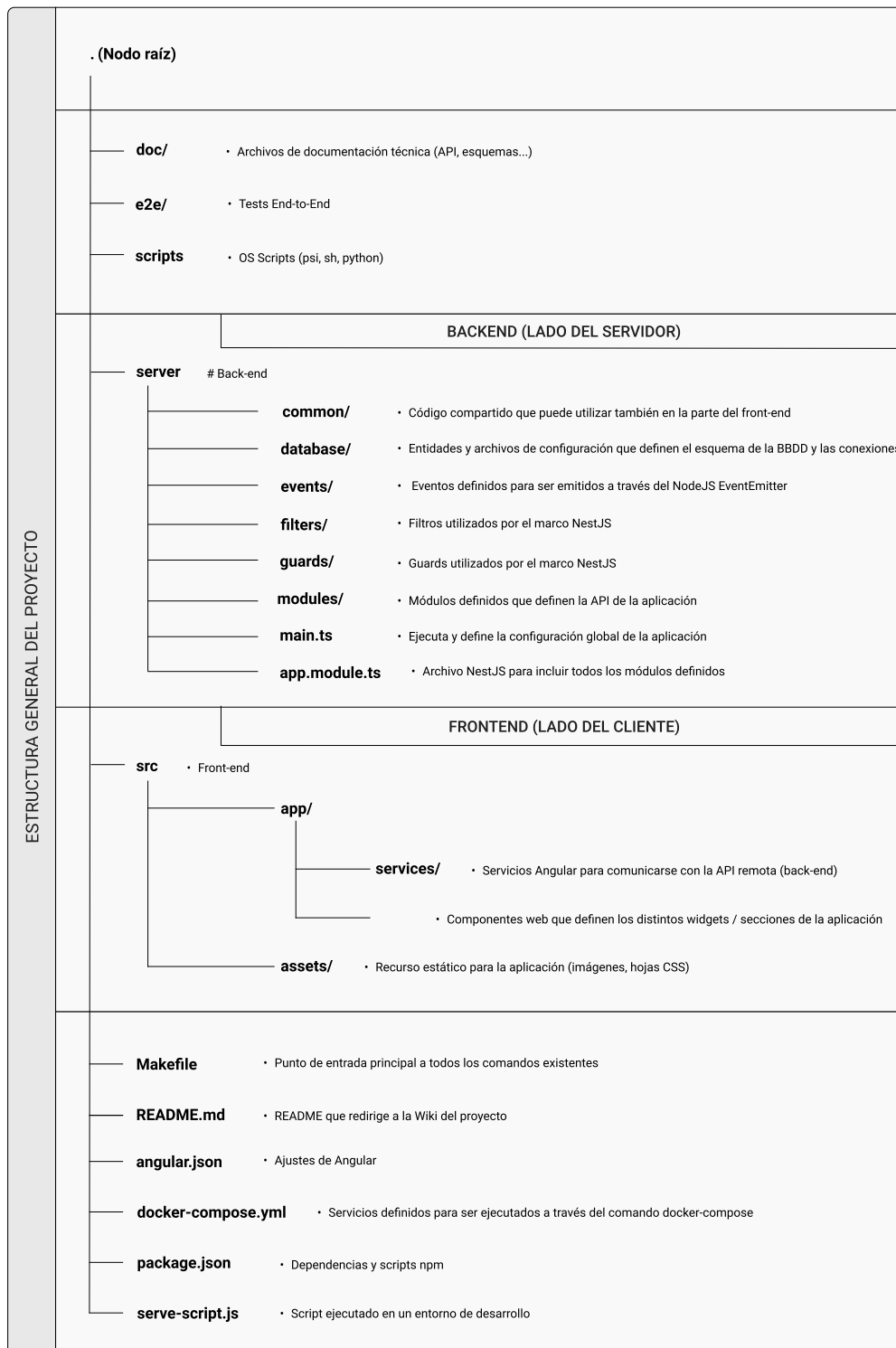


Figura 5.4: Estructura general del proyecto

5.2 Diseño de la base de datos

Se procedió a realizar el diseño global de la base de datos e integración en *My Client Genius (MCG)*. Para ello, se crearon las entidades específicas. Estas entidades permiten a los desarrolladores trabajar con datos en forma de objetos y propiedades específicas del dominio, sin tener que preocuparse de las tablas y columnas de bases de datos subyacentes en las que se almacenan estos datos.

Por tanto, gracias a las entidades se puede trabajar en un nivel más alto de abstracción y se pueden crear y mantener aplicaciones orientadas a datos con menor código que en las aplicaciones tradicionales. Estas entidades son tablas con un nivel de abstracción mayor que permiten definir tipos de datos y relaciones de forma más sencilla que con el propio Lenguaje de definición de datos (DDL) del Sistema Gestor de Base de Datos (SGBD). Las entidades utilizadas para este proyecto han sido, en concreto, las de *TypeORM*.

5.2.1 Elección de la Base de Datos

- **Base de datos relacional.** Es un tipo de base de datos que almacena y proporciona acceso a puntos de datos relacionados entre sí. Las bases de datos relacionales se basan en el modelo relacional. En una base de datos relacional cada fila en una tabla es un registro con un ID único llamado clave. Las columnas de la tabla contienen los atributos de los datos y cada registro suele tener un valor para cada atributo, lo que simplifica la creación de relaciones entre los puntos de datos.

Las transacciones en bases de datos relacionales están definidas por cuatro propiedades: Atomicidad, Consistencia, Aislamiento y Durabilidad (ACID).

Al inicio de este proyecto se plantearon diferentes opciones para el almacenamiento de datos del nuevo servicio, entre ellos estaban, *Oracle*, *Microsoft SQL Server* o *PostgreSQL*. Después de analizar los tipos de base de datos existentes y los recursos para este proyecto, se ha optado por seguir y diseñar una base de datos relacional [Her13], en concreto, *PostgreSQL*.

PostgreSQL es un potente sistema de base de datos relacional orientado a objetos de código abierto y clase empresarial que hace hincapié en la extensibilidad y la conformidad con los estándares. *PostgreSQL* presenta numerosas características sofisticadas y ejecuta procedimientos almacenados en numerosos lenguajes de programación. Este es el motivo fundamental para escoger a *PostgreSQL* como el SGBD del servicio web *Walking for Health*.

La elección de *PostgreSQL* como motor de bases de datos relacionales también ha sido tomada debido a las innumerables ventajas que ofrece su implantación.

Algunas de estas ventajas [Mar14] son las siguientes:

- **Alta escalabilidad.** Permite configurarlo en cada equipo adaptándose al hardware. Lo que permite ajustarse al número de CPU y a la memoria disponible en el sistema de

una manera óptima. Como resultado, se logra una mayor cantidad de peticiones a la base de datos en menos tiempo.

- **Estabilidad y confiabilidad.** Durante más de 20 años de desarrollo activo nunca ha presentado caídas de la base de datos.
- **Instalación ilimitada y gratuita.** Permite instalarlo en todos los equipos independientemente de la plataforma y la arquitectura que se use. *PostgreSQL* está disponible para diversos sistemas operativos, lo que hace que sea un sistema multiplataforma más rentable con instalaciones a gran escala.
- **Estándar SQL.** Implementa casi todas las funcionalidades del estándar¹ ISO/IEC 9075:2011, así pues, resulta sencillo realizar consultas e incluir *scripts* de otros Motores de Bases de Datos.
- **Extensibilidad.** Ofrece una gran variedad de extensiones, las cuales pueden ser incluso lenguajes de programación.
- **Robustez y potencia.** *PostgreSQL* cumple con la característica de ACID, lo cual permite que las transacciones no interfieran unas con otras. Con ello, se garantiza la información en las bases de datos y que los datos perduren en el sistema.

Por otro lado, las bases de datos relacionales se adaptan mejor que las no relacionales a las necesidades del sistema web de *Walking for Health*. Algunas de las ventajas de utilizar bases de datos relacionales son la independencia que existe entre los datos del sistema web, atomicidad y una mejor coherencia de datos evitando inconsistencias.

Posteriormente, se procedió a la creación de las operaciones básicas (*CRUD*) relacionadas con los grupos locales y se crearon los *endpoints* correspondientes al servicio de *Walking for Health*, que son las *URLs* de un API o un *backend* que responden a una petición.

También se utilizó la herramienta de *Postman* mediante la cual se pueden construir peticiones que utilicen las APIs implementadas y poder así probar la parte del *backend* implementada.

La base de datos es la que se encargará de recoger y almacenar la información necesaria, incluyendo las operaciones *Create, read, update, delete (CRUD)*. En primer lugar, se identifican los *endpoints* y los recursos que serían necesarios, y posteriormente, se decidirá la forma en la que se almacenan los datos del sistema web a desarrollar.

5.2.2 Anonimización de la Base de Datos

La anonimización de datos es un tipo de saneamiento de la información cuyo objetivo es la protección de la privacidad. Es el proceso de eliminar la información personal identi-

¹<https://www.iso.org/standard/53685.html>

ficable de los conjuntos de datos de modo que, las personas a las que describen los datos permanezcan en el anonimato.

En el contexto de este proyecto se utilizan datos médicos. Por este motivo, la anonimización es necesaria para proteger la información del caminante como, por ejemplo, el nombre, la dirección, el teléfono, la fecha de nacimiento o cualquier otra información relevante que comprometa la identidad de la persona en cuestión.

5.2.3 Entidades

Se define una entidad o instancia como una unidad de la base de datos que contiene información. Esta unidad es una representación dentro de la base de datos de un objeto y como tal, posee ciertos atributos que la diferencian del resto de entidades². En este servicio web, las entidades están relacionadas entre sí por distintos parámetros o atributos. Estas entidades pueden ser fuertes, que son aquellas que existen por sí mismas o débiles, su existencia depende de que exista otra entidad.

El modelo Entidad-Relación es uno de los más importantes a la hora de diseñar e implementar una base de datos con éxito. Mediante este modelo se relacionan una o varias entidades por sus atributos que pueden ser comunes o no a varias de ellas.

5.2.4 Diagrama de base de datos Walking for Health

A continuación, se hace uso de un diagrama de Lenguaje Unificado de Modelado (UML) usando la herramienta *Visual Paradigm*, comentada en el capítulo 4. Con este diagrama se pueden ver las relaciones que existen entre las distintas tablas creadas con la base de datos relacional, *PostgreSQL*.

El diagrama de la base de datos de *Walking for Health* se puede consultar en la Figura C.1.

5.3 Roles y permisos

Los roles y permisos permiten proteger partes de la aplicación tanto a nivel de cliente como de servidor. La protección debe hacerse a nivel de servidor, pero también a nivel de cliente para que al usuario final se le muestren mensajes de error convenientes y las diferentes páginas/elementos dependiendo de los roles que tengan.

A continuación, se muestra cómo utilizar la API desarrollada para proteger las diferentes partes de la aplicación a nivel de cliente y servidor.

5.3.1 Permisos a nivel de servidor

Para poder proteger las distintas partes del sistema a nivel de servidor se tienen que proteger los *endpoints* del *backend*. Los *endpoints* se encuentran en las clases *controlador* del

²<https://sistemas.com/entidades.php>

proyecto.

Un ejemplo de ello es el que se muestra en el Listado 5.1 en la que se puede ver cómo se protegen los endpoints mediante el decorador: ‘@Permissions(IPermission[], IPermissionsSettings)’.

Este ejemplo que se muestra a continuación permite que cuando un usuario intenta acceder a ese *endpoint*, es necesario que tenga los permisos **EPermission.ManageHWWalks** con la acción **EPermissionAction.Create**, en el caso de que la petición al servidor sea de tipo creacional, comúnmente definidas en nuestro caso mediante el método HTTP *POST*. Y **EPermission.ManageHWWalks** con la acción de **EPermissionAction.Edit** cuando la petición sea de tipo *PUT* o *PATCH*.

```
// Create
@Permissions([ { id: EPermission.ManageHWWalks, actions: [EPermissionAction.Create] } ])
@UsePipes(CustomValidationPipe)
@Post()
async create(@Body() data: CreateHWWalkDto): Promise<HWWalkDtoData> {
  const walk = await this.walkService.createHWWalk(data);
  return plainToClass(HWWalkDtoData, walk);
}

// Update
@Permissions([ { id: EPermission.ManageHWWalks, actions: [EPermissionAction.Edit] } ])
@UsePipes(CustomValidationPipe)
@Put('/:walkId')
async update(@Param('walkId') walkId: string, @Body() data: CreateHWWalkDto): Promise<
  HWWalkDtoData> {
  const updatedWalk = await this.walkService.updateHWWalk(walkId, data);
  return plainToClass(HWWalkDtoData, updatedWalk);
}
}
```

Listado 5.1: Permisos a nivel de servidor

Los usuarios del servicio pueden tener un conjunto de permisos. Estos permisos se conocen como *roles*. Los cuales son plantillas con permisos asignados previamente. De tal manera que la lista de permisos se extrae de los roles requeridos y se asienta junto con los permisos asignados al usuario. Es decir, los permisos que se asignan directamente a los usuarios anulan los permisos existentes en sus roles correspondientes. A continuación, todos los permisos asignados a un usuario y sus roles se comparan con los del decorador para decidir la concesión de acceso. Los permisos asignados se comprueban en un archivo llamado *permissions.guard.ts*

5.3.2 Permisos a nivel de cliente

Desde el lado del cliente se pueden proteger las distintas partes del *frontend* de distintas formas, las cuales se muestran a continuación.

Se puede proteger el acceso mediante el método: **hasPermissions()** como puede verse en el código del Listado 5.2. O bien optar por proteger el acceso a las distintas rutas del sistema en el archivo *app-routing.module.ts*. Estos permisos se comprueban en el archivo del lado del cliente llamado *auth.guard.ts*.

```
permissionsReadEpisode = [{ id: EPermission.ManageHWEpisodes,
actions: [EPermissionAction.Read] }];
permissionsCompleteEpisode = [{ id: EPermission.ManageHWEpisodes,
actions: [EPermissionAction.Edit] }];
permissionsCreateSession = [{ id: EPermission.ManageHWSessions,
actions: [EPermissionAction.Create] }];
permissionsClientStatus = [{ id: EPermission.ManageHWEpisodes,
actions: [EPermissionAction.Edit] }];

hasPermission(permissions: IPermission[]): boolean {
  const user = this.authService.userSubject.value;
  return hasPermissions(user, permissions);
}
```

Listado 5.2: Proteger el acceso mediante un método específico

5.3.3 Directivas personalizadas para proteger el frontend

Para proteger algunas características del *frontend* se utilizaron dos directivas personalizadas ya implementadas en el sistema.

La primera de ellas fue la *AppPermissionsDisable*. Esta directiva deshabilitará todos los componentes HTML dentro de ese *div* que tengan el atributo *disable* (inputs, selectores, botones, etc.) si el usuario no cumple con la lista de permisos especificada. Esto se puede observar en el Listado 5.3

```
<div class="mb-3 card main-card" [appIfPermissionsEnable]="permissionsBookWalk">
```

Listado 5.3: Directiva personalizada Book Walk

Y la segunda directiva personalizada que se utilizó fue la directiva '*PermissionCheckDirective*'.

Esta directiva funcionará como una directiva condicional **ngIf* de Angular, que ocultará el elemento que utiliza esa directiva si el usuario no tiene los permisos especificados, como puede verse en 5.4.

```
<app-program-tab
  *appIfPermissionsShow="adminPermissions"
  [icon]="''./assets/images/branding/my-admin-genius-icon.svg'"
  [title]="''<strong>MY</strong><br>ADMIN<br>GENIUS'"
  [path]="''/' + adminPath"
  [bgColor]="''var(--admin-color)'"
  [customButtonStyle]="{ padding: 0 }"
></app-program-tab>

adminPermissions = [{ id: EPermission.ManageAdminSettings, actions: [EPermissionAction.Read] }];
```

Listado 5.4: Uso de la directiva PermissionsCheckDirective

5.4 Filtrado de datos

El filtrado de datos permite recuperar fácilmente uno o varios elementos de un conjunto de información siguiendo unos requisitos especificados. Por ejemplo, el nombre o el teléfono.

A continuación, en el Listado 5.5 se muestra un ejemplo de cómo se utilizan estos filtros en el sistema web.

```
import { LookUpDataService } from '@client-services/look-up-data.service';
import { ELookUpDataResource } from '@common/enums/look-up-data-resource.enum';
import { EFilterType, IFilterCriteria } from 'src/app/filters/filter-types';

const filterVolunteer: IFilterCriteria = {
  criteria: [
    {
      group: 'Volunteer',
      filters: [
        {
          name: 'Name',
          filter: 'name',
          type: EFilterType.TextType,
        },
        {
          name: 'Telephone',
          filter: 'telephone',
          type: EFilterType.TextType,
        },
      ],
    },
  ],
};
```

Listado 5.5: Uso del filtrado de datos en voluntarios

5.5 Validación de formularios

Es importante agregar algún tipo de validación a los formularios que se utilicen en una aplicación o sistema. De esta manera, se asegura que la persona encargada de rellenarlos no deje campos vacíos o ingrese de manera incorrecta el formato de los datos. Es por ello que la validación es fundamental en la entrada de datos en cualquier sistema. En los formularios utilizados para desarrollar las secciones del servicio web se ha hecho uso de la validación. A continuación, se explican los tipos de validaciones que existen en *Angular*.

- **Validación asíncrona.** Se debe realizar una solicitud a la base de datos para poder validar los datos. Un ejemplo de ello es validar el nombre de usuario. Para ello, se deberá comprobar en la base de datos que el nombre de usuario esté disponible.
- **Validación sincrónica.** Es aquella en la que no se necesita consultar la base de datos para comprobar datos. Un ejemplo es poder verificar que el *e-mail* tiene el formato correcto.

La validación elegida para los formularios del servicio web ha sido la validación sincrónica. Esto ha sido así, ya que en los formularios se han comprobado datos en los que no era necesario consultar la base de datos. A lo largo de este capítulo se pondrán ejemplos de ello.

5.5.1 Ejemplos de validación en formularios

En primer lugar, es necesario establecer las reglas de validación que se requieren en el formulario. Para ello se pueden utilizar reglas predefinidas como funciones en el objeto *Validators* del *framework* o utilizar sobrecargas que resuelven casos de una única validación.

En segundo lugar, hay que aplicar las reglas establecidas previamente y avisar al usuario en caso de que esas reglas no se cumplan. Como la validación sincrónica hace uso de controles reactivos, estos se gestionan por medio de máquinas de estados que comprueban en todo momento cada control y el formulario en sí.

La máquina de estados de validación contempla los siguientes estados mutuamente excluyentes:

- **VALID.** El control ha pasado todas las reglas.
- **INVALID.** El control ha fallado al menos en una regla.
- **PENDING.** El control está en medio de un proceso de validación.
- **DISABLED.** El control está desactivado y exento de validación.

Cuando un control incumple con alguna regla de validación, estas se reflejan en su propiedad *errors* que será un objeto con una propiedad por cada regla que no se cumpla junto con un mensaje informativo guardado en dicha propiedad.

Un ejemplo de uso de la validación se puede ver en el formulario de grupos, donde se

5. ARQUITECTURA

añade la validación necesaria para que el número máximo de miembros por grupo sea 15. En una zona rural sólo se admiten 10 personas, mientras que en una zona pública 15. Si el usuario introdujese más de 15 miembros, el sistema avisaría mediante un mensaje de alerta, advirtiéndole que no se puede superar el número de miembros permitidos. Esto puede verse en el Listado 5.6.

```
if (this.maxMembersForm.value > 15) {  
  this.toastr.error('', 'The maximum number of members per group allowed in a group is 15');  
  return;  
}
```

Listado 5.6: Validación del máximo número de miembros por grupo de caminata

En este otro código se puede ver cómo se controla que no se pueda asignar una caminata rural con más de 10 personas a un grupo. Por ello, a la hora de reservar una caminata, no podrá reservarse si no cumple con dichas restricciones, como puede observarse en 5.7.

```
if (this.groups[this.groupForm.value].maxNumberOfMembers > 10 && this.walks[this.walkForm.value].  
  zone.id === ELUTHWZone.Rural) {  
  this.toastr.error('', 'It is not possible to assign a group with more than 10 members to a rural  
    walk.');
```

Listado 5.7: Validación de la zona rural y pública en la sección de reservar una caminata

5.6 Módulos

A continuación, se explican los módulos de los que se compone la arquitectura del proyecto a desarrollar. En cada módulo se explicarán además los problemas encontrados, solución a la problemática específica, así como las ventajas de haber elegido la solución planteada y no otra en cada funcionalidad implementada. En primer lugar, para poder entender el flujo desde que se reserva una caminata hasta que se registra en el sistema tras su realización es necesario explicar el proceso de pre-caminata y post-caminata.

El problema inicial al que enfrentarse es el de dividir las caminatas en dos procesos. Esto es necesario ya que se necesita un proceso que gestione los pasos previos que hay que seguir para poder realizar un paseo saludable. Y posteriormente, un proceso que permita guardar y actualizar los datos de la caminata que tenga lugar. A continuación, se describe cada uno de los procesos.

5.6.1 Pre-caminata

La pre-caminata consiste en todo el proceso de desarrollo previo a realizar una caminata saludable. Antes de que un caminante pueda realizar un paseo es necesario que se realicen

una serie de procedimientos.

En primer lugar, deben crearse las *venues* o lugares de encuentro de las caminatas. Posteriormente, el administrador deberá crear o seleccionar un grupo de caminata que va a asistir a esa ruta en específico.

A continuación, se podrán crear o seleccionar las caminatas que ya existen en el sistema. O bien, se podrán crear paseos nuevos asignándoles el grupo al que pertenecen en el formulario correspondiente. Después de crear las rutas, el administrador creará la reserva de esa ruta mediante el formulario apropiado para ello. Estas rutas programadas podrán visualizarse en el calendario del sistema. De esta manera, el administrador podrá visualizar más fácilmente todas las caminatas que haya programadas.

En este proceso también se asigna el caminante al grupo de caminatas. Dentro de este formulario se podrá ver el número de personas que están en el grupo de caminata específico.

El proceso de pre-caminata explicado se puede visualizar en la Figura 5.5.

5.6.2 Post-caminata

El proceso de post-caminata se lleva a cabo una vez que la caminata saludable ya ha tenido lugar. Para ello, es necesario registrar los datos necesarios referentes al paseo realizado.

Tras realizar la caminata el administrador registrará datos, como, por ejemplo, la fecha, hora, ruta realizada, si ha asistido o no el caminante. Estos paseos se pueden consultar en la sección de resumen de caminatas. En la Figura 5.6 puede visualizarse el diagrama correspondiente.

5.6.3 Walkers

Esta sección consta de 4 subsecciones: *my walkers*, *recent walkers*, *search walkers* y *waiting list*.

Después de sopesar varias opciones a la hora de implementar esta sección se decidió utilizar listas para poder visualizar los diferentes caminantes del sistema.

De esta manera, se pueden visualizar los caminantes según sea el interés del administrador. Por ejemplo, consultar los caminantes que ha creado en **My walkers**, la lista de espera (**Waiting list**), los caminantes recientes en **Recent Clients** o bien realizar una búsqueda global en **Search walkers**. Esto tiene la ventaja de poder estructurar mejor las partes en las que se dividen los caminantes y poder seleccionar la sección que se desee en cada momento.

Algunos de los problemas encontrados a la hora de realizar esta sección han estado relacionados con la forma en la que se muestran los caminantes. Es decir, en la subsección de la *waiting list* se tuvieron algunas dificultades ya que no se llegaba a conseguir que los caminantes que no estuvieran asignados a un grupo figuraran en esta lista de espera. Finalmente, se logró resolver con éxito.

Figura 5.5: Diagrama de secuencia pre-caminata

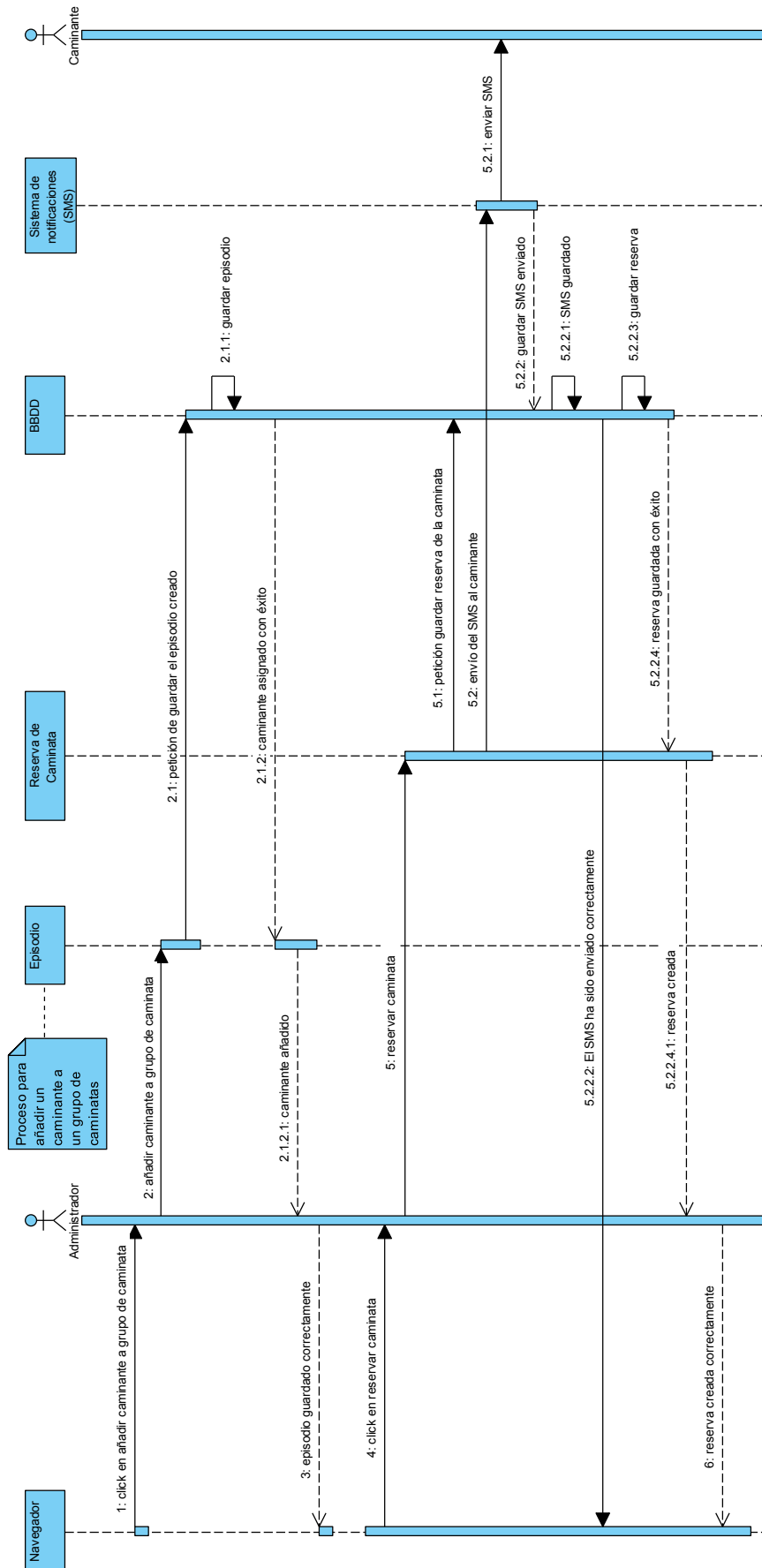
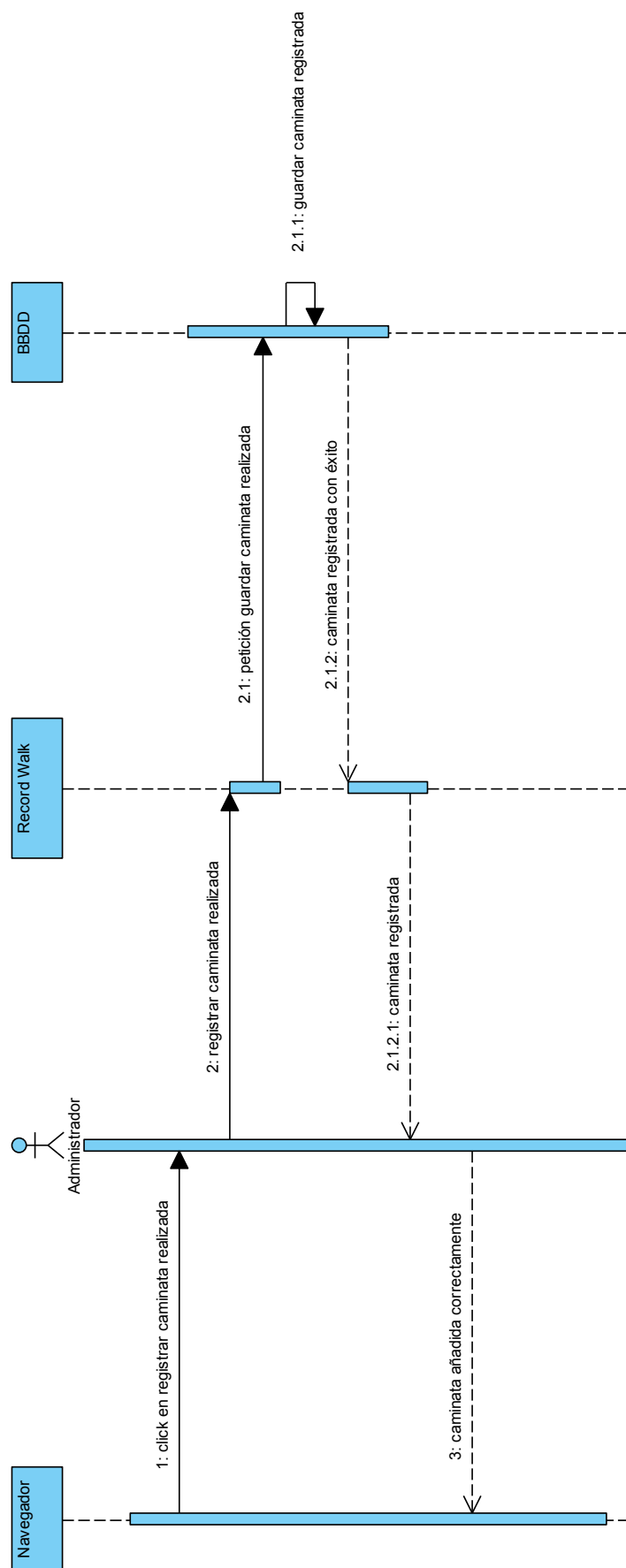


Figura 5.6: Diagrama de secuencia post-caminata



5.6.4 Groups

La problemática a la se debía hacer frente era la de crear grupos de caminatas para posteriormente poder asignar caminantes a estos grupos. La solución que se plantea para ello es poder crear un grupo mediante el uso de un formulario que recoja los datos necesarios. En la creación de estos grupos se hace también uso del filtrado de datos y la validación de formularios.

Las ventajas que tiene utilizar un formulario para crear grupos es que ahorra mucho tiempo al administrador a la hora de crear nuevos grupos ya que puede editarlos cuando desee de una manera muy sencilla y visual.

5.6.5 Volunteers

Para poder crear y listar voluntarios se hizo uso de soluciones similares a las comentadas en las secciones anteriores. Para crear un voluntario se podrá hacer mediante el uso de un formulario. Y mediante un listado se mostrarán todos los voluntarios creados en el sistema.

Uno de los problemas encontrados fue el de introducir el cuestionario de salud relacionado con los riesgos *COVID* dentro del formulario de voluntarios. Sin embargo, esto fue llevado a cabo con éxito como se explicará a continuación.

Evaluación de Riesgos COVID

Dentro del formulario de voluntarios era necesario introducir de algún modo el cuestionario que recogería la Evaluación de Riesgos *COVID*. Para ello, se tenían varias opciones, entre ellas la del uso de un modal, un nuevo formulario o implementarlo en el mismo formulario de voluntarios. Finalmente, la solución elegida fue la de un modal que se desplegará cuando el usuario pulse sobre un determinado botón en el formulario de creación de voluntarios.

Las ventajas que tiene utilizar este tipo de modales es que al posicionarse en una capa superpuesta con respecto a la página original, su aspecto visual y su tamaño no están limitados por las propiedades del *template*. Esto permite captar la atención del usuario mostrándole contenido que esté relacionado con la página actual como es este caso.

De esta forma, se rellena el cuestionario con los datos necesarios sin tener que implementar de nuevo otro formulario o implementarlo en el formulario principal de *add volunteer*. Ya que esto podría sobrecargar el contenido de la página web y en una pantalla demasiado pequeña el usuario tendría que bajar la barra de desplazamiento para rellenar todo el formulario.

5.6.6 Manage Walks

Esta sección se subdivide en otras 6 subsecciones más, las cuales son: creación y listado de lugares de encuentro, creación y listado de caminatas y, finalmente, reservar y listar estas reservas.

Crear y listar venues

La solución planteada en esta subsección era similar a otro servicio ya implementado en el sistema en producción. La creación y listado de lugares de encuentro ya se realizaba en el sistema, sin embargo, se han hecho algunos cambios mínimos que han permitido adaptarlas al servicio *Walking for Health*. Como se comentará en el capítulo de 6 se utilizará un formulario para crear las *venues* y una lista para poder mostrarlas en el sistema.

Crear y listar caminatas

Para dar solución a esta problemática se plantearon diversas opciones, entre las cuales se encontraban el uso de formularios o modales. Finalmente, se optó por utilizar un formulario para añadir caminatas nuevas y un listado para mostrarlas. Esta solución planteada es similar a la creación y listado de *venues*. Cabe destacar que se hizo uso de *Google Maps* para incluir un mapa sobre el que seleccionar el punto de partida del paseo.

La dificultad de esta solución fue la de incluir un mapa dentro de un formulario ya que se necesitaron realizar unos pasos previos para poder hacer uso de esta herramienta de *Google*.

Crear y listar reservas de caminatas

La reserva de caminatas ha sido difícil de implementar y llevar a cabo debido a la complejidad del proceso. Para poder reservar paseos se deben tener en cuenta muchas funcionalidades extra como, por ejemplo, su relación directa con el calendario para que queden programadas y sean visibles en el calendario del sistema.

Además, debido al espacio temporal que se debe manejar y el proceso de cancelación de ésta, convierte la gestión de reservar caminatas en una funcionalidad complicada.

5.6.7 Contact

En esta sección, se mostrarán los SMS enviados a los caminantes en forma de lista.

Notificaciones a través de SMS de la ruta programada

Antes de resolver esta problemática se barajaron varias opciones. Estas opciones fueron o bien hacerlo mediante el uso de SMS o mediante el envío de un *e-mail*. Finalmente se optó por seguir el sistema de notificaciones basado en SMS. Esta funcionalidad, como bien se explica en el capítulo de 6, permite avisar a los caminantes de la ruta programada.

Algunos de los problemas encontrados al llevar a cabo esta solución fue la de poder enviar un *SMS* desde el entorno de desarrollo. Sin embargo, esto fue probado con éxito en el entorno de pre-producción.

Cancelación de una caminata

Otro de los requisitos que se definieron al principio del desarrollo del proyecto era poder cancelar una caminata que ya ha sido programada. Bien porque al final no se lleve a cabo o porque se quiera programar otra en su lugar.

Para ello, el administrador podrá cancelarla mediante un botón que permitirá que esa ruta no aparezca en la subsección de la lista de rutas programadas del sistema. Y además, la caminata no aparecerá tampoco programada en el calendario.

Esta manera de implementarlo es más sencilla. El administrador solamente tiene que encargarse de pulsar un botón, mientras que la lógica de todo lo que implica cancelar esa caminata se hace de manera interna y ajena al mismo.

Al cancelar una caminata, ésta desaparecerá de la lista de reservas y del calendario del sistema, en el día y hora que estuviera programada. Además, se enviará un SMS a los caminantes que les afecte esa cancelación, notificándoles de que la ruta ha sido cancelada.

5.6.8 Otras funcionalidades

También se han realizado otras funcionalidades que apoyan el proceso de pre y post caminata, como por ejemplo la asignación de caminantes a grupos de caminatas, permitiendo así asignar las personas a sus grupos correspondientes. El calendario, donde el administrador puede visualizar las reservas programadas en una fecha y hora concretas. Las sesiones de formación de los voluntarios, que pueden ser creadas mediante un evento de calendario. Y el registro de caminata donde se ha hecho uso de los modales de *Angular*, los cuales ayudan a agregar contenido web adicional a una página cuando la página ya está cargada de elementos HTML. El uso de estos modales podrá visualizarse en los resultados obtenidos en el capítulo 6.

5.7 Testing

La arquitectura permite dar soporte al testing ya que resulta mucho más sencillo realizar tests sobre una arquitectura bien definida.

En el entorno donde se ha desarrollado el sistema *Walking for Health* existen tres tipos de tests: unitarios, de integración y end-to-end.

- **Test unitarios.** Permiten probar de manera individual funciones o métodos de las clases, componentes o módulos utilizados en el sistema. Estos test se ejecutan rápidamente por un servidor de integración continua.
- **Test de integración.** Estos tests permiten comprobar que los módulos utilizados en el sistema funcionan correctamente cuando trabajan en conjunto. Suelen ser más complejos que los tests unitarios, ya que es necesario que una mayor parte del sistema se configure y se encuentre en funcionamiento para poder ejecutarlos.

- **Test end-to-end.** Permiten verificar que el comportamiento que sigue el usuario sea el esperado. Estas pruebas son útiles pero difíciles de mantener si no se encuentran automatizadas.

Por tanto, suele ser más recomendable trabajar con pruebas unitarias o pruebas de integración para poder detectar más rápidamente aquellos cambios inesperados que afectan negativamente al sistema.

A lo largo del desarrollo de este proyecto se han diseñado una serie de pruebas unitarias o *unit testing* utilizando el *testing framework* de JS llamado *Jest*. Éste es un procedimiento de los que se llevan a cabo dentro de una metodología ágil de trabajo, como es el caso del servicio web desarrollado para este TFG. Así, se comprueba que el funcionamiento sea el esperado en ciertas partes del sistema.

Algunas de las ventajas de utilizar tests unitarios para probar el funcionamiento del sistema se exponen a continuación.

- Permiten aumentar la legibilidad del código, lo que facilita realizar cambios en el sistema mucho más rápido.
- Mejoran la calidad del código permitiendo que éste sea más ordenado.
- Permiten probar partes del proyecto sin tener que esperar a que otras ya estén finalizadas.

A continuación, se muestra un ejemplo de *unit testing* como puede verse en el Listado 5.8.

Este test realizado en la sección de *Book walk* permite comprobar que la concatenación de la fecha y hora de inicio y fin son correctas cuando se reserva un paseo saludable.

Se realiza la comprobación utilizando el método *getFormatDate* importado desde *book-walk-data-functions*, que es una clase que permite recoger las funciones más relevantes.

```
import { getFormatDate } from './book-walk-data-functions';

describe('Concat date and time testing', () => {
  it('correct concatenation of START date and time', () => {
    expect(getFormatDate('2021-06-15', '11:17')).toEqual(new Date(2021, 5, 15, 11, 17, 0, 0));
  });

  it('correct concatenation of END date and time', () => {
    expect(getFormatDate('2021-07-16', '12:17')).toEqual(new Date(2021, 6, 16, 12, 17, 0, 0));
  });
});
```

Listado 5.8: Concatenación de fecha y hora en una reserva de caminata

Capítulo 6

Resultados

La propiedad intelectual de este proyecto se enmarca en el contexto de la empresa, aún así se recoge en el repositorio la parte más relevante del código utilizado para *Walking for Health*. Además, se pueden ver las imágenes originales de los resultados obtenidos. El enlace del repositorio es el siguiente: https://pruebasaluuclm-my.sharepoint.com/:f:/g/personal/noeliamaria_granados_alu_uclm_es/EqIr2M1vIMZ0oBGXM8RUKpkBsfmIKSkyEJHUR5rnnPLXA?e=eDLzwY.

El proyecto se ha organizado en 5 *sprints* con una duración de 4 semanas cada uno, siguiendo una metodología *Scrum*. Al finalizar cada *sprint*, se ha organizado una reunión para discutir el desarrollo hasta el momento del *sprint*, así como las dificultades encontradas durante el mismo. Decidiendo también el/los objetivo/s a conseguir en el siguiente *sprint*. Al finalizar la reunión se crearon una serie de historias de usuario que servirían para fijar los requisitos y establecer los objetivos necesarios durante el desarrollo del *sprint*.

Como ya se ha comentado, en el capítulo de 5 se han explicado los principales problemas que existían a la hora de abordar el diseño de la solución para este proyecto, los problemas que han surgido durante el desarrollo del mismo y las ventajas que ha tenido elegir esa solución y no otras. De esta manera, en este capítulo se expondrán los resultados conseguidos del sistema web desarrollado para este TFG.

Para hacer más sencilla su explicación, los resultados conseguidos se dividen e incluyen en los *sprints* realizados a lo largo del proyecto.

6.1 Sprint 0

En este *sprint* se realizó una reunión para determinar los objetivos de desarrollo generales y específicos que acabaron implementándose en el nuevo servicio. Esto dio como resultado una serie de historias de usuario. Las cuales son la unidad de trabajo más pequeña en un *framework* ágil, como es el caso de este proyecto.

En *Scrum*, las historias de los usuarios se añaden a los *sprints* y se van completando a lo largo de éste. Estas historias de usuario dan al equipo un contexto importante y asocian las tareas con el valor que aportan. Esto es fundamental ya que algunos de los beneficios que tienen las historias de usuario son la mejora de la colaboración con el usuario, impulsar

6. RESULTADOS

soluciones creativas y centrar la atención en el usuario final. Por tanto, la historia de usuario que corresponde con este *sprint* se puede visualizar en la tabla A.1.

En este *sprint* también se instaló el entorno de trabajo requerido para poder desarrollar el nuevo servicio web, así como el estudio de la *Wiki* de la empresa que contenía todos los documentos y elementos necesarios para abordar el nuevo proyecto. También se realizó un aprendizaje mediante cursos *on-line* de *Angular* y *Node.js*.

Finalmente, se hizo un estudio del documento de diseño para *Walking for Health* v0.1 proporcionado por la empresa de OneLife Suffolk y se seleccionaron las posibles funcionalidades que debería tener el servicio, así como un primer diseño de la base de datos utilizando *PostgreSQL*.

6.2 Sprint 1

El siguiente *sprint* se dedicó especialmente a desarrollar la sección de **Caminantes**, incluyendo la *waiting list* o lista de espera, la sección de **Grupos** y el proceso de añadir un determinado caminante a un grupo de caminatas.

La historia de usuario correspondiente con la sección de caminantes y la *waiting list* se puede visualizar en la tabla A.2.

6.2.1 Sección de caminantes o walkers

En esta sección se mostrarán los distintos caminantes que haya asignado el administrador del sistema (Figura B.1), los caminantes recientes (Figura B.2) y en la subsección *search*, (Figura B.3) se podrá buscar de manera genérica los clientes del resto de servicios incluyendo el de *Walking for Health*.

Caminantes creados por el usuario admin (My walkers)

Mediante una lista se puede visualizar los caminantes que ha creado el administrador del sistema. En este caso, si es el administrador el que ha creado nuevos caminantes, le aparecerán en esta sección. Gracias a esta lista es más manejable y se puede ver de un simple vistazo todos aquellos caminantes creados por el propio administrador.

Caminantes recientes (Recent walkers)

Se mostrarán aquellos caminantes que se hayan añadido recientemente al servicio. Además, se podrá filtrar por período de tiempo estos caminantes usando filtros como: *All time*, *Today*, *This Week*, *This Year*, *Custom*.

Desde esta sección, también se podrá registrar una caminata si este caminante ya la ha realizado, gracias al botón de *Record Walk*. Esto puede visualizarse en la Figura B.2.

Búsqueda de caminantes (Search)

Mediante una lista, se podrán buscar y filtrar todos los caminantes o usuarios que hayan sido registrados en el sistema. También se podrán registrar y guardar los datos de una caminata al igual que en la anterior subsección. Esta subsección puede verse en la Figura B.3.

Lista de espera (Waiting List)

En esta subsección se añadirán aquellos caminantes que no tengan asignado ningún grupo de caminatas o estén a la espera de que algún grupo quede libre para poder entrar en él. Gracias a esta lista, el administrador puede llevar un control de los caminantes que no están en ningún grupo y poder asignarles uno cuando se queden disponibles. Un ejemplo de cómo funcionan estas listas de espera se puede ver en la Figura B.4.

6.2.2 Crear y listar grupos de caminatas

Estos grupos son necesarios para poder asignar a los caminantes a los paseos saludables. Para ello, se diseñó un formulario en el que se pueden introducir los datos necesarios referente a los grupos. De esta manera, el administrador del sistema puede crear grupos y posteriormente listarlos y editarlos si lo necesitase. La historia de usuario que corresponde con esta sección implementada puede visualizarse en la tabla A.3.

La solución implementada para esta sección puede visualizarse en la Figura 6.1.

6.2.3 Asignar caminante a un grupo de caminatas

En este proceso relacionado con la pre-caminata, el administrador asigna el caminante al grupo de caminatas que considere más apropiado siempre que queden huecos disponibles.

De esta manera, los caminantes quedan registrados en un grupo de caminatas. También se podrá consultar el grupo y el número de caminantes que tiene cada grupo mediante el modal específico para ello. Esto puede visualizarse mejor en la Figura 6.2.

Además, se puede consultar la historia de usuario que corresponde a este proceso en la tabla A.4.

6.3 Sprint 2

Durante este *sprint* se diseñaron e implementaron las secciones de **Crear y listar lugares de encuentro (venues)**, las cuales son explicadas en el Anexo A. También se implementó la sección de **Volunteers**, con sus dos subsecciones correspondientes, las cuales se explicarán a continuación.

Y finalmente, en las dos últimas semanas de este *sprint* se implementó el proceso que permitiría crear y listar caminatas saludables (*add walk* y *list walks*).

Figura 6.1: Creación y listado de grupos de caminatas saludables

WALKING FOR HEALTH

Walkers

Groups

Add group

List groups

Volunteers

Manage Walks

Contact

Group Data

Name *

Group 2

Contact First Name *

Martina

Contact Last Name *

Carrasco

Contact Telephone *

42334234234

Financial Year *

2021

Minimum recommended age *

0

Maximum recommended age *

100

Save

Other Data

Maximum number of members *

10

Group State *

Current

Skill level *

Beginner

Disability type *

Hearing Impairment

Advisor *

Anna Smith

Volunteer of this group

Mario Castro

Mario Castro

WALKING FOR HEALTH

Walkers

Groups

Add group

List groups

Volunteers

Manage Walks

Contact

Group List

These are the groups

Filter Criteria:

Add criteria

Total records: 4

<<<

<

1

>

>>>

Name	Fin Yr	Active Clients	Max Clients	State	Advisor
Group 1	2021	3	15	Current	Adele Flower
Group 2	2021	0	10	Current	Anna Smith
Group 3	2021	0	10	Current	Alastair Stewart
Group Only Childs	2021	0	15	Current	Amanda Baldwin

Figura 6.2: Añadir caminante a grupo de caminata

New WALKING FOR HEALTH Walking group - Pepe Pérez

General Walking group Information

Registration Date *
21/06/2021

Group Information

Group
Group 2
Adelle Stopher

Advisor *
Aaron Robinson

Employee Status *
Employed

Occupation *
Full-time student

Medical Condition
Asthma

Pregnancy Data
☐ Is Pregnant
☐ Is Breast Feeding

Comments

Cancel

Save

Click on a group to select it

Filter Criteria:

Add criteria

Name	Responsible	Max members	Current members	Info
Group 2	Adelle Stopher	15	2	<div></div>
Group 1	Anna Smith	15	0	<div></div>
Group 3	Chris Armstrong	10	0	<div></div>

Close

6.3.1 Crear y listar lugares de encuentro o venues

Otros de los requisitos necesarios en el proceso de pre-caminata era poder crear lugares de encuentro de las caminatas. Estos lugares de encuentro son necesarios para poder asignarlos a los grupos de caminatas y que los caminantes sepan la **venue** asociada a la ruta antes de empezarla. Estas *venues* podrán listarse y ser editados por el administrador cuando lo necesite.

Gracias al formulario específico para ello se puede crear la *venue*, B.5 donde va a tener lugar la caminata.

Como bien se ha comentado anteriormente, se podrán listar los distintos lugares de encuentro creados en el proceso de pre-caminata. Para ello se ha hecho uno de una lista que muestra todas las *venues* del sistema. Esto puede verse en B.6 y en la historia de usuario correspondiente (Tabla A.7).

6.3.2 Crear y listar voluntarios

Para llevar a cabo la implementación de estas dos funcionalidades se diseñó la historia de usuario que puede visualizarse en el Anexo A, tabla A.5.

En esta sección se permite al administrador o coordinador del sistema crear nuevos voluntarios, que son caminantes que deciden hacerse voluntarios para convertirse en líderes posteriormente, mediante una serie de sesiones de formación. Además de poder listar y actualizar estos voluntarios cuando se desee. Para ello, se debe rellenar un formulario en el que se introducirán los datos necesarios para registrar correctamente a ese voluntario. Este formulario también incluye un cuestionario salud, el cual debe ser completado con la información que el administrador tiene de este voluntario. El cuestionario es la Evaluación de Riesgos *COVID*, la cual debe ser cumplimentada correctamente para poder realizar de forma segura las caminatas saludables.

- **Listar voluntarios.** Se pueden listar los voluntarios que han sido creados para el servicio web de *Walking for Health*. Estos voluntarios tendrán un campo específico en la interfaz para poder ser editados y/o actualizados.

El administrador puede consultar en cualquier momento la lista de voluntarios creados en el sistema. Para ello, se diseñó una lista en la que aparecen los distintos voluntarios. Estos voluntarios se pueden filtrar gracias a los filtros que contiene el *dropdown add filter*. En la Figura 6.3 se muestra la sección de crear y listar voluntarios en el sistema.

Figura 6.3: Crear y listar voluntarios

WALKING FOR HEALTH

Walkers

Groups

Volunteers

List volunteers

Add volunteer

Manage Walks

Contact

Name *

Willy Rodriguez

Email *

willy@gmail.com

Telephone *

32423423423

Mobile

4234141432

Location *

West

Related advisor of this volunteer *

Amy Watling-Cash

Vulnerable person *

Yes

Living with a vulnerable person *

Yes

Covid Risk Questionnaire

Take Test

Comments

Enter a comment

Save

WALKING FOR HEALTH

Walkers

Groups

Volunteers

List volunteers

Add volunteer

Manage Walks

Contact

Volunteer List

These are the volunteers for HW service

Filter Criteria:

Add criteria

Total records

3

Name

Phone/Mobile

Email

Location

Actions

Carlos Jimenez

45352321241 - 45353333333

carlos@gmail.com

West

Catalina Carrasco

67767756654 - 32535233333

catalina1998@gmail.com

West

Mario Castro

09999999999 - 43432123213

mario@gmail.com

East

Marta Perez

00000000000 - 55757474747

marta@gmail.com

Waveney

Noelia Maria

24143242342 - 11111111111

volunteernoc@gmail.com

East

Cuestionario de salud: Evaluación de Riesgos COVID

Otro requisito definido al principio del desarrollo de este proyecto fue el de tener un cuestionario de salud que reuniese datos referentes al *COVID*. Para ello, se diseñó un formulario dentro de la sección de *Add volunteer* para que cuando el administrador rellene los datos del voluntario, pueda cumplimentar la Evaluación de Riesgos *COVID*. Con este formulario lo que se pretende es guardar datos importantes sobre el *COVID* para que los caminantes puedan realizar de manera segura las caminatas saludables.

Cabe destacar que este formulario solamente será visible si previamente se responde afirmativamente a alguna de preguntas sobre si el caminante es vulnerable o vive con una persona vulnerable. De ser así, el cuestionario será visible para que el administrador del sistema pueda cumplimentarlo. Este cuestionario se puede visualizar en la Figura 6.4. Su historia de usuario correspondiente se puede consultar la tabla A.6.

Figura 6.4: Cuestionario de salud de la Evaluación de Riesgos COVID

Covid Risk Questionnaire for Volunteers

Please tick the box that best describes your experience of walking *

	Rarely	Often	All the time	Some of the time
I've been feeling optimistic about the health walks	★ ★ ★ ★ ★	★ ★ ★ ★ ★	★ ★ ★ ★ ★	★ ★ ★ ★ ★
I've been feeling relaxed	★ ★ ★ ★ ★	★ ★ ★ ★ ★	★ ★ ★ ★ ★	★ ★ ★ ★ ★
I've been thinking clearly	★ ★ ★ ★ ★	★ ★ ★ ★ ★	★ ★ ★ ★ ★	★ ★ ★ ★ ★
I've been feeling close to other walkers	★ ★ ★ ★ ★	★ ★ ★ ★ ★	★ ★ ★ ★ ★	★ ★ ★ ★ ★

Score of the test: 0

☒ Guidelines for walkers

☒ Personal Protective Equipment (PPE)

☐ Received the guidelines for volunteers returning to walk

Yes

Covid Risk Questionnaire

Take Test

Cancel

Save

6.3.3 Crear y listar caminatas saludables

En esta sección se podrá crear una caminata en específico. La historia de usuario que corresponde con esta funcionalidad se puede visualizar en la tabla A.8.

Cabe destacar que estas caminatas no se les asigna el espacio temporal, ya que serán creadas para poder ser reutilizadas en el futuro por los distintos grupos de caminantes, sin una fecha y hora en específico.

La creación de estos paseos se realizará a través de un formulario mediante el cual, el administrador rellenará los datos de la caminata previa a realizarse. Este formulario incluye un mapa en el que se puede seleccionar el punto de encuentro de la caminata. Para poder utilizar este mapa se ha hecho uso de la *API* de *Google Maps*.

Se podrán filtrar las distintas caminatas creadas en la subsección anterior. Además, se podrán filtrar según unos filtros preestablecidos, como por ejemplo, por nombre de la ruta, código postal, distancia o lugar de encuentro.

Estas caminatas también podrán ser filtradas en la sección de *list walks*, como puede verse en la Figura 6.5.

6.4 Sprint 3

En este *sprint* se finalizó la sección de *Manage Walks*, implementando las subsecciones de *book walk* o reserva de caminatas y *list book walks*. También se implementó la funcionalidad de visualizar las rutas programadas en el calendario, así como el envío de SMS al caminante informándole de la ruta programada.

Finalmente, se terminó este *sprint* implementando la funcionalidad de poder visualizar las rutas programadas en el calendario del sistema.

6.4.1 Crear y listar reservas de caminatas

Esta subsección es en la que maneja toda la lógica de reserva de los paseos saludables. De esta manera, se hace uso de las caminatas creadas en la subsección *add walk*, asignándole un espacio temporal. La historia de usuario correspondiente se encuentra en el Anexo A, tabla A.9.

El administrador es quien se encarga de programar los paseos asignándoles una fecha y hora determinadas. Para la reserva de las caminatas saludables se ha utilizado también un formulario. Además, se dispondrá de una sección para poder visualizar en forma de lista todas las caminatas programadas. Se pueden visualizar estas reservas de caminatas en la Figura 6.6.

Figura 6.5: Creación y listado de una caminata saludable

WALKING FOR HEALTH

Walkers

Groups

Volunteers

Manage Walks

Contact

Walk Data

Name *

Thorpeness Windmill

Venue *

2129 Goodwin Points Apt. 175

Venue Address

118, Saint Avenue, Ipswich, Suffolk, England, LS53PY, UK.

Grade

Enter the walk grade

Marker Name

Enter the marker name

Walk distance (km) *

12

Description

Enter a description

Other Data

Walk Zone *

Rural


Walk Symbols *

Provisioning

Walk Risk Level *

Low

Meeting Point



Save

Walk List

These are the walks

Filter Criteria:

Add criteria

Total records:

4

Name	Venue Address	Distance (km)	Meeting Point	Risk Level
Walk 1	LS53PY.	22	52.0823444239944,-2.973126617187507	Medium
Walk 2	LS53PY.	22	52.38077643280481,-1.3471510605158876	Medium
Walk between 2 bridges	118, Saint Venue, Ipswich, Heath, Suffolk, England, LS53PY, UK.	14	52.165657,0.998431	High
Walk to the big bridge	118, Saint Venue, Ipswich, Heath, Suffolk, England, LS53PY, UK.	20	52.259217254226556,-0.25469705175781954	Low

WALKING FOR HEALTH

Walkers

Groups

Volunteers

Manage Walks

Contact

Add venue

List venues

Add walk

List walks

Book walk

List book walks

Contact

Figura 6.6: Crear y listar reservas de caminatas

Booking Name *

Booking 1

Walk *

Walk 1

Group *

Group 1

Start Date *

17/06/2021

End Date *

17/06/2021

Start Time *

17:10

End Time *

18:10

☒ Send SMS notification to walkers of this booking

Comments

This is a rural walk

Save

WALKING FOR HEALTH

Walkers

Groups

Volunteers

Manage Walks

Add venue

List venues

Add walk

List walks

Book walk

List book walks

Contact

Book Walks List

These are the book walks

Filter Criteria:

Add criteria

Total records: 5

<<<

<

1

>

>>>

Booking name	Walk	Group	Cancel Booking
Booking 1	Walk 1	Group 1	
Booking 2	Walk 1	Group 1	
Booking 5723	Walk 1	Group Only Childs	
Booking old bridge	Walk 1	Group 1	
Book walk 3	Walk 1	Group 3	

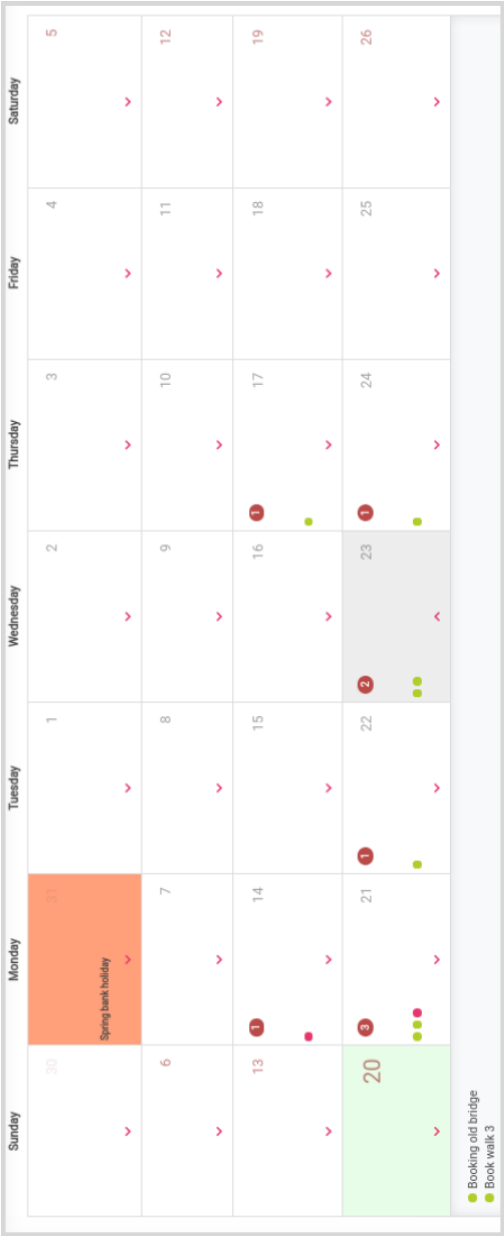
6.4.2 Visualizar rutas programadas en el calendario del sistema

Otro de los requisitos fundamentales a abordar en el desarrollo de este proyecto fue el de tener un calendario en el que se pudieran visualizar las rutas programadas. De esta forma, el administrador del sistema puede ver todas las caminatas programadas para poder llevar un control de éstas. En la Figura 6.7 se muestra el calendario del sistema con una serie de rutas programadas.

También se implementó la funcionalidad de poder actualizar estas rutas si se produjera algún cambio en el horario. Estas rutas se pueden cambiar o bien editando la ruta desde la lista de rutas programadas o desde el evento creado en el calendario.

Esta funcionalidad en el sistema web desarrollado tenía una prioridad media y un alto riesgo en desarrollo como puede verse en la historia de usuario A.10.

Figura 6.7: Visualización de las rutas programadas en el calendario



6.4.3 Envío de SMS con la información de la ruta programada

La problemática de poder ponerse en contacto o enviar de alguna manera la información a los caminantes podía ser abordada de diferentes formas. La opción elegida ha sido la del SMS debido a su alta eficiencia, sencillez y versatilidad.

Mediante este método de envío de notificaciones, cualquier caminante que vaya a asistir a una ruta será avisado con la información más relevante del paseo. De esta manera, se le recuerda al caminante asistir y puede planificarse mejor.

En esta sección se podrán consultar los SMS enviados a los caminantes informándoles de las rutas que van a realizar.

Esto también es una ventaja para el administrador ya que desde la lista implementada para ello en la subsección de *Contact* se pueden ver todos los mensajes enviados y a quién han sido enviados. En la Figura B.8 se puede visualizar un ejemplo del resultado del proceso de envío de SMS. La historia de usuario correspondiente se recoge en la tabla A.11.

6.5 Sprint 4

En este último *sprint* se llevó a cabo el desarrollo de la post-caminata. Es decir, se implementaron las secciones que tenían que ver con el registro de una caminata realizada. La funcionalidad de poder crear sesiones de formación para los voluntarios que lo prefieran. Y por último, permitir al administrador cancelar reservas de caminatas y enviarles a los caminantes su correspondiente notificación a través de SMS.

6.5.1 Registrar caminata realizada

Una vez que ha tenido lugar la caminata es necesario registrar en el sistema los datos de ésta. Para ello, se ha utilizado un modal en el que es el administrador el que se encarga de introducir los datos pertinentes de la ruta. Algunos de estos datos son sobre si ha asistido o no el caminante a la ruta, el paseo saludable que ha tenido lugar, tipo de ruta, minutos de actividad o la fecha y hora en la que se ha registrado.

La historia de usuario creada para este proceso de post-caminata queda recogida y puede visualizarse en la tabla A.12. También se puede consultar esta funcionalidad implementada en la Figura 6.8.

Figura 6.8: Registro de caminata en el sistema

Walking group #1 ^

Record Walk

In Progress

Walking status:

Active - In group

Registration date:

Mon 21 Jun 2021

Other Walker Data:

Employment Status: *Employed*
Occupation: *Full-time student*

Medical Condition:

Asthma

Adviser:

Aaron Robinson

Ambassador

No

Comments:

N/A

Walks

Walks Summary

Record Walk

Attendance: **Attended**

Walk #1 (22/06/2021 07:29 PM) ^

Walk #2 (22/06/2021 07:30 PM) ^

Attendance: **DNA - Forgot To Attend**

Walks

Walks Summary

Attendance: **Attended**

Walk #1 (22/06/2021 07:29 PM) ^

Walk Type

Regular Session

Duration

20 minutes

Comments:

N/A

Height

180 cm

Weight

55 kg

BMI

16

Blood Pressure (Systolic)

120

Blood Pressure (Diastolic)

80

Edit

Attendance: **DNA - Forgot To Attend**

Walk #2 (22/06/2021 07:30 PM) ^

72

6.5.2 Eventos de calendario para crear sesiones de formación

Las sesiones de formación consisten en una serie de reuniones instructivas con los voluntarios que se presentan para ser líderes en un futuro de paseos saludables. Estas sesiones se podrán programar mediante el uso de una serie de eventos del calendario del sistema.

El administrador se encargará de crear el evento y lo asignará a un *advisor*. Posteriormente, cuando ese *advisor* inicie sesión en el sistema con sus credenciales le aparecerá como evento en el calendario la sesión de formación que el administrador le había asignado. Esto puede visualizarse en la Figura B.7. También se puede consultar su historia de usuario en la tabla A.13.

6.5.3 Envío de SMS avisando de la cancelación de una ruta

Cuando el administrador decide cancelar una caminata, automáticamente se envía un SMS a aquellos caminantes que les afecte la cancelación de dicha ruta. En ese mensaje se especifica el paseo, la fecha y hora a la cual afecta esta cancelación, como puede verse en la Figura B.9. Además, en la tabla A.14 se puede consultar la historia de usuario correspondiente.

6.6 Costes y recursos del proyecto

El desarrollo del sistema web *Walking for Health* se inició el 2 de febrero de 2021 y finalizó el 4 de julio de 2021. En total son 26 semanas de trabajo. Se han dedicado 150 horas mensuales al desarrollo del proyecto. Por tanto, son 975 horas en total. Se estima entonces que el coste por hora de un desarrollador sin experiencia sería de 25€/hora, en el que se asume el coste que imputaría una empresa por un desarrollador junior. Aquí no sólo se incluye el sueldo bruto, sino también los costes de la seguridad social y la infraestructura de la empresa.

Dado que se ha integrado el servicio desarrollado dentro de los servicios de **OneLife Suffolk** se utiliza mismo hardware y recursos que en su despliegue para el resto de servicios. Por lo tanto, habría que añadir esos costes a los ya establecidos en esta sección.

El sistema se encuentra desplegado en AWS incluyendo los servicios de *Amazon EC2*, *AWS Elastic Beanstalk*, *Amazon RDS*, *Amazon S3*, *AWS SES*, *AWS Lambda*, *Amazon PinPoint*. En este caso, no se estima el precio de mantener el sistema desplegado pero sería un coste que habría que tener en cuenta. Además, se utilizan servicios extra como el servicio de SMS que tiene costes dependientes del uso.

Para el desarrollo de este proyecto se ha hecho uso de un ordenador *Asus* con el sistema operativo *GNU/Linux* instalado. Este *Asus* tiene un coste de unos 600€.

Cabe destacar que el coste total se va a calcular teniendo en cuenta estos parámetros. Por tanto, el coste final del proyecto ha sido de: 24.975€

En la Tabla 6.9 se puede observar un resumen de los costes del proyecto.

6. RESULTADOS

	Recurso	Unidad	Coste/unidad	Coste
Recursos humanos	Desarrollador	1	25 €/hora	24.375 €
Recursos desarrollo	Ordenador ASUS	1	600 €/unidad	600 €
Recursos proyecto	Servicio SMS	1	Propiedad intelectual <u>OneLife Suffolk</u>	Propiedad intelectual de <u>OneLife Suffolk</u>
	AWS	7	Propiedad intelectual de <u>OneLife Suffolk</u>	Propiedad intelectual de <u>OneLife Suffolk</u>
				24.975 €

Figura 6.9: Costes y recursos del proyecto

6.7 Estadísticas del proyecto

En esta sección se puede visualizar de manera gráfica algunas de las estadísticas recogidas del proyecto realizado. Durante el desarrollo del sistema web se utilizó un repositorio alojado en *Bitbucket* para guardar todo el software del sistema.

El repositorio en *Bitbucket* tiene 307 ramas hasta el 2 de julio. El sistema de control de versiones es *Git*. La mayor parte del código está escrito en *TypeScript* y *HTML* entre otros.

Los contribuyentes del repositorio fueron todo el equipo de desarrollo contratado por *MoreLife* y *FK*. Aún así, el código del servicio correspondiente a *Walking for Health* fue desarrollado exclusivamente por un único desarrollador, que es la autora de este TFG.

En la Figura 6.10 se muestra un resumen de todos los *commits* realizados en *FK* desde el mes de febrero hasta principios del mes de julio, incluidos aquellos que tienen que ver con el desarrollo de servicio web. Para *Walking for Health* se realizaron 50 *commits*, se añadieron 22.720 líneas y se borraron 3.566 líneas de código.



Figura 6.10: Estadísticas del repositorio generadas por Bitbucket

Conclusiones

EN este último capítulo se va a proceder a realizar una valoración final de los resultados alcanzados en este TFG. Para ello, se revisarán los objetivos expuestos en el capítulo 2, justificando de qué manera se han conseguido. Además, se comentarán las dificultades encontradas a lo largo de todo el desarrollo de este proyecto. Finalmente, se realizará una reflexión acerca de los posibles trabajos futuros que podrían seguirse para el progreso y avance del servicio web desarrollado para este proyecto.

7.1 Análisis de objetivos alcanzados

Como se estableció en el capítulo 2 de este TFG, el principal objetivo ha sido el desarrollo de un sistema web para la gestión de un programa de salud orientado a caminatas saludables. Este objetivo ha sido alcanzado gracias al desarrollo y despliegue en producción del nuevo servicio web *Walking for Health*.

A continuación, se muestran los objetivos que han sido alcanzados tras la realización de este proyecto.

1. **Estudio de las tecnologías y herramientas que soporten el desarrollo y despliegue del servicio, incluyendo los sistemas web que tratan de solucionar una problemática similar a la que se enfrenta este proyecto.** Tal y como se explica en el capítulo 5 se procedió a realizar un estudio exhaustivo de las tecnologías y herramientas para abordar con éxito este TFG. También se realizó un estudio comparativo de los sistemas web existentes en la actualidad que tenían relación con la problemática de este proyecto. Esto se encuentra explicado en detalle en el capítulo 3.
2. **Creación, desarrollo e integración del proceso de caminatas saludables.** Se diseñaron e implementaron dos procesos, el proceso de pre-caminata y el de post-caminata. Estos procesos se encuentran explicados mediante un diagrama de secuencia en el capítulo 5.

Cabe destacar que la integración fue compleja ya que se debía dividir dos procesos con numerosas funcionalidades. Pero al final, este objetivo fue llevado a cabo con éxito, consiguiendo integrar ambos procesos mencionados dentro del sistema real ya en producción.

3. **Desarrollo de formularios de salud que permitan registrar información relevante para su posterior estudio y análisis.** Este objetivo se logró con éxito y puede consultarse en detalle en el capítulo 6.

Uno de ellos es el formulario diseñado para la creación de voluntarios se incluye un modal que permite mostrar el formulario de la Evaluación de Riesgos *COVID*. En él se hacen una serie de preguntas relacionadas sobre cómo se siente el caminante al volver a hacer caminatas saludables, si ha recibido una guía, si tiene equipo personal de protección, etc.

4. **Creación de nuevas funcionalidades escalables y consistentes que permitan gestionar caminatas saludables y dar soporte para su correcta visualización.** Este objetivo se consiguió llevar a cabo con éxito debido a la implementación de las siguientes funcionalidades:

En primer lugar, se implementó la funcionalidad de notificaciones para poder avisar a los caminantes de los paseos programados. Esta funcionalidad se llevó a cabo utilizando el sistema de mensajería de SMS. En el módulo de *Contact* se listan los mensajes enviados por el administrador del sistema a los caminantes.

Por otro lado, el sistema permite la creación de eventos de calendario donde quedan registradas las caminatas que se programan previamente en la subsección *Book walk* del módulo *Manage walks*. También permite la creación de eventos que permitan gestionar las sesiones de formación de los voluntarios del sistema.

Además, se diseñó un formulario que permite recoger la información del paseo una vez que éste ha tenido lugar. Con él, se tienen guardados todos los datos necesarios para su posterior consulta por parte del administrador.

Estas funcionalidades comentadas pueden visualizarse mejor en el capítulo de 6, donde se muestran de manera gráfica los resultados conseguidos tras su implementación.

5. **Implementación de una arquitectura modular.** Este último objetivo se ha llevado a cabo mediante el uso de módulos a lo largo de todo el desarrollo del proyecto. Para ello, se ha dividido el sistema en 5 módulos en los que se han implementado sus funcionalidades correspondientes. Estos módulos dividen el proceso de pre-caminata y post-caminata, los cuales se encuentran recogidos en el capítulo 5.

En estos módulos se recoge toda la funcionalidad desarrollada para este TFG. En los 3 primeros módulos referentes a *Walkers*, *Groups* y *Volunteers* se implementó todo lo relacionado con la creación de caminantes, lista de espera, creación de grupos de caminatas y voluntarios del sistema.

En los dos últimos módulos se implementó la gestión de caminatas (*Manage walks*) en la que se maneja todo lo referente a la creación, actualización, reserva, cancelación de caminatas y creación de lugares de encuentro. Y *Contact*, en la que se pueden listar todos los SMS enviados a los caminantes.

7.2 Dificultades

A lo largo del desarrollo de este proyecto, se han presentado diversas dificultades que serán comentadas a continuación.

La mayor dificultad presentada ha sido la integración de *Walking for Health* en el sistema ya existente. Esta integración ha sido complicada ya que ha tenido lugar dentro de un equipo de trabajo con un proyecto en curso.

Sin embargo, esta dificultad también ha tenido sus ventajas. Y es que esto me permitido desarrollar aún más la proactividad y el esfuerzo. Además, ha supuesto un gran aprendizaje adaptarse a entornos cambiantes con los que al salir del grado universitario no se está tan acostumbrado.

Otra de las dificultades encontradas ha sido la de conseguir que el mapa utilizado para marcar el punto de encuentro al crear una ruta mediante la API de *Google* se visualizara correctamente. Finalmente, también se ha conseguido resolver con éxito esta dificultad.

7.3 Competencias académicas trabajadas

En esta sección se justifican las competencias académicas trabajadas en el desarrollo del proyecto y que tienen relación con la rama de TI. En la tabla 7.1 se pueden consultar las distintas competencias aplicadas.

Cuadro 7.1: Justificación de competencias

Competencia académica	Justificación
[TI2] Capacidad para seleccionar, diseñar, desplegar, integrar, evaluar, construir, gestionar, explotar y mantener las tecnologías de <i>hardware</i> , <i>software</i> y redes, dentro de los parámetros de coste y calidad adecuados.	En este TFG se ha hecho uso de las tecnologías <i>hardware</i> y <i>software</i> utilizadas para el resto de servicios ya en producción de OneLife Suffolk, integrando el nuevo servicio dentro de los ya existentes y adquiriendo las capacidades necesarias para construir, diseñar e integrar una solución que se adapte a las restricciones existentes dentro de los parámetros de coste y calidad que engloban al proyecto.

[TI3] Capacidad para emplear metodologías centradas en el usuario y la organización para el desarrollo, evaluación y gestión de aplicaciones y sistemas basados en tecnologías de la información que aseguren la accesibilidad, ergonomía y usabilidad de los sistemas.	Se ha aplicado la metodología <i>Agile Scrum</i> , expuesta en el capítulo 4. Esta metodología es abierta, flexible y se adapta a las necesidades de los clientes en cada momento. Seguir una metodología <i>Agile</i> ha ayudado a minimizar tiempo y costes del sistema, gracias a que se trabaja dentro de un modelo que tiene unas metas fijadas. Además, al trabajar en plazos de tiempo cortos, permite prestarle más atención al resultado mientras se ahorra en costes.
[TI7] Capacidad para comprender, aplicar y gestionar la garantía y seguridad de los sistemas informáticos.	El uso de directivas personalizadas y el uso de permisos para proteger tanto el <i>backend</i> como el <i>frontend</i> son claros ejemplos relacionados con la seguridad que se han tenido en cuenta en el desarrollo del proyecto.

7.4 Trabajos futuros

A continuación, se expondrán y plantearán una serie de posibles implementaciones o mejoras futuras para el servicio web desarrollado en este TFG.

1. **Convertir un caminante en voluntario de caminatas.** Durante la realización de los paseos saludables es probable que ciertos caminantes deseen convertirse en voluntarios para poder posteriormente dirigir caminatas y convertirse en líderes. Estos voluntarios se añadirían a una lista de caminantes interesados en convertirse en líderes de caminatas.

Para ello, se utilizarían las sesiones de formación de voluntarios en las que se forma a los nuevos voluntarios y se decidirá con ellas si son aptos o no para ser líderes y poder llevar grupos de caminatas.

2. **Evaluación de las caminatas saludables.** El proceso de evaluar las caminatas saludables podría realizarse mediante un formulario que rellenarían previamente los caminantes. Este formulario contendría una serie de preguntas sobre aspectos como qué tal ha sido la experiencia con el grupo de caminatas, la implicación del líder con el grupo, si ha surgido algún problema o incidente durante el paseo, mejoras o sugerencias de las rutas, etc.

El formulario seguiría el concepto y el estilo del ya implementado para la creación de un voluntario en el sistema.

De esta manera, el administrador podría posteriormente realizar una evaluación global de los paseos saludables llevados a cabo, basándose en la opinión recogida de los caminantes para poder mejorar ciertas características de las caminatas y ofrecerles una mejor experiencia en cada ruta.

3. **Penalizar a los caminantes que frecuentemente no asistan a las caminatas.** En el proceso de post-caminata, cuando el administrador registra los datos de la ruta realizada de un caminante en concreto, existe un desplegable que permite registrar si ese caminante ha asistido, no asistido, cancelado u olvidado presentarse a la caminata. De esta forma, teniendo esta información se podría llevar fácilmente un registro de los caminantes que con frecuencia se apuntan a los paseos saludables pero luego no asisten por motivos desconocidos.

Por tanto, se podría crear una sección en el sistema que permitiese listar a estas personas y llevar un control de estos comportamientos no deseados, informándoles en última instancia de este suceso mediante una notificación a través de SMS.

4. **Avisar a los voluntarios de las sesiones de formación.** Esta funcionalidad podría implementarse de manera similar a la funcionalidad desarrollada en este TFG cuando se les comunica a los caminantes mediante un SMS las rutas programadas. De esta forma, se enviaría a los voluntarios una notificación al crear una sesión de formación en el calendario del sistema.

Bibliografía

- [CCD⁺07] Facultad De Ciencias, Amaro Calderón, Sarah Dámaris, Valverde Rebaza, y Jorge Carlos. Universidad Nacional de Trujillo. *Lexus*, 4(None):37, 2007.
- [Her04] I. Hernández. Obesidad y salud pública. *Endocrinología y Nutrición*, 51(2):35–36, 2004. url: <https://www.elsevier.es/es-revista-offarm-4-pdf-13111061>.
- [Her13] M. J. Hernandez, MichaHernandez. *Database Design for Mere Mortals: A Hands-On Guide to Relational Database Design*. Addison-Wesley Professional; 3rd edition (February 14, 2013), 2013. url: https://books.google.es/books/about/Database_Design_for_Mere_Mortals.html?id=_n4hBQAAQBAJ&redir_esc=yhttps://books.google.com/books?id=MVApxsrcHiAC&pgis=1.
- [Jua04] Juan Salvador Castejón Garrido. Arquitectura y diseño de sistemas web modernos. *Revista de Ingeniería Informática del CIIRM (Colegio de ingenieros en informática de la región de Murcia)*, páginas 1–6, 01 2004.
- [Mar14] Azat Mardan. *Practical Node.js*. Apress, 2014.
- [PDB⁺09] Pete Por, Gabrielle Deemer, Craig Benefield, Bas Larman, y Versión Vodde. Información básica de SCRUM (The Scrum Primer). Technical report, Scrum Training Institute, 2009. url: www.ScrumTI.com.
- [PR09] Juan Palacio y Claudia Ruata. Scrum Manager: Gestión de proyectos. *Scrum Manager*, 1.4.0:1–113, 01 2009.
- [Roj17] Luis Alberto Rojas Pino. Integración de la arquitectura de la información dentro de un proceso ágil de desarrollo centrado en el usuario. *Universidad Autónoma de Madrid*, página 439, 2017.
- [Ser16] Sergio Mosquera. Requirements Engineering in Agile Development Methods Engenharia de Requisitos , em Métodos Ágeis de Desenvolvimento La Ingeniería de Requisitos en los Métodos de Desarrollo Agiles. *La ingeniería de requisitos en los métodos de desarrollos agiles resumen*, 2016.

7. BIBLIOGRAFÍA

- [SPL⁺11] Joseba Rabanales Sotos, Ignacio Párraga Martínez, Jesús López-Torres Hidalgo, Fernando Andrés Pretel, y Beatriz Navarro Bravo. ¿QUÉ ES LA TELEMEDICINA? Tecnologías de la Información y las Comunicaciones: Telemedicina. Technical Report 1, Unidad de Investigación de la Gerencia de Atención Primaria de Albacete, 2011.
- [TCGS14] Cristhian Leonardo Tolosa-Cuadrado y Juan Sebastián González-Sanabria. Amazon Web Services: alternativa para el almacenamiento de información-Amazon Web Services: an alternative for storing information. *Revista científica*, 2(19):134, 2014.
- [Tho11] P. Y. Thomas. Computación en la nube: un paradigma potencial para practicar la erudición de la enseñanza y el aprendizaje. *Electronic Library*, 29(2):214–224, 04 2011.
- [TV10] Stefan Tilkov y Steve Vinoski. Node.js: uso de JavaScript para crear programas de red de alto rendimiento. *IEE Internet Computing*, 14(6):80–83, nov 2010.

ANEXOS

Anexo A

Historias de usuario

En este anexo se describen las historias de usuario empleadas durante el desarrollo de este proyecto.

Introducción a los servicios web	
Número: 1	Sprint: 0
Prioridad: Alta	Riesgo en desarrollo: Bajo
Descripción: Aprender sobre los servicios web y cómo se comunican entre ellos para poder abordar con éxito este proyecto.	
Criterios de aceptación: <ul style="list-style-type: none">• Aprender las bases de los servicios web para poder desarrollar con éxito el proyecto.• Estudiar los fundamentos web y los protocolos de comunicación que existen.	

Cuadro A.1: HU1 Introducción a los servicios web

Creación de la sección de caminantes	
Número: 2	Sprint: 1
Prioridad: Alta	Riesgo en desarrollo: Bajo
Descripción: Como administrador quiero poder listar los caminantes que he creado, incluyendo los caminantes recientes, búsqueda global de clientes en el sistema y poder asociar las waiting list a las caminatas para llevar un registro de los caminantes que no tienen grupo asignado.	
Criterios de aceptación: <ul style="list-style-type: none">• Comprobar que se pueden listar correctamente los caminantes creados y los caminantes recientes.• Comprobar que funcionan los filtros para filtrar caminantes y la búsqueda global de ellos en el sistema.• Asociar las waiting list a las caminatas saludables.	

Cuadro A.2: HU2 Listado y búsqueda de caminantes

A. HISTORIAS DE USUARIO

Crear y listar grupos de caminatas	
Número: 3	Sprint: 1
Prioridad: Media	Riesgo en desarrollo: Medio
Descripción: Como administrador quiero poder crear y listar grupos de caminatas para asignarlas posteriormente a una ruta.	
Criterios de aceptación: <ul style="list-style-type: none"> • Crear grupos de caminatas con sus diferentes opciones. • Tener un listado con todas los los grupos de caminatas disponibles en el sistema. • Comprobar que se pueden actualizar los datos del grupo de la caminata. 	

Cuadro A.3: HU3 Crear y listar grupos de caminatas

Asignar caminante a grupo de caminatas	
Número: 4	Sprint: 1
Prioridad: Media	Riesgo en desarrollo: Alto
Descripción: Como administrador quiero asignar un caminante a un grupo de caminatas para poder registrar que ese caminante ha sido el que ha realizado el paseo en ese grupo determinado.	
Criterios de aceptación: <ul style="list-style-type: none"> • Comprobar que el caminante tiene un grupo asignado. • Poder actualizar los campos del caminante añadido a un grupo de caminatas. 	

Cuadro A.4: HU4 Asignar un caminante a un grupo de caminatas

Crear y listar voluntarios	
Número: 5	Sprint: 2
Prioridad: Media	Riesgo en desarrollo: Medio
Descripción: Como administrador quiero crear y listar voluntarios para posteriormente poder programar en el calendario sesiones de formación para convertirse en líder.	
Criterios de aceptación: <ul style="list-style-type: none"> • Comprobar que se pueden crear voluntarios. • Tener un listado con todos los voluntarios registrados en el sistema. • Comprobar que se pueden actualizar los datos de un voluntario. 	

Cuadro A.5: HU5 Crear y listar voluntarios

Cuestionario de riesgos Covid	
Número: 6	Sprint: 2
Prioridad: Media	Riesgo en desarrollo: Alto
Descripción: Como administrador quiero rellenar el cuestionario de salud de riesgos Covid de un voluntario para llevar un registro de vulnerabilidad ante el Covid.	
Criterios de aceptación: <ul style="list-style-type: none"> • Habilitar el cuestionario de salud cuando se conteste previamente a una serie de preguntas. • Comprobar que se pueden actualizar los datos del cuestionario. 	

Cuadro A.6: HU6 Evaluación de Riesgos COVID

Crear y listar venues / lugares de encuentro	
Número: 7	Sprint: 2
Prioridad: Media	Riesgo en desarrollo: Medio
Descripción: Como administrador quiero poder crear y listar venues para asignarlas a cada ruta que quiera crear.	
Criterios de aceptación: <ul style="list-style-type: none"> • Crear venues con sus diferentes opciones. • Tener un listado con todas las venues disponibles en el sistema. • Comprobar que se pueden actualizar los datos de una venue. 	

Cuadro A.7: HU7 Crear y listar venues

Crear y listar caminatas saludables	
Número: 8	Sprint: 2
Prioridad: Alta	Riesgo en desarrollo: Alto
Descripción: Como administrador quiero poder crear y listar caminatas saludables para llevar un control de las mismas.	
Criterios de aceptación: <ul style="list-style-type: none"> • Crear una caminata con sus diferentes opciones. • Tener un listado con todas las caminatas disponibles. • Comprobar que se pueden actualizar los datos de una caminata. 	

Cuadro A.8: HU8 Crear y listar caminatas saludables

A. HISTORIAS DE USUARIO

Reserva de caminatas saludables	
Número: 9	Sprint: 3
Prioridad: Alta	Riesgo en desarrollo: Alto
Descripción: Como administrador quiero reservar una caminata saludable para asociarle una fecha y hora concretas.	
Criterios de aceptación: <ul style="list-style-type: none"> • Crear una reserva con sus diferentes opciones. • Tener un listado con todas las reservas de caminatas programadas. • Asociar a la reserva una ruta creada previamente con una fecha y hora determinadas. • Asociar a la reserva un grupo de caminatas creado previamente. • Comprobar que se pueden actualizar los datos de una reserva. 	

Cuadro A.9: HU9 Reserva de caminatas saludables

Visualizar rutas programadas en el calendario	
Número: 10	Sprint: 3
Prioridad: Media	Riesgo en desarrollo: Alto
Descripción: Como administrador quiero poder visualizar en el calendario las caminatas programadas para poder llevar un control de éstas.	
Criterios de aceptación: <ul style="list-style-type: none"> • Comprobar que se pueden visualizar las rutas en el calendario. • Comprobar que las rutas se actualizan si se produce algún cambio temporal. 	

Cuadro A.10: HU10 Visualización de rutas programadas en el calendario

Envío de SMS a los caminantes con la información de la ruta	
Número: 11	Sprint: 3
Prioridad: Media	Riesgo en desarrollo: Medio
Descripción: Como administrador quiero poder enviar una notificación a los caminantes para informarles de los detalles de la caminata programada en una fecha y hora determinadas.	
Criterios de aceptación: <ul style="list-style-type: none"> • Enviar SMS a los caminantes incluyendo la información necesaria de la ruta programada. 	

Cuadro A.11: HU11 Envío de SMS con la información de la ruta programada

Registrar caminata realizada	
Número: 12	Sprint: 4
Prioridad: Alta	Riesgo en desarrollo: Alto
Descripción: Como administrador quiero registrar todos los datos de la caminata que ha tenido lugar para tener un control de las caminatas que ya se han llevado a cabo.	
Criterios de aceptación: <ul style="list-style-type: none"> • Registrar una caminata realizada con todos sus datos correspondientes. • Comprobar que se pueden actualizar los datos de la caminata realizada. • Comprobar que se puede visualizar un resumen de todas las caminatas realizadas. 	

Cuadro A.12: HU12 Registro de caminata realizada

Sesiones de formación para los voluntarios	
Número: 13	Sprint: 4
Prioridad: Media	Riesgo en desarrollo: Medio
Descripción: Como administrador quiero crear un evento en el calendario para programar las sesiones de formación de los voluntarios para que los voluntarios se instruyan antes de poder convertirse en líderes.	
Criterios de aceptación: <ul style="list-style-type: none"> • Poder crear una sesión de formación mediante un evento en el calendario. • Comprobar que las sesiones de formación se visualizan correctamente en el calendario. 	

Cuadro A.13: HU13 Sesiones de formación para voluntarios

Cancelación de una caminata programada	
Número: 14	Sprint: 4
Prioridad: Alta	Riesgo en desarrollo: Alto
Descripción: Como administrador quiero poder cancelar una caminata ya programada para que se elimine de la lista de rutas programadas y del calendario del sistema. Además, como administrador quiero que se envíe un SMS al caminante para informarle de la cancelación de la ruta.	
Criterios de aceptación: <ul style="list-style-type: none"> • Comprobar que las caminatas programadas canceladas no aparecen en el calendario. • Comprobar que las caminatas programadas canceladas no aparecen en la lista de reservas de rutas programadas. • Comprobar que el SMS se ha enviado correctamente al caminante con la información de la cancelación de la ruta. 	

Cuadro A.14: HU14 Cancelación de una caminata programada

Anexo B


Funcionalidades de Walking for Health

En este Anexo se expondrán las funcionalidades implementadas para el servicio web *Walking for Health*.

B.1 Sección de caminantes o walkers

B.1.1 Caminantes creados por el usuario admin (My walkers)

Figura B.1: Subsección para visualizar los caminantes creados por el administrador

**My Walkers**
List of walkers visible for you

Filter Criteria:

Add criteria

Total records:

3

<<<<

<

1

>

>>>>

ID	Full Name	Date of Birth	Home Address
0029604	Dorothy MONEKOSSO	23/09/2000	519, Riverside Island Marina, Ely, Suffolk, England, CB756L, United Kingdom.
0029611	Nadine ELMS	23/09/2000	519, Riverside Island Marina, Ely, Suffolk, England, CB756L, United Kingdom.
0029614	Pepe PÉREZ	04/05/1997	LS53PY.
0029618	Adam SMITH	09/08/1994	118, Saint Avenue, Ipswich, Suffolk, England, LS53PY, United Kingdom.
0029619	Elizabeth MILLER	18/09/1998	120, Ipswich, Suffolk, England, LS53PY, UK.


B.1.2 Caminantes recientes (Recent walkers)

Figura B.2: Caminantes añadidos recientemente al sistema

Recently Accessed Walkers List of walkers visible for you				
<div> <div>PERIOD</div> <div> <div>All Time</div> <div>All Time</div> <div>Today</div> <div>This Week</div> <div>This Year</div> <div>Custom</div> </div> </div> <div> <div>Total records</div> <div>1</div> <div><<<</div> <div>>>></div> </div>				
ID	Full Name	Date of Birth	Date Added	Actions
0029604	Dorothy MONEKOSSO	23/09/2000	08/06/2021	Record Walk
0029611	Nadine ELMS	23/09/2000	18/06/2021	Record Walk
0029614	Pepe PÉREZ	04/05/1997	21/06/2021	Record Walk
0029618	Adam SMITH	09/08/1994	29/06/2021	Record Walk
0029619	Elizabeth MILLER	18/09/1998	29/06/2021	Record Walk

B.1.3 Búsqueda de caminantes (Search)

Figura B.3: Búsqueda de caminantes y clientes del resto de servicios web



Search Walkers

Filter Criteria:

Add criteria

Total records: 262

<<<

<

1

2

3

...


>

>>>

ID	Full Name	Date of Birth	Practice	Actions
000012	Hilary JOHNSON	01/01/1990	THE REYNARD SURGERY	Record Walk
0000153	Jenice CLARK	06/07/1976		Record Walk
0000469	Marc DALE	26/11/1960		Record Walk
0000724	Jean ROBERTS	01/11/2001	VICTORIA SURGERY	Record Walk
0001073	Mary WEST	06/02/1992	ORCHARD MEDICAL PRACTICE	Record Walk
0002202	Sam WILLIAMS	25/04/1972	HIGH STREET SURGERY	Record Walk
0003928	Tracey CROSS	18/01/1975	BARRACK LANE MEDICAL CENTRE	Record Walk

B.1.4 Lista de espera (Waiting List)

Figura B.4: Lista de espera de caminantes en la sección de Waiting list

**Waiting List**
Manage waiting lists

Filter Criteria:

Add criteria

Total records: 3

<<<

<

1

>

>>>

Export as CSV

ID	Full Name	Date of Birth	Guardian name	Guardian contact number	Waiting List	Mobile Number
0029615	Pepe PÉREZ	03/03/1998	N/A	N/A	Waiting for Health - Waiting List	43432423423
0029616	Marta JIMÉNEZ	02/02/1997	N/A	N/A	Waiting for Health - Waiting List	42423423423
0029617	Jack WILLIAMS	06/06/1997	N/A	N/A	Waiting for Health - Waiting List	13213222131

94

B.2 Crear y listar lugares de encuentro o venues

Figura B.5: Creación de una venue

One Life Suffolk

WALKING FOR HEALTH

Walkers

Groups

Volunteers

Manage Walks

Add venue

List venues

Add walk

List walks

Book walk

List book walks

Contact

Suffolk

MY INFORMATION GENIUS

MANAGING MY WEIGHT

MY SMOKE FREE LIFE

MY HEALTH CHECKS

MY ACTIVE LIFE

WALKING FOR HEALTH

Enter venue

Insert or update a venue with this form

Venue Name *

2129 Goodwin Points Apt. 175

Venue Code *

2129 Goodwin Points Apt. 175

Full Address *

118, Saint Avenue, Ipswich, Suffolk, England, LS5 3PY, UK.

Working Area *

Suffolk

Save

Figura B.6: Listar venues

One Life Suffolk

WALKING FOR HEALTH

Walkers

Groups

Volunteers

Manage Walks

Add venue

List venues

Add walk

List walks

Book walk

List book walks

Contact

Suffolk

MY INFORMATION GENIUS

MANAGING MY WEIGHT

MY SMOKE FREE LIFE

MY HEALTH CHECKS

MY ACTIVE LIFE

WALKING FOR HEALTH

MY ADMIN GENIUS

QA TEST ADMIN

Venue List

These are the venues

Filter Criteria:

Add criteria

Total records: 2

<<<

<

1

>

>>>

Name	Code	Address	Working Area
2129 Goodwin Points Apt. 175	2129 Goodwin Points Apt. 175	118, Saint Avenue, Ipswich, Suffolk, England, LS53PY, UK.	Suffolk
2185 Britten Avenue	2185 Britten Avenue	Suffolk, England, LS53PY, UK.	Suffolk

96

B.3 Sesiones de formación para los voluntarios

Figura B.7: Sesión de formación programada para un advisor en el sistema

Custom event

Title *

First training session for volunteers

Description

This is the first training session. Remember send all the necessary information.

☐ All-day Event

Start *

25/06/2021 14:00

End *

25/06/2021 16:00

Repeat

M T W T F S S

Until

dd/mm/yyyy

Service

Health Walks

Type

Custom event

Advisor

Noelia Granados

Client Id

Write the client's public id to filter

Cancel Save

MY INFORMATION GENIUS

MY HEALTH CHECKS

WALKING FOR HEALTH

Noelia Granados HW_CRU

SERVICE PROVIDERS

JUN 2021

MONTH WEEK DAY PREVIOUS NOW NEXT

Sunday	Monday	Tuesday	Wednesday	Thursday	Friday	Saturday
30	31 Spring bank holiday	1	2	3	4	5
6	7	8	9	10	11	12
13	14	15	16	17	18	19
20	21	22	23	24	25	26

First training session for volunteers

B.4 Envío de SMS de una ruta programada

Figura B.8: SMS enviado a un caminante y lista de SMS enviados en el sistema

8:59

NOTICE

jueves, 1 de julio de 2021

The walk Alton Water, Shotley Peninsula is scheduled from 08/07/2021 13:27 to 08/07/2021 16:27

8:59

SMS Sent

Previously sent SMS

Filter Criteria: Add criteria

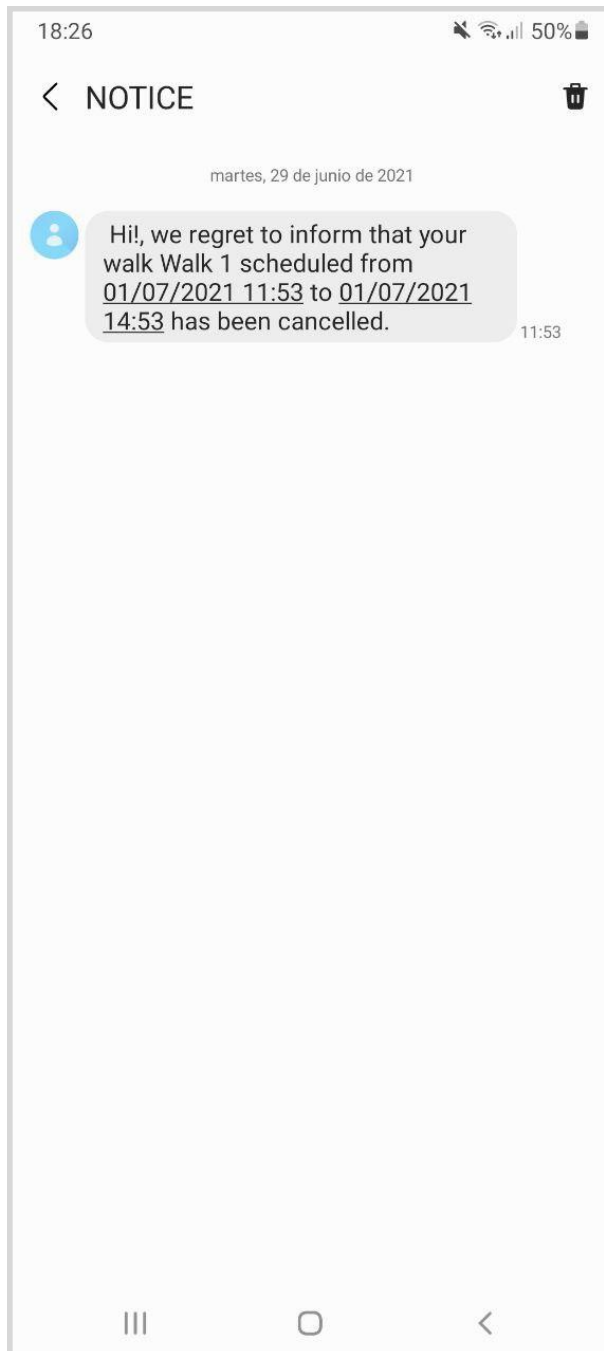
Total records: 3

Export as CSV

Sender	Date	Number	Client	Subject	Message	Delivery Status
Chris Armstrong	01/07/2021 09:13 AM	434644589744	29619	Hi, here is the information of your next scheduled walk	The walk Flatford is scheduled from 03/07/2021 14:18 to 03/07/2021 17:18	SUCCESSFUL
Chris Armstrong	01/07/2021 08:59 AM	434644589744	29619	Hi, here is the information of your next scheduled walk	The walk Alton Water, Shotley Peninsula is scheduled from 08/07/2021 13:27 to 08/07/2021 16:27	SUCCESSFUL
QA TEST	21/06/2021 02:14 PM	434644589744	29614	OneLife	Laura MacDonald	SUCCESSFUL

B.5 Envío de SMS con la cancelación de una ruta

Figura B.9: SMS de la cancelación de una caminata



Este documento fue editado y tipografiado con \LaTeX empleando
la clase **esi-tfg** (versión 0.20181017) que se puede encontrar en:
<https://github.com/UCLM-ESI/esi-tfg>

