



**Universidad de Valladolid**

**Escuela de Ingeniería Informática**  
**Trabajo Fin de Grado**

Grado en Ingeniería Informática

Mención en Tecnologías de la Información.

**APP DE PASEo-GAM: estudio y generación de rutas  
de paseo por tiempos**

**Autor: Alejandro Pescador González**





**Universidad de Valladolid**

**Escuela de Ingeniería Informática**

**Trabajo Fin de Grado**

Grado en Ingeniería Informática

Mención en Tecnologías de la Información.

**APP DE PASEo-GAM: estudio y generación de rutas  
de paseo por tiempos**

**Autor:** Alejandro Pescador González

**Tutora:** Margarita Gonzalo Tasis



## **Resumen**

Con el avance de la tecnología y el aumento de su uso, el objetivo de este proyecto es el desarrollo de una aplicación con la que facilitar la labor de decidir por donde poder dar un paseo en una ciudad desconocida cuando se dispone de un tiempo limitado.

Para llevar a cabo esta labor se hará uso de todo lo aprendido durante el estudio del grado en Ingeniería Informática junto con otras técnicas modernas que ayudarán con el desarrollo. Esto implica la adquisición de nuevos conocimientos y competencias, lo que resulta una manera ideal para acabar de cumplimentar el estudio del grado.

## **Abstract**

As technology and its usage keeps in progress, this project's purpose is to develop an app which helps to decide where they could go for a walk in an unknown city when their time is limited.

For its development, the knowledge acquired during the degree of computer engineering will be put on it apart from other modern techniques. This implies that the acquisition of new knowledge and skills have been get it, which is an excellent way to finish the degree study



## INDICE DE CONTENIDO:

Capítulo 1: Introducción y contexto. ....	1
1.1 Contexto y motivación. ....	1
1.2 Objetivos. ....	1
1.3 Estudio de mercado. ....	2
1.3.1 Estudio de aplicaciones del mercado. ....	2
1.3.2 Público objetivo y necesidades de este. ....	6
1.4 Estructura de la memoria. ....	6
Capítulo 2: Planificación del proyecto. ....	7
2.1 Metodología de trabajo (Scrum). ....	7
2.1.1 Motivos para la elección de Scrum. ....	7
2.1.2 Artefactos Scrum usados en el proyecto. ....	7
2.1.3 Roles Scrum usados en el proyecto. ....	8
2.1.4 Eventos Scrum utilizados en el proyecto. ....	8
2.1.5 Herramientas usadas para apoyar Scrum (Azure DevOps). ....	9
2.1.6 Seguimiento de Scrum en el desarrollo del proyecto. ....	9
2.2 Plan de gestión de riesgos. ....	11
2.2.1 Identificación, plan de contingencia y mitigación. ....	12
2.2.2 Seguimiento y monitorización de los posibles riesgos. ....	19
2.3 Presupuesto. ....	19
2.3.1 Presupuesto simulado. ....	20
2.3.2 Presupuesto real. ....	20
Capítulo 3: Análisis. ....	23
3.1 Especificación de requisitos. ....	23
3.1.1 Requisitos funcionales. ....	23
3.1.2 Requisitos de información. ....	24
3.1.3 Requisitos no funcionales. ....	24
3.2 Roles y actores del sistema. ....	25
3.3 Diagrama de casos de uso. ....	25
3.4 Descripción de los casos de uso. ....	26
3.5 Diagrama de secuencia de los casos de uso. ....	33
3.6 Modelo de dominio. ....	41
Capítulo 4: Diseño. ....	43
4.1 Arquitectura del sistema. ....	43
4.2 Diagrama de despliegue. ....	44

4.3 Frameworks utilizados.....	45
4.3.1 Ionic.....	45
4.3.2 Angular Js.....	45
4.4 Patrones arquitectónicos.....	46
4.4.1 Data Mapper.....	46
4.4.2 Capas y databinder.....	46
4.5 Modelo de base de datos.....	47
4.6 Diseño de la interfaz de usuario.....	47
4.7 Diagrama de clases de diseño.....	48
4.8 Diagrama de clases detallado.....	49
Capítulo 5: Implementación.....	51
5.1 Herramientas.....	51
5.1.1 Visual Studio Code.....	51
5.1.2 GitHub.....	51
5.1.3 Azure DevOps.....	51
5.1.4 Android Studio.....	52
5.1.5 Firebase.....	52
5.1.6 Turf.....	53
5.1.7 Mapbox.....	53
5.1.8 Draw.io.....	53
5.1.9 Astah.....	53
5.2 Decisiones de implementación y algoritmos utilizados.....	53
5.2.1 Versiones de Android soportadas.....	53
5.2.2 Algoritmos usados.....	54
5.2.3 Protección de datos y seguridad.....	56
5.3 Organización del código.....	57
Capítulo 6: Pruebas.....	59
6.1 Criterio de evaluación de las pruebas.....	59
6.2 Definición y resultado de las pruebas.....	60
6.3 Pruebas de usuarios reales.....	64
Capítulo 7: Conclusiones y trabajo futuro.....	65
7.1 Conclusiones.....	65
7.2 Trabajo futuro.....	65
Webgrafía y Bibliografía.....	67
Anexo I: Guía de Instalación.....	69



Anexo II: Guía de usuario.....	71
Registro de usuario e inicio de sesión. ....	71
Búsqueda de una localización.....	71
Creación y guardado de un paso .....	72
Seguimiento de un paseo. ....	74
Eliminar un paseo. ....	75
Cerrar sesión. ....	75



## INDICE DE FIGURAS:

Figura 1. Imágenes capturadas durante las pruebas de la aplicación Maps.me. ....	2
Figura 2. Imágenes capturadas durante las pruebas de la aplicación Just Draw It.....	3
Figura 3. Imágenes capturadas durante las pruebas de la aplicación Komoot.....	5
Figura 4. Ejemplo de cómo se ve la gestión de un sprint en una pizarra de Kanban ofrecida por Azure DevOps. Se puede ver el Sprint 7 correspondiente al proyecto. ....	9
Figura 5. Diagrama de casos de uso de la aplicación .....	26
Figura 6. Diagrama de secuencia del CU001. ....	34
Figura 7. Diagrama de secuencia del CU002. ....	35
Figura 8. Diagrama de secuencia del CU003. ....	35
Figura 9. Diagrama de secuencia del CU004. ....	36
Figura 10. Diagrama de secuencia del CU005. ....	36
Figura 11. Diagrama de secuencia del CU006. ....	37
Figura 12. Diagrama de secuencia del CU007. ....	38
Figura 13. Diagrama de secuencia del CU008. ....	38
Figura 14. Diagrama de secuencia del CU009. ....	39
Figura 15. Diagrama de secuencia del CU010. ....	39
Figura 16. Diagrama de secuencia del CU011. ....	40
Figura 17. Diagrama del modelo de dominio en análisis. ....	41
Figura 18. Modelo arquitectónico del sistema.....	43
Figura 19. Diagrama de despliegue. ....	44
Figura 20. Esquema del 'Two way data binding' <sup>[8]</sup> . ....	45
Figura 21. Diagrama patrón Data Mapper. ....	46
Figura 22. Esquema de la base de datos. ....	47
Figura 23. Bocetos de la interfaz de usuario. ....	48
Figura 24. Logotipos de la aplicación. ....	48
Figura 25. Diagrama de clases de diseño. Modules and uses style.....	49
Figura 26. Clases detalladas de la capa de lógica de negocio. ....	50
Figura 27. Clases detalladas de la capa de servicios. ....	50
Figura 28. Vista de la pizarra de Kanban en el seguimiento del Sprint 14 del proyecto.	52
Figura 29. Estadísticas de la distribución de usuarios en las diferentes versiones de Android. ....	54
Figura 30. Expresión para adaptar el tiempo de duración del paseo. ....	55
Figura 31. Representación de isócronas en una demo ofrecida por Mapbox. ....	55
Figura 32. Pagina de inicio de sesión y de registro de usuario, en la creación de un nuevo usuario. ....	71

Figura 33. Pagina Home con mapa y búsqueda de localización.....	72
Figura 34. Pagina Home en la creación de un paseo, añadiendo el punto de inicio. ....	73
Figura 35. Pagina Home, ejemplo de la selección de ritmo y duración en la creación de un paseo. ....	73
Figura 36. Pagina Home, añadiendo punto de fin, duración, ritmo, generando opciones de paseo y seleccionando una de ellas. ....	74
Figura 37. Manu lateral, página de paseos guardados y de seguimiento.....	75
Figura 38. Pagina de seguimiento, confirmar eliminar un paseo.....	75
Figura 39. Pagina de ajustes, cerrar sesión. ....	76

## INDICE DE TABLAS:

Tabla 1. Tabla explicativa de la exposición a los riesgos, relacionando la Probabilidad y el impacto. ....	12
Tabla 2. Análisis del riesgo R1. ....	13
Tabla 3. Análisis del riesgo R2. ....	13
Tabla 4. Análisis del riesgo R3. ....	14
Tabla 5. Análisis del riesgo R4. ....	15
Tabla 6. Análisis del riesgo R5. ....	15
Tabla 7. Análisis del riesgo R6. ....	16
Tabla 8. Análisis del riesgo R7. ....	16
Tabla 9. Análisis del riesgo R8. ....	17
Tabla 10. Análisis del riesgo R9. ....	17
Tabla 11. Análisis del riesgo R10. ....	18
Tabla 12. Análisis del riesgo R11. ....	18
Tabla 13. Análisis del riesgo R12. ....	19
Tabla 14. Requisitos funcionales. ....	24
Tabla 15. Requisitos de información. ....	24
Tabla 16. Tabla de requisitos no funcionales. ....	25
Tabla 17. Descripción CU-001.....	27
Tabla 18. Descripción CU-002.....	27
Tabla 19. Descripción CU-003.....	28
Tabla 20. Descripción CU-004.....	28
Tabla 21. Descripción CU-005.....	29
Tabla 22. Descripción CU-006.....	30
Tabla 23. Descripción CU-006.....	30
Tabla 24. Descripción CU-009.....	31
Tabla 25. Descripción CU-010.....	32
Tabla 26. Descripción CU-011.....	32
Tabla 27. Descripción CU-012.....	33
Tabla 28. Clasificación de los resultados obtenidos en las pruebas. ....	59
Tabla 29. Test 1, crear usuario. ....	60
Tabla 30. Test 2, iniciar sesión.....	60
Tabla 31. Test 3, cerrar sesión.....	60
Tabla 32. Test 4, recuperar contraseña.....	61
Tabla 33. Test 5, buscar localización. ....	61

Tabla 34. Test 6, ver mapa.....	62
Tabla 35. Test 7, geolocalizar usuario. ....	62
Tabla 36. Test 8, crear paseo.....	62
Tabla 37. Test 9, guardar paseo. ....	63
Tabla 38. Test 10, seguir paseo. ....	63
Tabla 39. Test 11, eliminar paseo.....	64

# Capítulo 1: Introducción y contexto.

Este capítulo se va a utilizar para explicar el contexto en el que se ha desarrollado el proyecto, su motivación y analizar el mercado actual para ver si la aplicación que se quiere desarrollar puede encajar en él.

## 1.1 Contexto y motivación.

Este proyecto se ha desarrollado como TFG del grado en ingeniería informática, donde el alumno ha decidido desarrollar un proyecto propuesto por una profesora. Concretamente la idea del proyecto quería dar solución a un problema que afectaba a dicha profesora y que sirvió como idea para una aplicación. La funcionalidad básica de esta aplicación consiste en el diseño de rutas adaptando el tiempo y el ritmo para que al realizarlas se ajustara lo más posible al tiempo deseado.

Un ejemplo de situación en la que usar esta aplicación sería tener que esperar durante un tiempo en una ciudad desconocida y querer dar un paseo que se ajuste al tiempo disponible, por ejemplo, si tuvieras media hora de tiempo libre antes de coger un bus. Pues con este software lo que se quiere conseguir es un sistema que te ayude a generar una ruta que se adecue a tus necesidades y que te ayude a seguir dicha ruta.

Además, la propuesta de proyecto cuadraba a la perfección con las características buscadas por el alumno. Ya que este buscaba profundizar en el desarrollo de aplicaciones para dispositivos móviles y sobre todo tenía la intención de trabajar con mapas y aprender como funcionaban este tipo de tecnologías.

## 1.2 Objetivos.

El objetivo de este proyecto consiste en el desarrollo de una aplicación Android con la que poder crear y gestionar rutas, que se adapten a sus necesidades de tiempo y de ritmo. Para ello, el proyecto estará formado de dos partes que se cumplimentarán entre sí.

Por un lado, tenemos el desarrollo de la aplicación en el que se hará uso de los conocimientos técnicos adquiridos durante el estudio del grado en ingeniería informática junto con una serie de nuevas técnicas que nos permitirán llegar al objetivo de desarrollar la aplicación.

Y por otro lado tenemos la creación del documento de la memoria técnica, en la que se plasmará todo lo necesario para comprender el desarrollo de la aplicación. En ella se puede encontrar análisis, diseño, posibles riesgos, explicaciones de las decisiones de implementación, etc. Con la lectura de este documento se debería de ser capaz de entender el funcionamiento de la aplicación, tanto a nivel

técnico, como de usuario. Permitiendo así hacer uso de ella y para aquellas personas con un cierto conocimiento técnico, ampliar la funcionalidad y seguir con el desarrollo.

### 1.3 Estudio de mercado.

En este apartado se analizarán las opciones existentes para realizar funciones similares a las de la aplicación que se quiere desarrollar. Del mismo modo se hará un estudio de cuál es el público objetivo de esta aplicación con el fin de ver cuál sería el lugar que ocuparía está en el mercado.

#### 1.3.1 Estudio de aplicaciones del mercado.

A continuación, se analizarán una serie de aplicaciones ya disponibles en el mercado, concretamente analizaremos opciones disponibles en Google Play. El análisis se hará con el fin de analizar las diferentes opciones existentes y ver implementaciones con funcionalidades similares a las que dispondrá nuestra aplicación.

#### Maps.me:

Se trata de una aplicación de mapas con funciones que yo consideraría básicas de una aplicación de este tipo, como puede ser la búsqueda de lugares de interés, búsqueda de direcciones, creación de rutas con puntos de inicio, final e intermedios y la descarga de mapas entre alguna otra.

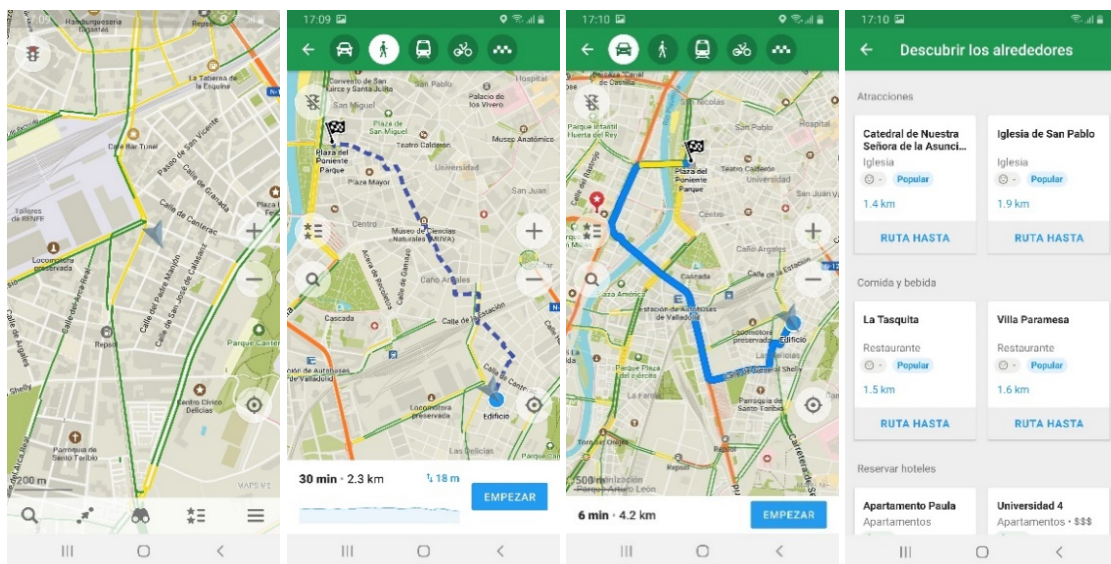


Figura 1. Imágenes capturadas durante las pruebas de la aplicación Maps.me.

#### Fortalezas de la aplicación:

- Esta aplicación permite la descarga de mapas de forma gratuita, además, esta descarga es bastante ligera y el uso de los mapas una vez descargados tiene muy buen rendimiento.
- La creación de rutas una vez sabes utilizarla es bastante rápida y cómoda ya que con mantener pulsado un punto del mapa puedes directamente usarlo como punto de inicio, del mismo modo con el punto final y los puntos de paso de la ruta.
- Uso de Open Street Map<sup>[1]</sup> para la gestión de los mapas, lo cual es una gran ventaja ya que se trata de un proyecto para la creación de mapas gratuito y que cuenta con una gran comunidad.
- Al hacer las pruebas, el rendimiento en términos de tiempos de respuesta ha sido bastante satisfactorio ya que a pesar de usar un móvil relativamente antiguo todo ha funcionado de forma fluida.



### Limitaciones de la aplicación:

- Las funcionalidades con las que cuenta son básicas comparado con otras aplicaciones populares como Google maps, y en cuanto a la funcionalidad básica de nuestro proyecto, no la cubre, pues no permite crear rutas adaptándose a un tiempo.
- Los atajos para la creación de las rutas no son intuitivos y tampoco hay ningún tutorial ni manual que lo explique, por lo que a ciertas personas con poca experiencia en el uso de móvil les puede ser complejo su uso.
- Por defecto no cuenta con la opción de mapa con vista de satélite que para mí personalmente es la más cómoda.

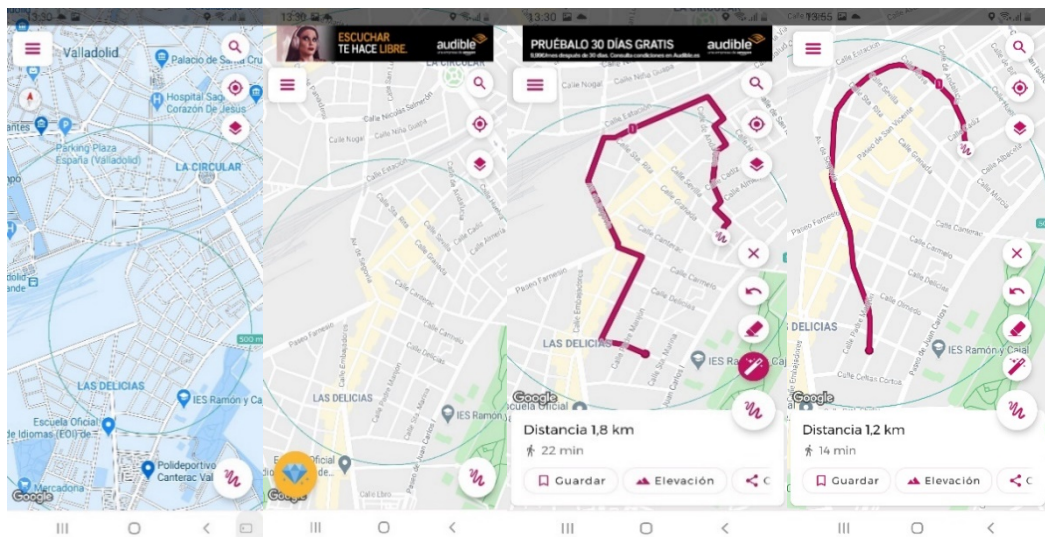
### Puntuación y valoración de la comunidad de usuarios en Google play:

La puntuación de esta aplicación es de 4'6 estrellas sobre 5 y teniendo en cuenta que han votado más de un millón doscientos mil usuarios, me parece una valoración bastante buena.

En cuanto a las valoraciones de los usuarios se puede ver que muchos de ellos están contentos con la aplicación y sus funcionalidades, pero en cuanto a lo negativo se repiten principalmente tres temas de queja, que son: la ausencia de indicaciones de voz cuando se realiza una ruta, problemas al descargar los mapas y ausencia o fallos a la hora de buscar o localizar ciertos puntos.

### **Just Draw It:**

Esta aplicación tiene un enfoque similar al de la aplicación que quiero desarrollar ya que la funcionalidad básica de esta es el diseño de paseos, concretamente esta aplicación lo que te permite es preparar tus paseos con la simple acción de dibujar en el mapa por donde quieres pasear. Cuenta con una versión de pago con la que aumentan sus funcionalidades permitiendo la exportación de las rutas en formato '.gpx', Soporte para mapas OpenStreetMap y la ausencia de límites a la hora de realizar búsquedas y compartir rutas.



**Figura 2.** Imágenes capturadas durante las pruebas de la aplicación Just Draw It.

### Fortalezas de la aplicación:

- La versión gratuita permite personalizar la interfaz de los mapas pudiendo elegir la que más cómoda o agradable resulte para el usuario.
- La versión de pago se puede conseguir a través de un pago único en vez de a través de una tarifa periódica.
- El hecho de poder preparar una ruta con el simple gesto de dibujar cual se desea que sea el recorrido resulta fácil e intuitivo, además no es necesario dibujar exactamente por donde se va a caminar ya que la propia aplicación es capaz de corregir el recorrido, de manera que no trazas el recorrido exactamente por encima de una calle, camino o carretera transitable es capaz de corregirlo.
- Permite compartir rutas creadas con otros usuarios.

#### Limitaciones de la aplicación:

- Alguna función interesante solo se puede acceder a través de la versión de pago.
- Solo permite la creación de rutas con el método de dibujar sobre el mapa, no permite establecer un punto de inicio y uno de final ni la búsqueda de puntos concretos para la creación de rutas.

#### Puntuación y valoración de la comunidad de usuarios en Google Play:

La puntuación de esta aplicación es de 4'4 estrellas sobre 5 por lo que parece una buena puntuación, aunque únicamente han valorado la aplicación unas tres mil personas.

En las valoraciones los usuarios la tendencia general es de agrado con la aplicación, destacando su buen funcionamiento y la facilidad de uso.

#### **Komoot:**

Se trata de una aplicación cuya finalidad principal es el diseño de rutas y la monitorización de estas durante su realización. A demás de esto también cuenta con opciones de red social en la que se puede añadir amigos, comunicarse con ellos y compartir la actividad. Como información adicional esta aplicación usa MapBox para la gestión de mapas que como ya hemos comentado se trata de una ventaja.

#### Fortalezas de la aplicación:

- Permite la conexión de la aplicación a través de bluetooth con una serie de dispositivo como bicis eléctricas, relojes inteligentes, ordenadores para bicicleta, etc.
- La interfaz resulta intuitiva y aporta información sobre cómo se deben de hacer ciertas acciones. En general resulta sencillo la creación de nuevas rutas.
- Permite adaptar las rutas teniendo en cuenta tu estado de forma, dependiendo del estado de forma que aportes, la aplicación adapta la velocidad media con la que se va a realizar la ruta.
- La aplicación dispone de un apartado de monitorización de la ruta.
- Para cada ruta aporta una ficha técnica muy completa con información como desnivel, tiempo, tipo de superficie por la que pasa la ruta, meteorología (se necesita versión premium).

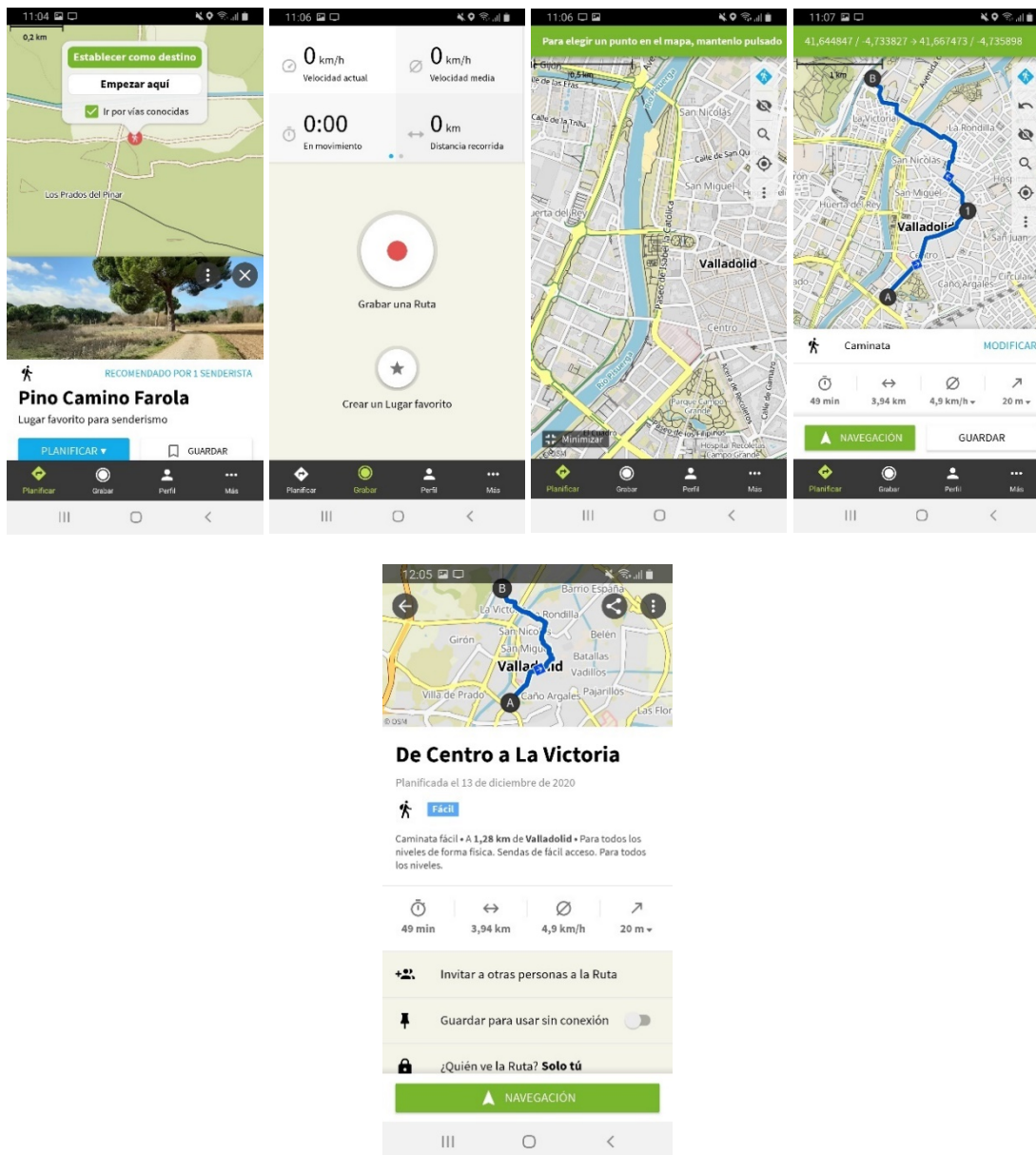


Figura 3. Imágenes capturadas durante las pruebas de la aplicación Komoot.

Limitaciones de la aplicación:

- No permite la edición del estilo del mapa.

Puntuación y valoración de la comunidad de usuarios en Google play:

La puntuación de la aplicación es de 4'6 puntos sobre 5 lo cual me parece una buena puntuación teniendo en cuenta que han votado casi dos cientos mil usuarios.

Sobre las valoraciones sí que se puede ver en este caso alguna en concreto de usuarios a los que no les funciona correctamente, pero en general parece que es todo lo contrario, de hecho, muchos de los usuarios que han añadido una valoración destacan funcionalidades de esta aplicación como la ficha técnica relativa a la ruta o la existencia de rutas que otras aplicaciones como Google Maps no tiene en cuenta. Por otro lado, también hay muchas quejas de que el precio de la versión premium es elevado para las funcionalidades que te ofrece.

### 1.3.2 Público objetivo y necesidades de este.

El público al que va dirigida esta aplicación es principalmente aquellas personas que desean dar un paseo y por alguna razón tienen tiempo de sobra y deciden usar un software que les ayude, con esta aplicación podrán crear variaciones de la ruta sin que esta sea necesariamente por el camino más corto, pudiendo elegir el tiempo en el que desean completar la ruta.

La intención con esta aplicación es que no sea necesario un control avanzado de dispositivos móvil, si no que cualquier persona que sepa manejar mínimamente un teléfono inteligente y sea capaz de usar las funciones básicas de aplicaciones de mapas pueda también hacer uso de las funciones más básicas de la aplicación.

En cuanto al sesgo de edad, esta aplicación no está orientada a un rango de edades en particular, aunque lógicamente hay que tener en cuenta que se trata de una aplicación para la generación de rutas por lo que su público comenzará normalmente con una edad mínima de unos 15-16 años ya que antes considero que lo normal no es hacer rutas solo, sino acompañado de algún adulto responsable.

Por otro lado, ha de considerarse que esta aplicación no está orientada para gente que hace rutas de forma muy exigente o incluso profesional, sino que está orientado más hacia aquellas personas que de forma ocasional deciden dar un paseo y por no hacer el recorrido de siempre usan la aplicación para generar la ruta a realizar.

### 1.4 Estructura de la memoria.

La organización del contenido del documento de la memoria va a ser el siguiente:

**Capítulo 1: Introducción y contexto.** Este capítulo se trata de una presentación del proyecto, el problema que va a cubrir, los objetivos y un estudio de soluciones similares.

**Capítulo 2: Planificación del proyecto.** En este capítulo se desarrolla la metodología de trabajo utilizada, la planificación de las actividades a realizar y se hace un seguimiento de cuales han sido los pasos seguidos en el desarrollo del proyecto.

**Capítulo 3: Análisis.** Se realiza un estudio de las características que debe tener el sistema a desarrollar en el proyecto y como se desea que funcione.

**Capítulo 4: Diseño.** Extiende el trabajo realizado en el capítulo anterior, adaptando lo obtenido en el análisis a un nivel más cercano a la implementación. También se explica la arquitectura utilizada por el sistema, la estructura de los datos, patrones de diseño software, etc.

**Capítulo 5: Implementación.** Se trata de un capítulo puramente técnico en el que se explican las particularidades que han acontecido durante la implementación del sistema. Se indagará más en las tecnologías y herramientas utilizadas.

**Capítulo 6: Pruebas.** Capítulo en el que se reflejan los ensayos hechos a la aplicación y los resultados obtenidos para comprobar que funciona correctamente.

**Capítulo 7: Conclusiones y trabajo futuro.** Resumen de lo obtenido durante el desarrollo del proyecto y las posibles mejoras que podrían implementarse en esta.

## Capítulo 2: Planificación del proyecto.

Capítulo dedicado a todo lo relacionado con la planificación del proyecto. Se explicará la metodología utilizada para trabajar, se realizará un análisis de los posibles riesgos relacionados con el proyecto, se estimará el presupuesto necesario para la realización del proyecto y por último se hará un seguimiento del trabajo realizado.

### 2.1 Metodología de trabajo (Scrum).

Para el desarrollo del proyecto se ha seguido en todo momento una metodología ágil, en la que se han adaptado el marco de trabajo Scrum a las circunstancias de personal y disponibilidad de este proyecto. Más concretamente en este proyecto ha sido imprescindible adaptar la distribución de los roles, porque al tratarse de un TFG el proyecto lo han formado dos personas, entre los que había que repartir dichos roles. Por tanto, también ha sido importante adaptarse a la disponibilidad ya que se ha compaginado el desarrollo del TFG con un trabajo a jornada completa.

#### 2.1.1 Motivos para la elección de Scrum.

Para la planificación del proyecto se ha hecho uso de metodologías ágiles, concretamente del marco de desarrollo Scrum. Los motivos<sup>[2]</sup> por los que se ha tomado la decisión de usar esta metodología son los siguientes:

- Una de las características básicas de Scrum es la realización de entregas periódicas y parciales, lo que permite ver de forma más clara el avance del proyecto y ayuda al equipo a mantenerse motivado, ya que con cada entrega se puede ver como el proyecto avanza.
- Scrum nos va a permitir evitar confusiones con los clientes, ya que periódicamente van a tener acceso a entregas incrementales del producto en las que paulatinamente aumenta su funcionalidad y les permite decidir si la dirección tomada por el proyecto es la correcta o no.
- Cuando el proyecto dio comienzo los requisitos no estaban definidos en su totalidad, sino que se irían adaptando a medida que avanzara el proyecto. Característica que encaja muy bien con la flexibilidad que otorga Scrum a la hora de ir redefiniendo y ajustando los requisitos.

#### 2.1.2 Artefactos Scrum usados en el proyecto.

Scrum está compuesto por una serie de artefactos<sup>[3]</sup> que se utilizan para el correcto desempeño del marco de trabajo, estos artefactos son:

- **Sprint:** Es el periodo de tiempo predefinido por el equipo en el que se realiza la entrega de un incremento del proyecto. Normalmente el tiempo de un sprint se define rondando una duración de entre 2 a 4 semanas, pero dada la situación en la que se encontraba el equipo de desarrollo (el alumno) en este proyecto no se ha establecido una duración fija si no que se ha adaptado la

duración a la disponibilidad del equipo, quedando como resultado **sprints** que tenían una duración de en torno a 20 horas de trabajo.

- **Backlog:** Lista dinámica compuesta de los requisitos y actividades que el equipo de desarrollo debe desempeñar para la realización del proyecto. Dentro del **backlog** se clasifican los elementos dependiendo de su criticidad, su prioridad, si se tratan de un bug, de una nueva funcionalidad, etc.
- **Sprint Backlog:** Lista formada a partir de elementos del **backlog** que conforma los requisitos y actividades que se deben cumplir durante un **sprint**.
- **Incrementos:** Fracción del proyecto que se ha desempeñado durante la actividad de un **sprint** y que se entrega como muestra del avance.

### 2.1.3 Roles Scrum usados en el proyecto.

Un equipo Scrum normalmente está compuesto por un grupo de personas de entre cinco y doce personas, cada una de estas personas desempeña un rol <sup>[4]</sup> diferente que se centra en unas responsabilidades concretas, con el fin común que es sacar a flote el proyecto. En este caso en concreto hemos tenido que adaptar estos roles a un equipo formado de dos personas, un alumno haciendo su trabajo de fin de grado y su tutora.

A continuación, se explica el reparto de los roles y las responsabilidades correspondientes a cada uno de ellos:

- **Product Owner:** Se trata del encargado de gestionar los requisitos a medida que avanza el proyecto manteniendo el **backlog** estructurado, priorizado y detallado. Además, tiene que entender a la perfección cuáles son los objetivos del proyecto y cuáles son los pasos a seguir en cada **sprint**. En el proyecto que nos ocupa este rol le pertenece a la profesora encargada de tutorizar el TFG, ya que es ella quien tiene la idea final de cómo tiene que ser la aplicación.
- **Scrum Master:** Es el encargado de que la metodología Scrum se lleve a cabo correctamente, para ello tiene que encargarse de eliminar todos aquellos contratiempos que aparezcan en el camino, con el fin de que en el proceso de scrum se sigan haciendo entregas de valor con cada iteración. Este rol será tomado principalmente por el alumno, pero no es un rol que desempeñe exclusivamente una persona por lo que será apoyado por el tutor cuando este lo considere.
- **Equipo de desarrollo:** Es el grupo de personas que siguiendo la metodología Scrum se encarga de desarrollar el proyecto, autogestionándose para realizar una entrega cada vez que sea necesario. El alumno será el encargado de desempeñar este rol en su totalidad.

### 2.1.4 Eventos Scrum utilizados en el proyecto.

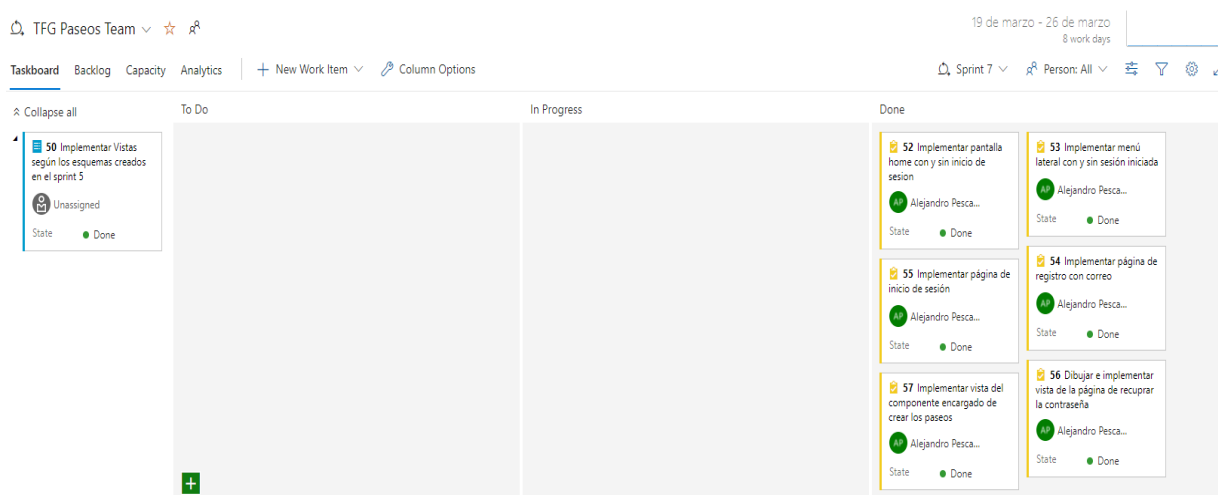
Scrum define cinco eventos básicos con los que apoyar al proyecto a lograr su éxito, entre estos eventos encontramos los Sprint Planning, los Daily Scrum, Sprint Review y reunión de refinamiento.

- **Sprint Planning:** Se trata de una reunión que se realiza antes de comenzar cada sprint en el que participan tanto el tutor como el alumno (roles de equipo de desarrollo, scrum master y Project Owner). En esta reunión se planifica el trabajo que se va a realizar en el sprint, siempre teniendo en mente que al final de este se debe haber una entrega parcial.
- **Daily Scrum:** Es una reunión que se realiza de forma diaria en la que participa el alumno (con el rol de equipo de desarrollo) y cuya finalidad es aclarar que hizo el día anterior para contribuir a las metas del sprint, que va a hacer en el día y si tiene algún impedimento para lograr la entrega. En este caso no se trata de una reunión como tal ya que solo participa una persona, pero antes de empezar con el trabajo cada día el alumno ha de tomarse unos minutos para pensar en estos tres aspectos.

- **Sprint review:** Reunión que se celebra al finalizar el sprint en la que se hace un análisis del incremento creado en el sprint. En esta reunión participan tanto alumno como tutor (con los roles de equipo de desarrollo y Product Owner). En esta implementación concreta de scrum el Sprint Review se ha celebrado junto con el Sprint planning, de forma que en una misma sesión alumno y tutor revisaban el trabajo realizado durante el sprint y posteriormente planificarán el siguiente.
- **Refinamiento:** Reunión en la que participan todos los implicados en el proyecto en la que se hace una revisión y filtrado del **Backlog**, consiguiendo un **backlog** más limpio y centrado en el objetivo del proyecto.

### 2.1.5 Herramientas usadas para apoyar Scrum (Azure DevOps).

Para llevar la gestión del **backlog** y todo lo relacionado con el proyecto se ha hecho uso de la herramienta proporcionada de forma gratuita por Microsoft llamada **Azure DevOps**. Se trata de una herramienta web con la que se puede llevar un seguimiento sencillo y visual de todas las actividades realizadas por el equipo de desarrollo, mantener un seguimiento actualizado de los sprints, el estado de cada tarea, asignación de estas, etc. Y todo esto a través de una interfaz de usuario fácil e intuitiva.



**Figura 4.** Ejemplo de cómo se ve la gestión de un sprint en una pizarra de Kanban ofrecida por Azure DevOps. Se puede ver el Sprint 7 correspondiente al proyecto.

### 2.1.6 Seguimiento de Scrum en el desarrollo del proyecto.

A continuación, se mostrarán las actividades realizadas en cada uno de los 16 sprints que han sido necesarios para desempeñar este proyecto:

#### **SPRINT 1** (01/01/2021-24/01/2021):

- Inicio de la memoria: incluyendo actividades como preparar el formato del documento, planificación del proyecto, primera versión de requisitos, análisis del mercado.
- Estudio de las tecnologías a utilizar en el proyecto: Estudiar funcionamiento de Mapbox, de las APIs rest de Mapbox y las bases de Ionic y su funcionamiento.
- Instalación de todo el software necesario para la realización del proyecto: Ionic, Visual Studio Code, Android Studio, etc.

#### **SPRINT 2** (25/01/2021-01/02/2021):

- Inicio de la aplicación con la creación de una pantalla en la que aparezca un mapa centrado en la localización del dispositivo.

**SPRINT 3 (01/02/2021-08/02/2021):**

- Estudiar la manera en la que se puede implementar Firebase en el proyecto, ver si es viable y en caso de que lo sea, estudiar su funcionamiento.
- Implementar cabecera en la página del mapa con un botón que abra un menú lateral con el que navegar entre las páginas de la aplicación.

**SPRINT 4 (08/03/2021-19/02/2021):**

- Implementar servicio de registro de usuario, inicio de sesión y cerrado de sesión con Firebase Authentication.

**SPRINT 5 (19/02/2021-07/03/2021):**

- Crear esquema de la base de datos e implementar métodos básicos para la utilización de esta.
- Crear bocetos de las pantallas que van a formar la aplicación.

**SPRINT 6 (08/03/2021-15/03/2021):**

- Implementar método para la obtención de información de Firestore.

**SPRINT 7 (16/03/2021-26/03/2021):**

- Implementación de las vistas de la aplicación siguiendo los bocetos creados en el sprint 5.

**SPRINT 8 (27/03/2021-23/04/2021):**

- Comenzar con la funcionalidad de creación de paseo con actividades como, la búsqueda de un punto según un texto identificativo de la localización y diseñar algoritmo para ajustar un paseo al tiempo.
- Ajustes en las vistas implementadas en el sprint 7.

**SPRINT 9 (24/04/2021-30/04/2021):**

- Escribir en la memoria información sobre la metodología de trabajo utilizada (SCRUM).
- Continuar con la creación de paseos. Implementar el algoritmo para que se obtengan los paseos, dibujar los paseos en el mapa e introducir comprobaciones para que todos los datos necesarios para la realización del paseo estén disponibles.

**SPRINT 10 (01/05/2021-07/01/2021):**

- Implementar mecanismo para la limpieza del mapa en el proceso de creación de paseo.
- Escribir en la memoria el análisis realizado de la aplicación. Estudio de roles y de los actores del sistema, actualizar requisitos, diagrama de casos de uso y Modelo de dominio.

**SPRINT 11 (07/05/2021-14/05/2021):**

- Implementar mecanismo para informar al usuario del tiempo mínimo para crear un paseo entre dos puntos.
- Adaptar paseo al ritmo introducido por el usuario.



- Mostrar paseos obtenidos como una lista al usuario.
- Describir la arquitectura del sistema en la memoria.
- Terminar la parte de análisis en la memoria con el modelo de clases inicial y el modelo de dominio.

**SPRINT 12** (14/05/2021-23/05/2021):

- Sprint dedicado a la memoria en su totalidad. Repasar el capítulo de análisis y acabar el capítulo de diseño.

**SPRINT 13** (23/05/2021-06/06/2021):

- Arreglar bug con la vista del mapa aparecido en el sprint anterior.
- Actualizar requisitos en el capítulo de Análisis.
- Escribir capítulo de conclusiones y trabajo futuro en la memoria.
- Guardar paseos creados en la base de datos.

**SPRINT 14** (07/06/2021-12/06/2021):

- Implementación de página de términos y condiciones de uso.
- Repasar lo que está escrito en la memoria.
- Arreglar bug de registro de usuario, dejó de funcionar.
- Implementar funcionalidad de la página de paseos guardados.

**SPRINT 15** (13/06/2021-19/06/2021):

- Escribir términos y condiciones de usuario.
- Implementar página de seguimiento de paseo y borrado de paseos.
- Escribir capítulo de implementación en la memoria.

**SPRINT 16 FINAL** (10/06/2021-28/06/2021):

- Escribir capítulo de pruebas en la memoria.
- Revisar memoria.
- Escribir Anexos y bibliografía en la memoria.
- Repasar el código limando detalles vistos por los usuarios que han probado la aplicación, limpiando el código de depuración e introducción de los logotipos de la aplicación.
- Generar archivo '.apk'.

## 2.2 Plan de gestión de riesgos.

Para la realización del plan de gestión de riesgos se va a seguir las indicaciones del libro **Software Project Management** <sup>[5]</sup>, según el cual se considera que los riesgos son aquellos eventos o condiciones que en el caso de materializarse provocarían un efecto con respecto a los objetivos definidos para el proyecto, siempre teniendo en cuenta que los riesgos hacen referencia a los posibles problemas del futuro, no a los problemas actuales y que siempre están compuestos de una causa y un efecto.

Para la realización de este proceso se llevarán a cabo cuatro actividades:

- Identificación y definición de los riesgos.
- Análisis de los riesgos y su impacto.
- Diseño de un plan de control de los riesgos en caso de que estos llegaran a materializarse.

- Realización de un seguimiento de los riesgos definidos durante la realización del proyecto.

A la hora de la realización del plan de gestión riesgos hay que tener claro que es el **impacto**, esto es el conjunto de consecuencias que conlleva la materialización de un riesgo, siendo importante clasificar los riesgos teniendo en cuenta este concepto, lo que nos permite controlar cuáles serán las consecuencias de cada uno de los riesgos sobre el desarrollo del proyecto. En esta clasificación haremos una distinción en cuatro niveles, de menos a mayor: insignificante, moderado, preocupante y crítico. Donde un riesgo insignificante significa que las consecuencias sobre la realización del proyecto son prácticamente inexistentes, es decir, que no afecta al desarrollo del proyecto, y por el contrario un riesgo crítico puede implicar el retraso de la realización de alguna de las actividades o incluso la modificación de la planificación.

También es importante tener en cuenta la probabilidad y la frecuencia con la que pueden materializarse estos riesgos, ya que, aunque un riesgo sea de impacto insignificante o moderado, si este tiene una alta probabilidad de aparecer obviamente afectará de una manera importante al desempeño del proyecto llegando incluso a afectar a decisiones de planificación.

Relacionando los dos parámetros anteriores (impacto y probabilidad) aparece una nueva forma de clasificar los riesgos que es la exposición al riesgo. Para ver de forma más clara cómo funciona la exposición se analiza en la **tabla 1**. La exposición se dividirá en tres categorías alta, media, y baja.

Impacto Probabilidad	Insignificante	Moderado	Preocupante	Crítico
<25%	BAJA	BAJA	MEDIA	MEDIA
>25% y <50%	BAJA	MEDIA	MEDIA	ALTA
>50% y <75%	MEDIA	MEDIA	ALTA	ALTA
>75%	MEDIA	ALTA	ALTA	ALTA

**Tabla 1.** Tabla explicativa de la exposición a los riesgos, relacionando la Probabilidad y el impacto.

Otro aspecto a tener en cuenta es la necesidad de tener un plan premeditado para poder evitar los riesgos, o bien una vez que ya no se pueden evitar su aparición, combatirlos para que causen el mínimo daño posible a la realización el proyecto. Para ello hay que crear un plan de mitigación y de contingencia para cada posible riesgo.

### 2.2.1 Identificación, plan de contingencia y mitigación.

En este apartado se identificarán y analizarán los posibles riesgos concretos que pueden afectar a este proyecto, para ello se hará una tabla para cada uno de ellos utilizando los siguientes parámetros.

- Identificador: Se trata de un identificador único asociado a cada riesgo.
- Descripción: Breve explicación sobre el riesgo al que nos referimos.
- Consecuencias: Implicaciones sobre el proyecto.
- Criticidad
- Probabilidad
- Exposición
- Plan de contingencia: Medidas que permitirán evitar la materialización el riesgo.
- Plan de mitigación: Plan a realizar una vez el riesgo se haya materializado.

A continuación, se pueden ver las tablas donde se ha hecho el análisis de los riesgos, donde se ha hablado de trabajadores y clientes pero que obviamente al tratarse de un TFG estos actores representan al alumno y al tutor respectivamente.

<b>Identificador</b>	<b>R1 – Imprevistos del personal.</b>		
<b>Descripción</b>	Situaciones no previstas como problemas de salud, problemas personales, ausencias justificadas, problemas laborales, etc.		
<b>Consecuencias</b>	Periodos de tiempo en los que la producción del personal es menor o incluso ausente, provocando retrasos en la realización de las actividades según la planificación.		
<b>Impacto</b>	Moderado.	<b>Probabilidad</b>	40%
<b>Exposición</b>	Media.		
<b>Plan de Contingencia</b>	Planificar las actividades teniendo en cuenta un posible margen con el que poder reasignar los tiempos, tener un claro conocimiento de cuáles son las actividades críticas de cara a la realización del proyecto y seguir de forma estricta los tiempos de realización planificados.		
<b>Plan de mitigación</b>	Reconstruir la planificación teniendo en cuenta lo ocurrido (utilizando fines de semana y festivos si fuera necesario), dar prioridad a aquellas actividades más importantes.		

**Tabla 2.** Análisis del riesgo R1.

<b>Identificador</b>	<b>R2- Fallo del servidor externo.</b>		
<b>Descripción</b>	La aplicación va a hacer uso de API rests por lo que depende de que los servidores en los que se ejecuta estén en funcionamiento.		
<b>Consecuencias</b>	No se podrá trabajar en el desarrollo de ciertos requisitos mientras que el servidor no funcione correctamente.		
<b>Impacto</b>	Crítico.	<b>Probabilidad</b>	10%
<b>Exposición</b>	Media.		
<b>Plan de Contingencia</b>	No es algo que dependa directamente del equipo por lo que antes de elegir qué servicio usar se debería estudiar bien cuál de los posibles servicios va a dar mejor funcionamiento.		
<b>Plan de mitigación</b>	Será necesario realizar una revisión de la planificación, reasignando las tareas de forma que mientras dure la caída del servidor se pueden hacer otras tareas.		

**Tabla 3.** Análisis del riesgo R2.

Identificador	R3 – Perdida de trabajo realizado		
Descripción	Perdida del trabajo realizado durante un periodo de tiempo más o menos grande.		
Consecuencias	Es necesario repetir el trabajo ya realizado lo que se traduce en una pérdida de tiempo y desajuste de la planificación.		
Impacto	Preocupante.	Probabilidad	70%
Exposición	Alta.		
Plan de Contingencia	<p>Fundamental el uso de sistemas de control de versiones como git o cualquiera de sus implementaciones.</p> <p>Durante el desarrollo del trabajo seguir de forma estricta un protocolo mediante el cual se guarde el trabajo cada cierto tiempo de trabajo.</p> <p>Documentar de forma clara el trabajo realizado.</p> <p>Trabajar haciendo uso de alguna plataforma que permita almacenar el trabajo en la nube (Onedrive, Google drive, mega, etc.)</p>		
Plan de mitigación	<p>Recuperar versiones anteriores guardadas en el sistema de control de versiones.</p> <p>En el caso de no tener versiones antiguas habrá que repetir el trabajo haciendo uso de la documentación creada.</p> <p>Implicará una reorganización de la planificación.</p>		

Tabla 4. Análisis del riesgo R3.

Identificador	R4 – Fallo de los sistemas de trabajo personales.		
Descripción	Problema técnico en herramienta de trabajo, como fallos de componentes, infecciones con virus, fallos de programas usados para trabajar, etc.		
Consecuencias	Esto provocará periodos de tiempo en los que no se pueda trabajar con el 100% de la productividad.		
Impacto	Preocupante.	Probabilidad	30%
Exposición	Media.		
Plan de Contingencia	<p>Mantener el sistema y sus programas actualizados.</p> <p>Tener en funcionamiento y actualizado un antivirus o sistema de seguridad.</p>		

	Instalar los programas necesarios para el trabajo en otro ordenador disponible en casas.
<b>Plan de mitigación</b>	Tratar de arreglar el problema en casa si fuera posible, tratando de no hacer crecer el presupuesto del proyecto.  Mientras dure las reparaciones hacer uso de otro equipo disponible en casa.

Tabla 5. Análisis del riesgo R4.

<b>Identificador</b>	<b>R5 – Desconocimiento de las tecnologías a utilizar.</b>		
<b>Descripción</b>	Hacer uso de tecnologías que los desarrolladores no conocen en profundidad o que sea la primera vez que trabajan con ella.		
<b>Consecuencias</b>	Retraso a la hora de avanzar con el trabajo.		
<b>Impacto</b>	Preocupante.	<b>Probabilidad</b>	70%
<b>Exposición</b>	Alta.		
<b>Plan de Contingencia</b>	Decidir en etapas tempranas del proyecto que tecnologías se van a utilizar para que los desarrolladores puedan prepararse.  Contar en la planificación con periodos de formación previos al uso de dichas tecnologías.		
<b>Plan de mitigación</b>	Actualizar la planificación.  Pedir ayuda a compañeros que tengan conocimientos de dichas tecnologías.		

Tabla 6. Análisis del riesgo R5.

<b>Identificador</b>	<b>R6 – Mala definición de los requisitos.</b>		
<b>Descripción</b>	Generar una descripción mala o insuficiente de algunos de los requisitos claves.		
<b>Consecuencias</b>	Los trabajadores del proyecto no conocen exactamente qué es lo que se tiene que implementar.		
<b>Impacto</b>	Crítico.	<b>Probabilidad</b>	20%
<b>Exposición</b>	Media.		

<b>Plan de Contingencia</b>	<p>Mantener tantas reuniones con el cliente como sean necesarias y asegurarse que lo que se ha entendido es exactamente lo que quiere el cliente.</p> <p>Hacer posibles revisiones periódicas del producto por parte del cliente para ver si se está haciendo lo que el realmente quiere.</p> <p>Uso de metodologías ágiles, de forma que se revise de forma periódica los planes del proyecto.</p>
<b>Plan de mitigación</b>	<p>Modificar la especificación de los requisitos.</p> <p>Revisar la planificación para cuadrar las nuevas especificaciones de los requisitos.</p>

Tabla 7. Análisis del riesgo R6.

<b>Identificador</b>	<b>R7 – Cambio de los requisitos o aparición de nuevos.</b>		
<b>Descripción</b>	Durante el desarrollo del proyecto el cliente o los trabajadores consideran necesario modificar los requisitos o añadir alguno nuevo.		
<b>Consecuencias</b>	Provoca confusión en los trabajadores y ralentización en el trabajo ya que es posible que haya que rehacer ciertos elementos y reorganizar la planificación.		
<b>Impacto</b>	Preocupante.	<b>Probabilidad</b>	30%
<b>Exposición</b>	Media.		
<b>Plan de Contingencia</b>	<p>Realizar una buena especificación de requisitos utilizando tantas reuniones con el cliente como sean necesarias y asegurarse de tener un buen feedback por parte del cliente.</p> <p>Mostrar el trabajo de forma periódica a los clientes para ver si lo que se está creando coincide con sus ideas.</p> <p>Uso de metodología ágil con periodos cortos de planificación(sprints), lo cual permite flexibilidad a la hora de ir marcando los requisitos ya que se repasan cada sprint.</p>		
<b>Plan de mitigación</b>	<p>Evaluar la viabilidad de la incorporación de los nuevos requisitos y priorizar aquellos que sean más importantes.</p> <p>Revisar la planificación teniendo en cuenta los nuevos requisitos.</p>		

Tabla 8. Análisis del riesgo R7.

<b>Identificador</b>	<b>R8 – Mal diseño.</b>		
<b>Descripción</b>	El diseño no cubre las necesidades especificadas en los requisitos.		
<b>Consecuencias</b>	La implementación del proyecto toma una dirección errónea en la que no se cumple con los objetivos.		
<b>Impacto</b>	Preocupante.	<b>Probabilidad</b>	20%
<b>Exposición</b>	Media.		
<b>Plan de Contingencia</b>	Tener claro las peticiones del cliente haciéndole saber lo que se va a implementar y asegurándose de que coincide con su idea del proyecto.		
<b>Plan de mitigación</b>	Revisión del diseño. Revisión de la planificación. Argumentar las decisiones tomadas ante el cliente para saber si es interesante para el cliente modificar su propuesta inicial		

Tabla 9. Análisis del riesgo R8.

<b>Identificador</b>	<b>R9 – Desajuste de la planificación.</b>		
<b>Descripción</b>	El avance del trabajo no coincide con el plan especificado desde el principio.		
<b>Consecuencias</b>	Si se adelanta lo que provoca es una relajación en el trabajador que hacer que se aproveche menos el tiempo. Si se retrase el trabajo provoca el aumento del estrés en el trabajador y probablemente la entrega tardía del trabajo.		
<b>Impacto</b>	Preocupante.	<b>Probabilidad</b>	50%
<b>Exposición</b>	Media.		
<b>Plan de Contingencia</b>	Analizar las tareas a realizar correctamente para tener una estimación precisa. Trabajar con disciplina para hacer que se cumpla el plan.		
<b>Plan de mitigación</b>	En caso de ir con retraso será necesario reajustar la planificación e incluso aumentar las horas de trabajo. En caso de adelantarse hay que tratar de revisar la planificación teniendo en cuenta el avance del trabajo y replantear las estimaciones.		

Tabla 10. Análisis del riesgo R9.

<b>Identificador</b>	<b>R10 – Problemas de comunicación entre cliente y trabajador.</b>		
<b>Descripción</b>	Falta de entendimiento entre cliente y trabajador e incluso desacuerdo en algún punto del proyecto.		
<b>Consecuencias</b>	Desmotivación y baja de la productividad del trabajador. Dudas a la hora de avanzar con el desarrollo.		
<b>Impacto</b>	Moderado.	<b>Probabilidad</b>	40%
<b>Exposición</b>	Media.		
<b>Plan de Contingencia</b>	Realizar de forma estricta y periódica reuniones en las que fluya la comunicación. Comunicar cualquier duda o petición existente sin miedo al reproche o la desaprobación.		
<b>Plan de mitigación</b>	Convocar una reunión en la que se trate como tema principal la comunicación entre cliente y trabajador, en la que cada uno exponga sus ideas y cómo ve la situación para tratar de llegar a un acuerdo y cambiar los métodos de comunicación.		

Tabla 11. Análisis del riesgo R10.

<b>Identificador</b>	<b>R11 – Falta de seguimiento del proyecto.</b>		
<b>Descripción</b>	No llevar a cabo un proceso de monitorización suficiente durante la realización del proyecto.		
<b>Consecuencias</b>	El proyecto toma un rumbo diferente el especificado en la planificación y la definición de los requisitos.		
<b>Impacto</b>	Preocupante.	<b>Probabilidad</b>	10%
<b>Exposición</b>	Media.		
<b>Plan de Contingencia</b>	Mantener revisiones periódicas en las que ver cómo está siendo el avance del proyecto. Uso de metodologías ágiles como SCRUM en la que se realizan reuniones diarias en las que se ve como avanza el trabajo.		
<b>Plan de mitigación</b>	Revisar el trabajo hecho hasta el momento en el que se ve el problema y revisar la planificación.		

Tabla 12. Análisis del riesgo R11.



<b>Identificador</b>	<b>R12 – Mal diseño.</b>		
<b>Descripción</b>	El diseño no se ajusta a las especificaciones requeridas por el cliente.		
<b>Consecuencias</b>	El producto que se obtiene no es el solicitado por el cliente o sí que lo es, pero no tiene el funcionamiento óptimo.		
<b>Impacto</b>	Preocupante.	<b>Probabilidad</b>	50%
<b>Exposición</b>	Media.		
<b>Plan de Contingencia</b>	Estudiar bien los requisitos solicitados por el cliente y ante cualquier duda acudir a ellos para que no haya mal entendidos. No precipitarse con la toma de decisiones de diseño.		
<b>Plan de mitigación</b>	Localizar los errores y repetir el diseño.		

Tabla 13. Análisis del riesgo R12.

### 2.2.2 Seguimiento y monitorización de los posibles riesgos.

Durante el desarrollo del proyecto se han materializado tres de los riesgos identificados en el apartado anterior, concretamente son los siguientes:

- R1 Improvistos personales. Al desarrollarse el proyecto al mismo tiempo que se trabajaba, ha habido días que teniendo planificado trabajar en el TFG, no se ha podido, lo que ha provocado que algunos sprints se hayan alargado más de lo esperado. Pero es algo que era esperable por lo que no ha resultado crítico y aplicando los planes de mitigación se ha solventado.

- R5 Desconocimiento de las tecnologías. Al igual que con el R1 esto ha provocado el retraso en la entrega de multitud de sprints. Para su mitigación como se ha indicado en el plan de gestión de riesgos lo que se ha hecho ha sido replanificar los sprints, priorizar aquellas tareas más críticas y dedicar tareas en los sprints más tempranos para estudio.

- R3 Pérdida del trabajo realizado. En el sprint 13 fue necesario actualizar una serie de paquetes utilizados en el desarrollo del proyecto, esto hizo que la aplicación dejara de funcionar. Gracias a las copias periódicas y al sistema de gestión de versiones se pudo recuperar el trabajo en menos de una hora.

### 2.3 Presupuesto.

En este apartado se hará un análisis estimado del coste del proyecto. Al tratarse de un TFG y tratarse de un proyecto que se desarrolla en un ámbito puramente académico, se va a realizar un análisis simulando que se tratara de un proyecto a realizar en un ámbito laboral lo más fiel a la realidad posible y por otro lado, se va a hacer un análisis más realista ajustando el presupuesto a las condiciones reales del proyecto.

### 2.3.1 Presupuesto simulado.

Lo primero que se va a analizar es el coste de la mano de obra. Teniendo en cuenta el salario base publicado en el BOE el 6 de marzo de 2018<sup>[6]</sup>, en el convenio colectivo de empresas de consultoría y teniendo en cuenta que la persona encargada de desarrollar el proyecto tiene una experiencia laboral inferior a tres años. Tenemos que el salario es de 7'71€/h. Analizando las horas de trabajo, tenemos que se van a cubrir las horas equivalentes a 12 créditos, lo cual nos da un total de 300 horas de trabajo por lo que si hacemos cuentas tenemos que la mano de obra tendrá un coste de **2.313€**.

Seguidamente analizamos los materiales que se utilizarán para trabajar en el proyecto. Para ello se necesitaría un ordenador con sus periféricos correspondientes a partir de 800€, un teléfono móvil para realizar las pruebas (200€) y material de oficina para apoyar en el trabajo (15€). En cuanto al material tenemos un gasto aproximado de **1.015€**.

También hay que ver cuál sería el coste asociado a pagar una estancia adecuada donde poder trabajar. Para ello tendría que contar con los servicios de internet, calefacción y electricidad cubiertos. El trabajo se va a desarrollar en la ciudad de Valladolid, por lo que para alquilar una casa con todo cubierto estaríamos hablando de un precio de unos 400€ al mes y teniendo en cuenta que para cumplir con las 300 horas de trabajo estimadas en el párrafo anterior se van a necesitar como mínimo dos meses si se realiza una jornada completa de ocho horas de trabajo diarias. Cubrir con el alojamiento y servicios de luz, agua, calefacción e internet tendría un coste de **800€**.

Por último, tendremos en cuenta los costes de las licencias software necesarias para el desarrollo del proyecto. La única licencia indispensable, será la de Astah profesional que tiene un precio de 7'50€ mensuales por lo que si contamos que el trabajo se desarrolla en dos meses tenemos un coste de licencias de **15€**.

Recapitulando los gastos comentados en el apartado tenemos que el proyecto requerirá un presupuesto estimado de **3.843€**.

### 2.3.2 Presupuesto real.

Al tratarse de un proyecto desarrollado en el contexto de un TFG muchos de los gastos comentados en el apartado anterior no tienen sentido.

Primeramente, el coste de la mano de obra se suprime ya que el proyecto ha sido desarrollado por un estudiante que necesitaba de su realización para finalizar su formación de grado, y en este caso concreto no ha recibido ningún tipo de ayuda o beca.

En cuanto a los gastos relacionados con los materiales son muy difíciles de calcular ya que el alumno ya dispone de un ordenador, los periféricos y material de oficina para la realización del TFG. A si que como no van a suponer ningún gasto nuevo se va a tomar como un gasto de 0€ en este aspecto.

En cuanto al alojamiento y los servicios de agua, internet, calefacción y electricidad. No se dispone de los datos exactos por que el alumno vive en casa de sus padres y no tiene acceso a ellos.

En lo que respecta a la licencia de Astah, es un gasto nulo ya que la Universidad de Valladolid proporciona una licencia de forma gratuita a sus alumnos.





## Capítulo 3: Análisis.

A continuación, se realizará el análisis del sistema. Para ello se hará un estudio del problema que quiere cubrir la aplicación realizando una especificación de requisitos, se identificarán los roles del sistema, se analizarán los posibles casos de uso y se creará un modelo de dominio relacionado con el problema.

### 3.1 Especificación de requisitos.

En este apartado se hará un análisis detallado de las necesidades que el sistema debe de satisfacer una vez esté desarrollado y puesto en producción.

#### 3.1.1 Requisitos funcionales.

A continuación, en la **tabla 14** se puede ver la especificación de requisitos funcionales.

ID	NOMBRE	DESCRIPCIÓN
RF-001	Registro de usuario.	El sistema permitirá a los usuarios registrarse con su correo en el sistema.
RF-002	Inicio de sesión.	El sistema requerirá que los usuarios a tengan sesión iniciada para poder hacer uso de este.
RF-003	Recuperar contraseña olvidada.	El sistema permitirá a los usuarios recuperar su contraseña si la han olvidado mediante el envío de un correo de recuperación.
RF-004	Cerrar sesión.	El sistema permitirá al usuario cerrar sesión en su cuenta.
RF-005	Creación de paseos.	El sistema permitirá al usuario generar un paseo ajustándose a una localización inicial, una localización final, un ritmo y un tiempo especificado por el usuario.
RF-006	Opciones de paseos.	El sistema proporcionará una lista de opciones de paseo en el proceso de creación de paseo siempre que sea posible.
RF-007	Almacenado de paseos.	El sistema permitirá almacenar los paseos creados por los usuarios y relacionados a su cuenta de usuario.
RF-008	Paseos en cualquier dispositivo.	El sistema permitirá acceder a los paseos asociados a una cuenta desde cualquier dispositivo en el que se inicie sesión.
RF-009	Eliminar paseos.	El sistema permitirá a los usuarios eliminar paseos.
RF-010	Ajustes de la aplicación.	El sistema permitirá a los usuarios almacenar ajustes personalizados en el sistema.

RF-011	Tiempo mínimo.	El sistema controlará que el usuario no genere un paseo con un tiempo menor que el de la ruta más rápida entre los puntos de inicio y final.
RF-012	Seguridad y privacidad.	El sistema debe garantizar la privacidad y seguridad de los datos de usuario.
RF-013	Términos de uso y condiciones	El sistema no podrá registrar un usuario si no acepta unos términos de uso y condiciones.
RF-014	Seguimiento del paseo	El sistema permitirá al usuario seguir un paseo mostrándole la ruta y su geolocalización en tiempo real.

Tabla 14. Requisitos funcionales.

### 3.1.2 Requisitos de información.

A continuación, en la **tabla 15** pueden verse los requisitos de información.

ID		
RI-001	Descripción	Almacenar información correspondiente a los usuarios del sistema.
	Datos	<ul style="list-style-type: none"> <li>- Id de usuario.</li> <li>- Nombre.</li> <li>- Correo.</li> <li>- Fecha de Creación.</li> <li>- Ultima conexión.</li> </ul>
RI-002	Descripción	Información correspondiente con los paseos generados por el sistema.
	Datos	<ul style="list-style-type: none"> <li>- Id del paseo.</li> <li>- Punto donde comienza el paseo.</li> <li>- Punto donde finaliza el paseo.</li> <li>- Duración del paseo.</li> <li>- Ritmo del paseo.</li> <li>- Conjunto de puntos por los que discurre el paseo.</li> <li>- Longitud del paseo.</li> </ul>

Tabla 15. Requisitos de información.

### 3.1.3 Requisitos no funcionales.

A continuación, en la **tabla 16** se puede ver la especificación de requisitos no funcionales.

ID	NOMBRE	DESCRIPCIÓN
RNF-001	Plataforma Android.	El sistema estará desarrollado para ser usado en sistema operativo Android.
RNF-002	GPS.	El sistema hará uso del GPS del dispositivo para geolocalizar al usuario.
RNF-003	Mapas de Mapbox.	El sistema usará Mapbox como sistema proveedor de mapas.
RNF-004	Autenticación con Firebase.	El sistema usará Firebase Authentication para la gestión de las cuentas de usuario.
RNF-005	Base de datos Firestore.	El sistema usará como base de datos Firebase Firestore.
RNF-006	Versión Android.	El sistema debe poder ejecutarse en versiones Android 5.0 en adelante.
RNF-007	No almacenar contraseña.	El sistema no almacenará la contraseña de los usuarios en plano.
RNF-008	Encriptar comunicaciones con la base de datos.	El sistema debe encriptar las comunicaciones con la base de datos para garantizar seguridad de los datos.

Tabla 16. Tabla de requisitos no funcionales.

### 3.2 Roles y actores del sistema.

En esta aplicación únicamente interaccionarán con ella los usuarios registrados, que a efectos de funcionalidad no existirá diferencia entre ellos por lo que se pueden identificar dos actores, los usuarios registrados y los usuarios no registrados cuyo único caso de uso será el de registrarse.

### 3.3 Diagrama de casos de uso.

El diagrama de la **Figura 5** representa la funcionalidad que proporciona la aplicación, para la elaboración de este, se ha usado la información recopilada en la especificación de requisitos y el análisis de los roles del sistema.

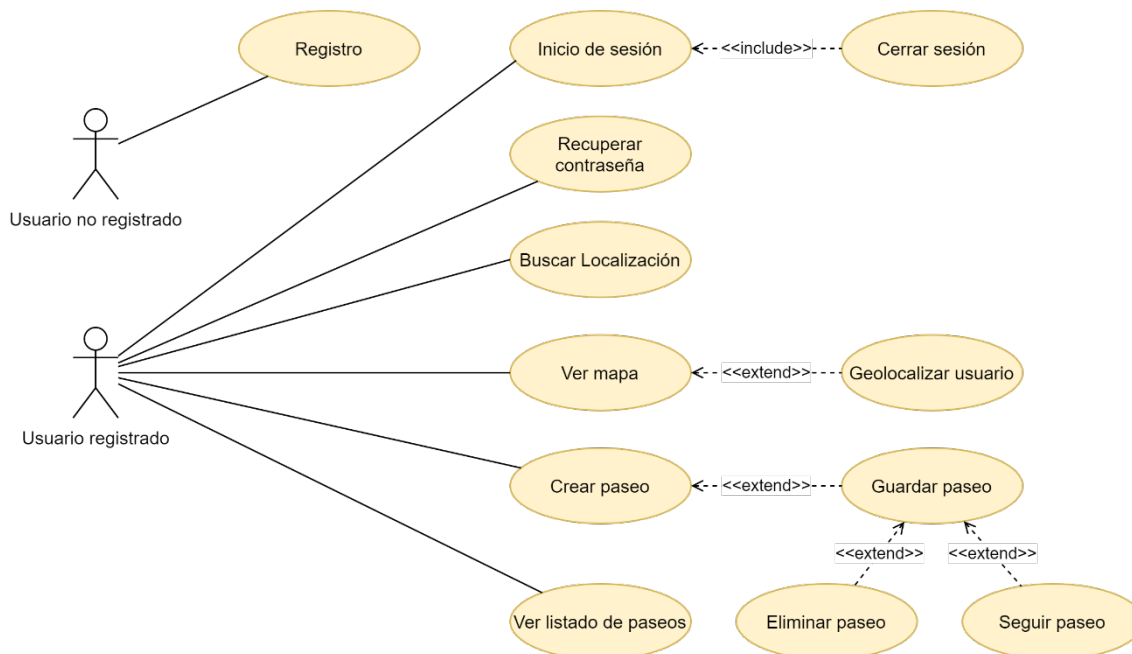


Figura 5. Diagrama de casos de uso de la aplicación

### 3.4 Descripción de los casos de uso.

En este apartado se detallará cada uno de los casos de uso identificados en el diagrama de casos de uso del apartado anterior.

<b>CU-001</b>	<b>Registro de usuario</b>
<b>Descripción</b>	El usuario se crea una cuenta de usuario en el sistema.
<b>Actor</b>	Usuario no registrado.
<b>Precondición</b>	El usuario que interactúa con la aplicación no está registrado en la aplicación.
<b>Flujo básico</b>	<ol style="list-style-type: none"> <li>1. El usuario selecciona la opción "Registro de usuario".</li> <li>2. El sistema solicita los datos requeridos para el registro de un usuario (correo electrónico, nombre y contraseña).</li> <li>3. El usuario introduce los datos y selecciona la opción "Crear cuenta".</li> <li>4. El sistema muestra la página de inicio tras hacer inicio de sesión con los datos proporcionados.</li> </ol>
<b>Postcondición</b>	No hay.
<b>Flujo alternativo</b>	<p>2.a, 3.a: El usuario cancela la operación, el caso de uso termina y queda sin efecto.</p> <p>3.a: Las contraseñas introducidas por el usuario no coinciden, por lo que se vuelve al paso 2.</p>



	3.b: El correo electrónico introducido por el usuario ya existe en el sistema por lo que el caso de uso vuelve al paso 2.
--	---

Tabla 17. Descripción CU-001

<b>CU-002</b>	<b>Iniciar sesión</b>
<b>Descripción</b>	El usuario inicia sesión con su cuenta de usuario del sistema.
<b>Actor</b>	Usuario registrado.
<b>Precondición</b>	El usuario ya está registrado en el sistema (CU-001). El dispositivo se encuentra sin ninguna cuenta de usuario iniciada.
<b>Flujo básico</b>	<ol style="list-style-type: none"> <li>1. El usuario selecciona la opción "Iniciar sesión".</li> <li>2. El sistema solicita correo electrónico y contraseña.</li> <li>3. El usuario introduce correo electrónico, contraseña y selecciona la opción "Enviar".</li> <li>4. El sistema realiza el inicio de sesión.</li> </ol>
<b>Postcondición</b>	No hay.
<b>Flujo alternativo</b>	<p>1.a, 3.a: El usuario cancela la operación, el caso de uso finaliza y queda sin efecto.</p> <p>3.a El usuario introduce datos erróneos, el sistema muestra un mensaje informativo y continua en el paso 2.</p> <p>3.b El usuario no recuerda las credenciales de su cuenta para iniciar sesión por lo que el caso de uso continua en el caso de uso CU-004.</p>

Tabla 18. Descripción CU-002

<b>CU-003</b>	<b>Cerrar sesión</b>
<b>Descripción</b>	El usuario desea cerrar sesión en su cuenta del sistema.
<b>Actor</b>	Usuario registrado.
<b>Precondición</b>	El usuario ha iniciado sesión previamente en el sistema.

<b>Flujo básico</b>	<ol style="list-style-type: none"> <li>1. El usuario selecciona la opción “cerrar sesión”.</li> <li>2. El sistema cierra la sesión del usuario en cuestión.</li> </ol>
<b>Postcondición</b>	No hay.
<b>Flujo alternativo</b>	No hay.

Tabla 19. Descripción CU-003\_

<b>CU-004</b>	<b>Recuperar contraseña.</b>
<b>Descripción</b>	El usuario desea iniciar sesión en el sistema, pero ha olvidado la contraseña por lo que el sistema le enviará un correo electrónico con un enlace donde podrá generar una nueva contraseña.
<b>Actor</b>	Usuario registrado.
<b>Precondición</b>	<p>El usuario no tiene una sesión iniciada en el dispositivo donde desea iniciar sesión.</p> <p>El usuario está registrado en el sistema.</p>
<b>Flujo básico</b>	<ol style="list-style-type: none"> <li>1. El usuario selecciona la opción de recuperar contraseña.</li> <li>2. El sistema solicita el correo electrónico al usuario.</li> <li>3. El usuario introduce el correo electrónico y selecciona la opción enviar.</li> <li>4. El sistema comprueba que existe una cuenta de usuario con ese correo asociado y envía un correo con el enlace donde el usuario podrá recuperar la contraseña.</li> <li>5. El usuario accederá a este enlace donde recuperar la contraseña.</li> <li>6. El sistema solicitará una nueva contraseña para el usuario.</li> <li>7. El usuario introducirá la contraseña y seleccionará la opción de “confirmar los cambios”.</li> <li>8. El sistema registra los cambios en la contraseña del usuario.</li> </ol>
<b>Postcondición</b>	No hay.
<b>Flujo alternativo</b>	<p>1.a, 3.a, 5.a, 7.a: El usuario cancela la operación, el caso de uso finaliza y queda sin efecto.</p> <p>3.a: El usuario introduce un correo que no está registrado en el sistema, el caso de uso vuelve al paso 2.</p> <p>7.a: El usuario introduce contraseñas distintas, el caso de uso vuelve al paso 6.</p>

Tabla 20. Descripción CU-004.

<b>CU-005</b>	<b>Buscar localización.</b>
<b>Descripción</b>	El usuario busca una localización introduciendo una cadena de texto identificativa de está.
<b>Actor</b>	Usuario registrado.
<b>Precondición</b>	El usuario está registrado en la aplicación y ha iniciado sesión.
<b>Flujo básico</b>	<ol style="list-style-type: none"> <li>1. El usuario selecciona la opción de buscar una localización.</li> <li>2. Sistema solicita una cadena de texto identificativa de la localización.</li> <li>3. El usuario introduce una cadena de texto identificativa del sitio que desea geolocalizar.</li> <li>4. El sistema muestra una lista de 5 opciones que se ajustan al texto buscado por el usuario.</li> <li>5. El usuario selecciona una de las opciones de la lista.</li> <li>6. El sistema marca en el mapa el punto exacto seleccionado y mueve el mapa hasta ese punto.</li> </ol>
<b>Postcondición</b>	No hay.
<b>Flujo alternativo</b>	<p>2.a, 4.a: El usuario cancela la operación, el caso de uso queda sin efecto.</p> <p>3.a: El sistema no encuentra coincidencias con la cadena introducida por el usuario. El sistema no muestra ninguna opción en la lista y queda a la espera de que el usuario modifique la cadena introducida en el paso 1.</p>

Tabla 21. Descripción CU-005.

<b>CU-006</b>	<b>Ver mapa</b>
<b>Descripción</b>	El usuario navega por el mapa.
<b>Actor</b>	Usuario.
<b>Precondición</b>	El usuario tiene sesión iniciada en el sistema.

<b>Flujo básico</b>	<ol style="list-style-type: none"> <li>1. El usuario selecciona la opción para abrir la página en la que se muestra el mapa.</li> <li>2. El sistema muestra el mapa.</li> <li>3. El usuario interacciona con el mapa (movimiento o cambio de zoom)</li> <li>4. El sistema actualiza la visualización del mapa.</li> </ol>
<b>Postcondición</b>	No hay.
<b>Flujo alternativo</b>	No hay.

Tabla 22. Descripción CU-006.

<b>CU-007</b>	<b>Geolocalizar usuario.</b>
<b>Descripción</b>	El sistema haya las coordenadas geodésicas en las que se encuentra el usuario y muestra el mapa centrado en la posición donde se encuentra el dispositivo.
<b>Actor</b>	Usuario.
<b>Precondición</b>	El usuario está registrado en la aplicación y ha iniciado sesión. El usuario ha realizado el caso de uso CU-006(Ver mapa).
<b>Flujo básico</b>	<ol style="list-style-type: none"> <li>1. El usuario selecciona la opción de geolocalizar al usuario en el mapa.</li> <li>2. El sistema halla la geolocalización del dispositivo, centra el mapa en dicha posición y pinta un marcador sobre el mapa.</li> </ol>
<b>Postcondición</b>	No hay.
<b>Flujo alternativo</b>	No hay.

Tabla 23. Descripción CU-006.

<b>CU-008</b>	<b>Crear paseo.</b>
<b>Descripción</b>	El usuario tiene la intención de preparar un paseo con un punto de inicio, un punto final y una duración, usa el sistema para que genere una ruta que se adapta a sus necesidades.
<b>Actor</b>	Usuario.
<b>Precondición</b>	El usuario está registrado en la aplicación y ha iniciado sesión.

<b>Flujo básico</b>	<ol style="list-style-type: none"> <li>1. El usuario selecciona la opción de empezar el proceso de crear un paseo.</li> <li>2. El sistema solicita una localización de inicio, una localización de finalización, una duración y un ritmo de andar.</li> <li>3. El usuario introduce una localización de inicio, una localización de finalización, una duración y un ritmo.</li> <li>4. El sistema calcula todas las opciones de paseos y se las muestra al usuario.</li> <li>5. El usuario selecciona el paseo.</li> <li>6. El usuario muestra un resumen con información relevante sobre el paseo.</li> </ol>
<b>Postcondición</b>	No hay.
<b>Flujo alternativo</b>	<p>1.a, 3.a, 5.a: El usuario cancela la operación, el caso de uso queda sin efecto.</p> <p>3b: El sistema no encuentra localizaciones adecuadas para la búsqueda del usuario, por lo que el sistema se queda a la espera de que el usuario introduzca una localización válida.</p> <p>6.a: una vez finalizado el caso de uso si el usuario lo desea este puede continuar en el CU-009(Guardar paseo).</p>

Tabla 24. Descripción CU-009.

<b>CU-009</b>	<b>Guardar paseo.</b>
<b>Descripción</b>	El usuario tras haber creado el paseo desea guardarlo en su cuenta para poder acceder a el cuando quiera.
<b>Actor</b>	Usuario.
<b>Precondición</b>	El usuario está registrado en la aplicación y ha iniciado sesión. El usuario ya ha creado el paseo en el CU-008.
<b>Flujo básico</b>	<ol style="list-style-type: none"> <li>1. El usuario selecciona la opción de guardar paseo.</li> <li>2. El sistema solicita un nombre con el que se desea guardar el paseo.</li> <li>3. El usuario introduce un nombre de paseo único.</li> <li>4. El sistema almacena el paseo y muestra un mensaje indicando que el proceso ha sido realizado correctamente.</li> </ol>
<b>Postcondición</b>	No hay.

<b>Flujo alternativo</b>	1.a, 3.a: El usuario cancela la operación, el caso de uso se cancela y queda sin efecto.
--------------------------	--

Tabla 25. Descripción CU-010.

<b>CU-010</b>	<b>Seguir paseo.</b>
<b>Descripción</b>	El usuario decide realizar un paseo de los que tiene guardados en su cuenta y el sistema le proporciona apoyo mostrándole la ruta y su ubicación.
<b>Actor</b>	Usuario.
<b>Precondición</b>	El usuario está registrado en la aplicación y ha iniciado sesión. El usuario ya tiene creado y guardado el paseo en el sistema.
<b>Flujo básico</b>	<ol style="list-style-type: none"> <li>1. El usuario selecciona uno de los paseos que tiene guardados.</li> <li>2. El sistema le muestra un mapa con, el paseo seleccionado dibujado en el mapa.</li> <li>3. El usuario selecciona la opción de geolocalizar el dispositivo en el mapa.</li> <li>4. El sistema geolocaliza el dispositivo y actualiza su posición periódicamente.</li> <li>5. El usuario sigue el paseo.</li> </ol>
<b>Postcondición</b>	No hay.
<b>Flujo alternativo</b>	<p>1.a, 3.a, 5.a: El usuario cancela la operación y el caso de uso queda sin efecto.</p> <p>4.a: el usuario no tiene la ubicación del dispositivo activada por lo que el caso de uso termina y queda sin efecto.</p>

Tabla 26. Descripción CU-011.

<b>CU-011</b>	<b>Eliminar paseo.</b>
<b>Descripción</b>	El usuario elimina paseos de su cuenta para que ya no aparezcan en esta.
<b>Actor</b>	Usuario.
<b>Precondición</b>	El usuario está registrado en la aplicación y ha iniciado sesión. El usuario tiene paseos creados en el sistema.
<b>Flujo básico</b>	<ol style="list-style-type: none"> <li>1. El usuario selecciona la opción de eliminar paseos.</li> </ol>

	<ol style="list-style-type: none"> <li>2. El sistema permite seleccionar una serie de paseos para que sean eliminados.</li> <li>3. El usuario selecciona una lista de paseos y selecciona la opción de eliminar.</li> <li>4. El sistema solicita una confirmación para eliminar el paseo.</li> <li>5. El usuario confirma que desea eliminar los paseos.</li> <li>6. El sistema elimina los paseos seleccionados por el usuario.</li> </ol>
<b>Postcondición</b>	No hay.
<b>Flujo alternativo</b>	1.a, 3.a, 5.a: El usuario cancela la operación, el caso de uso queda sin efecto.

Tabla 27. Descripción CU-012.

### 3.5 Diagrama de secuencia de los casos de uso.

En este apartado se muestran los diagramas de secuencia de los casos de uso en análisis, los cuales representan la interacción entre las clases definidas en el modelo de dominio del siguiente apartado y los actores.

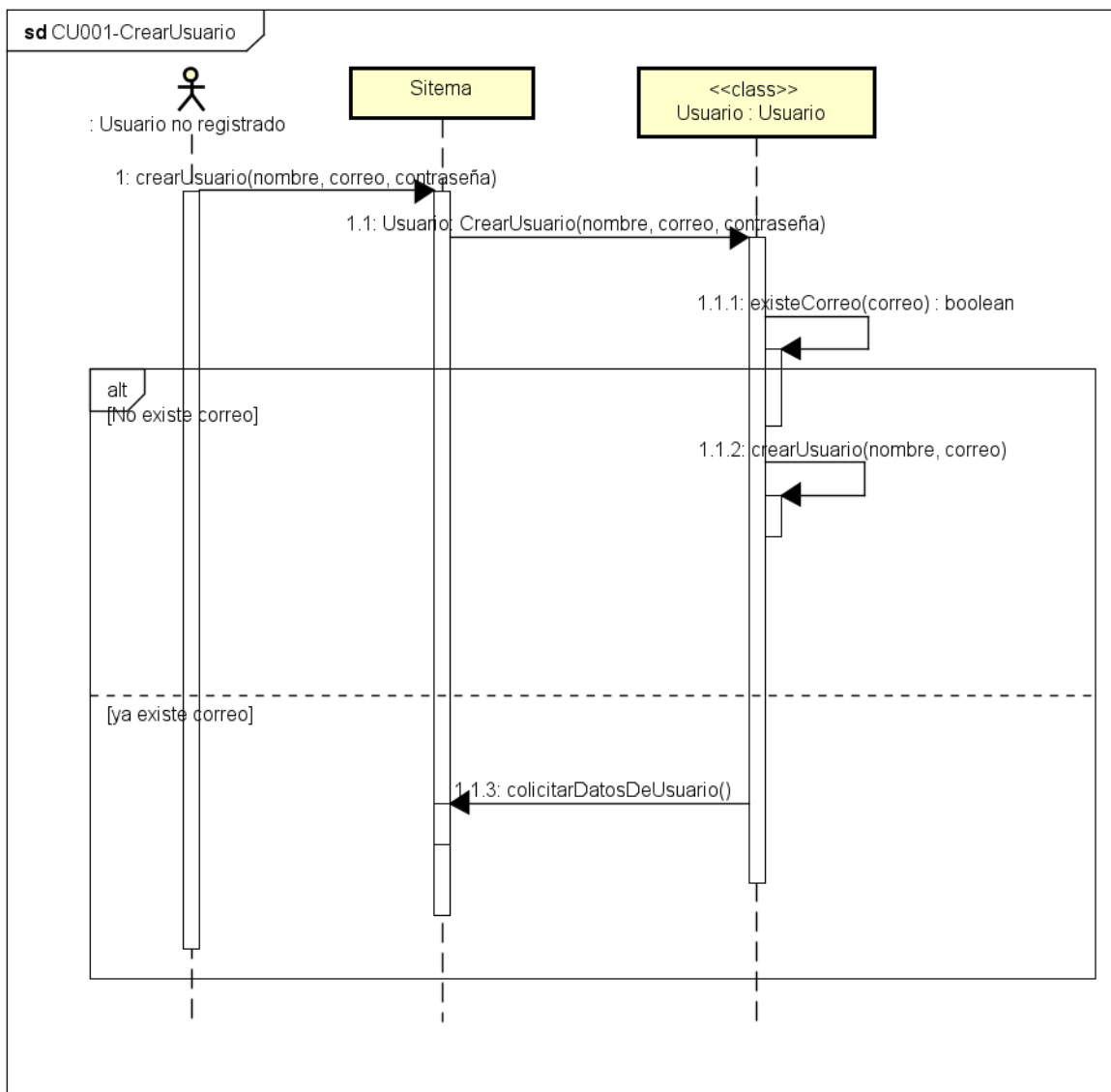


Figura 6. Diagrama de secuencia del CU001.



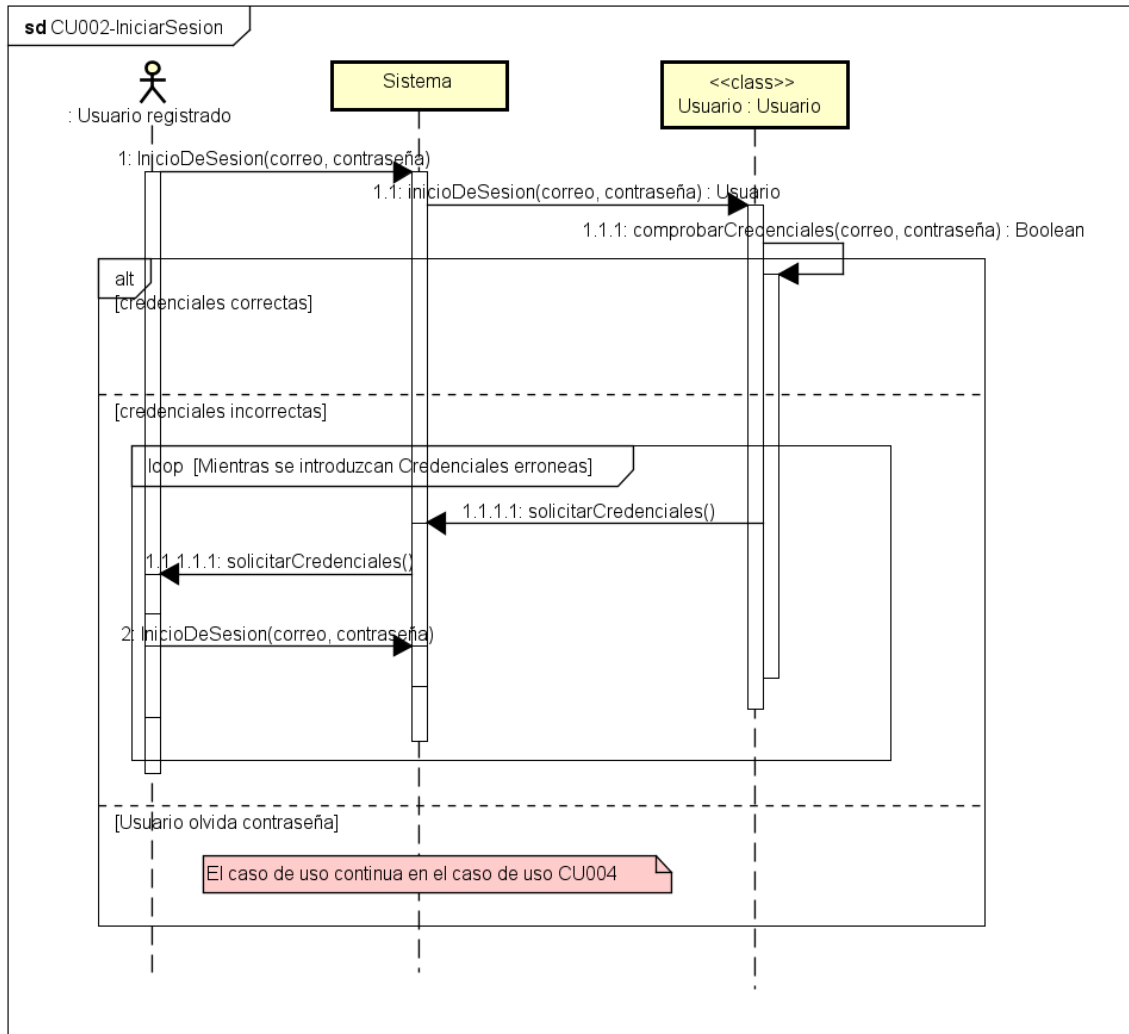


Figura 7. Diagrama de secuencia del CU002.

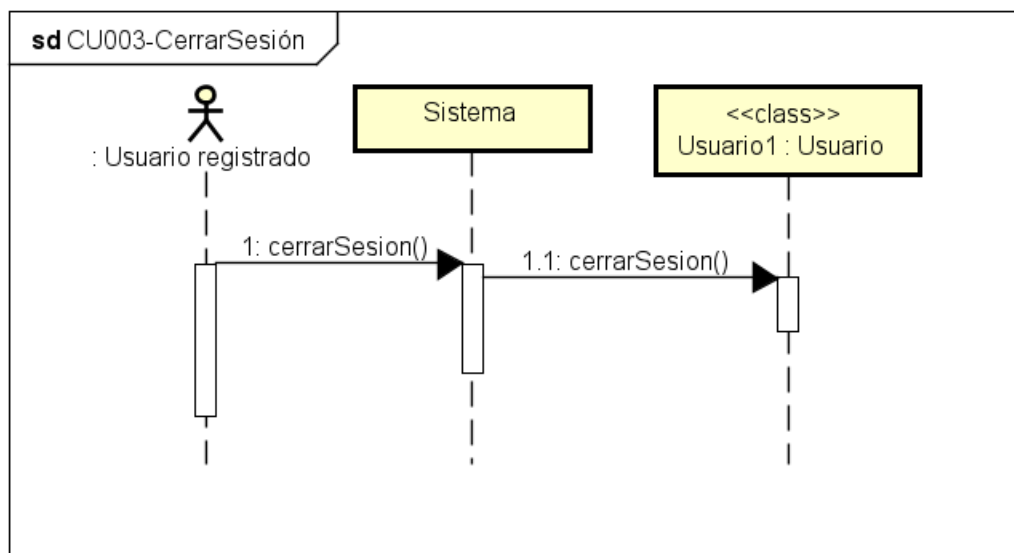


Figura 8. Diagrama de secuencia del CU003.

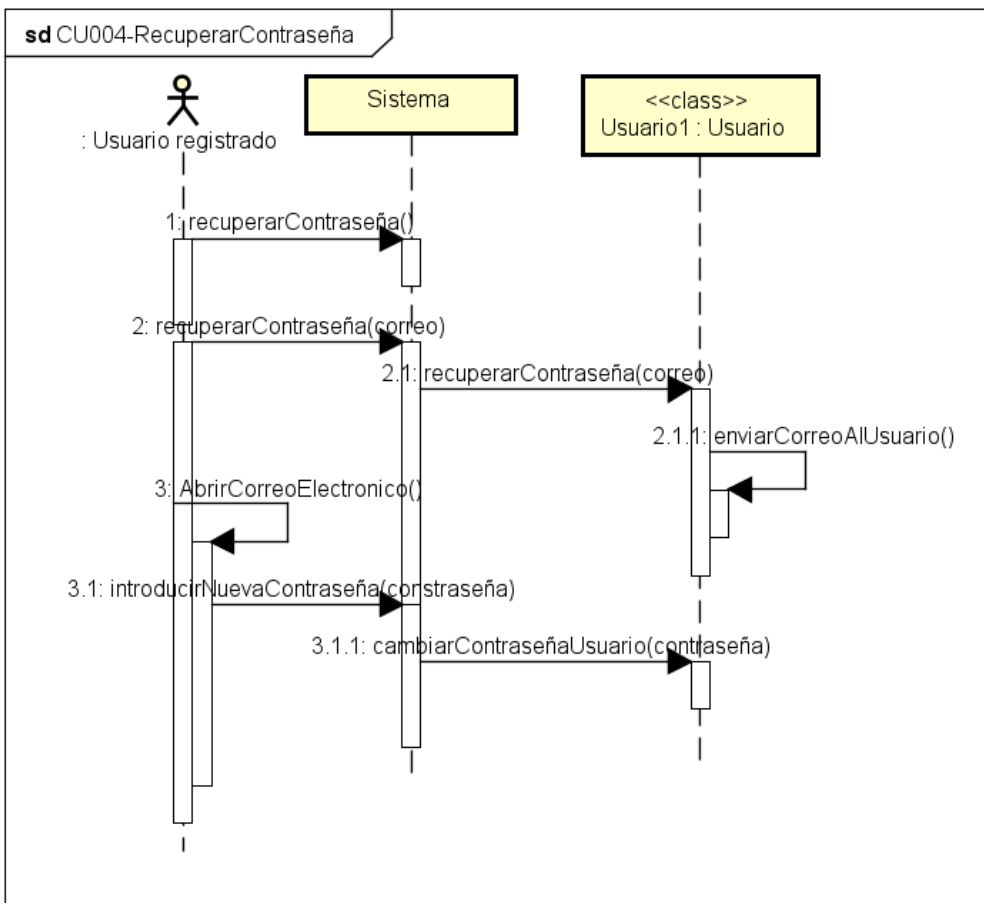


Figura 9. Diagrama de secuencia del CU004.

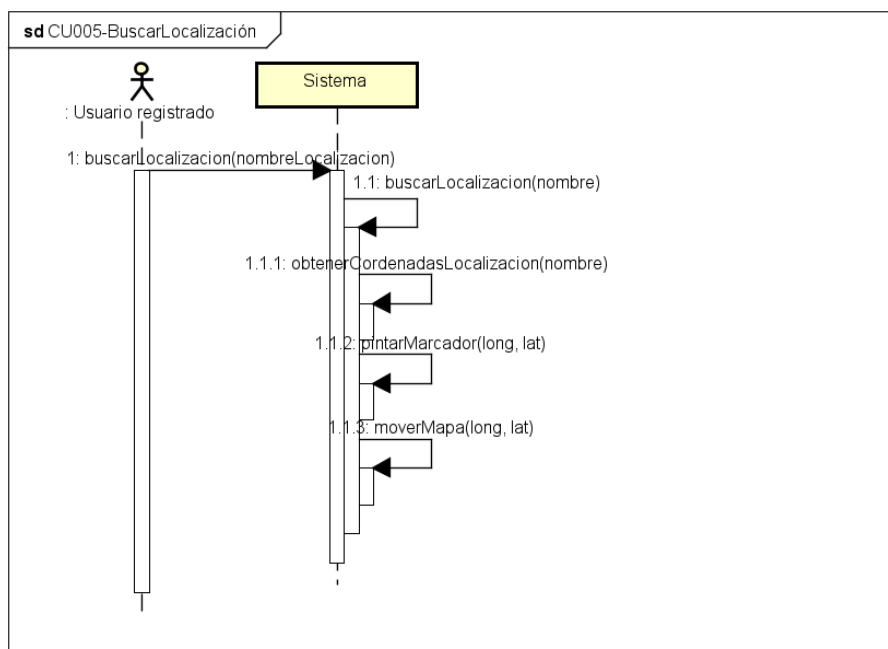


Figura 10. Diagrama de secuencia del CU005.

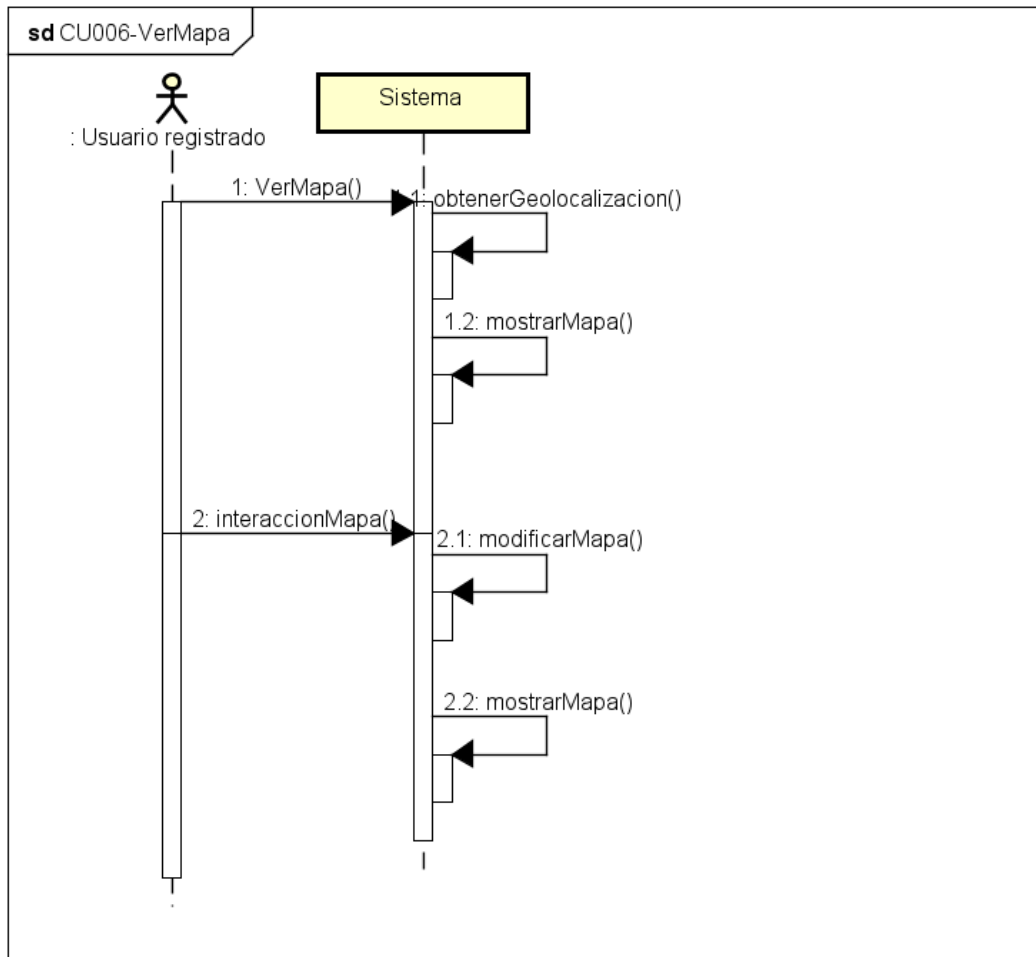


Figura 11. Diagrama de secuencia del CU006.

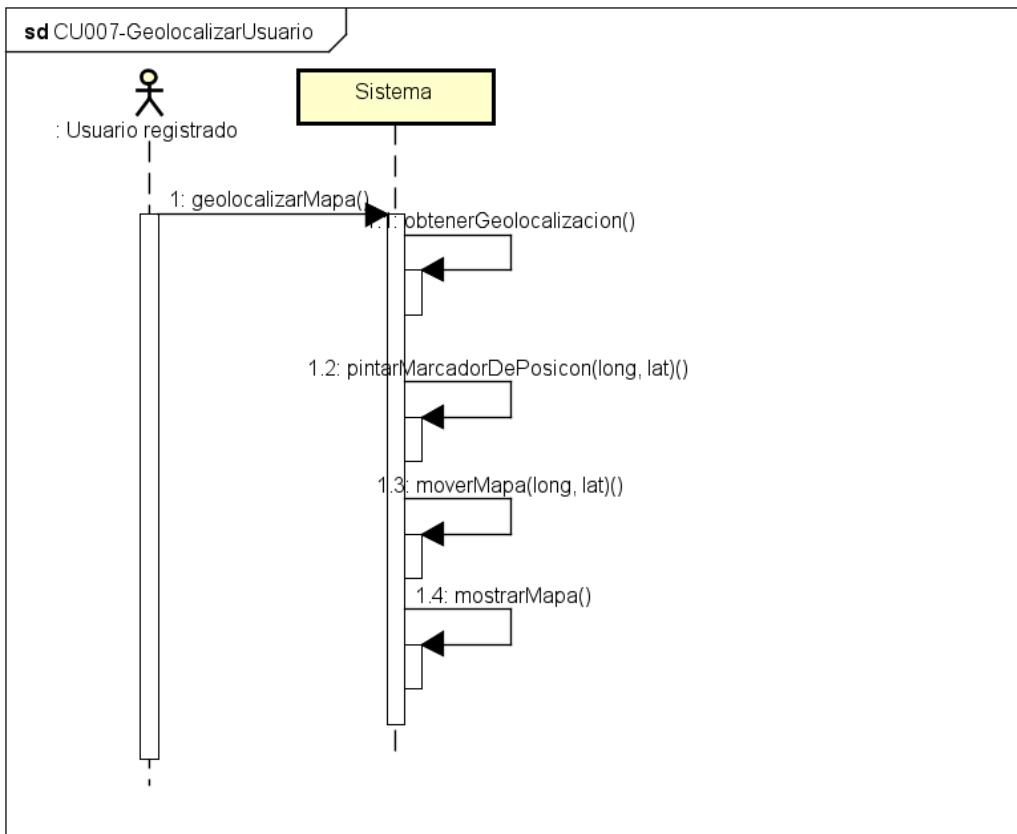


Figura 12. Diagrama de secuencia del CU007.

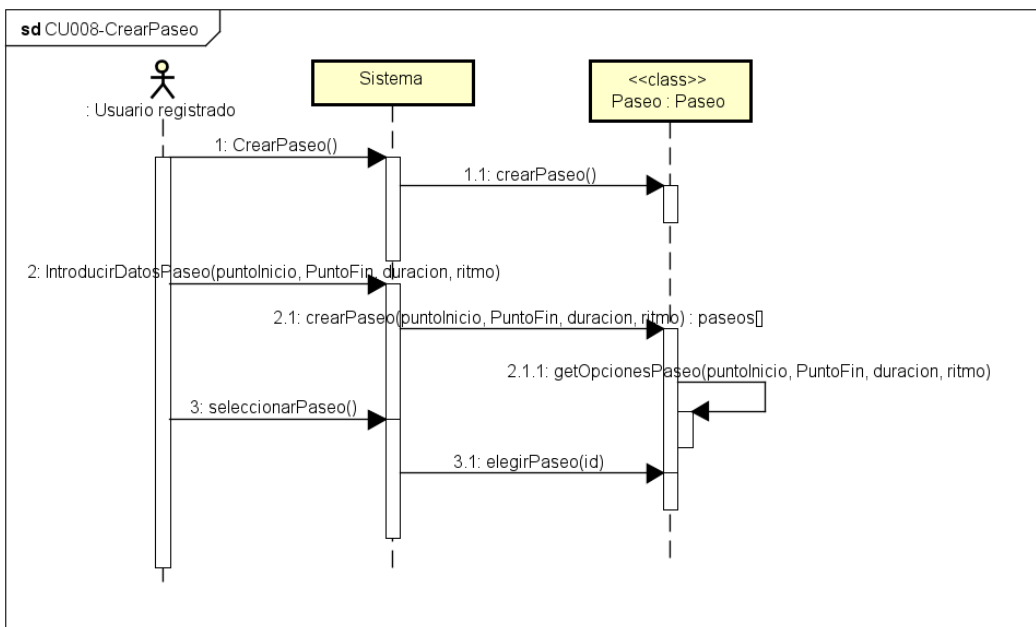


Figura 13. Diagrama de secuencia del CU008.

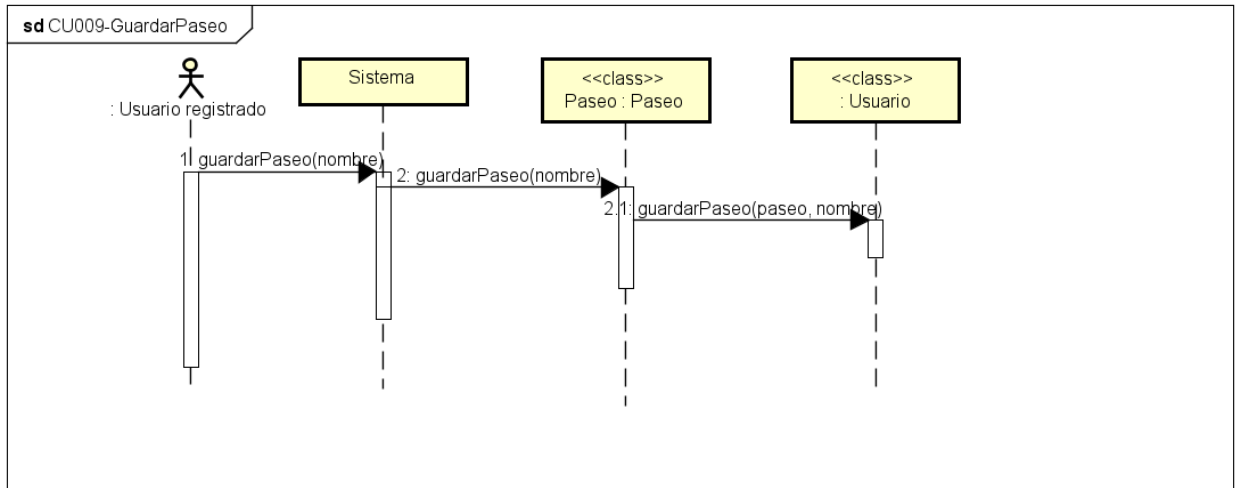


Figura 14. Diagrama de secuencia del CU009.

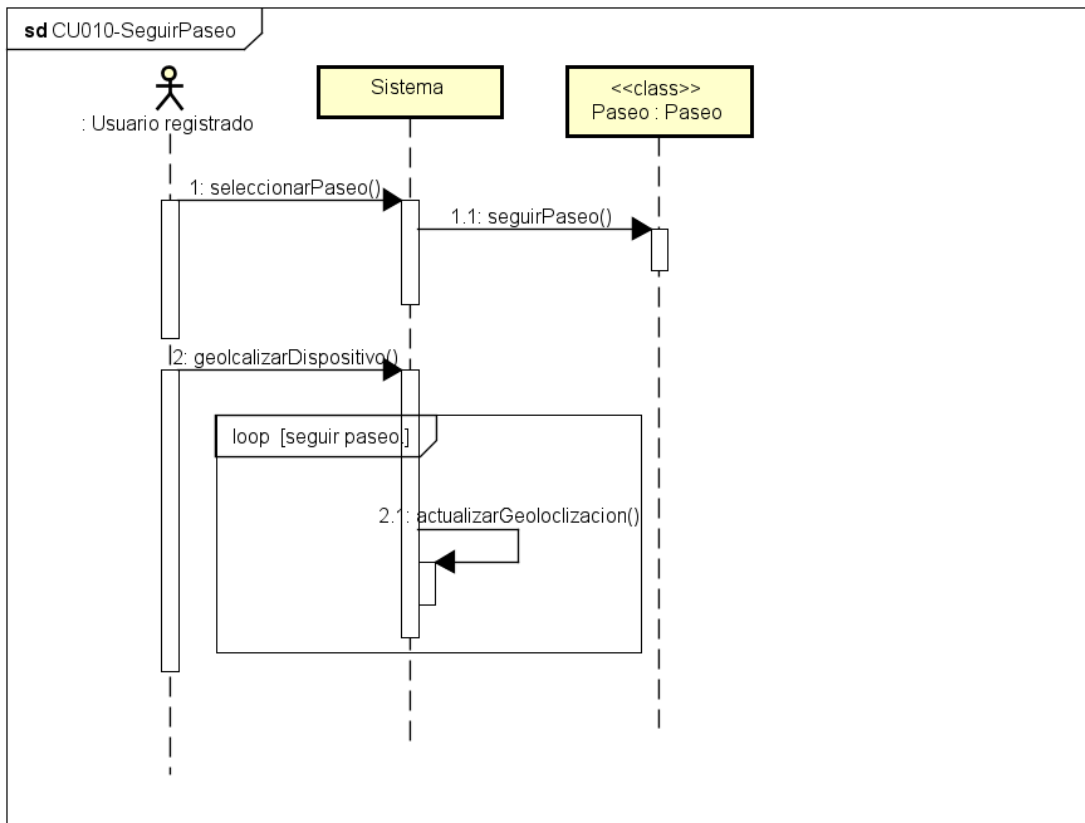


Figura 15. Diagrama de secuencia del CU010.

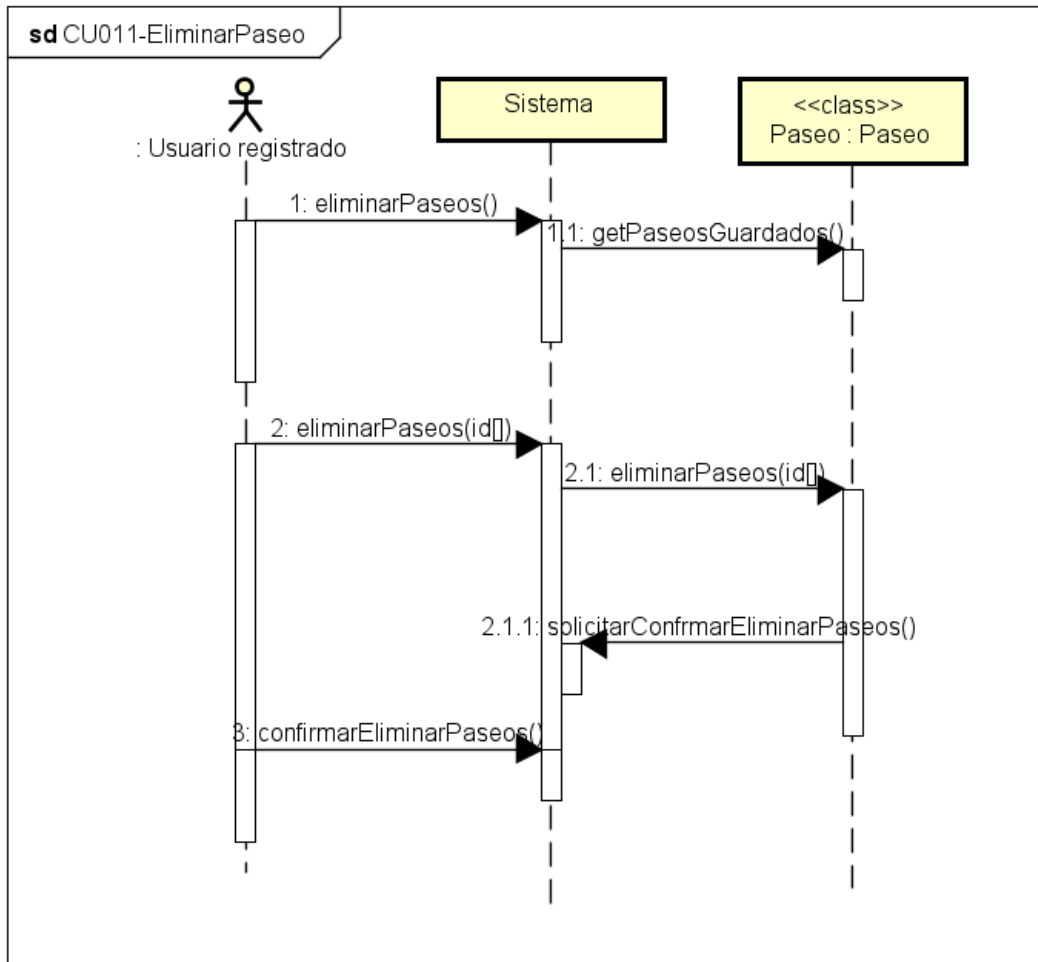
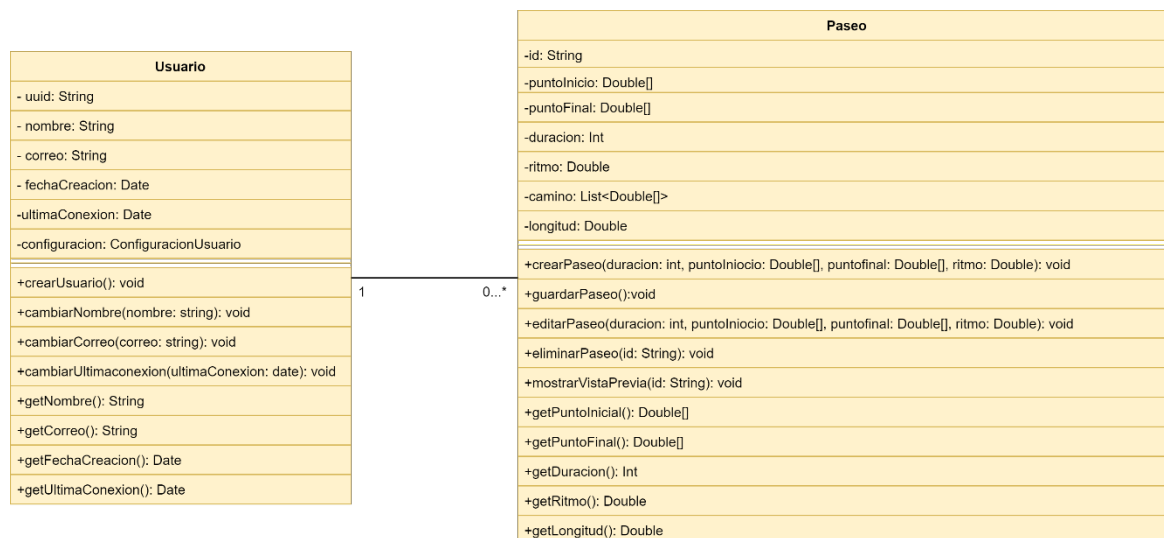


Figura 16. Diagrama de secuencia del CU011.

### 3.6 Modelo de dominio

En la **Figura 17** se puede ver un diagrama en el que se representa el modelo de dominio que de manera abstracta resume las entidades que forman parte del problema que va a cubrir la solución.



**Figura 17.** Diagrama del modelo de dominio en análisis.

A continuación, se explica las clases que forman el diagrama:

#### Usuario:

- Descripción: Clase que modela el perfil de los usuarios del sistema.
- Atributos:
  - Uuid: identificador único de cada usuario en el sistema.
  - nombre: Nombre del usuario en el sistema.
  - correo: Correo electrónico del usuario con el que iniciará sesión.
  - FechaDeCreacion: Fecha en la que el usuario se creó la cuenta.
  - ultimaConexion: Fecha en la que el usuario se conectó al sistema por última vez.
  - configuración: Conjunto de parámetros de configuración que el usuario desea tener en el sistema.

#### Paseo:

- Descripción: Clase que modela los paseos generados por los usuarios.
- Atributos:
  - Id: identificador único de cada paseo.
  - puntoInicial: Punto geodésico en el que da comienzo el paseo.
  - puntoFinal: Punto geodésico en el que finaliza el paseo.
  - duración: Tiempo que el usuario desea que lleve realizar la ruta del paseo.
  - ritmo: Velocidad que con la que el usuario estima que va a caminar y con la que se calculará el paseo.
  - camino: Conjunto de coordenadas por las que transcurrirá el paseo.
  - longitud: Distancia que cubrirá el paseo.





## Capítulo 4: Diseño.

En este capítulo se presentará el diseño del sistema, haciendo énfasis en la solución del problema. Se analizará con detalle la arquitectura del sistema, tanto lógica como física, el modelo de datos necesario para cubrir las necesidades de la aplicación y los bocetos iniciales de la interfaz de usuario.

### 4.1 Arquitectura del sistema.

La arquitectura utilizada por el sistema es una arquitectura en capas, como se puede ver en la **Figura 18**, para la que se ha utilizado una distribución de tres capas y un elemento llamado 'binder' que aparece entre las capas de presentación y Lógica de negocio debido al uso del framework web Angular, esto se explica con más detalle en el apartado 4.3.2.

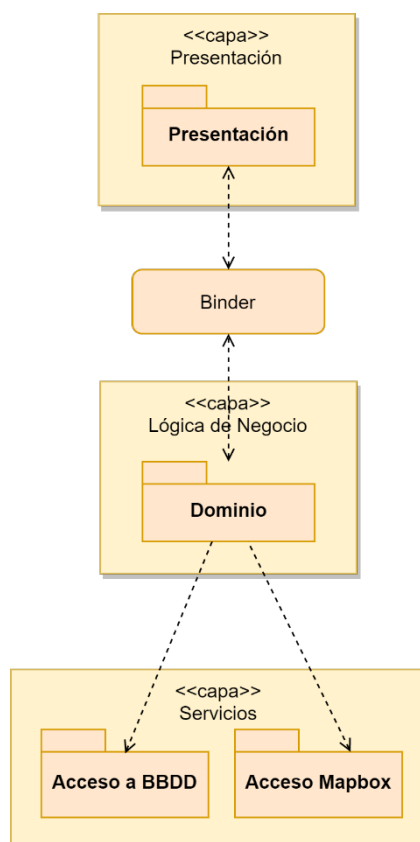


Figura 18. Modelo arquitectónico del sistema.

Las capas usadas en el sistema son:

- Capa de presentación: Es la capa de más alto nivel, en ella se encapsula la interfaz de la aplicación y la interacción del usuario con el sistema.
- Lógica de negocio: Es la capa intermedia, en ella se encuentra el dominio y toda su lógica, es decir, encapsula todo el marco conceptual utilizado en el sistema junto con los cálculos requeridos para cumplir con los requisitos.
- Servicios: Es la capa inferior y es la encargada de realizar la comunicación con los servicios externos utilizados por la aplicación. En este caso los servicios utilizados serán la API rest de Mapbox utilizada para obtener la información de mapas y por otro lado tenemos la conexión con la base de datos, en este caso será una base de datos NoSQL gestionada por Firebase Firestore.

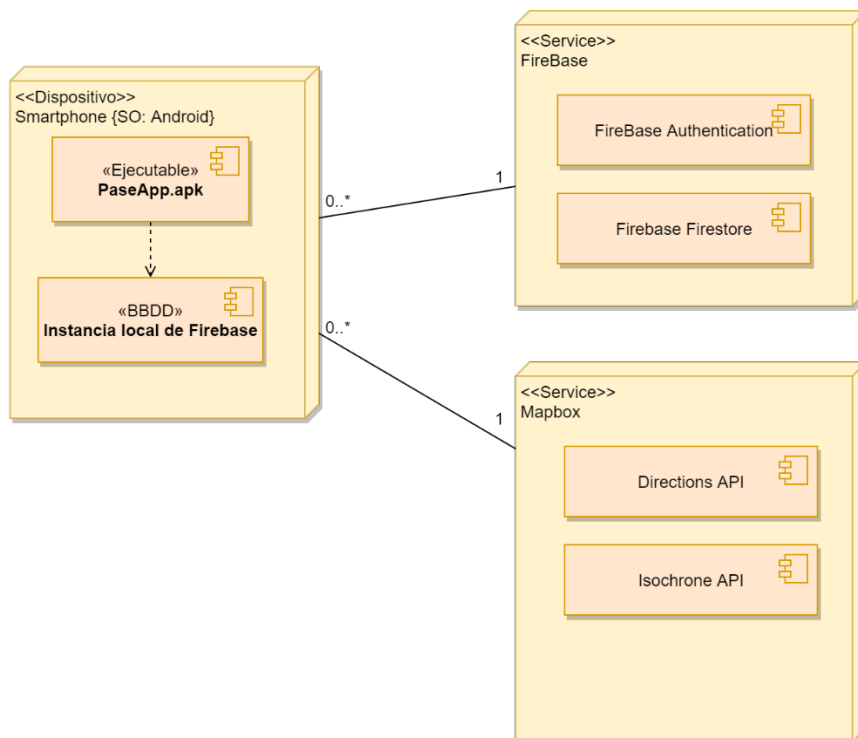
## 4.2 Diagrama de despliegue.

El diagrama de despliegue del sistema se muestra en la **Figura 19** y en él se modela la disposición física de los artefactos software usados por el sistema. En este caso podemos ver como el sistema se encuentra dividido en tres grandes grupos.

El primero de ellos sería el grupo formado por el dispositivo, donde se ejecuta el 'apk' de la aplicación y una instancia local de la base de datos con los datos del usuario.

También podemos ver un grupo formado por los servicios proporcionados por Firebase, autenticación (Authentication) y base de datos no relacional (Firestore).

Por último, tenemos al grupo formado por las APIs de Mapbox que serán usadas para obtener información de los mapas y poder llevar a cabo las funciones básicas del sistema.



**Figura 19.** Diagrama de despliegue.

### 4.3 Frameworks utilizados.

Para ayudar con el desarrollo de la aplicación se van a usar los Frameworks Ionic 4 y Angular, los cuales van a condicionar el diseño del sistema.

#### 4.3.1 Ionic.

Ionic <sup>[7]</sup> es un framework de código abierto que proporciona herramientas y servicios para el desarrollo de aplicaciones híbridas. El objetivo de usar Ionic es desarrollar una aplicación para dispositivos móviles usando como método de desarrollo técnicas web, es decir, usando HTML, CSS y JavaScript.

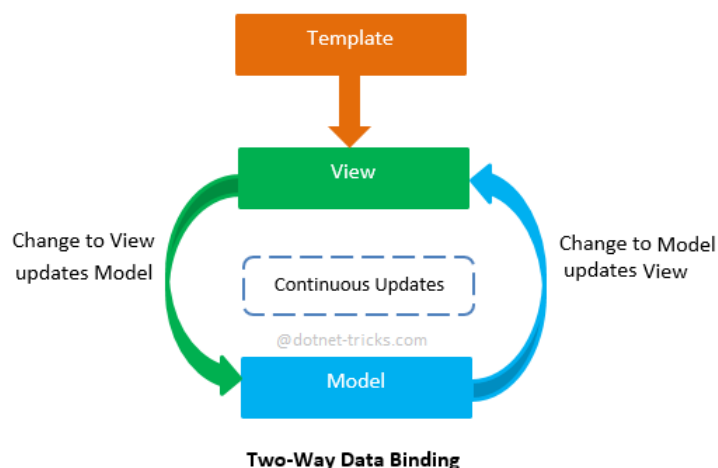
Para el desarrollo del proyecto hemos usado la versión de Ionic 6.12.3 junto con Angular 11.0.9 del que se hablará en el próximo apartado ya que condiciona el diseño del sistema.

Por último, hay que destacar que en este proyecto Ionic sólo se ha usado para la creación de una aplicación para Android, ya que los requisitos no especificaban la necesidad de implementarlo para otro tipo de SO.

#### 4.3.2 Angular Js.

Angular es un framework de código abierto cuyo fin es proporcionar herramientas y servicios para el desarrollo de aplicaciones web. Está mantenido por Google y es uno de los frameworks web con una comunidad más grande actualmente, por lo que resulta una buena opción para este proyecto.

El uso de Angular en el desarrollo de la aplicación ha condicionado el diseño de esta. Ya que una de las características básicas de angular es el ‘two way data binding’ <sup>[8]</sup>, esto es una función muy potente que consiste en la compartición de datos entre las plantillas de la vista y los componentes del modelo de dominio, es decir, cuando se produce un cambio en el dominio, inmediatamente el cambio se verá reflejado el cambio en la vista. Esto se puede ver representado en la **Figura 20**.



**Figura 20.** Esquema del ‘Two way data binding’<sup>[8]</sup>.

Con todo esto, es importante entender que la arquitectura explicada en el apartado 4.1 no es estrictamente una arquitectura por capas, si no que como se muestra en la **Figura 18** entre la capa de presentación y la de dominio existe un elemento llamado ‘binder’ proporcionado por Angular que se encarga de enlazar los datos de la vista con los del dominio, permitiendo actualizar la vista de forma dinámica con los datos del dominio.

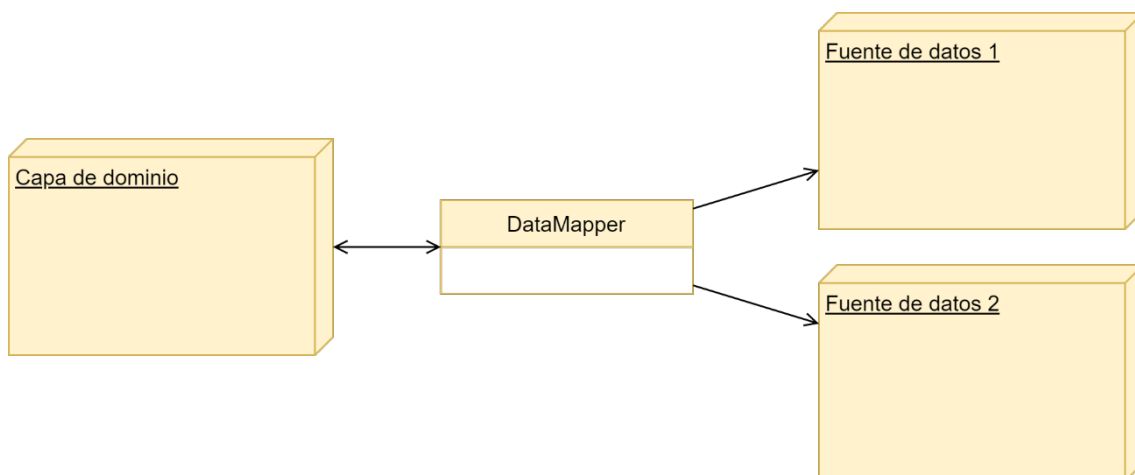
## 4.4 Patrones arquitectónicos.

En el diseño de este proyecto se ha hecho uso de patrones software con el fin de obtener soluciones conocidas y testadas a problemas genéricos como puede ser el acceso a la API rest de Mapbox o el acceso a la base de datos Firestore.

### 4.4.1 Data Mapper.

Para el acceso a datos se ha hecho uso del patrón **data mapper**, con el fin de simplificar el acceso a datos, este patrón lo que busca es aislar la capa de dominio y lógica de negocio de las capas de servicios externos y sistemas de almacenamiento de datos. El funcionamiento de este patrón se basa en el uso de una clase intermedia entre las capas de dominio y datos, la cual se hará cargo de todo el flujo de datos entre estas.

Este patrón nos aporta mayor aislamiento a la hora de hacer cambios en alguna de las capas, es decir, si fuera necesario hacer cambios en la capa de negocio o en los servicios externos, facilita hacerlo sin tener que hacer cambios en ambas capas. Un ejemplo sería el cambio de base de datos, no será necesario modificar el modelo de dominio ni su lógica, sino que habría que realizar los cambios en el **data mapper**, que es el que contiene toda la información técnica sobre cómo se ha de acceder a las fuentes de datos.



**Figura 21.** Diagrama patrón Data Mapper.

En el caso concreto de este sistema el patrón Data Mapper se ha utilizado para el uso de los servicios externos utilizados, Mapbox para la gestión de mapas y geolocalización y Firebase para la base de datos no relacional.

### 4.4.2 Capas y databinder.

Este ha sido el patrón utilizado para estructurar la arquitectura del sistema, se ha explicado en detalle en el **apartado 4.1**. De forma resumida, lo que implica este patrón es dividir el sistema en tres capas, capa de presentación, capa de dominio y capa de servicios. Lo que implican las capas es que cada una de ellas sólo se puede comunicar con sus capas inmediatamente inferior. Y en este caso tenemos la particularidad del databinder que lo que hace es crear un canal de comunicación directo e instantáneo entre la capa de presentación y la de dominio.

## 4.5 Modelo de base de datos.

Para la base de datos se ha usado un modelo de datos no relacional lo que permite una mayor flexibilidad, escalabilidad y mejor rendimiento, aunque para ello se sacrifican algunas de las restricciones de coherencia existentes en las bases de datos relacionales.

La base de datos usada ha sido Firebase Firestore, que es una base de datos no relacional basada en documentos. Esto significa que los datos se almacenan en formato de objetos del estilo a JSON o XML, lo cual facilita el desarrollo y las consultas, ya que todas las peticiones a la API de Mapbox trabaja también con datos en formato JSON.

El esquema de la base de datos se muestra en la **Figura 22**.

Usuario	Paseo
-uid: string	-pid: string
-correo: string	-putnos:double[double[]]
-nombre	-wayPoints:string
-fechaAlta: date	-idUsuario: string
-ultimaConexion: date	-fechaCración:date
-Ajustes: ajustesDeUsuario	-nombre:string
	-duración:int
	-ritmo:int

**Figura 22.** Esquema de la base de datos.

## 4.6 Diseño de la interfaz de usuario.

Para el diseño de la interfaz la actividad principal fue la realizada en el **apartado 1.3.1**, donde se realizó un análisis de aplicaciones ya existentes en el mercado. Con esto, se pudo analizar el estilo que usan este tipo de aplicaciones que tienen mapas como centro de sus casos de uso principales de la aplicación. A partir de esto, elaborar un diseño de la interfaz que sea intuitivo para los usuarios al tener una clara inspiración en ejemplos ya existentes. La primera versión de los bocetos se puede ver en la **figura 23**.

Para concretar el estilo y el logotipo de la aplicación se ha pedido ayuda a una estudiante del grado de Publicidad en Segovia, ella se ha encargado del diseño de los logotipos de la aplicación, se puede ver en la **figura 24**. Esta estudiante también ha ayudado con los colores usados en la aplicación, orientándonos a la utilización de una gama que usara azul y naranja (más concretamente en hexadecimal son los colores #006AFE y #FF9A00). La estudiante en cuestión se llama Lara Suarez.

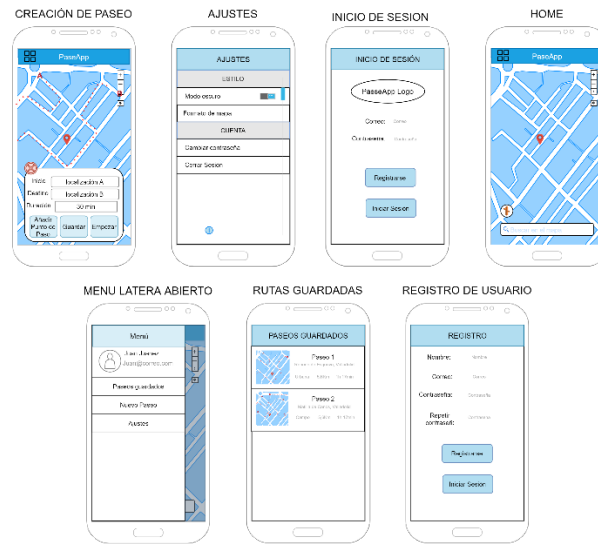


Figura 23. Bocetos de la interfaz de usuario.



Figura 24. Logotipos de la aplicación.

#### 4.7 Diagrama de clases de diseño.

En la **figura 25** se puede ver el diagrama de clases de diseño que representa conceptualmente cómo será la futura implementación de este. En él se puede ver la mezcla de la arquitectura del sistema (usando patrón capas), el modelo de dominio explicado a fondo en el análisis y la implementación del patrón **data mapper** en la relación existente entre las clases del dominio y los servicios encargados de la gestión de la información con los sistemas externos.

También hay que destacar que entre las capas de Presentación y la de Domini existe un mecanismo llamado **two way data binding** proporcionado por angular y que está representado como un paquete. La función de este elemento es mantener actualizado el flujo de información entre la capa de presentación y la lógica de negocio.

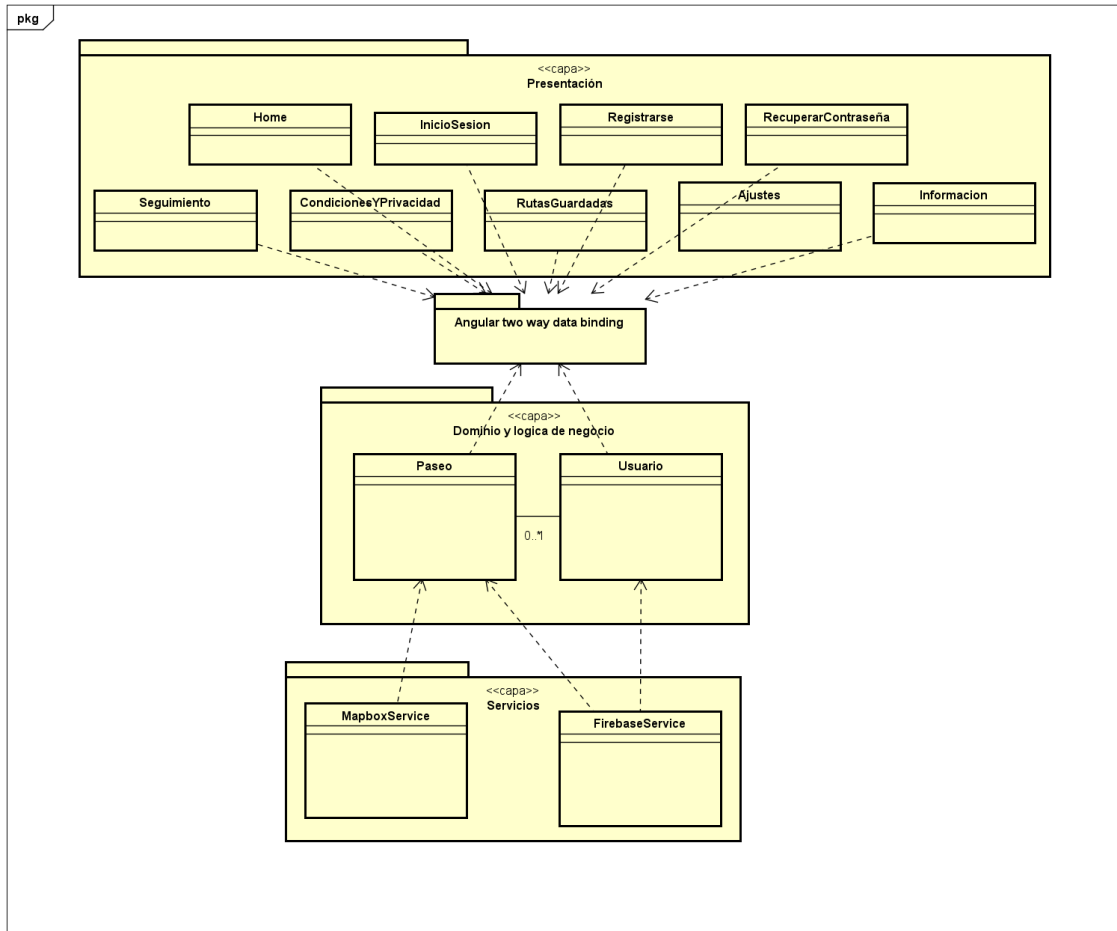


Figura 25. Diagrama de clases de diseño. Modules and uses style.

#### 4.8 Diagrama de clases detallado.

En este apartado se presentarán los diagramas de clases detallados, en ellos se representa en detalle las clases existentes en el sistema junto con sus métodos y atributos. Esto se puede ver en las **figuras 26 y 27**.

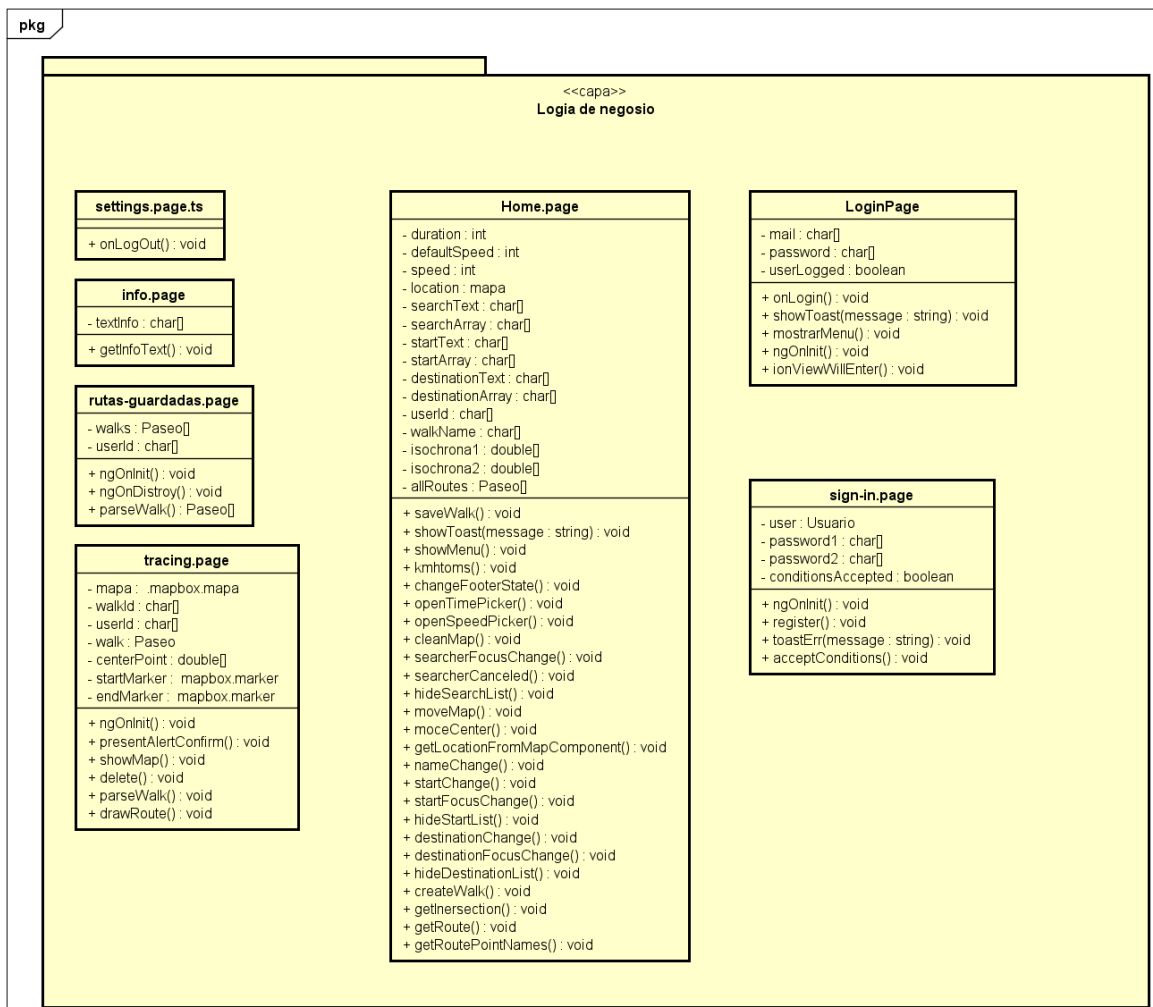


Figura 26. Clases detalladas de la capa de lógica de negocio.

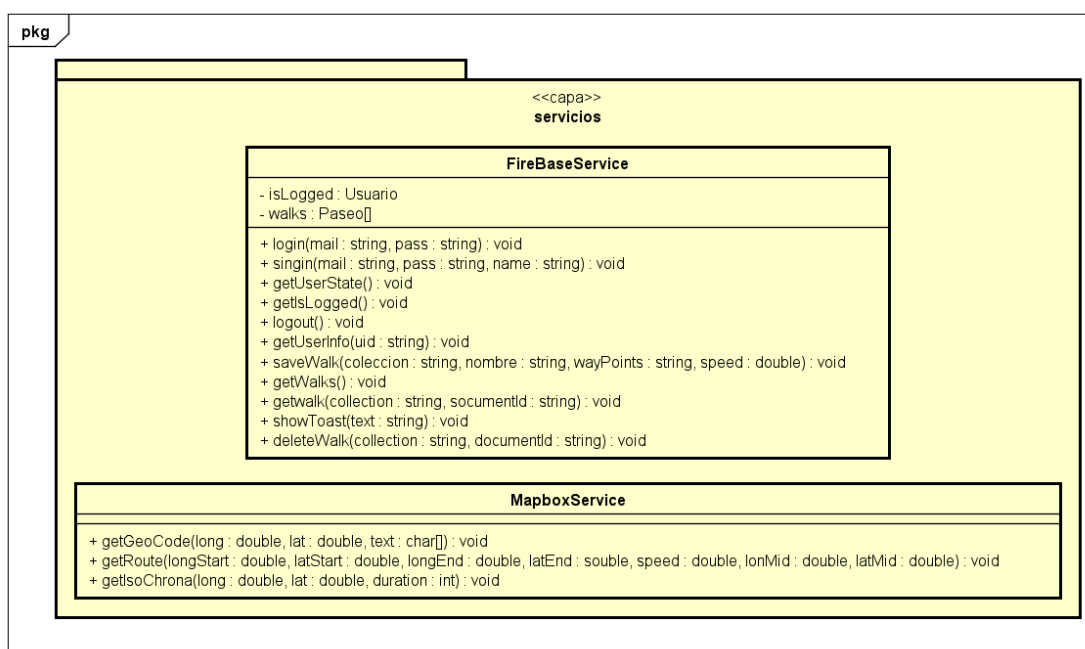


Figura 27. Clases detalladas de la capa de servicios.



## Capítulo 5: Implementación.

En este capítulo se tratan las técnicas, herramientas, algoritmos y decisiones tomadas a lo largo del proceso de implementación de la aplicación.

### 5.1 Herramientas.

A continuación, se describen todas aquellas herramientas utilizadas para el desarrollo del proyecto.

#### 5.1.1 Visual Studio Code.

Ha sido el editor de código elegido para el desarrollo del proyecto, se trata de un editor desarrollado y mantenido por Microsoft. El motivo principal por el que se ha usado es porque el alumno ya había trabajado con él y estaba familiarizado de antemano con su funcionamiento.

Por otro lado, también cuenta con funcionalidades muy potentes que ayudan al desarrollo de este tipo de aplicaciones como la integración con Git para llevar el control de versiones, la gestión de línea de comandos del sistema, soporte para depuración, existencia de gran variedad de plugins desarrollados por la comunidad ... lo que han hecho de esta herramienta la piedra angular sobre la que se ha desarrollado el proyecto.

La versión utilizada ha sido la 1.57.1.

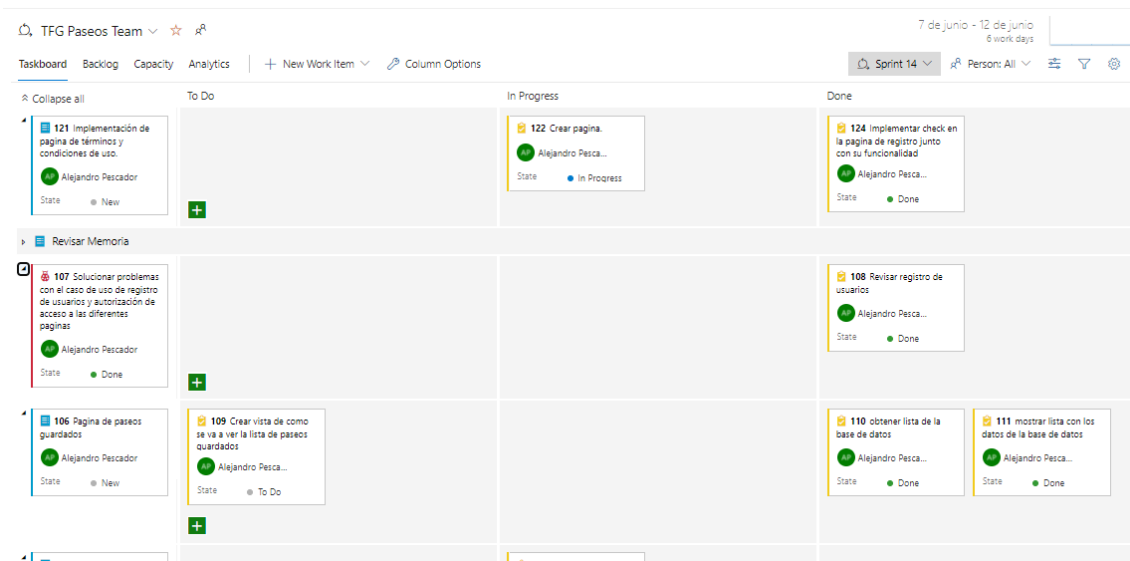
#### 5.1.2 GitHub.

Se trata de una compañía sin ánimo de lucro que ofrece alojamiento de repositorios en la nube utilizando el método de control de versiones de Git. GitHub cuenta con una página web y software de escritorio que permiten llevar el control de gestiones a través de una interfaz gráfica de usuario, lo que facilita el trabajo. También es importante su compatibilidad para sincronizar los repositorios remotos a través de Visual Studio Code, lo que se trata de una funcionalidad muy útil ya que en este proyecto se ha trabajado desde diferentes dispositivos y en diferentes localizaciones.

#### 5.1.3 Azure DevOps.

Herramienta proporcionada por Microsoft que en el ámbito de este proyecto ha sido utilizada para la gestión de Scrum en el proyecto. Esta herramienta proporciona una aplicación Web con todo lo necesario para hacer el seguimiento de un proyecto que use Scrum (Entre otros métodos de gestión). La aplicación permite personalización de la gestión, pero en el caso de este proyecto se ha usado la opción de Scrum por defecto existente en la propia Herramienta. Además, esta herramienta está

pensada para facilitar el trabajo en equipo por lo que facilita mucho la comunicación con el tutor ya que en todo momento este va a poder ver el estado de todas las actividades definidas en el backlog e incluso añadir o eliminar si fuera necesario. El seguimiento del estado de actividades es muy visual, utiliza una pizarra de Kanban como se puede ver en la **Figura 28**.



**Figura 28.** Vista de la pizarra de Kanban en el seguimiento del Sprint 14 del proyecto.

#### 5.1.4 Android Studio.

Android Studio es el entorno de desarrollo oficial para aplicaciones Android. En este proyecto el uso de Android estudio ha sido principalmente para usar el emulador integrado y hacer pruebas sin necesidad de un dispositivo físico. También se ha usado para generar los archivos ejecutables que posteriormente serán instalados en los dispositivos donde se quiera hacer uso de la aplicación.

#### 5.1.5 Firebase.

Plataforma gestionada por Google<sup>[9]</sup>, formada por una serie de herramientas en la nube orientadas a facilitar el desarrollo de aplicaciones móvil y páginas web. Entre los servicios que ofrece Firebase, podemos encontrar: Autenticación, base de datos no relacional, almacenamiento, alojamiento de servicios, servicios de machine learning y muchos más. Pero en lo que concierne a este proyecto hemos hecho uso de dos servicios, **Firestore Database** y **Firestore Authentication**.

- **Firestore Database:** Se trata de un servicio de base de datos no relacional que se ha usado en el proyecto para almacenar todos los datos necesarios para el funcionamiento de la aplicación. La estructura de datos usada se ha detallado en el apartado 4.4 del capítulo de Diseño.
- **Firestore Authentication:** Servicio que se ha usado para gestionar de forma sencilla y segura la creación, autenticación y gestión de cuentas de usuario.

Para el uso de los servicios de Firebase ha sido necesario instalar el SDK para JavaScript, crearse una cuenta de Google, crear un proyecto de Firebase, registrar la aplicación y obtener el objeto de configuración de Firebase, en el que se encuentra toda la información necesaria para que la aplicación pueda hacer uso de los servicios de Firebase a través del SDK de forma segura.

### 5.1.6 Turf.

Turf<sup>[11]</sup> es una librería escrita en JavaScript cuyo fin es el procesamiento de datos geoespaciales a través de objetos GeoJson y puede ser ejecutado tanto lado del servidor como en lado del cliente.

En este proyecto se ha hecho uso de esta librería para cubrir dos necesidades, que son: El mapeo de datos geoespaciales a objetos GeoJson cuando son devueltos de las API rest de Mapbox y por otro lado se ha usado para obtener los puntos de intersección entre Isócronas usados para calcular las rutas y que se explicarán más adelante con más detenimiento.

### 5.1.7 Mapbox.

Mapbox<sup>[18]</sup> es un proveedor de mapas que ofrece una gran variedad de servicios relacionados con la geolocalización y los mapas. Para ello, hemos instalado la librería Mapbox GL JS, que se trata de una librería enfocada al trabajo con mapas vectoriales en aplicaciones web, del mismo modo se va a hacer uso de tres de las API REST que proporcionan. Las API REST que se usarán en este proyecto son las siguientes:

- Geocoding: API que ofrece el servicio de transformar un texto relacionado con una ubicación en unas coordenadas y al revés, lo cual se ha usado para obtener las localizaciones de los lugares buscados por los usuarios en la aplicación.
- Isochrone: API que ofrece el servicio de obtener perímetros isócronos desde unas coordenadas determinadas. En el próximo apartado se explicará más en detalle lo que es un perímetro isócrono, pero básicamente es que cada punto perteneciente al perímetro está a al mismo tiempo del punto especificado.
- Directions: API que ofrece servicios para la creación de rutas, permite seleccionar entre diversos atributos como el medio de transporte, la velocidad, utilizar más de un punto de paso, formato de la respuesta, etc.

### 5.1.8 Draw.io.

Es una herramienta de creación y edición de diagramas multiplataforma. La aplicación cuenta con multitud de posibilidades y muchos recursos ya creados que se pueden reutilizar. En este proyecto, se ha utilizado para hacer los bocetos iniciales de la interfaz y para la creación de algunos de los diagramas descritos en la memoria, como por ejemplo el de la **Figura 18** con la arquitectura del sistema.

### 5.1.9 Astah.

Se trata de una herramienta de modelado de software con la que se puede hacer diagramas UML, de secuencia, de flujo, mapas conceptuales, etc. Es uno de los softwares más estandarizados para el apartado de diseño. También cuenta con la posibilidad de usar plug-ins que aumentan su potencia y sus posibilidades. La universidad nos proporciona una licencia para hacer uso de el.

## 5.2 Decisiones de implementación y algoritmos utilizados.

### 5.2.1 Versiones de Android soportadas.

La elección de versiones de Android en las que funcionará la aplicación se trata de una decisión fundamental a la hora de hacer una aplicación móvil, ya que de forma muy directa limitará el número de usuarios que podrán usar esta.

En la **Figura 29**, se puede ver las proporciones de usuarios que soporta cada versión de Android en la actualidad. Además de analizar los datos de uso de cada versión, al tratarse de una aplicación desarrollada con un framework orientado a la creación de aplicaciones híbridas, para poder ejecutarla es necesario que los dispositivos cuenten con Android Web View <sup>[15]</sup>. Si accedemos al portal de descarga de software Uptodown podemos ver que para instalar el componente web View exige una versión de Android superior a la 5.0 Lollipop.

Con la información que hemos comentado en el párrafo anterior, lo más sensato es implementar la aplicación de manera que podamos asegurar que cualquier usuario que posea Android 5.0 o superior y el componente Web View instalado, pueda hacer uso de la aplicación. Por eso, en el capítulo 3 se especificó como requisito no funcional que el sistema debe de poder correr en versiones de Android 5.0 o mayor.

ANDROID PLATFORM VERSION	API LEVEL	CUMULATIVE DISTRIBUTION
4.0 Ice Cream Sandwich	15	
4.1 Jelly Bean	16	99,8%
4.2 Jelly Bean	17	99,2%
4.3 Jelly Bean	18	98,4%
4.4 KitKat	19	98,1%
5.0 Lollipop	21	94,1%
5.1 Lollipop	22	92,3%
6.0 Marshmallow	23	84,9%
7.0 Nougat	24	73,7%
7.1 Nougat	25	66,2%
8.0 Oreo	26	60,8%
8.1 Oreo	27	53,5%
9.0 Pie	28	39,5%
10. Android 10	29	8,2%

**Figura 29.** Estadísticas de la distribución de usuarios en las diferentes versiones de Android.

### 5.2.2 Algoritmos usados.

Este apartado resulta ser uno de los más importantes para comprender el funcionamiento de la aplicación, pues resuelve el problema de la creación de las rutas que se usarán como paseos, lo que responde al caso de uso principal del sistema. Para ello se ha creado un algoritmo, que haciendo uso de las APIS de Mapbox y sus limitaciones, nos permite generar un paseo que se adapte al tiempo y al ritmo que desee el usuario.

Pero antes de explicar los pasos que componen este algoritmo es necesario tener claro el concepto de **isócrona** <sup>[10]</sup>, este concepto lo que viene a representar un conjunto de puntos que a partir de un punto central resultan estar al mismo tiempo. En la **Figura 31** se pueden ver dos áreas dibujadas de color azul y amarillo, donde el perímetro que rodea a estas áreas se trata de un conjunto de puntos **isócronos** con respecto de sus puntos centrales, pintados con un marcador azul. Con este concepto entendido ya podemos pasar a explicar el algoritmo:

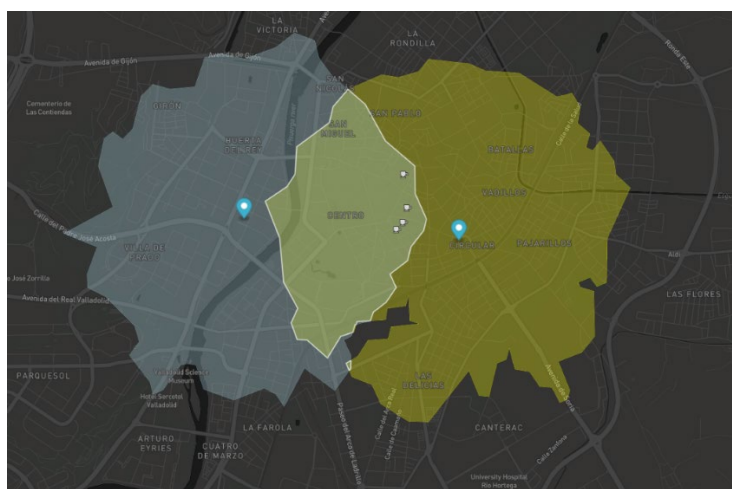
- **Datos de entrada:** Son los datos a partir de los cuales el algoritmo obtiene las rutas. Estos datos los conforman el punto geodésico de inicio de la ruta, el punto final de la ruta, la duración de la ruta (en segundos) y el ritmo con el que se quiere realizar la ruta (en metros por segundo). Todos estos datos serán datos que el usuario introduzca en la aplicación a través de la interfaz de usuario.

- **Datos de salida:** El resultado del algoritmo será un grupo de rutas, formadas por un conjunto de coordenadas geodésicas que representan la ruta que cumple con las restricciones impuestas por los datos de entrada.
- **Precondiciones:** La única precondición que exige este algoritmo es que el tiempo usado en la ruta más rápida entre el punto de inicio y el de final tiene que ser menor que la duración usada como dato de entrada.
- **Pasos del algoritmo:** Serie de pasos que garantizan que si se cumple se obtiene una ruta que cumple con las limitaciones deseadas por el usuario:
  1. Obtener valor de la duración adaptada en relación con el ritmo introducido para calcular la ruta. El motivo de este paso es que el método de la API de Mapbox encargada de calcular los isócronos usada más adelante en el algoritmo no permite adaptar la velocidad si no que lo hace con una velocidad por defecto de 1,42m/s o 5,1m/s. Para obtener este valor se utiliza la expresión de la **Figura 30**, donde  $v'$  es la velocidad de entrada del algoritmo,  $t$  es el tiempo de entrada del algoritmo y  $t'$  es la duración adaptada.

$$t' = \frac{v'}{5,1} \cdot t$$

**Figura 30.** Expresión para adaptar el tiempo de duración del paseo.

2. Con el valor obtenido en el primer punto y los puntos de inicio y de final hay que obtener los isócronos correspondientes a estos puntos y a  $t'/2$ . Para obtener estos isócronos se ha hecho a través de la API rest de Mapbox dedicada a este tipo de cálculos.
3. Obtenemos las intersecciones existentes entre los dos isócronos obtenidos. Si se cumple la precondición siempre va a existir como mínimo una. Para calcular estas intersecciones hemos usado el método *lineIntersect* proporcionado por la librería especializada en cálculos geoespaciales **Turf**.
4. Una vez obtenidos los puntos de las intersecciones lo que hay que hacer es calcular las rutas. Se obtendrán tantas rutas alternativas como puntos de intersección encontrados y para obtener estas rutas hay que hacer una petición a la API rest de Mapbox dedicada a este tipo de tareas, especificando la velocidad, el punto de inicio, un punto de intersección, el punto final e indicarle que se quiere obtener una ruta caminando. A esta petición la API respondería con los datos correspondientes a la ruta que se va a ofrecer al usuario.



**Figura 31.** Representación de isócronos en una demo ofrecida por Mapbox.

### 5.2.3 Protección de datos y seguridad.

Al tratarse de una aplicación que utiliza y almacena datos personales es importante proporcionar protección y seguridad sobre estos. Y no solo como buena práctica si no que está legislado en el Reglamento General de Protección de Datos (GDPR).

Por ello, lo primero es saber que datos personales se utilizan en la aplicación, cuáles de ellos se almacenan y de qué manera se almacenan estos datos.

La aplicación desarrollada en el proyecto hace uso de los siguientes datos personales: nombre del usuario, correo electrónico, contraseña, la geolocalización de dispositivo y los paseos creados por los usuarios. De estos datos mencionados anteriormente se almacenan: el nombre, el correo, la contraseña y los paseos creados por el usuario. Mientras que la geolocalización sí que se usa, pero en ningún momento se almacena.

Los datos de los que se ha hablado en el párrafo anterior se almacenan usando los servicios de Firebase Authentication y Firestore, pertenecientes a Google, entidad que en este proyecto tendrá el rol de **procesador de datos**, rol definido en el GDPR y que representa la persona física o virtual, autoridad pública, agencia u organismo que se encargará de procesa datos personales en nombre del responsable de los datos, en este caso el dueño del proyecto.

Una vez especificados los roles y los datos involucrados en el sistema, es momento de comprobar que se cumple con los derechos de los usuarios definidos el GDPR:

- Obligación por parte del dueño de los datos de comunicar de forma clara y concisa al usuario, los datos que se van a utilizar y el tratamiento de estos. Para ello se ha implementado una opción en el proceso de registro de usuario donde el usuario podrá leer una declaración con toda la información al respecto y que no podrá, finalizar con el proceso si no acepta expresamente dicha declaración.
- Obligación de proporcionar a los usuarios la posibilidad de retirar su consentimiento, para ello se ha habilitado un buzón de correo electrónico donde se podrá comunicar a los responsables de los datos dicha intención para posteriormente retirar los datos del usuario.
- Implementación del derecho al olvido, esto significa que, si el usuario decide eliminar su cuenta, el sistema debe ser capaz de eliminar sus datos de forma permanente. Para ello al igual que en el apartado anterior se ha habilitado un buzón donde comunicar a los responsables de los datos dicha intención.
- Existencia de un delegado de protección de datos. Será la persona encargada de supervisar como se tratan los datos. En este caso el alumno será el que desempeñe este rol.
- Existe un encargado del tratamiento de los datos que es la persona que dicta el fin y la forma en la que se van a tratar estos datos, en este caso el rol lo desempeñara el alumno.

En cuanto a la seguridad<sup>[21]</sup> de los datos, se pueden destacar dos momentos bien diferenciados. El primero de estos momentos sería el transporte y la comunicación de los datos entre el dispositivo donde corre la aplicación y el almacenamiento físico de los datos, sobre esto Google nos garantiza que encripta los datos con HTTPS asegurando la integridad de los datos. Por otro lado, tenemos el momento en el que los datos se almacenan en reposo en sus correspondientes servidores, en concreto en los servicios usados por esta aplicación (Authentication y Firestore) también se nos garantiza que los datos se encuentran encriptados.

Por último, sobre la seguridad se ha de destacar que los servicios de Google utilizados en la aplicación están certificados por los principales estándares de seguridad y privacidad ISO y SOC, más

concretamente Google nos garantiza el cumplimiento de los procesos de certificación ISO 27001, ISO 27017, ISO 27018, SOC 1, SOC 2 y SOC 3.

### 5.3 Organización del código.

La organización del código viene marcada por la implementación de un proyecto con el framework Angular y la arquitectura de capas usada en el proyecto. Todo el código y los recursos usados en la implementación de la aplicación se encuentran en la carpeta *src/app* dentro del proyecto y dentro de ella se organiza en diferentes directorios:

- Recursos multimedia: en la aplicación se hace uso de una serie de imágenes que se plasman en las diferentes vistas. Estos recursos se verán alojados en la carpeta *src/app/assets/images*.
- Capa de dominio y de presentación: La implementación de estas capas se encuentra en lo que en Angular se conoce como componentes, cada componente está formado por dos archivos de presentación (.html y .scss) y de uno de lógica de negocio que se trata de un archivo escrito en typescript (derivación de Javascript usada en Angular). El código se encuentra en los directorios *src/app/components* y *src/app/pages*, en este último hay un directorio dedicado para cada ventana de la vista.
- Capa de servicios: La implementación de esta capa se encuentra en el directorio *src/app/services*, donde se encuentran dos archivos, uno para la conexión con Firebase y otro para la conexión con Mapbox.
- Carpeta de Guards: Es un directorio que contiene la lógica para restringir las rutas a las que pueden acceder los usuarios, se encuentra en la dirección *src/app/guards*





## Capítulo 6: Pruebas.

En este capítulo se describirá el proceso llevado a cabo para hacer las pruebas de funcionamiento a la aplicación desarrollada. Durante la implementación se han ido realizando pruebas de caja negra con cada método que aquí no se describen. En este apartado, nos centraremos en lo que son pruebas de caja negra guiadas por caso de uso, lo que significa que no se va a tocar el código, sino que se va a comprobar que cada caso de uso definido en el capítulo de análisis, está implementado y funciona correctamente. Las pruebas se llevarán a cabo en el dispositivo móvil personal del alumno con sistema operativo Android. Por último, en el apartado final del capítulo se hablará de las pruebas realizadas por usuarios reales a los que se les ha instalado la aplicación en sus móviles personales.

### 6.1 Criterio de evaluación de las pruebas.

Lo primero, antes de ponerse a realizar las pruebas es definir las métricas que nos van a permitir entender el grado de éxito o de fracaso que se ha obtenido de ellas. En la **tabla 28** se puede ver especificado el criterio que se va a seguir para evaluar las pruebas.

Evaluación	Criterio
Buena	El caso de uso se completa con éxito, obteniendo el resultado esperado y con un funcionamiento adecuado.
Aceptable	El caso de uso se completa con éxito, el resultado obtenido es bueno, pero existen problemas de rendimiento.
Mejorable	El caso se completa de una forma parcial, dando un resultado cercano al esperado.
Bug	No se obtiene el resultado esperado.

**Tabla 28.** Clasificación de los resultados obtenidos en las pruebas.

## 6.2 Definición y resultado de las pruebas.

<b>TEST 1</b>	<b>Crear usuario</b>
Resultado esperado.	Mostrar al usuario un formulario con campos de texto de nombre, correo y contraseña necesarios además de un enlace para ver los términos y condiciones de uso que será indispensable su aceptación para poder proceder. Una vez introducidos los datos y aceptados los términos se deberá crear el usuario y pasará a la pantalla Home con el mapa.
Resultado obtenido.	Se crea el usuario y se pasa a la pantalla home de forma inmediata.
Evaluación.	Buena.

**Tabla 29.** Test 1, crear usuario.

<b>TEST 2</b>	<b>Iniciar sesión.</b>
Resultado esperado.	La aplicación mostrará un formulario donde solicitará correo electrónico y contraseña. El usuario con las credenciales usadas en el <b>TEST 1</b> iniciará sesión y la aplicación irá a la pantalla home con el mapa.
Resultado obtenido.	Se han introducido las credenciales y la aplicación ha mostrado la página de home inmediatamente.
Evaluación.	Buena.

**Tabla 30.** Test 2, iniciar sesión.

<b>TEST 3</b>	<b>Cerrar sesión.</b>
Resultado esperado.	El usuario accederá a la pantalla de ajustes y pulsará sobre el botón de cerrar sesión. Se cerrará la sesión y se abrirá la página de Inicio de sesión.
Resultado obtenido.	Se ha pulsado en el botón de cerrar sesión y se ha cerrado sesión correctamente y abierto la página de Inicio de sesión.
Evaluación.	Buena.

**Tabla 31.** Test 3, cerrar sesión.

<b>TEST 4</b>	<b>Recuperar contraseña.</b>
Resultado esperado.	<p>En el proceso de Inicio de sesión, accedemos a la página de recuperar la contraseña, donde la aplicación nos pedirá el correo con el que estamos registrados para enviarnos un correo con un enlace donde cambiar la contraseña.</p> <p>A continuación, accederemos a nuestro correo, para pulsar en el enlace que se nos ha enviado y cambiar la contraseña.</p> <p>Por último, iniciaremos sesión en la aplicación con la nueva contraseña para comprobar que todo ha ido bien.</p>
Resultado obtenido.	<p>Se ha introducido el correo y la aplicación nos ha reportado un mensaje diciendo que se nos enviaría un correo.</p> <p>Efectivamente, teníamos un correo con un enlace donde poder cambiar la contraseña, hemos introducido la nueva contraseña y finalmente hemos accedido a la cuenta correctamente con esta.</p>
Evaluación.	Bug.

Tabla 32. Test 4, recuperar contraseña.

<b>TEST 5</b>	<b>Buscar Localización.</b>
Resultado esperado.	<p>En la página home el usuario va a introducir en la entrada de texto que aparece sobre el mapa el texto 'Madr' para buscar Madrid, se desplegará una lista de opciones entre las que se encuentra Madrid, se selecciona y el mapa se mueve a la localización de Madrid y pone un Marcador azul sobre el mapa.</p>
Resultado obtenido.	<p>Se ha introducido el texto 'Madr' en la entrada de texto y se ha seleccionado la opción de Madrid en el desplegable, entonces el mapa se ha movido hasta la localización y ha pintado un marcador sobre el mapa.</p>
Evaluación.	Buena

Tabla 33. Test 5, buscar localización.

<b>TEST 6</b>	<b>Ver Mapa.</b>
Resultado esperado.	En la página home, donde aparece el mapa, se va a hacer zoom-in y zoom-out con dos dedos.
Resultado obtenido.	Se ha hecho zoom con dos dedos, la aplicación ha respondido correctamente.
Evaluación.	Buena.

Tabla 34. Test 6, ver mapa.

<b>TEST 7</b>	<b>Geolocalizar usuario.</b>
Resultado esperado.	En la pantalla de home, se va a pulsar el botón de geolocalizar usuario. La aplicación debe mover el mapa hasta la localización donde nos encontramos y pintar un punto sobre el mapa.
Resultado obtenido.	Se ha pulsado el botón de geolocalización y la aplicación ha movido el mapa hasta el punto donde nos encontrábamos probando la aplicación y ha dibujado un punto.
Evaluación.	Buena.

Tabla 35. Test 7, geolocalizar usuario.

<b>TEST 8</b>	<b>Crear paseo.</b>
Resultado esperado.	En la página de home vamos a pulsar el botón de crear paseo que aparece sobre el mapa, se introducirán los datos requeridos (punto de inicio, final, tiempo y ritmo) y se pulsará en el botón de vista previa. Después de esto, la aplicación pintará sobre el mapa todas las opciones obtenidas.  Antes de pulsar en el botón de vista previa se probará si faltando algún dato la aplicación te avisa de ello.
Resultado obtenido.	Primero se ha ido a crear un paseo sin punto de inicio y sin tiempo, la aplicación ha reaccionado correctamente y ha avisado de que no se puede.  Luego se han introducido todos los datos y ha pintado tres opciones de rutas tal y como se esperaba.
Evaluación.	Buena.

Tabla 36. Test 8, crear paseo.

<b>TEST 9</b>	<b>Guardar paseo.</b>
Resultado esperado.	Entre las opciones ofrecidas por la aplicación, se seleccionará una de ellas y se pulsará el botón de guardar paseo. El sistema nos debe informar si la operación ha sido exitosa. Por último, abriremos el menú lateral para abrir la página de paseos guardados y comprobar que aparece el nuevo paseo llamado 'Pruebas'.  Antes de guardar el paseo se comprobará que no se puede guardar un paseo sin un nombre introducido.
Resultado obtenido.	Se ha pulsado el botón de guardar paseo sin introducir un nombre y efectivamente la aplicación no ha permitido guardarlo.  Luego se ha introducido un nombre, se ha pulsado el botón de guardar y al abrir la página de paseos guardados ahí estaba.
Evaluación.	Buena.

Tabla 37. Test 9, guardar paseo.

<b>TEST 10</b>	<b>Seguir paseo.</b>
Resultado esperado.	Desde la página de paseos guardados se seleccionará el paseo guardado en el <b>TEST 9</b> . Se abrirá la página de seguimiento de paseo centrada en el paseo y con la ruta pintada sobre el mapa. Pulsaremos en el botón de geolocalizar y al movernos el punto de geolocalización nos seguirá.  También se controlará el tiempo que se ha tardado en realizar la ruta dibujada en el mapa y al ritmo elegido.
Resultado obtenido.	Hemos abierto el paseo creado en el <b>TEST 9</b> y ha abierto el paseo sobre el mapa correctamente. A continuación, hemos pulsado el botón de geolocalización y nos ha seguido correctamente.  En cuanto al tiempo tardado en realizar la ruta ha habido un error de dos minutos en un total de veinte.
Evaluación.	Aceptable.

Tabla 38. Test 10, seguir paseo.

<b>TEST 11</b>	<b>Eliminar un paseo.</b>
Resultado esperado.	En la página de seguimiento, se pulsará en el icono de la papelera, la aplicación pedirá una confirmación para eliminar el paseo, confirmaremos y la aplicación se irá a la página de home con el mapa.
Resultado obtenido.	El esperado, y luego se ha comprobado en la lista de paseos guardados y efectivamente ya no aparecía.
Evaluación.	Buena.

Tabla 39. Test 11, eliminar paseo.

### 6.3 Pruebas de usuarios reales.

En este apartado lo que se ha hecho ha sido pasar el '.apk' a diez personas del entorno del alumno para que probaran la aplicación y ofrecieran feedback desde un punto de vista de alguien que no ha desarrollado la aplicación. Los comentarios obtenidos han sido los siguientes:

- Cuatro de los usuarios han reportado que en la página de seguimiento de paseo quedaría mejor y sería más claro si se usara un color que no fuera naranja para la ruta y que además se introdujeran marcadores tal y como funcionaban en la creación del paseo. Respecto a estos comentarios se ha actuado introduciendo los marcadores con el mismo código de color que en la creación de los paseos y dibujando la ruta sobre el mapa con un gradiente de color que va del azul al rojo, pasando por el amarillo y el verde.
- Un usuario se dio cuenta que si tras abrir la aplicación y se pulsaba en el botón de retroceder del dispositivo, la aplicación volvía a la página de inicio de sesión y aparentemente no se había iniciado sesión cuando en verdad la sesión sí que estaba iniciada. Para solucionar esto lo que se ha hecho ha sido que en caso de que la sesión esté iniciada, si se accede a la pagina de inicio de sesión, se muestra un mensaje que informa de que la sesión ya está iniciada.
- El total de los diez usuarios reportaron que al guardar estaría bien saber que todo ha ido bien y se ha guardado correctamente. Para ello se ha añadido un Toast, que se lanza cada vez que se guarda un paseo satisfactoriamente.
- Uno de los usuarios que probó la aplicación reportó que estaría bien que en el seguimiento de los paseos la aplicación informara sobre el ritmo y guiara al usuario con indicaciones sobre la ruta. Esta implementación nos pareció muy interesante, pero sin embargo se ha tenido que descartar ya que implicaba mucho tiempo de desarrollo.
- Por último, también ha sido general el comentario reportando que sería interesante que la aplicación diera la opción de editar un paseo ya creado, ya sea añadiendo un punto intermedio por el que pasar o cambiando los atributos usados al crear el paseo. Al igual que con el tema del seguimiento del paseo, son desarrollos que conllevarían mucho desarrollo, por lo que se han añadido al apartado de trabajo futuro.

## Capítulo 7: Conclusiones y trabajo futuro.

Tras la realización del trabajo, en este capítulo se echará la vista atrás para ver qué se ha aprendido durante su realización y qué posibles implementaciones mejorarían el producto final.

### 7.1 Conclusiones.

A modo de resumen se puede ver como el trabajo realizado en este proyecto ha dado como resultado una aplicación Android desarrollada mediante técnicas web, haciendo uso de un Framework especializado en el desarrollo de aplicaciones híbridas (Ionic). Por lo que el paso a tener una aplicación funcional en múltiples plataformas debería de ser relativamente directo a partir del trabajo ya hecho.

Por otro lado, en este proyecto se ha trabajado en profundidad con el gestor de mapas Mapbox y sus APIs, lo que ha supuesto un reto dado que era la primera vez y, aunque la línea de aprendizaje al principio ha sido lenta, al final se han llegado a entender con claridad los conceptos usados, y gracias a la gran comunidad de desarrolladores se ha conseguido implementar la aplicación con éxito.

También, cabe destacar el uso de una base de datos no relacional, algo que durante el estudio del grado no se aborda en profundidad, pero el hecho de poder trabajar en un proyecto completo permite entender la gran cantidad de ventajas que este tipo de bases de datos aporta. Como puede ser la flexibilidad a la hora del desarrollo, la velocidad de estas, la escalabilidad, etc. Además, permite entender por qué actualmente el uso de este tipo de base de datos está aumentando y cada vez está presente en más sistemas de los que usamos en nuestro día a día.

Por último, y no menos importante, la metodología usada para la gestión del proyecto ha sido Scrum, lo que ha permitido una mayor flexibilidad con respecto a los requisitos iniciales del sistema y ha ayudado a mantener la motivación en todo momento ya que una de las filosofías básicas de Scrum consiste en tener un avance del producto cada cierto tiempo (Sprint), lo cual permite a las personas que trabajan en el proyecto ver que su esfuerzo va a buen puerto y que cada vez se está más cerca del objetivo.

### 7.2 Trabajo futuro.

En este apartado se expondrán una serie de desarrollos que se podrían realizar en un futuro para mejorar la aplicación y aumentar sus funcionalidades:

- Con el trabajo ya hecho y gracias a Ionic 4 sería interesante adaptar la aplicación para que funcionara en plataformas web e IOS. Esto debería ser un trabajo sencillo de lograr, pero sería necesario adaptar las interfaces a estas otras plataformas lo cual llevaría más trabajo.
- Mejorar el seguimiento de los paseos, como reportó uno de los usuarios que probó la aplicación, estaría bien que la aplicación te informara si el ritmo está siendo bueno y que te diera indicaciones de por dónde va el paseo.
- Para aumentar el público objetivo de la aplicación sería interesante introducir algún tipo de gamificación en los paseos, ya sea llevando la cuenta de los paseos e introduciendo un sistema de puntos y socialización con otras cuentas, o bien que durante el seguimiento del paseo se introdujeran funciones que hagan del paseo parecer un juego.
- Añadir opciones de socialización en la aplicación, introduciendo funcionalidades para compartir paseos e importar paseos de otros usuarios.
- Mejorar el algoritmo de creación de rutas evitando que un paseo pase dos veces por la misma calle o añadiendo la posibilidad de que el usuario pueda seleccionar un punto intermedio por el que quiere que pase el paseo.
- Añadir funcionalidad para editar un paseo ya creado, cambiando los valores introducidos en la creación.



# Webgrafía y Bibliografía.

- [1] Martínez, S. (2020, 7 octubre). Qué es Open Street Maps y cómo descargar sus datos. Cursos GIS | TYC GIS Formación. <https://www.cursosgis.com/que-es-open-street-maps-y-como-descargar-sus-datos/>
- [2] Beneficios de Scrum. (2008, 4 agosto). Beneficios de Scrum. <https://proyectosagiles.org/beneficios-de-scrum/>
- [3] Qué es SCRUM. (2018, 9 octubre). Proyectos Ágiles. <https://proyectosagiles.org/que-es-scrum/>
- [4] Scrum: roles y responsabilidades. (2019, 14 enero). Deloitte Spain. <https://www2.deloitte.com/es/es/pages/technology/articles/roles-y-responsabilidades-scrum.html>
- [5] Hughes, B., & Cotterell, M. (2009). Software Project Management (5.a ed.). McGraw-Hill Education.
- [6] Ministerio de Empleo y Seguridad Social. (2018, marzo). Resolución de 22 de febrero de 2018, de la Dirección General de Empleo, por la que se registra y publica el XVII Convenio colectivo estatal de empresas de consultoría y estudios de mercado y de la opinión pública. BOE. <https://www.boe.es/boe/dias/2018/03/06/pdfs/BOE-A-2018-3156.pdf>
- [7] Open-Source UI Toolkit to Create Your Own Mobile or Desktop Apps. (2020, 7 diciembre). Ionic Docs. <https://ionicframework.com/docs>
- [8] Patil, S. (2019). Two-way Binding in Angular. Two way data binding. <https://www.linkedin.com/pulse/two-way-binding-angular-sagar-patil/>
- [9] Add Firebase to your JavaScript project. (2021, 18 junio). Documentación Firebase. <https://firebase.google.com/docs/web/setup>
- [10] Documentation. (s. f.). Documentación Mapbox. Recuperado 25 de junio de 2021, de <https://docs.mapbox.com/>
- [11] Turf.js | Advanced Geospatial Analysis. (s. f.). Documentación Turf.js. Recuperado 25 de junio de 2021, de <https://turfjs.org/>
- [12] I. (2021). Your First Ionic App: Angular - Ionic Documentation. Ionic Docs. <https://ionicframework.com/docs/angular/your-first-app>
- [13] Kumar, V. (2017). How to Integrate Mapbox GL JS Maps with an Ionic Application. Information and Technology Blog - Gowriter. <https://www.gowriter.in/2017/02/how-to-integrate-mapbox-gl-js-maps-with.html>
- [14] Protección de Datos conforme al reglamento RGPD. (2021, 26 marzo). Your Europe. [https://europa.eu/youreurope/business/dealing-with-customers/data-protection/data-protection-gdpr/index\\_es.htm#shortcut-0](https://europa.eu/youreurope/business/dealing-with-customers/data-protection/data-protection-gdpr/index_es.htm#shortcut-0)
- [15] Fernández, Y. (2021, 23 marzo). WebView del sistema Android: qué es, qué beneficios tiene y por qué hay que tenerlo actualizado. Xataka. <https://www.xataka.com/basics/webview-sistema-android-que-que-beneficios-tiene-que-hay-que-tenerlo-actualizado>
- [16] Download Android Studio and SDK tools | Android Studio. (s. f.). Android Developers. Recuperado 25 de junio de 2021, de <https://developer.android.com/studio?hl=es>
- [17] Primeros pasos: Escribe, prueba e implementa tus primeras funciones. (s. f.). Firebase, primeros pasos. Recuperado 25 de junio de 2021, de <https://firebase.google.com/docs/functions/get-started?hl=es>
- [18] API Reference | Mapbox GL JS. (s. f.). Mapbox. Recuperado 25 de junio de 2021, de <https://docs.mapbox.com/mapbox-gl-js/api/>
- [19] colaboradores de Wikipedia. (2020, 2 abril). Mapa isócrono. Wikipedia, la enciclopedia libre. [https://es.wikipedia.org/wiki/Mapa\\_is%C3%B3crono](https://es.wikipedia.org/wiki/Mapa_is%C3%B3crono)

- [20] Privacidad y seguridad en. (s. f.). Firebase. Recuperado 25 de junio de 2021, de [https://firebase.google.com/support/privacy?hl=es-419#data\\_encryption](https://firebase.google.com/support/privacy?hl=es-419#data_encryption)

## Anexo I: Guía de Instalación.

En este anexo, se describirá el proceso requerido para poner la aplicación en marcha en un smartphone con sistema operativo Android superior a la versión 5.0 y tener el WebView instalado, que por defecto viene en todos los dispositivos Android.

Lo primero será hacerse con el archivo 'PaseAppV1.0.apk' que podemos encontrar en el GitLab dedicado al proyecto y tener controlada su ubicación en el dispositivo. El archivo se puede encontrar en la última versión de entrega existente (Resumen del proyecto > Versiones) o bien en el directorio raíz del proyecto. La dirección de acceso a este repositorio es la siguiente:

<https://gitlab.inf.uva.es/alepesc/PaseApp>

El segundo paso consiste en activar los permisos para instalar aplicaciones desconocidas lo cual será necesario ya que el '.apk' no se encuentra firmado de forma oficial y no está distribuido por los servicios de Google Play. Para activar estos permisos hay que acceder a la siguiente opción dentro de los ajustes del dispositivo: Ajustes > Protección y privacidad > Permisos especiales > Instalar aplicaciones desconocidas. Este proceso puede variar dependiendo del dispositivo, pero en general suele ser algo parecido a esto.

El tercer paso es abrir un gestor de archivos instalado en el teléfono, ir a la ubicación donde se encuentra el archivo '.apk', pulsar sobre él y aceptar la instalación de la aplicación en el dispositivo. Con este paso la aplicación ya estaría instalada en el teléfono.

Por último, para poder utilizar la aplicación hay que darle permisos para utilizar la ubicación del dispositivo, asegurarnos de tener activa la ubicación y tener una conexión a internet estable. Para dar el permiso de acceso a la ubicación a la aplicación, cuando se vaya a abrir la aplicación ésta te solicitará el permiso y automáticamente te abrirá los ajustes donde darle esos permisos. Sin embargo, si se quisieran dar los permisos manualmente hay que acceder a la siguiente opción dentro de los ajustes del dispositivo: Ajustes > Aplicaciones > PaseApp > Permisos > Permisos de Ubicación. Al igual que antes, las opciones para dar permisos pueden variar entre dispositivos, pero en la mayoría serán muy parecidas.



## Anexo II: Guía de usuario.

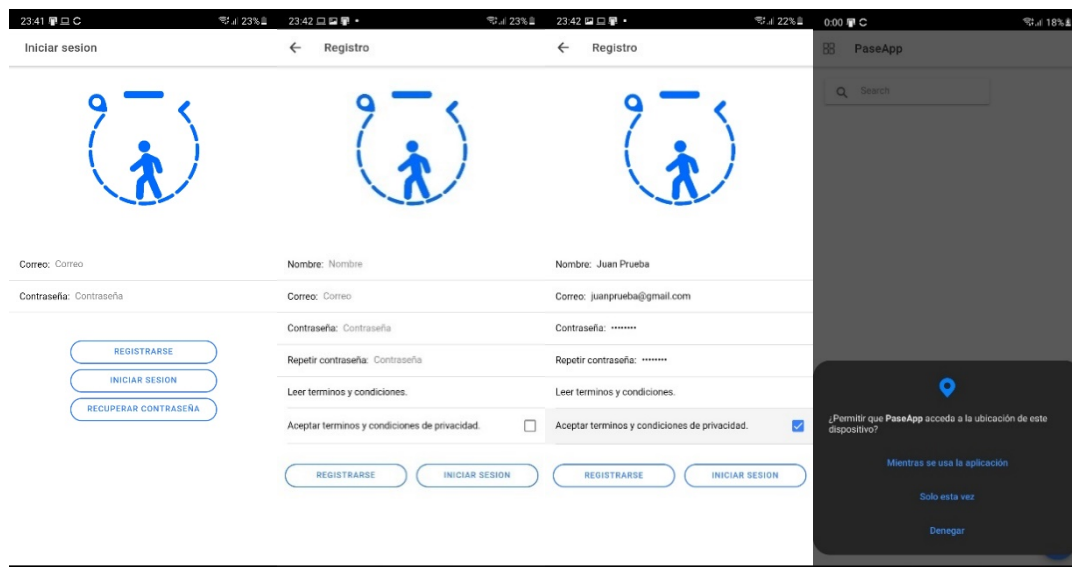
Una vez seguidos los pasos del **anexo I** y ya teniendo la aplicación instalada y funcionando en un dispositivo, vamos a explicar con ejemplos el funcionamiento de la aplicación.

### Registro de usuario e inicio de sesión.

Lo primero que nos encontraremos al entrar en la aplicación será con la página de inicio de sesión (**Figura 32** a la izquierda) donde se nos pedirá un correo asociado a una cuenta y su contraseña. En caso de no tener aún una cuenta lo primero que se debe hacer es registrarse. Para realizar el registro, desde la página de inicio de sesión se pulsará en el botón de registro y se accederá a la pantalla de registro (**Figura 32** las dos capturas centrales) de usuario donde el usuario deberá introducir su nombre, correo electrónico, contraseña y aceptar los términos y condiciones de uso de la aplicación. Si quisiera leer los términos, en esta página tenemos un botón donde pone “Leer términos y condiciones”, el cual nos llevará a una página con el texto.

Una vez introducidos los datos y pulsado al botón de registrar usuario, se nos abrirá la página de Home con el mapa (**Figura 32** a la derecha del todo), y en caso de no haber dado aún permisos para acceder a la ubicación, se nos abrirá un cuadro de diálogo donde activar dichos permisos y la sesión de la nueva cuenta se iniciará automáticamente.

En el caso de ya tener una cuenta creada en el sistema lo único que habría que hacer para iniciar sesión sería introducir tu correo y contraseña en la página de inicio de sesión.



**Figura 32.** Pagina de inicio de sesión y de registro de usuario, en la creación de un nuevo usuario.

### Búsqueda de una localización.

Una vez iniciada la sesión y estando en la página de Home (**Figura 33**) podemos hacer búsqueda de localizaciones. Para ello lo que hay que hacer es introducir el nombre de la localización que queremos en el cuadro de búsqueda de la parte superior izquierda lo que desplegará una lista de posibles localizaciones debajo del cuadro de texto, seleccionaremos la que más se ajuste a nuestra búsqueda y entonces el mapa dibujará un marcador sobre dicha localización y se trasladará hasta ella. Se puede ver un ejemplo en la **Figura 33** haciendo una búsqueda de la ciudad de Valladolid.

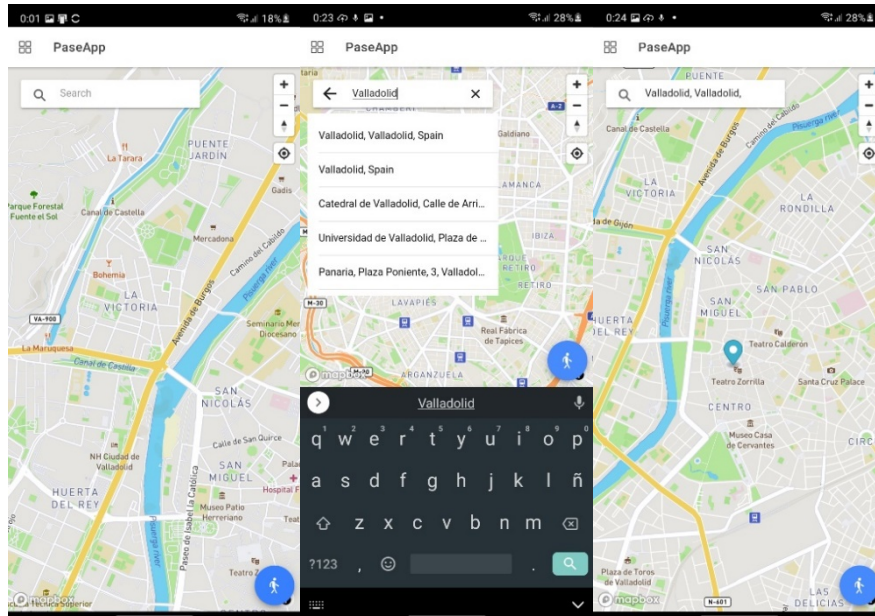


Figura 33. Pagina Home con mapa y búsqueda de localización.

## Creación y guardado de un paso

La creación de un paseo se hace desde la página Home, a través del menú que se despliega desde la parte inferior de la pantalla al pulsar el botón azul que aparece sobre el mapa en la parte inferior derecha.

Una vez desplegado el menú de creación de paseos (**Figura 34**) lo que hay que hacer es introducir en formato de texto la ubicación donde queremos que comience el paseo, con la misma dinámica que la búsqueda de una localización. Una vez añadido el punto de inicio se pintará sobre la localización deseada un marcador de color verde que se podrá ajustar arrastrándolo con el dedo por el mapa, y ahí donde se deje el marcador, será el punto de inicio del paseo. Para el punto de finalización del paseo, hacemos el mismo proceso que con el punto de origen, esta vez el marcador que aparecerá será rojo (**Figura 34**).

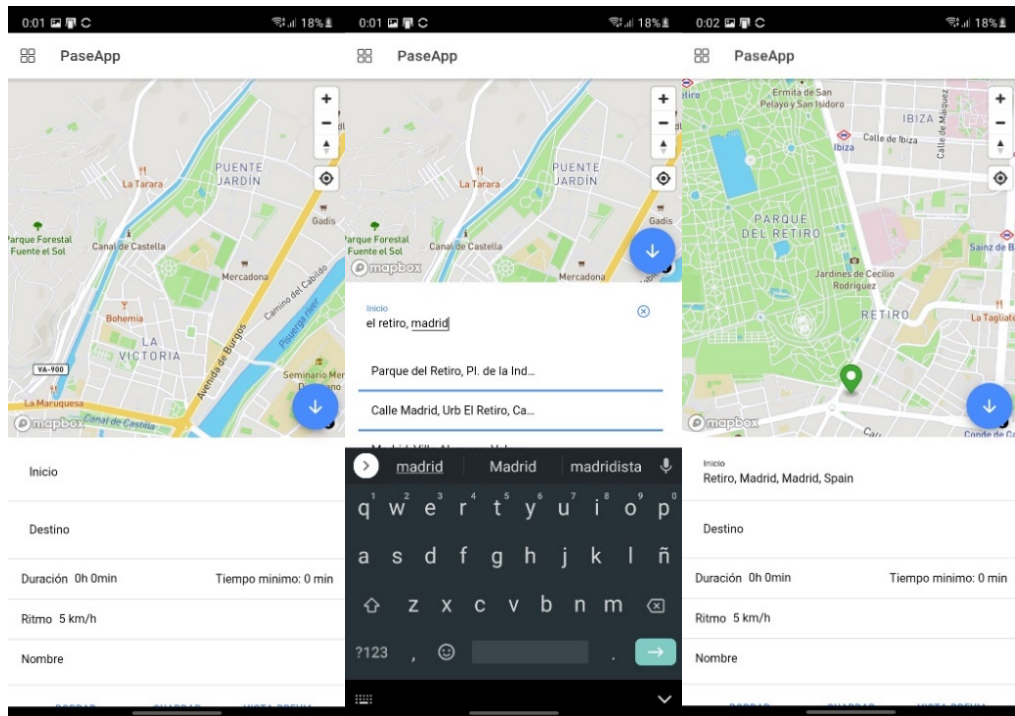


Figura 34. Pagina Home en la creación de un paseo, añadiendo el punto de inicio.

Lo siguiente, será seleccionar el tiempo y el ritmo que se desea que dure el paseo. Para ello hay que pulsar donde pone “Duración” y seleccionar una de las opciones posibles, y hacer lo mismo con el ritmo. En la figura 35 se puede ver un ejemplo de cómo se selecciona la duración y el ritmo del paseo.

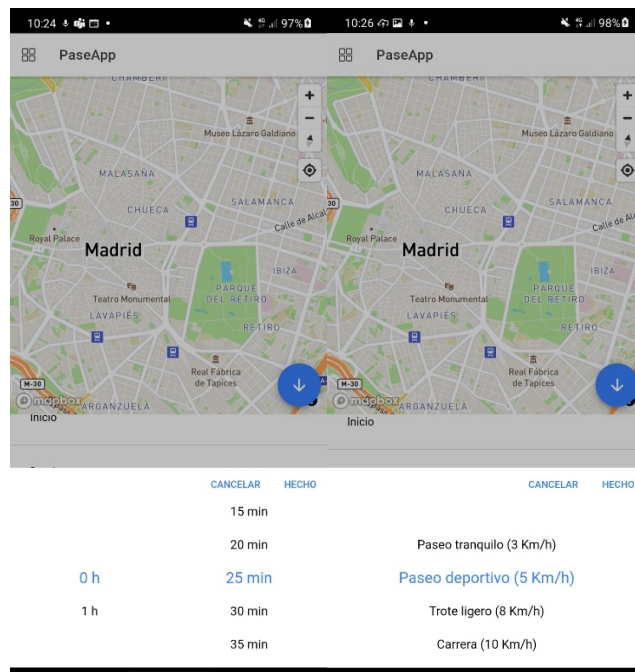
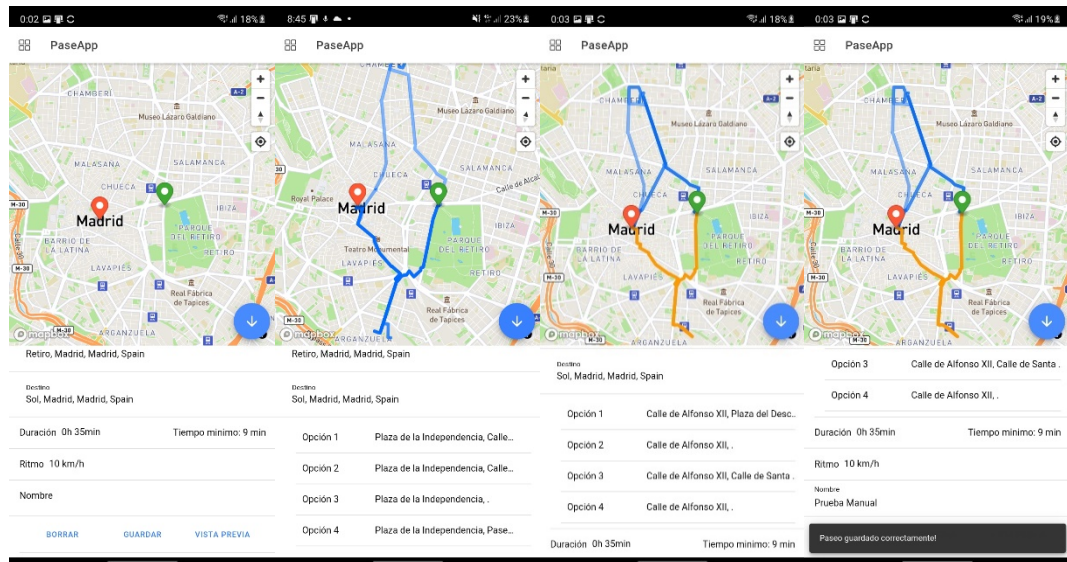


Figura 35. Pagina Home, ejemplo de la selección de ritmo y duración en la creación de un paseo.

Por último, para crear un paseo, hay que pulsar en el botón de “vista previa”, la aplicación calculará todas las opciones y las dibujará sobre el mapa. El usuario tendrá que seleccionar una de ellas, ya sea pulsando sobre la ruta que quiere o bien pulsando en la lista de opciones que se desplegará en el menú de creación de paseos.





**Figura 36.** Pagina Home, añadiendo punto de fin, duración, ritmo, generando opciones de paseo y seleccionando una de ellas.

Una vez se ha creado el paseo, si se desea guardar, lo único que hay que hacer es añadir un texto en el campo de nombre y pulsar sobre el botón “guardar”, el paseo se guardará y si todo ha ido bien la aplicación mostrará un mensaje indicándolo.

### Seguimiento de un paseo.

Para hacer el seguimiento de un paseo, previamente éste debe de haber sido creado y guardado. Con el paseo creado y guardado, accedemos a la lista de paseos guardados desde del menú lateral, que se puede abrir desde cualquier punto de la aplicación deslizando desde el lateral izquierdo o bien desde la página de Home, pulsando en el botón que aparece en la parte izquierda de la cabecera.

En el menú lateral pulsamos sobre el texto “Paseos Guardados”, lo que nos llevará a la página de paseos guardados donde seleccionaremos el paseo que deseamos seguir. Esto abrirá la pagina de seguimiento donde se verá la ruta pintada sobre el mapa y los dos marcadores de inicio y final. Una vez llegados a este punto, si pulsamos en el botón de geolocalización que aparece en la parte superior derecha del mapa, se dibujará en el mapa un punto situado en el lugar donde nos encontramos y que nos seguirá según nos vayamos moviendo, pudiendo ver en todo momento donde nos encontramos y por dónde hay que seguir el paseo.



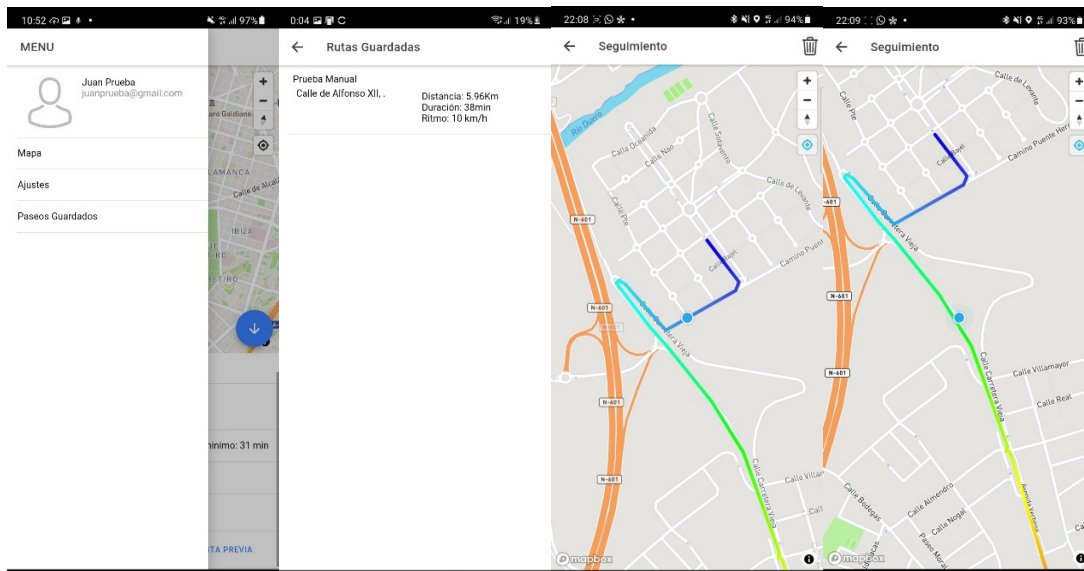


Figura 37. Manu lateral, página de paseos guardados y de seguimiento.

### Eliminar un paseo.

Si el usuario desea eliminar un paseo, lo debe hacer desde la página de seguimiento de paseo, en la parte derecha de la cabecera aparecerá un icono de una papelera, el usuario deberá pulsar sobre él e inmediatamente le aparecerá una ventana que le pedirá confirmar si desea eliminar el paseo. Una vez confirmas, el paseo se elimina y la aplicación nos lleva a la página Home.

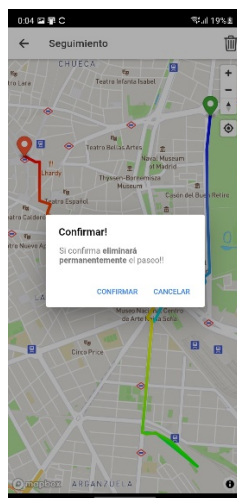
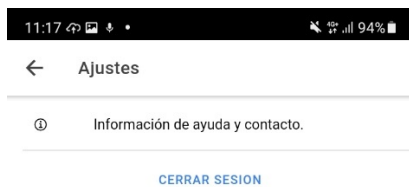


Figura 38. Pagina de seguimiento, confirmar eliminar un paseo.

### Cerrar sesión.

Para cerrar sesión hay que acceder a la página de Ajustes desde el menú lateral, y una vez abierta esta pestaña pulsar sobre el botón de cerrar sesión. La aplicación cerrará la sesión y se irá a la página de inicio de sesión.



**Figura 39.** Pagina de ajustes, cerrar sesión.