



Aplicación de navegación GPS para iOS

GRADO EN INGENIERÍA INFORMÁTICA DEL SOFTWARE

TRABAJO FIN DE GRADO

AUTOR

Marcos Fernández González

TUTOR

Cesar Fernández Acebal

Junio 2023

Copyright (C) 2020 **ELENA ALLEGUE GONZÁLEZ, JOSÉ MANUEL REDONDO LÓPEZ**

Teaching Innovation Project: PINN-19-A-029 (University of Oviedo)

This work has been published in [1] [2]

Esta versión de la plantilla para Trabajos de Fin de Grado ha sido posible gracias a la donación de la ex-alumna Elena Allegue González de su documentación de Trabajo de Fin de Grado, que ha servido como base para elaborar esta versión. Aquí podréis encontrar todos los títulos y subtítulos de las secciones, pero las explicaciones se mantendrán en la versión *Word* de la plantilla (se proporciona una versión PDF de la misma para facilitar el acceso a las mismas). No obstante, del trabajo de Elena se han conservado ejemplos de como hacer elementos clave como imágenes, tablas, etc.

Desarrollar una versión *Latex* de la plantilla desde cero es una trabajo bastante largo, pero gracias al trabajo de Elena se ha podido equiparar esta versión con las de *Word* mucho más rápidamente.

Agradecimientos

Lo primero de todo agradecerle a César su labor como tutor. De él nació la idea de este proyecto que he disfrutado llevándolo a cabo. Sus directrices han sido muy importantes y valiosas a lo largo de todo el proceso.

Continuando, escribo este capítulo como el último de mi trabajo, no porque sea lo más fácil ni lo que menos importa. Al contrario. Lo escribo ahora para dedicarle el tiempo que se merece, ya que en él están las personas más importantes mi vida, por las que he escrito cada una de las palabras de este documento, porque sé que me desean todo el éxito del mundo en mi carrera académica, laboral o en cualquier aspecto de mi vida y esto solo es una pequeña manera de corresponderles.

Gracias a mi madre, por su paciencia y apoyo en todo momento. Aunque a veces se desespere conmigo y forma de trabajar, sé que hace un gran esfuerzo por entender cómo soy y por qué hago las cosas como las hago.

Gracias a mi padre por tener en mí una fe ciega. Incluso en los momentos más difíciles he sentido una gran confianza en mí por su parte.

Gracias a mi hermana por todos sus consejos, que aunque crea que no le escucho, sí le termino haciendo caso y valoro mucho todo lo que me dice.

Gracias a mis amigos Javi y Toni, sin ellos no hubiera hecho este trabajo ni nada de lo que hago en mi vida ni la mitad de bien. Me fuerzan a ser la mejor versión de mi mismo y mejorar cada día.

Gracias a Laura, mi novia, la cual me ha acompañado todo el camino, desde el día que elegí este trabajo hasta el día que lo entregué. Cerramos juntos esta etapa del grado para empezar y continuar otras, en las que seguiremos compartiendo todos los momentos, buenos y malos, donde nos apoyaremos mutuamente como llevamos haciendo ya mucho tiempo.

Y aunque al final todos los que nos graduamos, ya sea de Ingeniería Informática o cualquier otra carrera, hacemos un Trabajo Fin de Grado, quiero agradecerme a mí mismo el no escoger el camino fácil, dar el 100% de mí e ir aprendiendo cada día a ser una mejor persona.

Índice general

1. ¿Qué es este trabajo?	11
1.1. Resumen	11
1.2. Palabras Clave	11
1.3. Abstract	12
1.4. Keywords	12
2. PLANIFICACIÓN DEL SISTEMA DE INFORMACIÓN	13
2.1. INICIO DEL PLAN DE SISTEMAS DE INFORMACIÓN	14
2.1.1. Análisis de la Necesidad del PSI	14
2.1.2. Identificación del Alcance del PSI	14
2.2. DEFINICIÓN Y ORGANIZACIÓN DEL PSI	16
2.2.1. Especificación del Ámbito y Alcance	16
2.2.2. Organización del PSI	17
2.3. ESTUDIO DE LA INFORMACIÓN RELEVANTE	18
2.3.1. Selección y Análisis de Antecedentes	18
3. DEFINICIÓN DE LA ARQUITECTURA TECNOLÓGICA	19
3.1. Identificación de las Necesidades de Infraestructura Tecnológica	20
3.2. Selección de la Arquitectura Tecnológica	23
4. ESTUDIO DE VIABILIDAD DEL SISTEMA	24
4.1. ESTUDIO Y VALORACIÓN DE ALTERNATIVAS DE SOLUCIÓN. SELECCIÓN DE ALTERNATIVA FINAL	25
4.1.1. Wikiloc	25
4.1.2. Strava	28
4.1.3. AllTrails	30
4.1.4. Selección final	32
5. PLANIFICACIÓN Y GESTIÓN DEL TFG	33
5.1. PLANIFICACIÓN DEL PROYECTO	34
5.1.1. Identificación de Interesados	34
5.1.2. OBS y PBS	34
5.1.3. Planificación Inicial. WBS	37
5.1.4. Riesgos	43
5.1.5. Presupuesto Inicial	45

5.2.	EJECUCIÓN DEL PROYECTO	50
5.2.1.	Bitácora de Incidencias del Proyecto	50
5.3.	CIERRE DEL PROYECTO	51
5.3.1.	Planificación Final	51
5.3.2.	Presupuesto Final de Costes	51
6.	ANÁLISIS DEL SISTEMA DE INFORMACIÓN	53
6.1.	DEFINICIÓN DEL SISTEMA	54
6.1.1.	Determinación del Alcance del Sistema	54
6.2.	ESTABLECIMIENTO DE REQUISITOS	55
6.2.1.	Obtención de los Requisitos del Sistema	55
6.2.2.	Identificación de Actores del Sistema	58
6.2.3.	Especificación de Casos de Uso	58
6.3.	ANÁLISIS DE LOS CASOS DE USO	61
6.3.1.	Caso de Uso 1: Importar ruta GPX	61
6.3.2.	Caso de Uso 2: Ver rutas importadas	62
6.3.3.	Caso de Uso 3: Enviar ruta importada	62
6.3.4.	Caso de Uso 4: Seleccionar ruta importada	63
6.3.5.	Caso de Uso 5: Registrar actividad	63
6.3.6.	Caso de Uso 6: Seleccionar una actividad registrada	64
6.3.7.	Caso de Uso 7: Ver actividades registradas	64
6.3.8.	Caso de Uso 8: Enviar actividad registrada	65
6.3.9.	Caso de Uso 9: Añadir capa al mapa	65
6.3.10.	Caso de Uso 10: Añadir waypoint	66
6.4.	ANÁLISIS DE CLASES	67
6.4.1.	Diagrama de Clases	67
6.4.2.	Descripción de las Clases	68
6.5.	DEFINICIÓN DE INTERFACES DE USUARIO	71
6.5.1.	Definición del aspecto de la interfaz	71
6.5.2.	Diagrama de Navegabilidad	76
6.6.	ESPECIFICACIÓN DEL PLAN DE PRUEBAS	77
6.6.1.	Pruebas unitarias	77
6.6.2.	Pruebas de usabilidad	79
7.	DISEÑO DEL SISTEMA DE INFORMACIÓN	80
7.1.	DISEÑO DE CASOS DE USO REALES	81
7.1.1.	Caso de Uso 5: Registrar actividad	81
7.2.	DISEÑO DE LA ARQUITECTURA DE MÓDULOS DEL SISTEMA	84
7.2.1.	Diseño de módulos del sistema	84
7.3.	DISEÑO DE CLASES	85
7.3.1.	Diagrama de Clases	85
7.4.	ESPECIFICACIÓN TÉCNICA DEL PLAN DE PRUEBAS	88
7.4.1.	Pruebas Unitarias y del Sistema	88
7.4.2.	Pruebas de Usabilidad y Accesibilidad	93

8. CONSTRUCCIÓN DEL SISTEMA DE INFORMACIÓN	97
8.1. PREPARACIÓN DEL ENTORNO DE GENERACIÓN Y CONSTRUCCIÓN	98
8.1.1. Estándares y normas seguidos	98
8.1.2. Lenguajes de programación	99
8.1.3. Librerías y frameworks	100
8.1.4. Herramientas y programas usados para el desarrollo	102
8.2. GENERACIÓN DEL CÓDIGO DE LOS COMPONENTES Y PROCEDIMIENTOS	106
8.2.1. Track	106
8.2.2. Map	108
8.3. EJECUCIÓN DE LAS PRUEBAS UNITARIAS	110
8.4. EJECUCIÓN DE LAS PRUEBAS DEL SISTEMA	114
8.4.1. Prueba de Usabilidad y accesibilidad	114
8.5. ELABORACIÓN DE LOS MANUALES DE USUARIO	126
8.5.1. Manual de Instalación	126
8.5.2. Manual de Ejecución	127
8.5.3. Manual de Usuario	127
8.5.4. Manual del Programador	140
9. CONCLUSIONES Y AMPLIACIONES	142
9.1. CONCLUSIONES	143
9.2. AMPLIACIONES	144
9.2.1. Usuarios	144
9.2.2. Base de datos	144
9.2.3. Trabajar con otro tipo de archivos (KML, GeoJSON...)	144
9.2.4. Otros sistemas de visualización de mapas	144
ANEXOS	145
PLAN DE GESTIÓN DE RIESGOS	146
9.2.5. Registro y valoración de los riesgos	146
9.2.6. Gestión de los riesgos	150

Índice de figuras

4.1. Logo de Wikiloc	25
4.2. Pantalla inicial de Wikiloc	26
4.3. Visualización del mapa de Wikiloc	27
4.4. Capas aplicables al mapa de Wikiloc	27
4.5. Logo de Strava	28
4.6. Pantalla inicial de Strava	29
4.7. Visualización del mapa de Strava	29
4.8. Logo de AllTrails	30
4.9. Pantalla inicial de AllTrails	31
4.10. Visualización del mapa de AllTrails	31
4.11. Capas aplicables al mapa de AllTrails	32
5.1. OBS	34
5.2. PBS - Primera parte	35
5.3. PBS - Segunda parte	36
5.4. PBS - Tercera parte	37
5.5. Planficación general, tabla de tareas	38
5.6. Planficación general, diagrama de Gantt	38
5.7. Planficación de la planificación, tabla de tareas	39
5.8. Planficación de la planificación, diagrama de Gantt	39
5.9. Planficación de la documentación, tabla de tareas	40
5.10. Planficación de la documentación, diagrama de Gantt	40
5.11. Planficación del análisis, tabla de tareas	41
5.12. Planficación del análisis, diagrama de Gantt	41
5.13. Planficación del diseño, tabla de tareas	42
5.14. Planficación del diseño, diagrama de Gantt	42
5.15. Planficación del desarrollo, tabla de tareas	42
5.16. Planficación del desarrollo, diagrama de Gantt	43
5.17. Planficación de las pruebas, tabla de tareas	43
5.18. Planficación de las pruebas, diagrama de Gantt	43
5.19. Tabla de precios de los recursos de personal	45
5.20. Presupuesto agregado	45
5.21. Partida Planificación	46
5.22. Partida Documentación	46
5.23. Partida Análisis	47

5.24. Partida Diseño	47
5.25. Partida Desarrollo	47
5.26. Partida Pruebas	48
5.27. Partida Otros Costes	48
5.28. Tabla con datos para cálculos de presupuesto de cliente	48
5.29. Presupuesto de cliente	49
5.30. Planficación final	51
5.31. Presupuesto agregado final	51
5.32. Tabla con datos para cálculos de presupuesto final de cliente	52
5.33. Presupuesto de cliente final	52
6.1. Diagrama de casos de uso	58
6.2. Diagrama de casos de uso	67
6.3. Prototipo de la pantalla inicial	71
6.4. Prototipo de vista de rutas importadas	72
6.5. Prototipo de vista registrar actividad	73
6.6. Prototipo de vista registrar actividad	74
6.7. Prototipo de vista importar ruta	75
6.8. Diagrama de navegabilidad	76
7.1. Diagrama de secuencia del Caso de Uso Ver rutas importadas	81
7.2. Diagrama de secuencia del Caso de Seleccionar ruta importada	82
7.3. Diagrama de secuencia del Caso de Uso Registrar actividad	83
7.4. Diagrama de paquetes	84
7.5. Paquete de clases relacionadas con el mapa	85
7.6. Paquete de clases relacionadas con el manejo de ficheros	87
8.1. MVVM	98
8.2. GPX	99
8.3. Logo de Swift	99
8.4. Logo de SwiftUI	100
8.5. Logo de CoreGPX	100
8.6. Logo de Google Maps	101
8.7. Logo de Swinject	101
8.8. Ejemplos de AlertToast	102
8.9. Logo de Xcode	102
8.10. Logo de CocoaPods	103
8.11. Logo de Overleaf	103
8.12. Logo de MS Project	104
8.13. Logo de Excel	104
8.14. Logo de PowerPoint	105
8.15. Logo de Visual Paradigm	105
8.16. Primera parte de la documentación de la estructura Track	106
8.17. Segunda parte de la documentación de la estructura Track	107
8.18. Tercera parte de la documentación de la estructura Track	107
8.19. Cuarta parte de la documentación de la estructura Track	108
8.20. Primera parte de la documentación del protocolo Map	108

8.21. Segunda parte de la documentación del protocolo Map	109
8.22. Resultado de pruebas unitarias	110
8.23. Desplegable para seleccionar el móvil donde instalar la aplicación	127
8.24. Aviso de uso localización al usuario	128
8.25. Aviso de uso localización al usuario	129
8.26. Capas aplicables al mapa desplegadas	130
8.27. Capa seleccionada aplicada sobre el mapa	130
8.28. Sistema de archivos del sistema para seleccionar archivo GPX	131
8.29. Mensaje de éxito de importación	132
8.30. Selección de medio multimedia para enviar un archivo GPX desde la aplicación . . .	133
8.31. Selección de una ruta	134
8.32. Ruta iniciada	134
8.33. Registro de actividad pausado	135
8.34. Resumen de registro de actividad con nombre introducido	136
8.35. Mensaje de éxito a guarda la actividad	136
8.36. Ruta guardada en el registro de actividades	137
8.37. Permiso de cámara al guardar waypoint	138
8.38. Introducir nombre waypoint y guardar	138
8.39. Información sobre un waypoint	139
8.40. Selección de visualización de datos	140
9.1. Desglose de las categorías de riesgos de la metodología Boehm	146
9.2. Probabilidad de que ocurra un riesgo	147
9.3. Escala de impacto relativo de cada riesgo	147
9.4. Matriz de probabilidad e impacto	148
9.5. Registro y valoración de riesgos	149

Índice de tablas

2.1. Organización del PSI	17
6.1. Especificación Caso de Uso 1: Importar ruta GPX	59
6.2. Especificación Caso de Uso 2: Ver rutas importadas	59
6.3. Especificación Caso de Uso 3: Enviar ruta importada	59
6.4. Especificación Caso de Uso 4: Seleccionar ruta importada	60
6.5. Especificación Caso de Uso 5: Registrar actividad	60
6.6. Especificación Caso de Uso 6: Seleccionar actividad registrada	60
6.7. Especificación Caso de Uso 7: Ver actividades registradas	60
6.8. Especificación Caso de Uso 8: Enviar actividad registrada	60
6.9. Especificación Caso de Uso 9: Añadir capa al mapa	61
6.10. Especificación Caso de Uso 10: Añadir waypoint	61
6.11. Análisis del Caso de Uso 1: Importar ruta GPX	61
6.12. Análisis del Caso de Uso 2: Ver rutas importadas	62
6.13. Análisis del Caso de Uso 3: Enviar ruta importada	62
6.14. Análisis del Caso de Uso 4: Seleccionar ruta importada	63
6.15. Análisis del Caso de Uso 5: Registrar actividad	63
6.16. Análisis del Caso de Uso 6: Seleccionar una actividad registrada	64
6.17. Análisis del Caso de Uso 7: Ver actividades registradas	64
6.18. Análisis del Caso de Uso 8: Enviar actividad registrada	65
6.19. Análisis del Caso de Uso 9: Añadir capa al mapa	65
6.20. Análisis del Caso de Uso 10	66
6.21. Análisis de la clase Mapa	68
6.22. Análisis de la clase ContenedorCapas	68
6.23. Análisis de la clase Capa	69
6.24. Análisis de la clase Ruta	69
6.25. Análisis de la clase Waypoint	70
6.26. Análisis de la clase GPX	70
6.27. Pruebas unitarias: Caso de Uso 1 - Importar ruta GPX	77
6.28. Pruebas unitarias: Caso de Uso 2 - Ver rutas importadas	77
6.29. Pruebas unitarias: Caso de Uso 3 - Enviar ruta importada	77
6.30. Pruebas unitarias: Caso de Uso 4 - Seleccionar ruta importada	77
6.31. Pruebas unitarias: Caso de Uso 5 - Registrar actividad	78
6.32. Pruebas unitarias: Caso de Uso 6 - Seleccionar actividad registrada	78
6.33. Pruebas unitarias: Caso de Uso 7 - Ver actividades registradas	78

6.34. Pruebas unitarias: Caso de Uso 8 - Enviar actividad registrada	78
6.35. Pruebas unitarias: Caso de Uso 9 - Añadir capa al mapa	78
6.36. Pruebas unitarias: Caso de Uso 10 - Añadir waypoint	79
7.1. Diseño de pruebas: Importar ruta GPX	88
7.2. Diseño de pruebas: Ver rutas importadas	88
7.3. Diseño de pruebas: Enviar ruta importada	89
7.4. Diseño de pruebas: Seleccionar ruta importada	89
7.5. Diseño de pruebas: Registrar actividad	90
7.6. Diseño de pruebas: Seleccionar actividad registrada	91
7.7. Diseño de pruebas: Ver actividades registradas	92
7.8. Diseño de pruebas: Enviar actividad registrada	92
7.9. Diseño de pruebas: Añadir capa al mapa	92
7.10. Diseño de pruebas: Añadir waypoint	93
7.11. Pruebas de usabilidad: preguntas de carácter general	94
7.12. Pruebas de usabilidad: preguntas cortas sobre la aplicación y observaciones	95
7.13. Pruebas de usabilidad: cuestionario para el responsable de las pruebas	96
8.1. Ejecución de pruebas: Importar ruta GPX	111
8.2. Ejecución de pruebas: Ver rutas importadas	111
8.3. Ejecución de pruebas: Enviar ruta importada	111
8.4. Ejecución de pruebas: Seleccionar ruta importada	111
8.5. Ejecución de pruebas: Registrar actividad	112
8.6. Ejecución de pruebas: Seleccionar actividad registrada	112
8.7. Diseño de pruebas: Ver actividades registradas	113
8.8. Diseño de pruebas: Enviar actividad registrada	113
8.9. Diseño de pruebas: Añadir capa al mapa	113
8.10. Diseño de pruebas: Añadir waypoint	113
8.11. Pruebas de usabilidad: preguntas de carácter general para el usuario 1	114
8.12. Pruebas de usabilidad: preguntas cortas sobre la aplicación y observaciones, usuario 1	115
8.13. Pruebas de usabilidad: prueba del usuario 1 para la actividad Importar un archivo GPX	116
8.14. Pruebas de usabilidad: prueba del usuario 1 para la actividad Ver los datos de un archivo importado previamente	116
8.15. Pruebas de usabilidad: prueba del usuario 1 para la actividad Registrar una actividad a partir de una ruta importada previamente	116
8.16. Pruebas de usabilidad: prueba del usuario 1 para la actividad Compartir una activi- dad registrada	116
8.17. Pruebas de usabilidad: preguntas de carácter general para el usuario 2	117
8.18. Pruebas de usabilidad: preguntas cortas sobre la aplicación y observaciones, usuario 2	118
8.19. Pruebas de usabilidad: prueba del usuario 2 para la actividad Importar un archivo GPX	119
8.20. Pruebas de usabilidad: prueba del usuario 2 para la actividad Ver los datos de un archivo importado previamente	119
8.21. Pruebas de usabilidad: prueba del usuario 2 para la actividad Registrar una actividad a partir de una ruta importada previamente	119

8.22. Pruebas de usabilidad: prueba del usuario 2 para la actividad Compartir una actividad registrada	119
8.23. Pruebas de usabilidad: preguntas de carácter general para el usuario 3	120
8.24. Pruebas de usabilidad: preguntas cortas sobre la aplicación y observaciones, usuario 3	121
8.25. Pruebas de usabilidad: prueba del usuario 3 para la actividad Importar un archivo GPX	122
8.26. Pruebas de usabilidad: prueba del usuario 3 para la actividad Ver los datos de un archivo importado previamente	122
8.27. Pruebas de usabilidad: prueba del usuario 3 para la actividad Registrar una actividad a partir de una ruta importada previamente	122
8.28. Pruebas de usabilidad: prueba del usuario 3 para la actividad Compartir una actividad registrada	122
8.29. Pruebas de usabilidad: preguntas de carácter general para el usuario 4	123
8.30. Pruebas de usabilidad: preguntas cortas sobre la aplicación y observaciones, usuario 4	124
8.31. Pruebas de usabilidad: prueba del usuario 4 para la actividad Importar un archivo GPX	125
8.32. Pruebas de usabilidad: prueba del usuario 4 para la actividad Ver los datos de un archivo importado previamente	125
8.33. Pruebas de usabilidad: prueba del usuario 4 para la actividad Registrar una actividad a partir de una ruta importada previamente	125
8.34. Pruebas de usabilidad: prueba del usuario 4 para la actividad Compartir una actividad registrada	125

Capítulo 1

¿Qué es este trabajo?

1.1. Resumen

El proyecto consiste en una aplicación nativa de iOS (fundamentalmente iPhone) dedicada a la realización de deportes al aire libre. La idea es implementar las funciones que un dispositivo GPS tradicional realiza en el teléfono móvil.

Además de la ya clásica función de registrar la actividad, la aplicación se enfoca hacia permitir importar rutas en formato GPX; las cuales se cargarán en un mapa de manera que el usuario pueda visualizarlas y seguirlas. Otra posibilidad que será poder visualizar estas rutas sobre mapas especializados para este tipo de actividades: Google Maps (relieve, satélite...), OpenStreetMap y Instituto Geográfico Nacional.

Un aspecto que se ha tenido en cuenta al realizar la aplicación, es la priorización de mantener una interfaz visual limpia y fácil de entender, proporcionando datos sobre las rutas importadas y realizadas que puedan ser de utilidad, como pueden ser: desnivel (en formato gráfica), tiempo (en caso de ser una ruta importada, tiempo aproximado, en caso de ser una grabada por el usuario, tiempo utilizado para realizar la ruta), ritmo medio, altura máxima y mínima...

Los clientes de este proyecto son aquellos que estén interesados en el mundo del deporte al aire libre, especialmente el relacionado con las rutas de montaña. A pesar de que esta sea una versión inicial no comercial, podría evolucionar hacia ello, ajustándose a las necesidades y requisitos del mercado.

1.2. Palabras Clave

GPS, ruta, waypoint, desnivel, mapa, GPX, iOS, móvil, Swift, SwiftUI, iPhone, app, Apple

1.3. Abstract

This project consists of an iOS native iOS app (mainly for iPhone) made for sports and outdoor activities. The idea is to implement the functions that a traditional GPS device performs on the cell phone.

Besides the already classic functionality of registering an activity, the app focuses on allowing the import of tracks in GPX format, which will be loaded in a map in order for the user to see and follow them. Another possibility will be seeing these tracks on outdoor activity specialized maps: Google Maps (terrain, satellite...), OpenStreetMap and the Spanish Geographical Institute.

One aspect that has been taken into account when developing the application is the prioritization of maintaining a clean and easy to understand visual interface, providing data on imported and completed tracks that may be useful, such as: slope (in graphic format), time (in case of an imported track, approximate time, in case of a track recorded by the user, time used to complete the track), average pace, maximum and minimum altitude...

The customers of this project are those who are interested in the world of outdoor sports, especially those related to mountain tracks. Although this is an initial non-commercial version, it could evolve towards it, adjusting to the needs and requirements of the market.

1.4. Keywords

GPS, track, waypoint, slope, map, GPX, iOS, mobile, Swift, SwiftUI, iPhone, app, Apple

Capítulo 2

PLANIFICACIÓN DEL SISTEMA DE INFORMACIÓN

FASE DE PLANIFICACIÓN

PSI

2.1. INICIO DEL PLAN DE SISTEMAS DE INFORMACIÓN

2.1.1. Análisis de la Necesidad del PSI

El objetivo de este proyecto es crear una aplicación nativa de iOS, orientada principalmente a dispositivo móvil (iPhone) en la que se permita registrar datos de actividades al aire libre usando la localización GPS. Se permitirá la importación de rutas en formato GPX[3], las cuales se podrán seguir a través de un mapa y que también registrará los datos relacionados con la actividad. Esta aplicación será utilizada por cualquier aficionado a este tipo de deportes, especialmente de las rutas de montaña. El software permitirá a los usuarios registrar los datos relacionados con la actividad deportiva mientras la realizan, como pueda ser tiempo, distancia o ritmo medio. Estos datos se podrán visualizar en todo momento, tanto en el proceso de realización como una vez finalizado.

2.1.2. Identificación del Alcance del PSI

La funcionalidad principal consiste en recopilar datos a partir de que un usuario haya cargado una ruta en formato GPX y posteriormente la haya seleccionado, indicando que ha iniciado su actividad. Esta ruta se visualizará en el mapa seleccionado: Google Maps[4] (relieve, satélite), OpenStreetMap[5] o uno proporcionado por el Instituto Geográfico Nacional[6]. El objetivo final, es que el usuario pueda seguir esta ruta registrando además sus propios datos: tiempo necesitado, ritmo medio... Estos datos permitirán crear una imagen final de la ruta realizada por el usuario. A pesar de que los datos incluidos parecen ser los necesarios para tener un plano general de la actividad, en caso de que el usuario no necesite alguno de ellos, se le da también la posibilidad de ocultar algunos de ellos, cumpliendo así la premisa propuesta de tratar de mantener la usabilidad de la aplicación al máximo, incluso si esto consiste en adaptarla a este.

Una segunda funcionalidad importante dentro de la aplicación consiste en registrar una actividad sin ninguna ruta previamente cargada.

Otra materia a tener en cuenta son los conocidos como *waypoints* (que se podrían entender como puntos de interés dentro de la ruta). En caso de que la ruta importada los contenga, estos se mostrarán en el mapa como marcadores pulsables, los cuales mostrarán información tanto estática (nombre, descripción) como dinámica (distancia hasta el punto, desnivel). Además, en caso de que se esté grabando una ruta o siguiendo una previamente importada, se podrán añadir nuevos *waypoints* junto con sus respectivos datos e incluso una imagen tomada con el teléfono móvil.

Una vez finalizados los registros de actividad, tanto si como se inician de cero, como si son a partir de rutas importadas, se podrán visualizar los datos recopilados de estas, incluso pudiendo cargar el contenido en el mapa o seguir la ruta como si fuese una importada a través de un archivo y registrando una nueva actividad.

Estas actividades registradas podrán además enviarse a otros usuarios través de los medios de comunicación que el móvil disponga y sean compatibles (correo electrónico, servicio de mensajería...)

Teniendo esto en cuenta, estos podrían ser los objetivos resumidos a cumplir en este proyecto:

- Registrar datos de una actividad
- Importar archivos GPX
- Cargar datos de archivos importados
- Mostrar datos de una actividad
- Manejo de waypoints
- Envío de rutas registradas

2.2. DEFINICIÓN Y ORGANIZACIÓN DEL PSI

2.2.1. Especificación del Ámbito y Alcance

El sistema consiste en una aplicación móvil para la realización de actividades deportivas al aire libre. Los usuarios podrán registrar actividades y cargar archivos GPX en la aplicación con el objetivo de poder seguir estos como rutas cargadas en un mapa.

Fase 1: Planificación inicial

En esta primera fase se discute con el tutor los objetivos y alcance del proyecto. Una vez aclarados, se examinan las alternativas que existen a lo que se propone, observando en estas cuáles son sus puntos fuertes y debilidades, de manera que se pueda aplicar este conocimiento al sistema.

A continuación, se realiza la planificación inicial, definición de la arquitectura tecnológica y el análisis del sistema de información.

Objetivos

- Realizar una planificación inicial del proyecto
- Estudiar las alternativas existentes en el mercado
- Plantear los objetivos generales del proyecto

Fase 2: Integración con sistemas externos

A continuación, se prevee que una parte compleja de la aplicación será integrarse con los sistemas externos. Por ello, se comienza por realizar una integración con el sistema de visualización de mapas elegido (incluyendo la posibilidad de añadir capas sobre este, que a su vez son otros sistemas externos). Por otro lado, se trabaja también con los archivos GPX, explorando las posibilidades que este ofrece y su correcto uso.

Objetivos

- Integrarse con el sistema de visualización de mapas elegido y sus respectivas capas
- Trabajar con archivos GPX

Fase 3: Representación visual de elementos

La tercera fase consiste en conseguir una representación visual adecuada que use los elementos obtenidos en las fases previas, esto es, visualización de los datos de los archivos GPX que se hayan importado en la aplicación, incluyendo sus respectivos datos sobre el mapa.

Objetivos

- Visualización de los archivos GPX importados

- Permitir selección de cada uno de los archivos, con el objetivo de visualizar su contenido sobre el mapa
- Permitir importar archivos GPX

Fase 4: Funcionalidades finales

La cuarta y última fase consiste en realizar las funciones restantes e integrarlas con lo que ya tenemos previamente. El centro de la aplicación consiste en registrar una actividad, por lo que se realiza esta funcionalidad. La visualización de las actividades registradas es prácticamente igual que la de las rutas importadas, se realizan los cambios necesarios y se añade esta funcionalidad. La última funcionalidad consiste en enviar rutas importadas y actividades registradas. Una vez más, el funcionamiento es el mismo para las dos, por lo que facilita el trabajo

Objetivos

- Realizar la funcionalidad de registrar actividad, combinándola con lo hecho en fases anteriores
- Permitir la visualización de las actividades registradas
- Permitir el envío de actividades registradas y rutas importadas

2.2.2. Organización del PSI

Tabla 2.1: Organización del PSI

Usuario	Perfil	Función
Equipo de supervisión		
Tutor	Jefe de proyecto	Orientar y supervisar la realización del proyecto, uso de buenas prácticas y el cumplimiento de los objetivos.
Equipo de desarrollo		
Alumno	Desarrollador Software	Desarrollar el software necesario para realizar las funcionalidades planteadas en el proyecto.
Alumno	Arquitecto Software	Plantear la arquitectura del sistema en función de los objetivos propuestos en el proyecto. Obtener requisitos y realizar la planificación.
Alumno	Analista	Realizar el análisis del sistema propuesto.
Equipo de pruebas y carga de datos		
Alumno	Tester	Realizar las pruebas unitarias y de usabilidad.

2.3. ESTUDIO DE LA INFORMACIÓN RELEVANTE

Una de las funcionalidades del sistema consiste en la importación de rutas a través de archivos GPX. Este estándar de intercambio de información geospacial basado en XML, no sólo se utiliza para este propósito dentro de la aplicación, sino que también se registrarán los datos en un archivo de este formato.

Debido a esto se revisa primeramente este estándar: en qué consiste, cómo utilizarlo...

2.3.1. Selección y Análisis de Antecedentes

GPX es el acrónimo de “GPS Exchange Format” (Formato de Intercambio de GPS, en español). Es un formato estándar utilizado para almacenar datos de rutas, puntos de interés (*waypoints*) y pistas de GPS.

GPX es un formato de archivo de texto que permite intercambiar datos de ubicación y rutas entre diferentes aplicaciones, dispositivos y servicios que utilizan tecnología GPS. Los archivos GPX contienen información geospacial, como coordenadas de latitud y longitud, altitud, tiempo, velocidad y otros datos relacionados.

Los archivos GPX son ampliamente utilizados en actividades al aire libre, como senderismo, ciclismo, geocaching y deportes acuáticos, ya que permiten a los usuarios registrar, compartir y seguir rutas y ubicaciones en un mapa. Los archivos GPX también se utilizan en aplicaciones de navegación, software de seguimiento de actividades y servicios en línea relacionados con la geolocalización (como es el caso de este sistema).

Capítulo 3

DEFINICIÓN DE LA ARQUITECTURA TECNOLÓGICA

FASE DE PLANIFICACIÓN

PSI 7

3.1. Identificación de las Necesidades de Infraestructura Tecnológica

Estudiando las tecnologías disponibles para desarrollo móvil para aplicaciones nativas de iOS aparecen tres alternativas: SwiftUI[7], React Native[8] y Flutter[9].

1. **SwiftUI** es un framework de interfaz de usuario desarrollado por Apple que permite crear aplicaciones para iOS, macOS, watchOS y tvOS.
 - *Declarativo*: SwiftUI utiliza un enfoque declarativo para crear interfaces de usuario, lo que significa se describe cómo se deben ver y comportar los elementos de la interfaz en lugar de escribir código imperativo para manipularlos.
 - *Diseño basado en componentes*: utiliza un enfoque basado en componentes para construir interfaces de usuario, lo que permite crear y reutilizar componentes de manera eficiente, lo que simplifica el desarrollo y el mantenimiento del código.
 - *Preview en tiempo real*: ofrece una característica de vista previa en tiempo real que permite ver cómo se verá y se comportará la interfaz de usuario en diferentes dispositivos.
 - *Integración nativa con Swift[10]*: está integrado nativamente con el lenguaje de programación Swift de Apple[11], lo que proporciona una sintaxis y una experiencia de desarrollo coherente con el ecosistema de desarrollo de Apple.
 - *Integración con otros frameworks de Apple*: se integra con otros frameworks de Apple, como Core Data, Core Animation, MapKit y otros.
2. **React Native** es un marco de desarrollo de aplicaciones móviles creado por Facebook que permite construir aplicaciones nativas para iOS y Android utilizando JavaScript y React.
 - *Desarrollo multiplataforma*: React Native permite escribir código una vez y utilizarlo en múltiples plataformas, como iOS y Android, lo que reduce la necesidad de escribir código nativo para cada plataforma por separado.
 - *Componentes reutilizables*: React Native utiliza componentes de interfaz de usuario reutilizables que se comportan de manera similar a los componentes de React para la web, lo que permite compartir fácilmente el código entre aplicaciones web y móviles.
 - *Rendimiento nativo*: A diferencia de las aplicaciones híbridas, que se ejecutan dentro de un contenedor web, las aplicaciones construidas con React Native se compilan en código nativo, lo que proporciona un rendimiento y una apariencia similares a las aplicaciones nativas.
 - *Acceso a API nativas*: React Native permite acceder a las API nativas de los dispositivos, como la cámara, el GPS y los sensores, a través de módulos de JavaScript, lo que permite crear aplicaciones móviles con funcionalidades nativas.
3. **Flutter** es un framework de desarrollo de aplicaciones móviles creado por Google que permite construir aplicaciones nativas para iOS, Android y otras plataformas utilizando un lenguaje de programación llamado Dart.

- *Desarrollo multiplataforma*: permite escribir código una vez y utilizarlo en múltiples plataformas, lo que reduce la necesidad de escribir código nativo para cada plataforma por separado. Flutter utiliza un enfoque de "pintura" para la interfaz de usuario, lo que le permite renderizar widgets personalizados de manera rápida y eficiente en todas las plataformas.
- *Interfaz de usuario declarativa*: Flutter utiliza una interfaz de usuario declarativa, lo que significa que define la interfaz de usuario mediante la construcción de widgets, que son objetos de Dart. Los widgets en Flutter son altamente personalizables y se pueden componer para crear interfaces de usuario complejas y atractivas.
- *Rendimiento nativo*: compila su código en código nativo para la plataforma de destino, lo que ofrece un rendimiento y una apariencia similares a las aplicaciones nativas. Además, Flutter cuenta con un motor de renderizado propio llamado Skia, que permite un rendimiento de pintura rápido y suave.
- *Acceso a API nativas*: Flutter permite acceder a las API nativas de los dispositivos, como la cámara, el GPS y los sensores, a través de plugins, lo que permite crear aplicaciones móviles con funcionalidades nativas.

Otro estudio a realizar es la elección de servicio de visualización de mapas. A priori, aparecen dos opciones con un SDK para iOS: Google Maps[12] y MapBox[13].

1. El **SDK de Google Maps** ofrece las siguientes características:

- Los mapas se pueden personalizar con diferentes estilos, marcadores, polilíneas y polígonos. Los usuarios también pueden interactuar con los mapas mediante gestos como pellizcar, deslizar y hacer zoom.
- Acceder a la ubicación del dispositivo del usuario.
- Mostrar imágenes panorámicas en 360 grados de Google Street View. Los usuarios pueden explorar el entorno en modo de vista panorámica y moverse en cualquier dirección.
- Se integra fácilmente con otros servicios de Google, como Places API para buscar lugares, Directions API para obtener direcciones, y Geocoding API para convertir direcciones en coordenadas geográficas.
- Funciones adicionales, como la visualización de información de tráfico en tiempo real, la personalización de la apariencia de los mapas, la implementación de rutas de navegación paso a paso, la gestión de marcadores y eventos de mapas además de la posibilidad de usar mapas sin conexión.

2. El **SDK de Mapbox** ofrece las siguientes características:

- Los mapas se pueden personalizar con diferentes estilos de mapas, capas, marcadores, polilíneas y polígonos. Los usuarios también pueden interactuar con los mapas mediante gestos como pellizcar, deslizar y hacer zoom.
- Acceder a la ubicación del dispositivo del usuario. Esto incluye la obtención de la ubicación actual del dispositivo, el seguimiento de la ubicación en tiempo real, la búsqueda de lugares cercanos y la obtención de direcciones entre ubicaciones.

- Ofrece una amplia gama de opciones de personalización de mapas, lo que permite a crear estilos de mapas personalizados para adaptarse a la apariencia y la experiencia de usuario deseada en sus aplicaciones. Esto incluye la posibilidad de personalizar colores, fuentes, iconos...
- Se integra con otros servicios de Mapbox, como Mapbox Studio para la creación y gestión de estilos de mapas personalizados y Mapbox Geocoding API para la búsqueda de lugares y la obtención de direcciones.
- Funciones adicionales, como la visualización de información de tráfico en tiempo real, la implementación de rutas de navegación paso a paso, la gestión de eventos de mapas y la posibilidad de usar mapas sin conexión.

3.2. Selección de la Arquitectura Tecnológica

Tras analizar las diferentes opciones con sus correspondientes pros y contras, la elección final es SwiftUI, ya que, al requerir la aplicación de varios sistemas de comunicación con el propio sistema operativo iOS, y SwiftUI utilizar el lenguaje nativo de Apple, Swift, además de tener numerosas librerías de la propia empresa para integrarse con este, parece la elección más adecuada a las necesidades, proporcionando facilidades, incluyendo menos código y tiempo de integración.

Otra ventaja que tiene peso en la decisión es que SwiftUI tienen los componentes, fuentes y animaciones clásicas de Apple por lo que la aplicación tendrá un aspecto y funcionamiento conocido por todos los usuarios de iOS, facilitando así la premisa de conseguir una interfaz de usuario limpia y altamente usable.

Por otro lado, para el servicio de visualización de mapas, se elige Google Maps. Ambas posibilidades estudiadas ofrecen características muy parecidas, pero debido al conocimiento previo de Google Maps y la mayor dificultad de desarrollo que a priori presenta MapBox, la opción más conveniente para este proyecto es la primera. Además, esta nos permitirá una integración con el Instituto Geográfico Nacional, uno de los objetivos del proyecto.

Capítulo 4

ESTUDIO DE VIABILIDAD DEL SISTEMA

FASE DE DESARROLLO

EVS

4.1. ESTUDIO Y VALORACIÓN DE ALTERNATIVAS DE SOLUCIÓN. SELECCIÓN DE ALTERNATIVA FINAL

Al ya existir aplicaciones de navegación GPS para dispositivos móviles, parece importante observar cómo las aplicaciones de más éxito en el mercado resuelven los problemas planteados previamente. Además de esto, será útil para saber a nivel de interfaz, algunos estándares relacionados con la situación de algunos elementos gráficos que todas las aplicaciones de este estilo tienen (como podría ser, por ejemplo en las aplicaciones de compra, la situación del botón carrito en la parte de arriba a la derecha de la pantalla). Otra información a obtener de estas aplicaciones, es el servicio de mapas que utilizan, ya que nos mostrarán diferentes alternativas a estudiar.

4.1.1. Wikiloc



Figura 4.1: Logo de Wikiloc

Esta aplicación[14] suele ser la primera elección de la mayoría de los usuarios. Es posiblemente la más famosa y dominante del mercado, con más de doce millones de miembros en todo el mundo.

Wikiloc permite a los usuarios descubrir, crear y compartir rutas de senderismo, ciclismo, carrera y otras actividades al aire libre. Los usuarios pueden buscar y descubrir rutas en una amplia variedad de destinos alrededor del mundo. Además, pueden crear y compartir sus propias rutas utilizando herramientas de trazado de mapas en línea. Wikiloc utiliza tecnología de seguimiento GPS para registrar rutas, lo que permite a los usuarios ver su progreso en tiempo real, medir la distancia y el tiempo, y obtener información detallada sobre la ruta, como la altitud y la dificultad.

Wikiloc también ofrece funciones adicionales, como la descarga de mapas offline, la integración con dispositivos de seguimiento GPS, la posibilidad de crear listas de rutas favoritas y la opción de suscripción premium para acceder a características avanzadas.

Dos de las grandes ventajas de Wikiloc son, por un lado la gran comunidad de usuarios que tiene y, por ende, muchas rutas disponibles, y por otro, un buen uso del servicio de visualización de

mapas, ya que es bastante rico, usable y con funcionalidades interesantes, entre la que se encuentra la integración con el Instituto Geográfico Nacional, que ofrece mapas específicos de rutas de montaña.

La desventaja de Wikiloc es su interfaz de usuario. Es bastante pobre, por lo que la experiencia de usuario empeora.

Wikiloc utiliza como servicio de mapas Google Maps, pudiendo añadir a este diferentes capas.

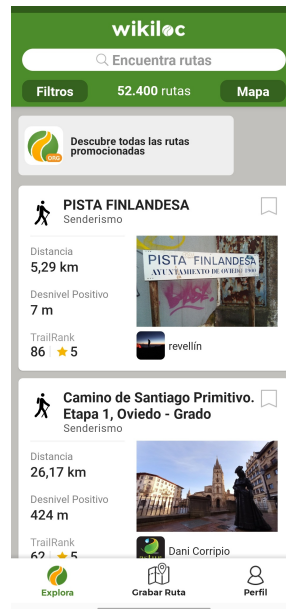


Figura 4.2: Pantalla inicial de Wikiloc

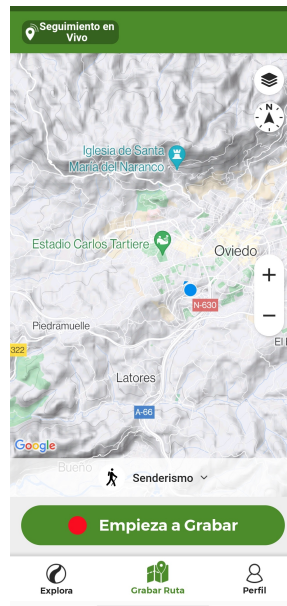


Figura 4.3: Visualización del mapa de Wikiloc

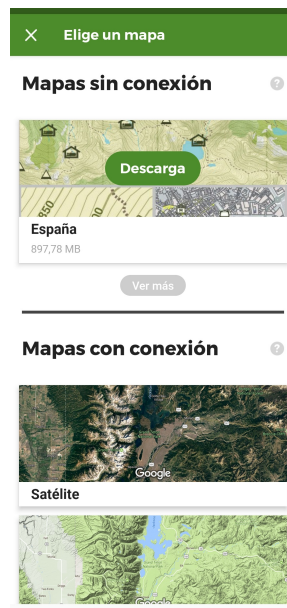


Figura 4.4: Capas aplicables al mapa de Wikiloc

4.1.2. Strava



Figura 4.5: Logo de Strava

Strava[15] es una plataforma en línea y una aplicación móvil dedicada a la actividad física y deportiva, especialmente para corredores y ciclistas. Permite a los usuarios registrar y seguir sus actividades deportivas, como correr, andar en bicicleta, nadar y más. Los usuarios pueden utilizar la aplicación móvil o dispositivos GPS compatibles para registrar sus actividades y obtener información detallada como la distancia, el tiempo, la velocidad, la altitud...

Ofrece además herramientas de seguimiento y análisis del rendimiento para ayudar a los usuarios a mejorar su condición física y alcanzar sus objetivos. Los usuarios pueden ver estadísticas detalladas de sus actividades, establecer metas, comparar su rendimiento con el de otros atletas y recibir retroalimentación sobre su progreso.

La aplicación también ofrece retos y competiciones para motivar a los usuarios a alcanzar nuevos objetivos y superarse a sí mismos. Los usuarios pueden unirse a desafíos mensuales, participar en competiciones virtuales y ganar medallas y reconocimientos por sus logros.

Strava permite a los usuarios descubrir y explorar rutas populares cerca de su ubicación, así como crear y compartir sus propias rutas.

Al igual que Wikiloc, Strava se integra con una amplia variedad de dispositivos de seguimiento GPS y aplicaciones externas, lo que permite a los usuarios importar y sincronizar automáticamente sus actividades desde diferentes fuentes, como relojes deportivos.

A diferencia de Wikiloc, el servicio de mapas de Strava no está explotado tan bien. No incluye capas a añadir al mapa en todo momento.

La gran ventaja de Strava es su interfaz. Esta, es muy rica, fácil de usar y con elementos que se integran muy bien entre sí.

Strava utiliza MapBox como servicio de visualización de mapas.

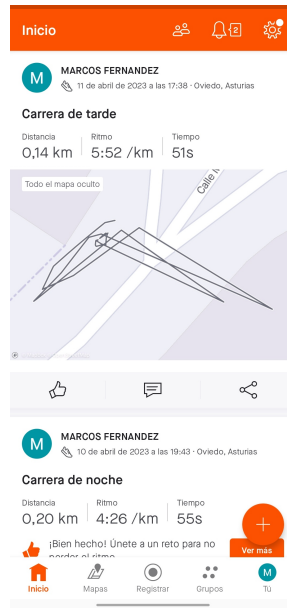


Figura 4.6: Pantalla inicial de Strava

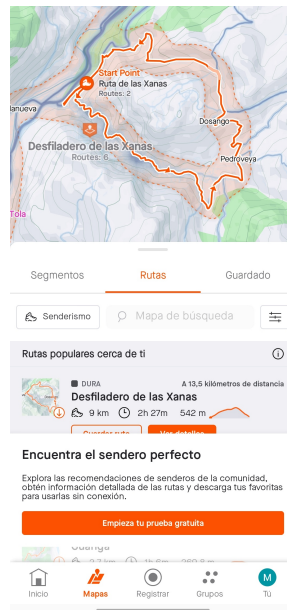


Figura 4.7: Visualización del mapa de Strava

4.1.3. AllTrails



Figura 4.8: Logo de AllTrails

AllTrails[16] es una plataforma en línea y una aplicación móvil dedicada a la planificación, descubrimiento y registro de rutas de senderismo y actividades al aire libre. Ofrece una amplia colección de rutas de senderismo y actividades al aire libre en todo el mundo, con información detallada sobre la distancia, la elevación, la dificultad, el tiempo estimado y más. Los usuarios pueden buscar y descubrir rutas basadas en la ubicación, la dificultad, la longitud...

Proporciona mapas interactivos y herramientas de navegación para ayudar a los usuarios a planificar y seguir rutas. Los usuarios pueden ver mapas topográficos, descargar mapas offline, seguir la ruta en tiempo real con el GPS de su dispositivo y recibir notificaciones de navegación.

Los usuarios pueden leer y escribir reseñas, dar "me gusta" comentar en rutas además de seguir a otros miembros para obtener recomendaciones personalizadas.

Ofrece la posibilidad de registrar actividades con detalles como la distancia, la duración, la altitud y la velocidad, y guardarlas en su perfil para referencia futura.

AllTrails ofrece una opción de suscripción premium que ofrece características adicionales, como mapas offline ilimitados, alertas de seguridad, filtros avanzados de búsqueda, análisis detallado de actividades y más.

Esta aplicación tiene un mayor parecido a Strava que a Wikiloc, ya que también tiene una interfaz muy buena y usable, por lo que sería una de las ventajas.

Quizá sea de las tres la menos usada y conocida por la comunidad, que aunque no sea pequeña, puede ser un inconveniente a la hora de usarla.

Al igual que Strava, el servicio de mapas utilizado por AllTrails es MapBox.

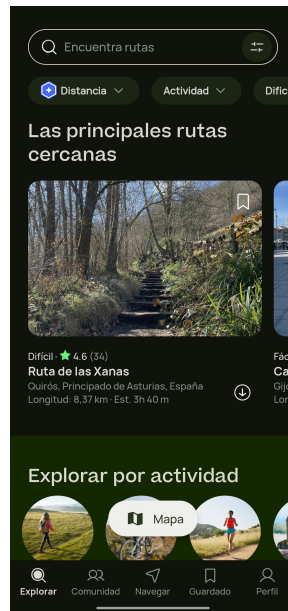


Figura 4.9: Pantalla inicial de AllTrails

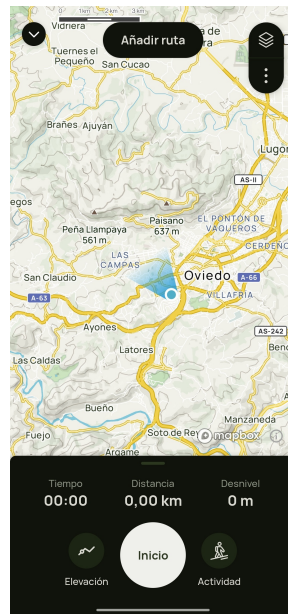


Figura 4.10: Visualización del mapa de AllTrails

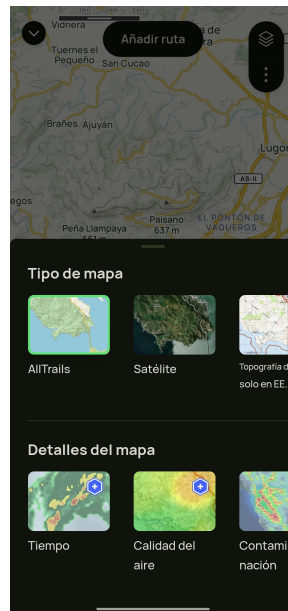


Figura 4.11: Capas aplicables al mapa de AllTrails

4.1.4. Selección final

Como se puede observar todas las aplicaciones ofrecen prácticamente los mismo servicios, simplemente con alguna diferencia o característica nueva en alguna de ellas. Por tanto, elegir una como alternativa a seguir no tiene mucho sentido. En su defecto, se observará cómo estas tres aplicaciones mencionadas resuelven los problemas y funcionalidades planteados para este proyecto, analizando cada una de ellas y tratando de juntar lo mejor de cada una añadiendo características que se observe que faltan o que a criterio personal podrían mejorar la funcionalidad y usabilidad.

Capítulo 5

PLANIFICACIÓN Y GESTIÓN DEL TFG

5.1. PLANIFICACIÓN DEL PROYECTO

5.1.1. Identificación de Interesados

- **Usuarios interesados en actividades relacionadas con los deportes al aire libre:** Personas interesadas en realizar actividades al aire libre utilizando dispositivos móviles iOS.
- **Tutor del proyecto:** Persona que ha propuesto esta temática de trabajo.
- **Autor del trabajo:** Persona encargada de realizar el análisis, diseño y desarrollo del proyecto, junto con su documentación asociada.

5.1.2. OBS y PBS

En el siguiente diagrama podemos ver el OBS (Organization Breakdown Structure) en el que se representa el organigrama del proyecto.

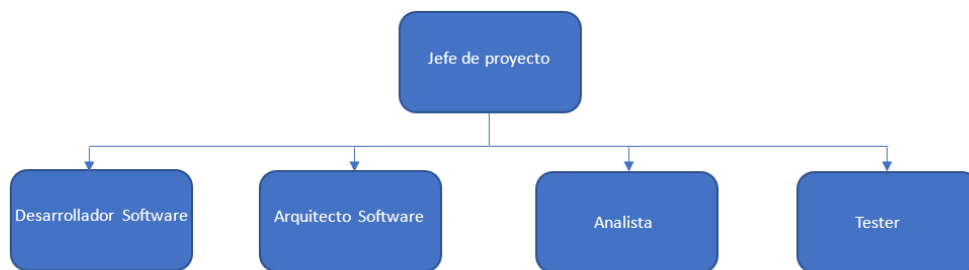


Figura 5.1: OBS

A continuación, se presenta el PBS (Product Breakdown Structure) donde podemos observar los objetivos del proyecto representados como productos. Se ha dividido en tres partes para que sea más legible

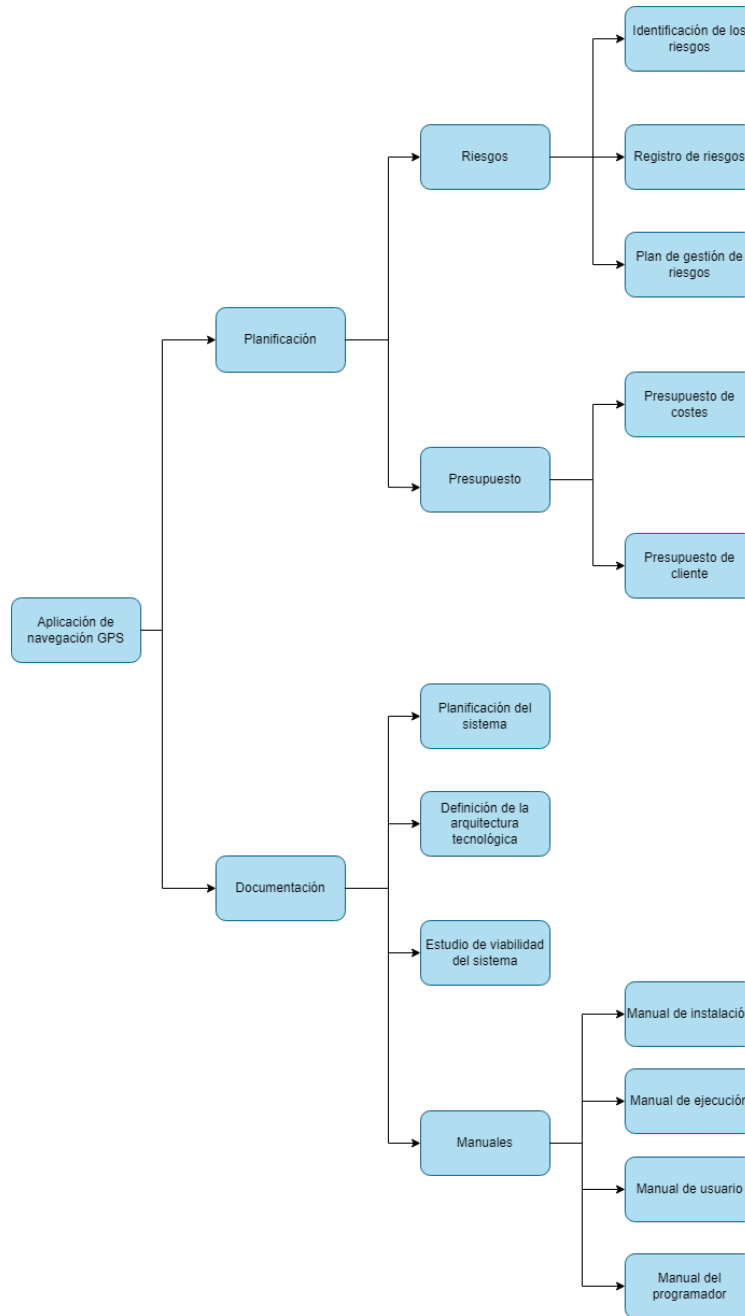


Figura 5.2: PBS - Primera parte

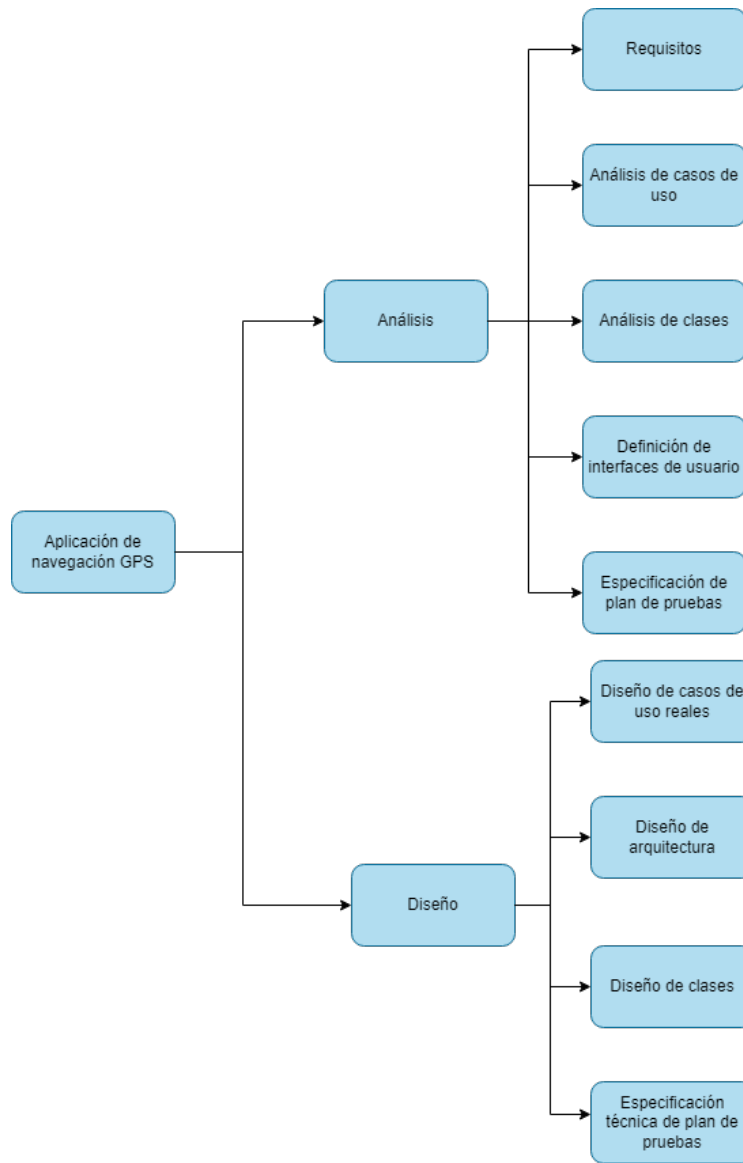


Figura 5.3: PBS - Segunda parte

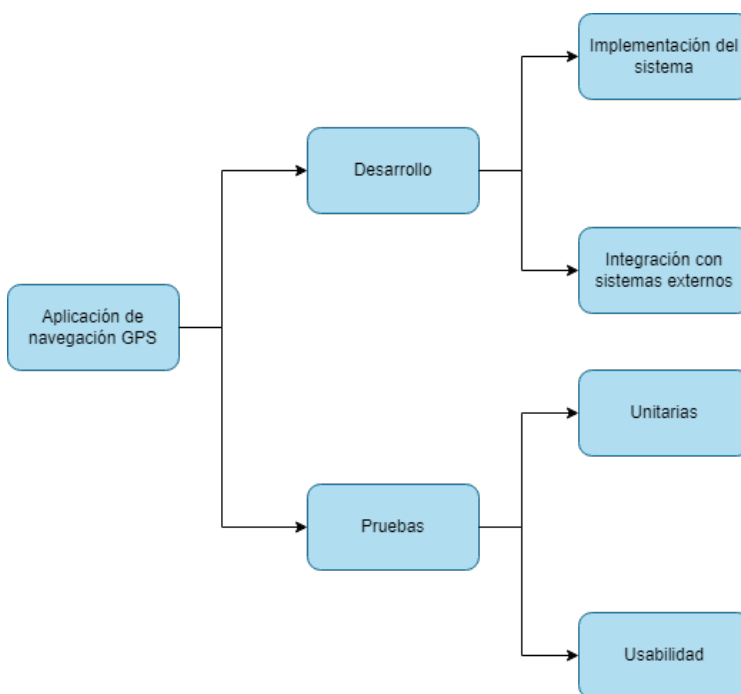


Figura 5.4: PBS - Tercera parte

5.1.3. Planificación Inicial. WBS

Una vez tenemos el PBS y el OBS, pasamos a ver el WBS (Work Breakdown Structure) donde veremos cómo estos dos elementos se combinan en una planificación temporal. Los elementos del organograma trabajarán en conjunto para conseguir los productos representados por el OBS.

En la siguiente ilustración se puede ver una visión general del WBS, representado en formato tabla y diagrama de Gantt. La semana laboral de las tareas planificadas es de lunes a viernes, con una jornada de 4 horas diarias.

Nombre de tarea	Duración	Comienzo	Fin
Aplicación móvil GPS iOS	28,88 días	jue 10/02/22	jue 09/06/22
Planificación	210 horas	jue 10/02/22	mar 24/05/22
Documentación	229 horas	vie 11/02/22	jue 09/06/22
Análisis	39 horas	vie 11/03/22	lun 28/03/22
Diseño	62 horas	mar 29/03/22	mar 26/04/22
Desarrollo	83 horas	mié 27/04/22	lun 06/06/22
Pruebas	82 horas	vie 29/04/22	mar 07/06/22

Figura 5.5: Planificación general, tabla de tareas

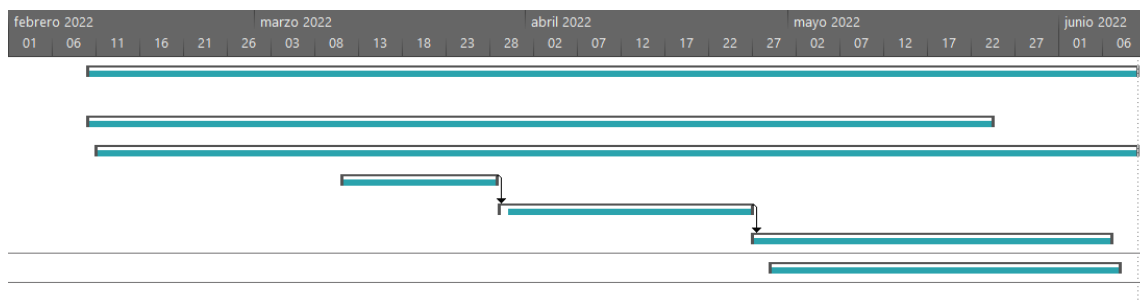


Figura 5.6: Planificación general, diagrama de Gantt

5.1.3.1. Planificación

Es la fase inicial del proyecto, donde se plantearán los objetivos y tiempos. Además, se planifica una reunión periódica con el Jefe de proyecto (tutor) para ir revisando el trabajo realizado.

Nombre de tarea	Duración	Comienzo	Fin
Planificación	210 horas	jue 10/02/22	mar 24/05/22
Reunión con tutor	182 horas	mar 15/02/22	mar 24/05/22
Reunión con tutor 1	1 hora	mar 15/02/22	mar 15/02/22
Reunión con tutor 2	1 hora	mar 01/03/22	mar 01/03/22
Reunión con tutor 3	1 hora	mar 15/03/22	mar 15/03/22
Reunión con tutor 4	1 hora	mar 29/03/22	mar 29/03/22
Reunión con tutor 5	1 hora	mar 12/04/22	mar 12/04/22
Reunión con tutor 6	1 hora	mar 26/04/22	mar 26/04/22
Reunión con tutor 7	1 hora	mar 10/05/22	mar 10/05/22
Reunión con tutor 8	1 hora	mar 24/05/22	mar 24/05/22
Plan de gestión de riesgos	42 horas	jue 10/02/22	lun 28/02/22
Identificación de riesgos	14 horas	jue 10/02/22	lun 14/02/22
Registro de riesgos	16 horas	mar 15/02/22	vie 18/02/22
Gestión de riesgos	21 horas	lun 21/02/22	lun 28/02/22
Presupuestos	29 horas	mar 01/03/22	vie 11/03/22
Presupuesto de costes	21 horas	mar 01/03/22	lun 07/03/22
Presupuesto de cliente	19 horas	lun 07/03/22	vie 11/03/22

Figura 5.7: Planificación de la planificación, tabla de tareas

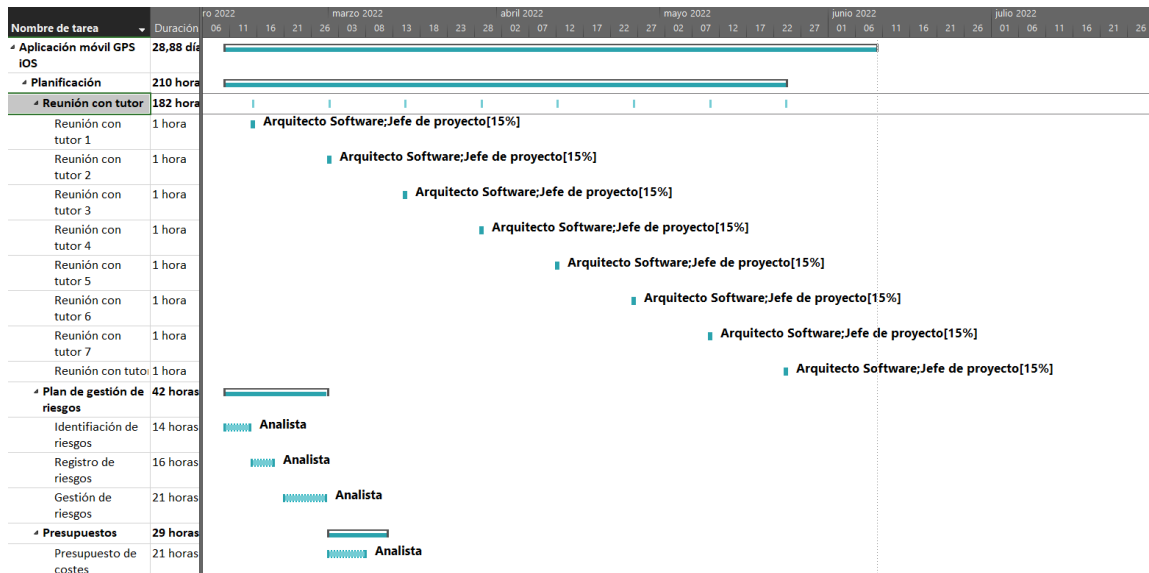


Figura 5.8: Planificación de la planificación, diagrama de Gantt

5.1.3.2. Documentación

La documentación se irá realizando a lo largo de toda la vida del proyecto de forma iterativa.

Nombre de tarea	Duración	Comienzo	Fin
Documentación	229 horas	vie 11/02/22	jue 09/06/22
Planificación del sistema	16 horas	mar 24/05/22	mié 01/06/22
Definición de la arquitectura tecnológica	19 horas	vie 11/02/22	mié 16/02/22
Estudio de viabilidad	32 horas	jue 17/02/22	mar 01/03/22
Manuales	17 horas	lun 06/06/22	jue 09/06/22
Manual de instalación	7 horas	lun 06/06/22	lun 06/06/22
Manual de ejecución	8 horas	mar 07/06/22	mar 07/06/22
Manual de usuario	8 horas	mié 08/06/22	mié 08/06/22
Manual del programador	8 horas	jue 09/06/22	jue 09/06/22

Figura 5.9: Planficación de la documentación, tabla de tareas

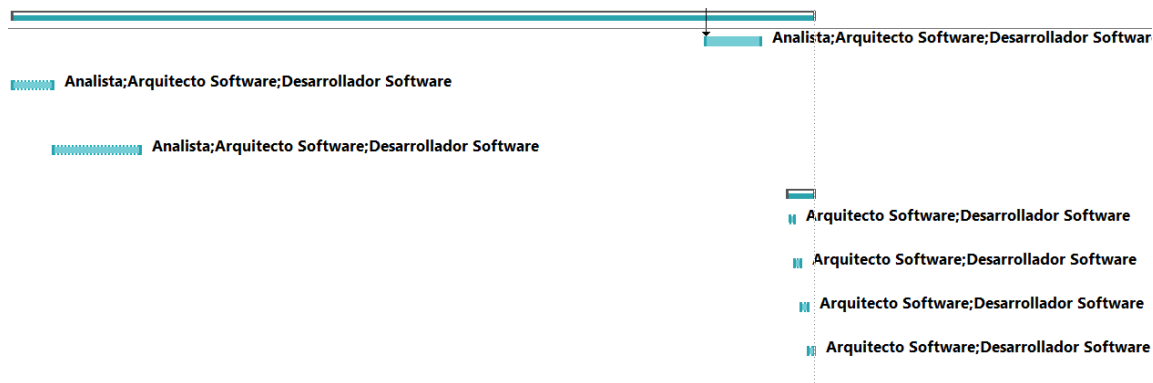


Figura 5.10: Planficación de la documentación, diagrama de Gantt

5.1.3.3. Análisis

En la fase de análisis se realizarán las primeras aproximaciones al proyecto, estudio de la situación y alternativas.

Nombre de tarea	Duración	Comienzo	Fin
Análisis	39 horas	vie 11/03/22	lun 28/03/22
Requisitos	14 horas	vie 11/03/22	lun 14/03/22
Análisis caso de uso	23 horas	mié 16/03/22	mié 23/03/22
Análisis de clases	17 horas	lun 21/03/22	jue 24/03/22
Definición de interfaces de usuario	23 horas	lun 14/03/22	lun 21/03/22
Especificación de plan de pruebas	16 horas	vie 18/03/22	lun 28/03/22

Figura 5.11: Planficación del análisis, tabla de tareas

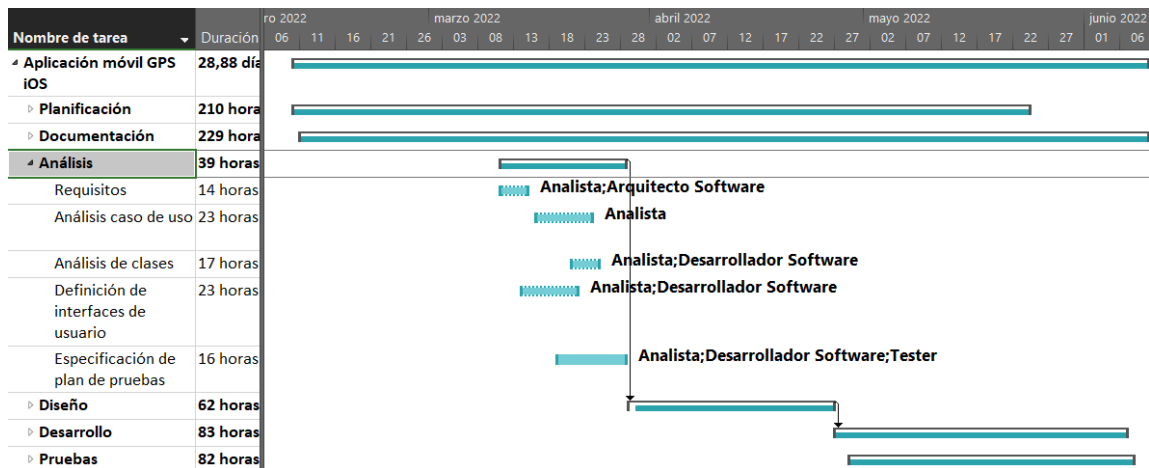


Figura 5.12: Planficación del análisis, diagrama de Gantt

5.1.3.4. Diseño

En el diseño el sistema ya tendrá una forma más concreta y comenzaremos a ajustarnos a las tecnologías previamente seleccionadas.

Nombre de tarea	Duración	Comienzo	Fin
Diseño	62 horas	mar 29/03/22	mar 26/04/22
Diseño de casos de uso	18 horas	mié 30/03/22	lun 04/04/22
Diseño de la arquitectura	23 horas	lun 04/04/22	lun 11/04/22
Diseño de clases	28 horas	lun 11/04/22	mié 20/04/22
Especificación técnica del plan de pruebas	21 horas	mié 20/04/22	mar 26/04/22

Figura 5.13: Planificación del diseño, tabla de tareas

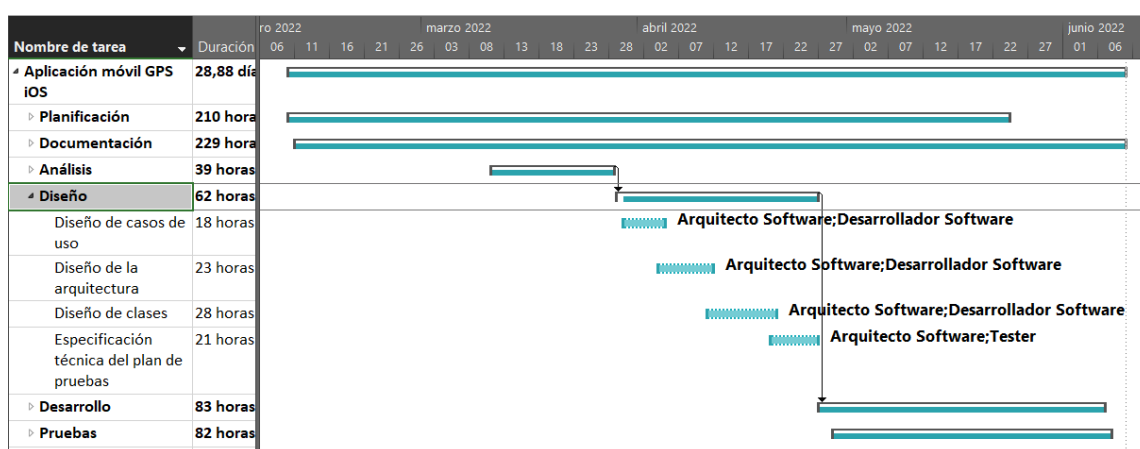


Figura 5.14: Planificación del diseño, diagrama de Gantt

5.1.3.5. Desarrollo

En el desarrollo se aplicará todo lo previamente planteado, plasmando en código y finalmente en una aplicación todo lo realizado en fases anteriores.

Nombre de tarea	Duración	Comienzo	Fin
Desarrollo	83 horas	mié 27/04/22	lun 06/06/22
Implementación del sistema	83 horas	mié 27/04/22	lun 06/06/22
Integración con sistemas externos	49 horas	mié 27/04/22	mié 18/05/22

Figura 5.15: Planificación del desarrollo, tabla de tareas

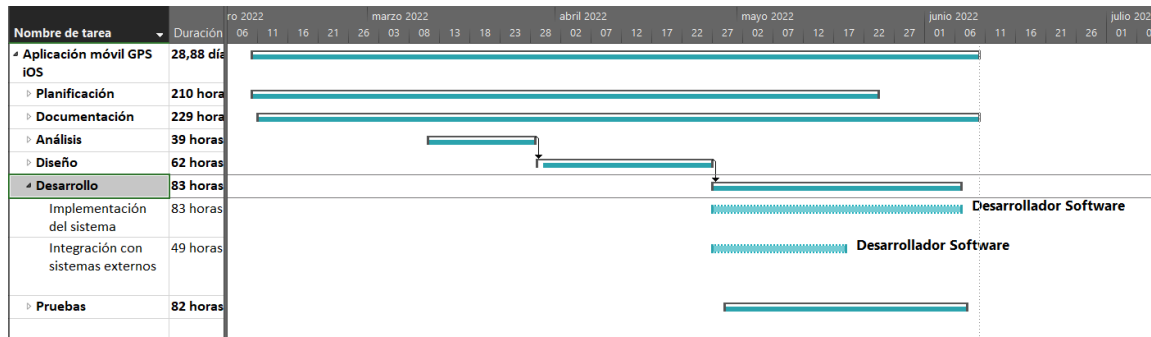


Figura 5.16: Planficación del desarrollo, diagrama de Gantt

5.1.3.6. Pruebas

Las pruebas unitarias se irán efectuando a lo largo de todo el desarrollo, tratando de seguir una metodología al estilo TDD (Test Driven Development) tratando de conseguir un sistema robusto.

Nombre de tarea	Duración	Comienzo	Fin
Pruebas	82 horas	vie 29/04/22	mar 07/06/22
Pruebas de usabilidad	26 horas	lun 30/05/22	mar 07/06/22
Pruebas unitarias	82 horas	vie 29/04/22	mar 07/06/22

Figura 5.17: Planficación de las pruebas, tabla de tareas

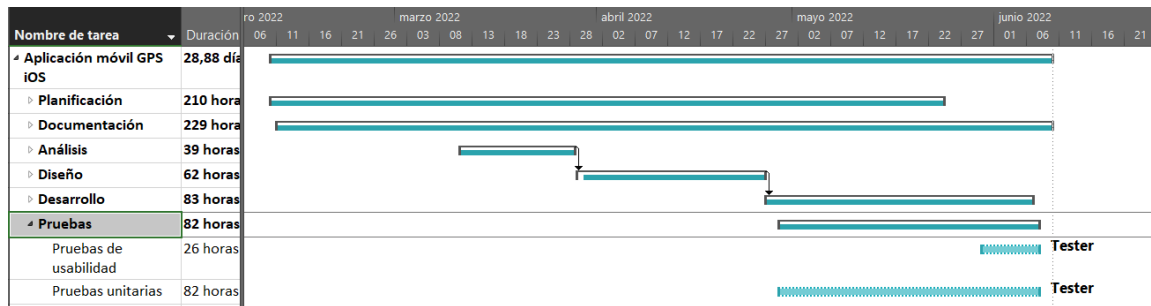


Figura 5.18: Planficación de las pruebas, diagrama de Gantt

5.1.4. Riesgos

A continuación veremos los riesgos que pueden afectar la proyecto, cómo pueden afectar y cómo reaccionaremos ante la posibilidad de que ocurran.

5.1.4.1. Plan de Gestión de Riesgos

Para ver el Plan de Gestión de riesgos, leer el anexo Plan de Gestión de Riesgos 9.2.4.

5.1.4.2. Identificación de Riesgos

Se han identificado cinco riesgos que pueden afectar al proyecto:

1. **Fallo en la estimación de tiempo:** Se planifica el proyecto con un tiempo que finalmente no se cumple.
2. **Falta de formación del desarrollador:** El programador software no tiene suficientes conocimientos sobre las tecnologías que se van a usar.
3. **Falta de usabilidad:** Los usuarios pueden no guiarse y situarse a través de la aplicación como se desearía.
4. **Pérdida de código:** Se pierde el código, total o parcialmente de la aplicación.
5. **Cambios en el framework:** El framework se actualiza y varias de las características utilizadas dejan de funcionar

5.1.4.3. Registro de Riesgos

Para ver el Registro de riesgos, leer el anexo Plan de Gestión de Riesgos 9.2.4.

5.1.5. Presupuesto Inicial

A continuación, utilizando los elementos anteriores se procede a realizar los presupuestos. Para ello, hay que plantear primeramente el precio por hora del personal. En la siguiente tabla se puede observar.

Personal	Precio/Hora
Jefe de proyecto	60,00 €
Arquitecto de Software	50,00 €
Analista	40,00 €
Desarrollador Software	40,00 €
Tester	35,00 €

Figura 5.19: Tabla de precios de los recursos de personal

5.1.5.1. Presupuesto de Costes

Lo primero será realizar una partida por elemento visto en el OBS y planificado en el WBS, apareciendo una más correspondiente a otros costes que un proyecto tiene indirectamente. Primeramente, veremos el presupuesto agregado y luego un desglose de cada una de las partidas.

Presupuesto agregado					
I1	I2	I3	Descripción		Total
	1		Planificación		4.520,00 €
	2		Documentación		11.500,00 €
	3		Análisis		5.380,00 €
	4		Diseño		7.995,00 €
	5		Desarrollo		5.280,00 €
	6		Pruebas		3.780,00 €
	7		Otros costes		4.045,80 €
TOTAL					42.500,80 €

Figura 5.20: Presupuesto agregado

Planificación									
I1	I2	I3	Descripción	Cantidad	Unidades	Precio	Subtotal(3)	Subtotal(2)	Total
1			Reunión con tutor						880,00 €
	1		Jefe de proyecto	8	horas	60,00 €		480,00 €	
		2	Arquitecto de software	8	horas	50,00 €		400,00 €	
2			Plan de Gestión de Riesgos						2.040,00 €
		1	Identificación de Riesgos						
		1	Analista	14	horas	40,00 €		560,00 €	
		2	Registro de Riesgos						
		1	Analista	16	horas	40,00 €		640,00 €	
		3	Gestión de riesgos						
		1	Analista	21	horas	40,00 €		840,00 €	
3			Presupuestos						1.600,00 €
		1	Presupuesto de costes					840,00 €	
		1	Analista	21	horas	40,00 €	840,00 €		
		2	Presupuesto de cliente						760,00 €
		1	Analista	19	horas	40,00 €	760,00 €		
TOTAL									4.520,00 €

Figura 5.21: Partida Planificación

Documentación									
I1	I2	I3	Descripción	Cantidad	Unidades	Precio	Subtotal(3)	Subtotal(2)	Total
1			Planificación del sistema						2.080,00 €
		1	Analista	16	horas	40,00 €		640,00 €	
		2	Arquitecto de software	16	horas	50,00 €		800,00 €	
		3	Desarrollador Software	16	horas	40,00 €		640,00 €	
2			Definición de la arquitectura tecnológica						2.470,00 €
		1	Analista	19	horas	40,00 €		760,00 €	
		2	Arquitecto de software	19	horas	50,00 €		950,00 €	
		3	Desarrollador Software	19	horas	40,00 €		760,00 €	
3			Estudio de la viabilidad						4.160,00 €
		1	Analista	32	horas	40,00 €		1.280,00 €	
		2	Arquitecto de software	32	horas	50,00 €		1.600,00 €	
		3	Desarrollador Software	32	horas	40,00 €		1.280,00 €	
4			Manuales						2.790,00 €
		1	Manual de instalación					630,00 €	
		1	Arquitecto de software	7	horas	50,00 €	350,00 €		
		2	Desarrollador Software	7	horas	40,00 €	280,00 €		
		2	Manual de ejecución					720,00 €	
		1	Arquitecto de software	8	horas	50,00 €	400,00 €		
		2	Desarrollador Software	8	horas	40,00 €	320,00 €		
		3	Manual de usuario					720,00 €	
		1	Arquitecto de software	8	horas	50,00 €	400,00 €		
		2	Desarrollador Software	8	horas	40,00 €	320,00 €		
		4	Manual del Programador					720,00 €	
		1	Arquitecto de software	8	horas	50,00 €	400,00 €		
		2	Desarrollador Software	8	horas	40,00 €	320,00 €		
TOTAL									11.500,00 €

Figura 5.22: Partida Documentación

Análisis									
I1	I2	I3	Descripción	Cantidad	Unidades	Precio	Subtotal(3)	Subtotal(2)	Total
1			Requisitos						1.260,00 €
	1		Analista	14	horas	40,00 €		560,00 €	
	2		Arquitecto de software	14	horas	50,00 €		700,00 €	
2			Análisis casos de uso						920,00 €
	1		Analista	23	horas	40,00 €		920,00 €	
3			Análisis de clases						1.360,00 €
	1		Analista	17	horas	40,00 €		680,00 €	
	2		Desarrollador Software	17	horas	40,00 €		680,00 €	
4			Definición de interfaces de usuario						1.840,00 €
	1		Analista	23	horas	40,00 €		920,00 €	
	2		Desarrollador Software	23	horas	40,00 €		920,00 €	
5			Especificación de plan de pruebas						1.840,00 €
	1		Analista	16	horas	40,00 €		640,00 €	
	2		Desarrollador Software	16	horas	40,00 €		640,00 €	
	3		Tester	16	horas	35,00 €		560,00 €	
TOTAL									5.380,00 €

Figura 5.23: Partida Análisis

Diseño									
I1	I2	I3	Descripción	Cantidad	Unidades	Precio	Subtotal(3)	Subtotal(2)	Total
1			Diseño de casos de uso						1.620,00 €
	1		Desarrollador Software	18	horas	40,00 €		720,00 €	
	2		Arquitecto de software	18	horas	50,00 €		900,00 €	
2			Diseño de la arquitectura						2.070,00 €
	1		Desarrollador Software	23	horas	40,00 €		920,00 €	
	2		Arquitecto de software	23	horas	50,00 €		1.150,00 €	
3			Diseño de clases						2.520,00 €
	1		Desarrollador Software	28	horas	40,00 €		1.120,00 €	
	2		Arquitecto de software	28	horas	50,00 €		1.400,00 €	
4			Especificación técnica del plan de pruebas						1.785,00 €
	1		Tester	21	horas	35,00 €		735,00 €	
	2		Arquitecto de software	21	horas	50,00 €		1.050,00 €	
TOTAL									7.995,00 €

Figura 5.24: Partida Diseño

Desarrollo									
I1	I2	I3	Descripción	Cantidad	Unidades	Precio	Subtotal(3)	Subtotal(2)	Total
1			Implementación del sistema						3.320,00 €
	1		Desarrollador Software	83	horas	40,00 €		3.320,00 €	
2			Integración con sistemas externos						1.960,00 €
	1		Desarrollador Software	49	horas	40,00 €		1.960,00 €	
TOTAL									5.280,00 €

Figura 5.25: Partida Desarrollo

Pruebas									
I1	I2	I3	Descripción	Cantidad	Unidades	Precio	Subtotal(3)	Subtotal(2)	Total
1			Pruebas de usabilidad						910,00 €
		1	Tester	26	horas	35,00 €		910,00 €	
2			Pruebas unitarias						2.870,00 €
		1	Tester	82	horas	35,00 €		2.870,00 €	
TOTAL									3.780,00 €

Figura 5.26: Partida Pruebas

Otros costes									
I1	I2	I3	Descripción	Cantidad	Unidades	Precio	Subtotal(3)	Subtotal(2)	Total
1			Consumo eléctrico	229	horas	0,20 €			45,80 €
2			Dietas	400	dietas	10,00 €			4.000,00 €
TOTAL									4.045,80 €

Figura 5.27: Partida Otros Costes

5.1.5.2. Presupuesto de Cliente

En este apartado se presenta el presupuesto para el cliente. En él tendremos en cuenta las partidas vistas anteriormente junto con el beneficio que se pretendería obtener.

En la siguiente tabla, vemos el total del proyecto sin tener en cuenta la partida *Otros costes*, el beneficio que se pretende obtener (20 %) y la partida mencionada. Debemos aplicar la siguiente fórmula para obtener el porcentaje a compensar:

Total del proyecto	38.455,00 €
Beneficio (20%)	7.691,00 €
Otros costes	4.045,80 €
Porcentaje a compensar	31%

Figura 5.28: Tabla con datos para cálculos de presupuesto de cliente

$$\text{Incremento} \equiv \frac{\text{Beneficio} + \text{Otros costes}}{\text{Coste total}} \equiv \frac{7.691,00 + 4.045,80}{38.455,00} \equiv 31\%$$

Aplicando el porcentaje a compensar, este sería el presupuesto final presentado al cliente.

Presupuesto de cliente					
I1	I2	I3	Descripción		Total
1			Planificación		5.899,54 €
2			Documentación		15.009,90 €
3			Análisis		7.022,02 €
4			Diseño		10.435,14 €
5			Desarrollo		4.933,69 €
6			Pruebas		5.280,61 €
				TOTAL	48.580,91 €

Figura 5.29: Presupuesto de cliente

5.2. EJECUCIÓN DEL PROYECTO

5.2.1. Bitácora de Incidencias del Proyecto

El proyecto se inicia con una reunión con el tutor, donde se habla sobre la perspectiva de la aplicación, los objetivos y un esbozo inicial de la planificación. Se comienza a elaborar un poco más esta con la intención de terminar en la fecha prevista.

El tiempo comienza a pasar y las fechas y tiempos no se cumplen, se cambian las fechas y el proyecto se empieza a retrasar mucho.

La falta de conocimiento sobre el lenguaje de programación y el framework utilizado hacen que los tiempos sean realmente largos y se vaya retrasando todo mucho más de lo deseado.

Una vez se consiguen los conocimientos suficientes para completar los objetivos del sistema, los tiempos propuestos finalmente se comienzan a cumplir, proponiendo la fecha de finalización a junio de 2023. A lo largo del tiempo planificado, se va a hablando con el tutor en diferentes reuniones para revisar el trabajo realizado, además de diferentes versiones de la documentación, donde se va concretando qué camino seguir y qué decisiones se han de tomar.

5.3. CIERRE DEL PROYECTO

5.3.1. Planificación Final

Como se comentó previamente, hubo retrasos en la planificación, por lo que se tuvo que ir adaptando. Viendo que lo que fallaba era falta de tiempo para el desarrollo, se adaptó la planificación para aumentar el tiempo de este.



Figura 5.30: Planificación final

5.3.2. Presupuesto Final de Costes

Al haber cambiado la planificación, el presupuesto lo hace también. Para no repetir demasiados datos, se mostrará únicamente el presupuesto agregado y el presupuesto de cliente. En caso de ver el desglose de las partidas se puede ver en el archivo *PresupuestoFinal.xlsx*.

Como se puede observar, al haber tenido que aumentar el tiempo de vida del proyecto el coste ha aumentado.

Presupuesto agregado					
I1	I2	I3	Descripción		Total
1			Planificación		3.880,00 €
2			Documentación		12.280,00 €
3			Análisis		5.060,00 €
4			Diseño		7.390,00 €
5			Desarrollo		8.240,00 €
6			Pruebas		6.790,00 €
7			Otros costes		4.045,80 €
TOTAL					47.685,80 €

Figura 5.31: Presupuesto agregado final

Total del proyecto	43.640,00 €
Beneficio (20%)	8.728,00 €
Otros costes	4.045,80 €
Porcentaje a compensar	29%

Figura 5.32: Tabla con datos para cálculos de presupuesto final de cliente

Presupuesto de cliente					
I1	I2	I3	Descripción		Total
1			Planificación		5.015,71 €
2			Documentación		15.874,46 €
3			Análisis		6.541,11 €
4			Diseño		9.553,12 €
5			Desarrollo		8.777,49 €
6			Pruebas		5.230,04 €
TOTAL					50.991,92 €

Figura 5.33: Presupuesto de cliente final

Capítulo 6

ANÁLISIS DEL SISTEMA DE INFORMACIÓN

FASE DE DESARROLLO

ASI

6.1. DEFINICIÓN DEL SISTEMA

6.1.1. Determinación del Alcance del Sistema

En la actualidad ya existen numerosas aplicaciones de navegación GPS que tienen un dominio bastante grande del mercado y a las que los usuarios ya están acostumbrados. El objetivo de este proyecto es realizar una aplicación móvil con las funciones generales ya conocidas por el usuario promedio, tratando de juntar las mejores características de las ya existentes, manteniendo una interfaz limpia y fácil de usar, añadiendo además alguna funcionalidad extra que parece de utilidad para los usuarios.

Para el intercambio de información geoespacial dentro de la aplicación se utilizará el estándar GPX.

La aplicación permitirá registrar la actividad del usuario, guardando el recorrido seguido, tiempo, distancia, desnivel y ritmo medio. Una vez finalizada la actividad, el usuario podrá visualizar los datos de esta misma, con la posibilidad de enviar a través de un medio compatible el archivo GPX correspondiente.

El usuario podrá importar rutas GPX previamente descargadas en el teléfono móvil. Una vez importadas, el usuario podrá visualizar los datos de estas mismas como distancia, tiempo... además de la propia ruta sobre un mapa, al cual se le podrán aplicar capas de visualización, como por ejemplo, alguna proporcionada por el Instituto Geográfico Nacional.

Además, el usuario podrá seguir estas rutas importadas, registrando también los datos dinámicamente, al igual que al grabar una ruta propia.

6.2. ESTABLECIMIENTO DE REQUISITOS

6.2.1. Obtención de los Requisitos del Sistema

6.2.1.1. Requisitos de Interfaces Externas

- RIE-1.** El sistema deberá utilizar el estándar GPX para el intercambio de información geoespacial
- RIE-2.** El sistema utilizará el SDK de Google Maps para la visualización en mapas
- RIE-3.** El sistema utilizará el sistema operativo para obtener la ubicación del usuario
- RIE-4.** El sistema utilizará el sistema operativo para acceder al sistema de archivos
- RIE-5.** El sistema se deberá conectar con los servicios de mapas del IGN

6.2.1.2. Requisitos Funcionales

6.2.1.2.1 Importación de rutas GPX

- RF-IRGPX-1.** El sistema debe permitir importar rutas en formato GPX
 - RF-IRGPX-1.1.** El sistema mostrará el explorador de archivos del sistema operativo
 - RF-IRGPX-1.2.** El usuario debe escoger un archivo a importar
 - RF-IRGPX-1.2.1.** En caso de ser un archivo de otro tipo que no sea GPX se mostrará un mensaje de aviso al usuario notificando que el formato no es correcto
 - RF-IRGPX-1.2.2.** En caso de ser un archivo GPX
 - RF-IRGPX-1.2.2.1.** El sistema importará el archivo en la aplicación
 - RF-IRGPX-1.2.2.2.** El sistema notificará al usuario de importación con éxito
- RF-IRGPX-2.** El sistema debe permitir a los usuarios consultar las rutas importadas (ver **RF-IRGPX-1.**) desde el dispositivo
 - RF-IRGPX-2.1.** El usuario deberá poder observar los siguientes datos de cada ruta
 - RF-IRGPX-2.1.1.** Nombre
 - RF-IRGPX-2.1.2.** Distancia hasta el inicio de la ruta (en kilómetros)
 - RF-IRGPX-2.1.3.** Distancia total de la ruta (en kilómetros)
 - RF-IRGPX-2.1.4.** Tiempo (en hh:mm)
 - RF-IRGPX-2.1.5.** Desnivel positivo (en metros)
 - RF-IRGPX-2.1.6.** Perfil de elevación (en formato gráfica)
 - RF-IRGPX-2.1.7.** Altura máxima (en metros)
 - RF-IRGPX-2.1.8.** Altura mínima (en metros)
 - RF-IRGPX-2.1.9.** Enlace a compartir
 - RF-IRGPX-2.1.10.** Visualización del camino sobre un mapa
 - RF-IRGPX-2.1.10.1.** Línea representativa del camino
 - RF-IRGPX-2.1.10.2.** Punto de inicio
 - RF-IRGPX-2.1.10.3.** Punto de fin

- RF-IRGPX-2.1.10.4.** Waypoints (como marcadores)
- RF-IRGPX-2.1.10.5.** El sistema deberá ofrecer la posibilidad de ver los siguientes datos sobre cada Waypoint:
 - RF-RA-1.** Nombre
 - RF-RA-2.** Distancia hasta él
 - RF-RA-3.** Desnivel hasta él
 - RF-RA-4.** Imágenes (en caso de que tenga)
- RF-IRGPX-2.1.10.6.** Capas aplicables al mapa
 - RF-RA-1.** Google Maps Relieve
 - RF-RA-2.** Google Maps Satellite
 - RF-RA-3.** Raster Instituto Geográfico Nacional
 - RF-RA-4.** Open Street Map

6.2.1.2.2 Registro de Actividad

- RF-RA-1.** El sistema debe permitir registrar una nueva actividad
 - RF-RA-1.1.** El sistema deberá crear un archivo GPX para almacenar los datos de la actividad
 - RF-RA-1.2.** El sistema deberá comenzar a detectar la localización del usuario
 - RF-RA-1.2.1.** Cada vez que se actualice, se guardará la localización en el archivo GPX abierto en **RF-RA-1.1.**
 - RF-RA-1.3.** El sistema comenzará a cronometrar el tiempo
 - RF-RA-1.4.** El sistema deberá permitir añadir Waypoints con los siguientes datos
 - RF-RA-1.4.1.** Nombre
 - RF-RA-1.4.2.** Foto(s)
 - RF-RA-1.5.** El sistema mostrará los siguientes datos al usuario:
 - RF-RA-1.5.1.** Tiempo desde que inició la actividad (en hh:mm:ss)
 - RF-RA-1.5.2.** Distancia recorrida (en kilómetros)
 - RF-RA-1.5.3.** Ritmo medio (en kilómetros/hora)
 - RF-RA-1.6.** El usuario podrá pausar el cronómetro.
 - RF-RA-1.7.** Una vez pausado:
 - RF-RA-1.7.1.** El tiempo se pausará
 - RF-RA-1.7.2.** La localización dejará de actualizarse
 - RF-RA-1.7.3.** El usuario podrá continuar la actividad o finalizarla
 - RF-RA-1.7.4.** En caso de finalizarla:
 - RF-RA-1.7.4.1.** El sistema pedirá al usuario por un nombre para la ruta
 - RF-RA-1.7.4.2.** El sistema guardará los datos de la ruta que faltan (nombre obtenido de **RF-RA-1.7.4.1.**, fecha, tiempo obtenido de **RF-RA-1.3.**) en el archivo GPX
 - RF-RA-1.7.4.3.** El sistema mostrará un resumen de la actividad al usuario

- RF-RA-1.7.4.3.1.** Tiempo (en hh:mm:ss)
- RF-RA-1.7.4.3.2.** Distancia recorrida (en kilómetros)
- RF-RA-1.7.4.3.3.** Ritmo medio (en kilómetros/hora)
- RF-RA-1.7.4.3.4.** Desnivel (en metros)
- RF-RA-1.7.4.3.5.** Perfil de elevación (en formato gráfica)
- RF-RA-1.7.4.3.6.** Elevación máxima (en metros)
- RF-RA-1.7.4.3.7.** Elevación mínima (en metros)
- RF-RA-1.7.5.** En caso de continuar:
 - RF-RA-1.7.5.1.** El tiempo continuará contándose desde el valor desde el que se pausó
 - RF-RA-1.7.5.2.** La localización continuará actualizándose
- RF-RA-2.** El sistema debe permitir registrar una nueva actividad siguiendo una ruta importada (ver **RF-IRGPX-1.**, **RF-IRGPX-2.1.**)
 - RF-RA-2.1.** El usuario deberá poder ver un mapa con los respectivos datos
 - RF-RA-2.1.1.** Localización del usuario
 - RF-RA-2.1.2.** Línea representativa de la ruta
 - RF-RA-2.1.3.** Waypoints
 - RF-RA-2.1.4.** El sistema deberá ofrecer la posibilidad de ver los siguientes datos sobre cada Waypoint:
 - RF-RA-2.1.4.1.** Distancia hasta el punto
 - RF-RA-2.1.4.2.** Desnivel hasta el punto
 - RF-RA-2.1.4.3.** Nombre
 - RF-RA-2.1.4.4.** Fotos (en caso de que tenga)
 - RF-RA-2.1.5.** Capas aplicables al mapa al igual que en **RF-IRGPX-2.1.10.6.**
 - RF-RA-2.2.** El sistema deberá permitir lo mismo que en **RF-RA-1.**
- RF-RA-3.** El sistema deberá mostrar los registros de actividad realizados por el usuario
 - RF-RA-3.1.** Por cada ruta mostrará los mismos datos que en **RF-IRGPX-2.1.**
- RF-RA-4.** El sistema deberá permitir registrar una nueva actividad a siguiendo un registro previos
 - RF-RA-4.1.** El sistema deberá permitir lo mismo que en **RF-RA-2.**

6.2.1.2.3 Atributos del sistema

- RNF-SIS-1.** El sistema debe poder ser usado por cualquier usuario aficionado al deporte al aire libre sin conocimientos previos en informática
- RNF-SIS-2.** El sistema debe ser una aplicación nativa de iOS orientada a dispositivos móviles
- RNF-SIS-3.** El sistema debe tener una interfaz adaptable a cualquier dispositivo móvil iOS

6.2.2. Identificación de Actores del Sistema

6.2.2.1. Usuario

En el sistema propuesto tan solo existe un usuario genérico. Este actor tiene la capacidad de usar todas las funcionalidades del sistema.

6.2.3. Especificación de Casos de Uso

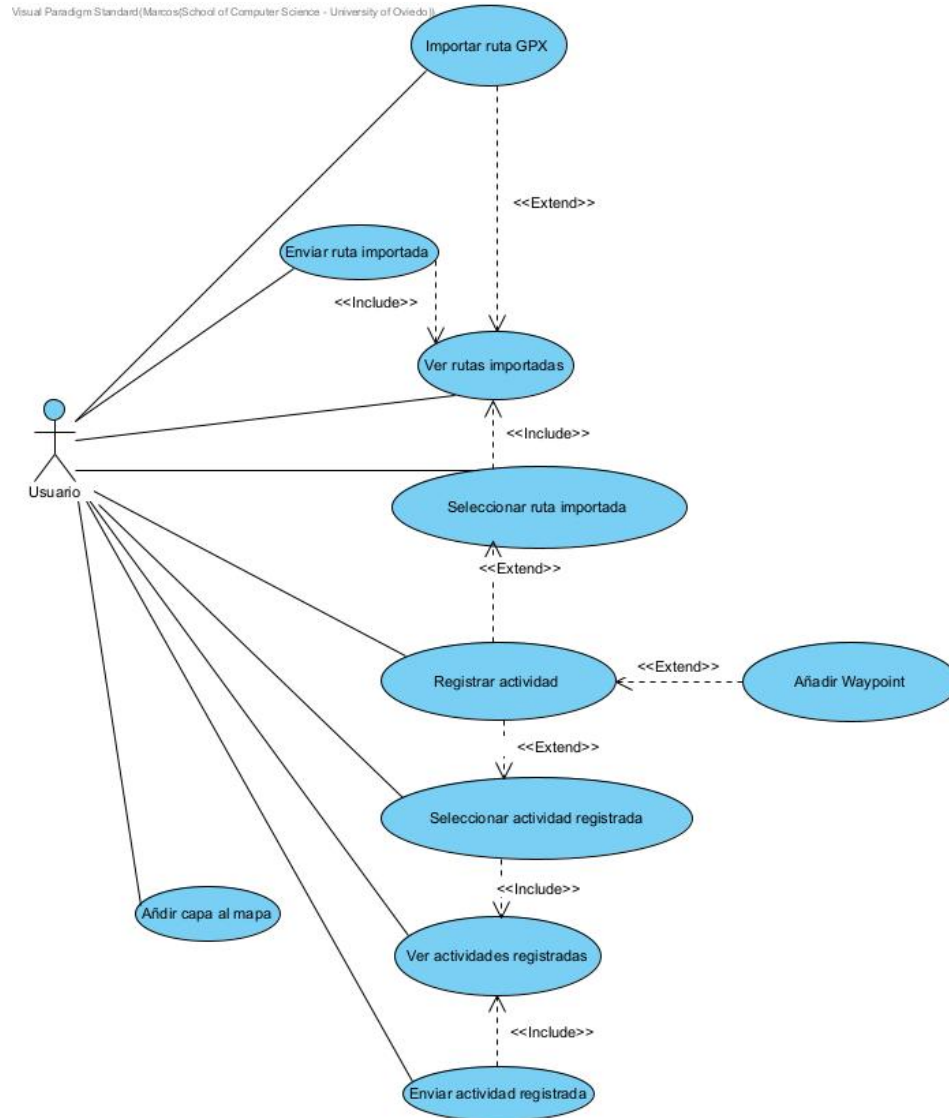


Figura 6.1: Diagrama de casos de uso

Tabla 6.1: Especificación Caso de Uso 1: Importar ruta GPX

Nombre del caso de uso
Importar ruta GPX
Descripción
Un usuario debe poder importar una ruta en formato GPX descargada previamente en el dispositivo.

Tabla 6.2: Especificación Caso de Uso 2: Ver rutas importadas

Nombre del caso de uso
Ver rutas importadas
Descripción
Un usuario debe poder ver todas las rutas GPX importadas previamente.

Tabla 6.3: Especificación Caso de Uso 3: Enviar ruta importada

Nombre del caso de uso
Enviar ruta importada
Descripción
Un usuario debe poder enviar a través de medios compatibles una ruta GPX previamente importada en la aplicación.

Tabla 6.4: Especificación Caso de Uso 4: Seleccionar ruta importada

Nombre del caso de uso
Seleccionar ruta importada
Descripción
Un usuario debe poder seleccionar una ruta importada para poder ver sus datos en detalle.

Tabla 6.5: Especificación Caso de Uso 5: Registrar actividad

Nombre del caso de uso
Registrar actividad
Descripción
Un usuario debe poder registrar una nueva actividad a la que, además, podrá añadir waypoints.

Tabla 6.6: Especificación Caso de Uso 6: Seleccionar actividad registrada

Nombre del caso de uso
Seleccionar actividad registrada
Descripción
Un usuario debe poder seleccionar una actividad registrada para poder ver sus datos en detalle.

Tabla 6.7: Especificación Caso de Uso 7: Ver actividades registradas

Nombre del caso de uso
Ver actividades registradas
Descripción
Un usuario debe poder ver todas sus actividades registradas previamente.

Tabla 6.8: Especificación Caso de Uso 8: Enviar actividad registrada

Nombre del caso de uso
Enviar actividad registrada
Descripción
Un usuario debe poder enviar una actividad registrada a través de los medios compatibles disponibles en el dispositivo.

Tabla 6.9: Especificación Caso de Uso 9: Añadir capa al mapa

Nombre del caso de uso
Añadir capa al mapa
Descripción
Un usuario debe poder añadir diferentes capas al mapa.

Tabla 6.10: Especificación Caso de Uso 10: Añadir waypoint

Nombre del caso de uso
Añadir waypoint
Descripción
Un usuario debe poder añadir waypoints a la actividad que está registrando.

6.3. ANÁLISIS DE LOS CASOS DE USO

6.3.1. Caso de Uso 1: Importar ruta GPX

Tabla 6.11: Análisis del Caso de Uso 1: Importar ruta GPX

Importar ruta GPX	
Precondiciones	El usuario debe haber descargado previamente la ruta y debe estar disponible en el sistema de archivos del dispositivo.
Postcondiciones	El archivo se guardará en los datos internos de la aplicación y se mostrará un mensaje de éxito.
Actores	Usuario
Descripción	El usuario seleccionará en el menú de navegación la opción de importar archivo. A continuación, el explorador de archivos del sistema se abrirá y el usuario seleccionará el archivo que desea importar en la aplicación.
Escenarios Secundarios	1. El usuario selecciona un archivo que no está en formato GPX. En este caso se mostrará un mensaje de error.

6.3.2. Caso de Uso 2: Ver rutas importadas

Tabla 6.12: Análisis del Caso de Uso 2: Ver rutas importadas

Ver rutas importadas	
Precondiciones	-
Postcondiciones	-
Actores	Usuario
Descripción	El usuario seleccionará en el menú de navegación la opción de ver rutas importadas. A continuación, el sistema mostrará un listado de todas las rutas importadas en la aplicación con sus respectivos datos.
Escenarios Secundarios	-

6.3.3. Caso de Uso 3: Enviar ruta importada

Tabla 6.13: Análisis del Caso de Uso 3: Enviar ruta importada

Enviar ruta importada	
Precondiciones	El usuario debe haber importado previamente la ruta en la aplicación.
Postcondiciones	El archivo se enviará y será accesible a través del medio por el cual se ha enviado.
Actores	Usuario
Descripción	El usuario seleccionará en el menú de navegación la opción de ver rutas importadas. A continuación, pulsará el botón compartir de la ruta que desea enviar. El sistema mostrará los medios compatibles a través de los cuales el usuario puede enviar el archivo. Finalmente, el usuario seleccionará el medio deseado.
Escenarios Secundarios	-

6.3.4. Caso de Uso 4: Seleccionar ruta importada

Tabla 6.14: Análisis del Caso de Uso 4: Seleccionar ruta importada

Seleccionar ruta importada	
Precondiciones	El usuario debe haber importado previamente la ruta en la aplicación.
Postcondiciones	El sistema mostrará en un mapa una línea representativa de la ruta incluyendo punto de inicio y fin, además de los waypoints (como marcadores) contenidos.
Actores	Usuario
Descripción	El usuario seleccionará en el menú de navegación la opción de ver rutas importadas. A continuación, pulsará sobre la ruta que desea ver en el mapa.
Escenarios Secundarios	-

6.3.5. Caso de Uso 5: Registrar actividad

Tabla 6.15: Análisis del Caso de Uso 5: Registrar actividad

Registrar actividad	
Precondiciones	-
Postcondiciones	Se guardará un archivo GPX en los datos de la aplicación con toda la información recogida a lo largo de la actividad.
Actores	Usuario
Descripción	El usuario iniciará el registro de una actividad, pudiendo haber seleccionado previamente una ruta importada o una actividad registrada.
Escenarios Secundarios	1. El usuario cancela el registro de la actividad. En este caso, no se guardarán los datos recogidos durante el tiempo que el usuario registró la actividad.

6.3.6. Caso de Uso 6: Seleccionar una actividad registrada

Tabla 6.16: Análisis del Caso de Uso 6: Seleccionar una actividad registrada

Seleccionar una actividad registrada	
Precondiciones	-
Postcondiciones	El sistema mostrará en un mapa una línea representativa de la ruta incluyendo punto de inicio y fin, además de los waypoints (como marcadores) contenidos.
Actores	Usuario
Descripción	El usuario seleccionará en el menú de navegación la opción de ver actividades registradas. A continuación, seleccionará la ruta que desea cargar.
Escenarios Secundarios	-

6.3.7. Caso de Uso 7: Ver actividades registradas

Tabla 6.17: Análisis del Caso de Uso 7: Ver actividades registradas

Ver actividades registradas	
Precondiciones	-
Postcondiciones	El sistema mostrará en un listado todas las actividades registradas por el usuario.
Actores	Usuario
Descripción	El usuario seleccionará en el menú de navegación la opción de ver actividades registradas.
Escenarios Secundarios	-

6.3.8. Caso de Uso 8: Enviar actividad registrada

Tabla 6.18: Análisis del Caso de Uso 8: Enviar actividad registrada

Enviar actividad registrada	
Precondiciones	-
Postcondiciones	El archivo se enviará y será accesible a través del medio por el cual se ha enviado.
Actores	Usuario
Descripción	El usuario seleccionará en el menú de navegación la opción de ver actividades registradas. A continuación, pulsará en el botón de compartir de la ruta que desea enviar. El sistema mostrará los medios disponibles para el usuario y este seleccionará el deseado.
Escenarios Secundarios	-

6.3.9. Caso de Uso 9: Añadir capa al mapa

Tabla 6.19: Análisis del Caso de Uso 9: Añadir capa al mapa

Añadir capa al mapa	
Precondiciones	-
Postcondiciones	La capa se añadirá al mapa.
Actores	Usuario
Descripción	El usuario pulsará el botón de capas del mapa. A continuación, seleccionará la capa deseada.
Escenarios Secundarios	-

6.3.10. Caso de Uso 10: Añadir waypoint

Tabla 6.20: Análisis del Caso de Uso 10

Añadir waypoint	
Precondiciones	El usuario debe estar registrando una actividad.
Postcondiciones	Los waypoints que haya añadido se guardarán junto con el resto de datos de la ruta, además de sus fotos. La localización de cada waypoint será la correspondiente a la del usuario en el momento en que lo guardó.
Actores	Usuario
Descripción	El usuario pulsará el botón de añadir waypoint. Sacará una foto y a continuación introducirá el nombre deseado.
Escenarios Secundarios	-

6.4. ANÁLISIS DE CLASES

6.4.1. Diagrama de Clases

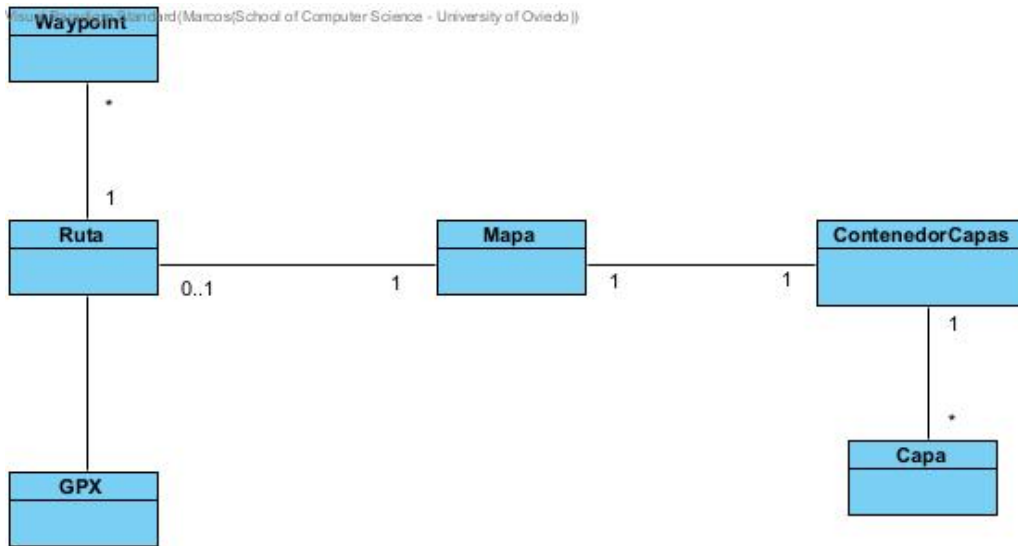


Figura 6.2: Diagrama de casos de uso

6.4.2. Descripción de las Clases

Tabla 6.21: Análisis de la clase Mapa

Mapa	
Descripción	
Representa el elemento mapa de la interfaz de usuario. Es la encargada de mostrar la ruta (en caso de que se añada una) en la vista mapa. Encapsulará también la capa que se le esté aplicando.	
Atributos propuestos	
ruta	Representa la ruta a ser representada en el mapa.
contenedorCapas	Representa un contenedor donde se almacenan las capas que son aplicables al mapa.
Métodos propuestos	
addRuta	Recibe una ruta a ser añadida al mapa. Se sustituirá la que potencialmente está almacenada en el atributo.
iniciarRuta	Reliza las operaciones necesarias para comenzar a grabar una actividad.

Tabla 6.22: Análisis de la clase ContenedorCapas

ContenedorCapas	
Descripción	
Representa un contenedor de las capas que se pueden aplicar a la clase Mapa asociada.	
Atributos propuestos	
mapa	Representa el mapa al que se le pueden aplicar las capas contenidas.
capas	Un listado de las capas disponibles aplicables al mapa asociado.
Métodos propuestos	

Tabla 6.23: Análisis de la clase Capa

Capa	
Descripción	
Representa una capa que se puede aplicar al mapa asociado a su contenedor.	
Atributos propuestos	
nombre	Nombre representativo de la capa.
Métodos propuestos	
addCapa	Aplica la capa que representa al mapa asociado a su contenedor.

Tabla 6.24: Análisis de la clase Ruta

Ruta	
Descripción	
Representación de un camino a visualizar en el mapa o guardar en los datos de la aplicación.	
Atributos propuestos	
path	Lista de coordenadas que se corresponde con la ruta.
distancia	Distancia calculada a partir de las coordenadas.
desnivel	Valor del desnivel calculado a partir del punto de altitud máximo frente al mínimo.
altitudMáxima	Valor de la altitud máxima en la ruta
altitudMínima	Valor de la altitud mínima en la ruta
tiempo	Tiempo necesitado para realizar la ruta (calculado en caso de que sea una ruta importada y directo en caso de ser una actividad registrada).
waypoints	Lista objetos Waypoint donde cada una de ellas representa un punto de interés de la ruta.
Métodos propuestos	
addWaypoint	Recibe un Waypoint y lo añade a la lista.

Tabla 6.25: Análisis de la clase Waypoint

Waypoint	
Descripción	
Representa un punto de interés.	
Atributos propuestos	
coordenadas	Latitud, longitud y altitud del punto.
nombre	Nombre representativo del punto de interés.
descripción	Descripción del punto de interés.
Métodos propuestos	
-	

Tabla 6.26: Análisis de la clase GPX

GPX	
Descripción	
Representa una clase que encapsula las operaciones relacionadas con archivos GPX.	
Atributos propuestos	
-	
Métodos propuestos	
getRutasImportadas	Lee en los datos de la aplicación los archivos GPX importados y devuelve una lista de Rutas que representan estos archivos.
getRutasUsuario	Lee en los datos de la aplicación los archivos GPX de los registros de actividad del usuario y devuelve una lista de Rutas que representan estos archivos.
guardarRuta	Recibe una Ruta, la transforma en archivo GPX y la guarda en los datos de la aplicación.

6.5. DEFINICIÓN DE INTERFACES DE USUARIO

6.5.1. Definición del aspecto de la interfaz

A continuación, se presentan los prototipos de la interfaz diseñados para el sistema. Como se puede observar, todas tienen en común la barra de navegación inferior, que servirá al usuario para moverse a través de toda la aplicación (los botones que se encuentran en este componente tienen unas siglas, que significan lo siguiente: RI - Rutas Importadas, RA - Registrar Actividad, AR - Actividades registradas, Imp - Importar ruta).

Pantalla inicial

Esta será la primera pantalla que observe el usuario una vez se haya cargado la aplicación. El usuario verá la vista del mapa y la barra de navegación.

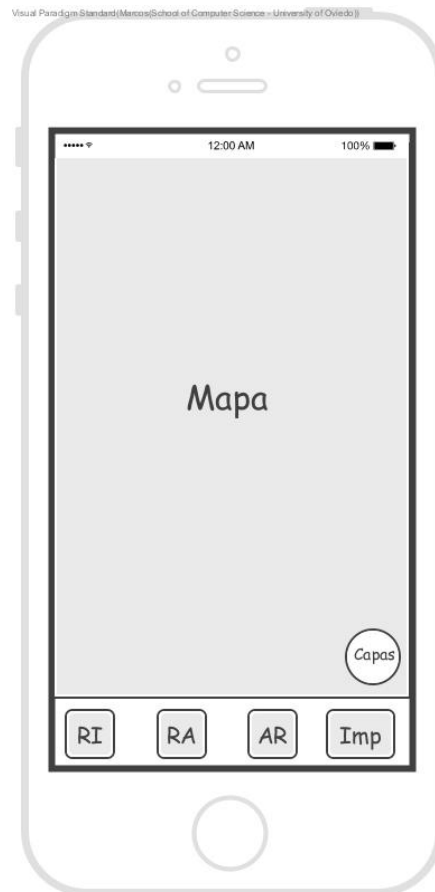


Figura 6.3: Prototipo de la pantalla inicial

Rutas importadas

En la vista de rutas importadas el usuario podrá ver un listado de los archivos GPX que haya importado en la aplicación con sus respectivos datos.



Figura 6.4: Prototipo de vista de rutas importadas

Registrar actividad

En la vista de registrar actividad el usuario tendrá disponible el mapa, los datos generados dinámicamente en la actividad y el menú de navegación.



Figura 6.5: Prototipo de vista registrar actividad

Actividades registradas

En la vista de actividades registradas, el usuario verá un listado de sus actividades registradas con el mismo formato que la vista de rutas importadas. (6.5.1).



Figura 6.6: Prototipo de vista registrar actividad

Importar ruta

En la vista importar ruta, el sistema abrirá el explorador de archivos del dispositivo y el usuario podrá seleccionar el que desea importar.

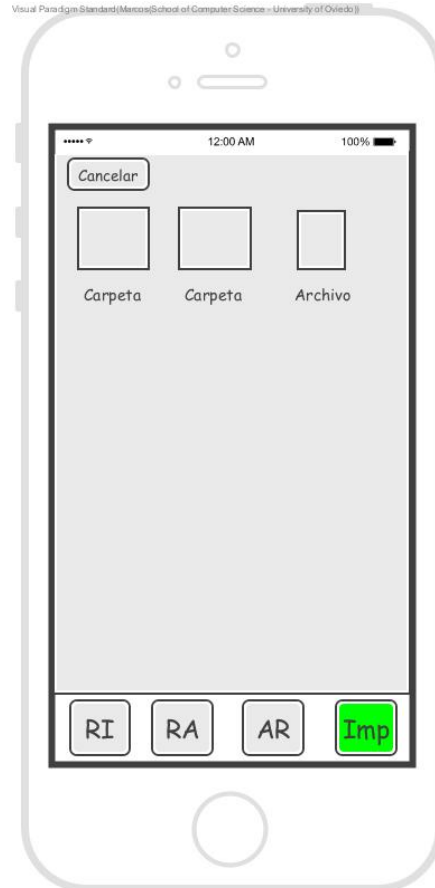


Figura 6.7: Prototipo de vista importar ruta

6.5.2. Diagrama de Navegabilidad

En el siguiente diagrama se puede observar cómo un usuario puede moverse a través de la aplicación.



Figura 6.8: Diagrama de navegabilidad

6.6. ESPECIFICACIÓN DEL PLAN DE PRUEBAS

6.6.1. Pruebas unitarias

Para la realización de pruebas unitarias se utilizará el framework XCTest[17]. Este permitirá pruebas automatizadas de la interfaz.

Tabla 6.27: Pruebas unitarias: Caso de Uso 1 - Importar ruta GPX

Caso de uso 1: Importar ruta GPX		
Prueba		Resultado esperado
Importar archivo con formato correcto	ar- con- co- rrecto	El archivo se guardará en los datos internos de la aplicación y mostrará un mensaje de éxito.
Importar archivo con formato incorrecto	ar- con inco- rrecto	El archivo no se guardará y se mostrará un mensaje de error.

Tabla 6.28: Pruebas unitarias: Caso de Uso 2 - Ver rutas importadas

Caso de uso 2: Ver rutas importadas		
Prueba		Resultado esperado
Obtener rutas importadas		Se mostrará un listado con todas las rutas importadas previamente por el usuario.

Tabla 6.29: Pruebas unitarias: Caso de Uso 3 - Enviar ruta importada

Caso de uso 3: Enviar ruta importada		
Prueba		Resultado esperado
Enviar ruta importada	ruta	El archivo se enviará.

Tabla 6.30: Pruebas unitarias: Caso de Uso 4 - Seleccionar ruta importada

Caso de uso 4: Seleccionar ruta importada		
Prueba		Resultado esperado
Seleccionar ruta importada		El sistema mostrará el botón de inicio de la ruta además de la representación de esta sobre el mapa.

Tabla 6.31: Pruebas unitarias: Caso de Uso 5 - Registrar actividad

Caso de uso 5: Registrar actividad	
Prueba	Resultado esperado
Registro de nueva actividad	El sistema mostrará un resumen de la actividad al terminarla. Además, habrá guardado los datos y los mostrará en el listado de actividades registradas.

Tabla 6.32: Pruebas unitarias: Caso de Uso 6 - Seleccionar actividad registrada

Caso de uso 6: Seleccionar actividad registrada	
Prueba	Resultado esperado
Seleccionar actividad registrada	El sistema mostrará el botón de inicio de la ruta además de la representación de esta sobre el mapa.

Tabla 6.33: Pruebas unitarias: Caso de Uso 7 - Ver actividades registradas

Caso de uso 7: Ver actividades registradas	
Prueba	Resultado esperado
Obtener actividades registradas	El sistema mostrará todas las actividades realizadas previamente por el usuario.

Tabla 6.34: Pruebas unitarias: Caso de Uso 8 - Enviar actividad registrada

Caso de uso 8: Enviar actividad registrada	
Prueba	Resultado esperado
Enviar actividad registrada	El sistema enviará por el medio seleccionado el archivo GPX correspondiente a la ruta seleccionada.

Tabla 6.35: Pruebas unitarias: Caso de Uso 9 - Añadir capa al mapa

Caso de uso 9: Añadir capa al mapa	
Prueba	Resultado esperado
Añadir capa al mapa	El sistema añadirá la capa seleccionada por el usuario al mapa.

Tabla 6.36: Pruebas unitarias: Caso de Uso 10 - Añadir waypoint

Caso de uso 10: Añadir waypoint		
Prueba		Resultado esperado
Añadir point	way-	El sistema guardará el punto de interés y sus datos junto con los de la ruta.

6.6.2. Pruebas de usabilidad

Se realizarán pruebas de usabilidad de la aplicación, en las que, a través de cuestionarios se obtendrán resultados tras observar y anotar cómo una serie usuarios a los que se les pedirá realizar una serie de tareas interactúan con el sistema

Capítulo 7

DISEÑO DEL SISTEMA DE INFORMACIÓN

FASE DE DESARROLLO

DSI

7.1. DISEÑO DE CASOS DE USO REALES

7.1.1. Caso de Uso 5: Registrar actividad

A continuación, se puede observar los diagramas de secuencia representando el caso de uso *Registrar actividad*. Para ser más exactos, es la extensión de este mismo, realizando previamente el caso de uso *Seleccionar ruta importada* que su vez incluye *Ver rutas importadas*.

Esta selección se debe a que todo el conjunto de acciones que forman este caso de uso y su extensión, conforma prácticamente toda la funcionalidad principal del sistema. El objetivo de la aplicación, es decir, seguir una ruta previamente importada en archivo GPX, está representado casi al completo gracias a ello.

Entrando en detalle de lo que se puede ver en los propios diagramas, se puede observar como el usuario le pide al sistema ver las rutas que haya importado previamente. Este, llegando hasta el almacenamiento interno de la aplicación (donde estarán guardados todos los archivos GPX), tratará adecuadamente los archivos necesarios hasta finalmente mostrar una representación de ellos en el listado. Llegados a este punto, el usuario podrá seleccionar el archivo deseado en el listado e iniciar la actividad.

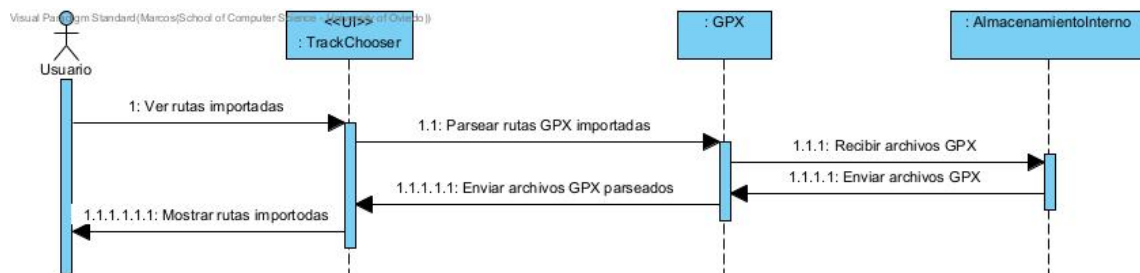


Figura 7.1: Diagrama de secuencia del Caso de Uso Ver rutas importadas

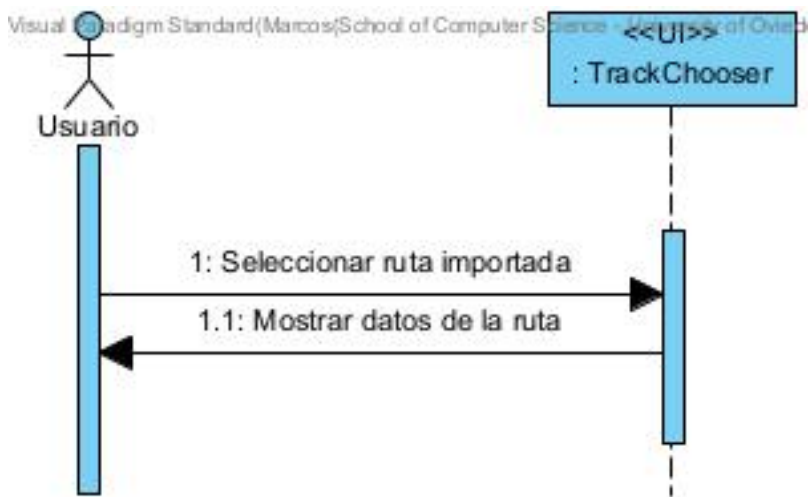


Figura 7.2: Diagrama de secuencia del Caso de Seleccionar ruta importada

Tras estas acciones, el listado deja de ser el receptor de las acciones y ahora el usuario trata con los diferentes registros. Al hacer la llamada de finalizar la ruta, primeramente el usuario verá un resumen de esta, pudiendo introducir un nombre y finalmente terminando su interacción con el sistema. Este, guardará en un archivo GPX todos los datos recogidos previamente y notificará al usuario con el resultado de la operación.

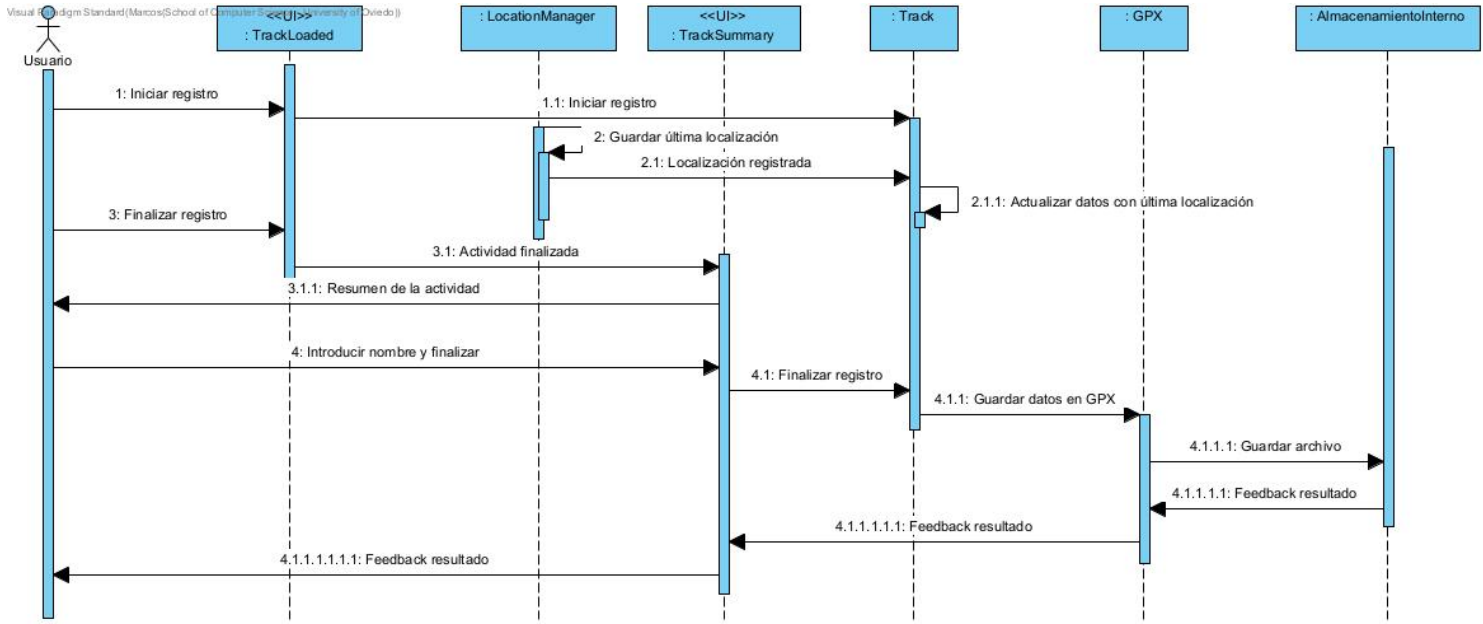


Figura 7.3: Diagrama de secuencia del Caso de Uso Registrar actividad

7.2. DISEÑO DE LA ARQUITECTURA DE MÓDULOS DEL SISTEMA

Una vez más, entra en juego las características del framework a la hora de representar los diagramas. Al haber seleccionado *SwiftUI*, la arquitectura recomendada para las aplicaciones (puede haber excepciones que no usen este modelo, pero la gran mayoría, y en este caso también, lo usarán) es el MVVM[18] (Model View View-Model). Lo que implica esta arquitectura es que, por un lado tendremos todas las entidades que representan el dominio (modelo), por otro lado las vistas (vista) y finalmente los elementos que unen lo primero con lo segundo (vista-modelo).

7.2.1. Diseño de módulos del sistema

7.2.1.1. Diagrama de paquetes

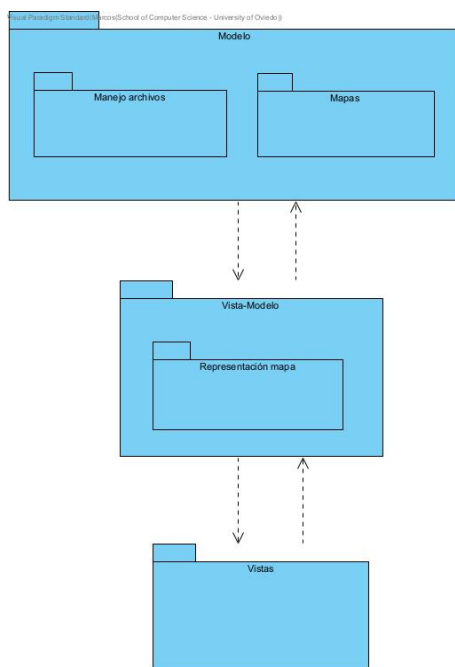


Figura 7.4: Diagrama de paquetes

Lo que se consigue con este diseño de la arquitectura, es separar las vistas del modelo que representan.

7.3. DISEÑO DE CLASES

7.3.1. Diagrama de Clases

A continuación, se presenta el diagrama de clases representativo del sistema. Lo que se puede observar en estos diagramas, es una evolución de lo visto previamente en la fase de análisis: se añaden abstracciones a través de interfaces, se especifican atributos y operaciones de las entidades y se relacionan con más detalle.

Para evitar confusiones, hay que tener en cuenta que a pesar de que en este diagrama se presente todo como clases e interfaces, debido a las características del lenguaje *Swift* y el framework *SwiftUI*, esto al llevarlo a la implementación no será así. Este lenguaje no se basa en interfaces, como sería el caso de Java, sino que es orientado a protocolos, por lo que todo lo que aparece como interfaz en estos diagramas, es realmente un protocolo. Además, SwiftUI recomienda el uso de estructuras para representar el modelo e implementar los protocolos. Debido a esto, al plasmar este diagrama en el código, prácticamente todas las entidades no serán clases, sino estructuras.

7.3.1.1. Paquete de clases relacionadas con el mapa

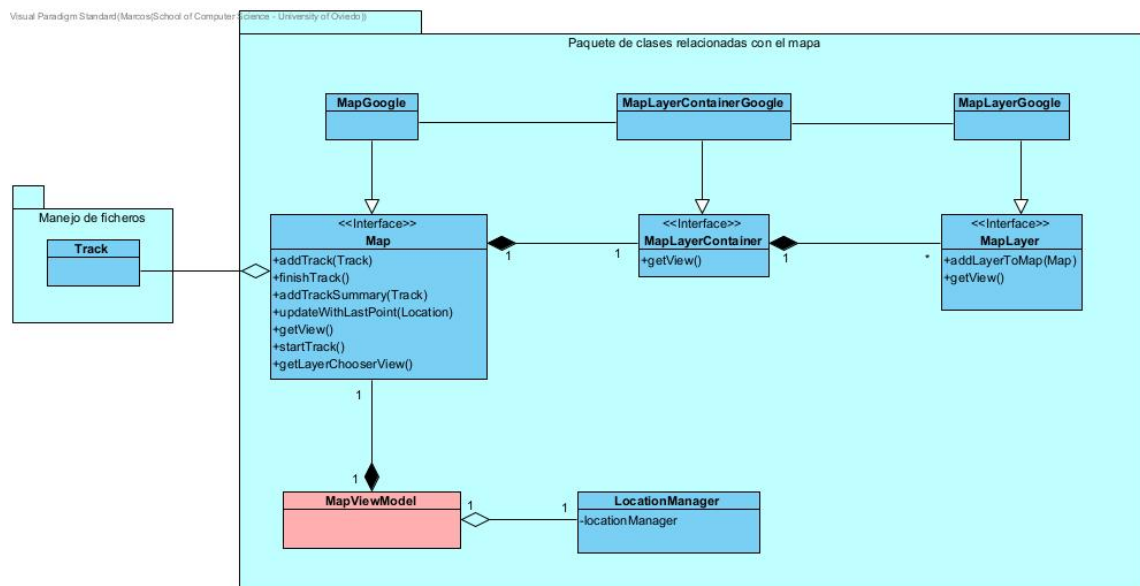


Figura 7.5: Paquete de clases relacionadas con el mapa

Cabe destacar que, como se puede observar, hay una entidad de diferente color al resto. Esto ocurre debido a que, no es realmente necesaria dicha entidad para el sistema, sino que, se origina debido a las características del framework seleccionado previamente, *SwiftUI*. La finalidad de esta clase es servir de contenedor para la entidad *Map* de manera que, cada vez que ocurra un cambio

en este objeto, se notifique a todas las vistas que estén escuchando los mensajes enviados por esta.

Por otro lado, vemos como *Map* es realmente quien recibe todas las llamadas. Esto se debe a que al trabajar cada SDK de visualización de mapas de forma diferente, debemos delegar en la implementación específica que hagamos para cada uno de ellos todas las operaciones. En el caso de este sistema, como ya se comentó previamente, solo se soporta *Google Maps* por lo que es la entidad concreta *MapGoogle* quien sabe como actuar. En caso de que se quisiera añadir otro SDK, como por ejemplo, el de *MapBox*, sería la entidad concreta que creemos de *MapBox* quien supiera como actuar con cada operación: puede ser que para *Google Maps* haya que hacer llamadas a su SDK en ocasiones en las que en el de *MapBox* no haga falta y viceversa.

En este diagrama entra en juego otra característica de las tecnologías que estamos usando. En este caso, no es por *SwiftUI* como en las otras ocasiones, sino por *Swift*. Como podemos ver, las clases concretas *MapGoogle*, *MapLayerContainerGoogle* y *MapLayerGoogle* están asociadas cuando las habíamos abstraído previamente. Esto se debe a un problema que nos encontramos generalmente a la hora de programar usando librerías, frameworks o SDKs externos. Llevándolo a este caso, en tiempo de compilación, yo como programador sé qué mapa o capa del mapa estoy voy a estar usando en ejecución, pero no el compilador. ¿Cómo resuelve *Swift* esto? Con una característica llamada *Associated Types*[19][20], la cual nos permite crear variables o argumentos en las interfaces que su tipo no es una interfaz, si no “algo” (escrito en el código *some*) que implementa esa interfaz. De esta manera, al declarar la clase concreta, estaremos obligados a declarar el tipo de esa variable o argumento, sabiendo en tiempo de compilación con qué tipo estamos trabajando pudiendo acceder a todas sus operaciones.

7.3.1.2. Paquete de clases relacionadas con el manejo de ficheros y su representación dentro del dominio

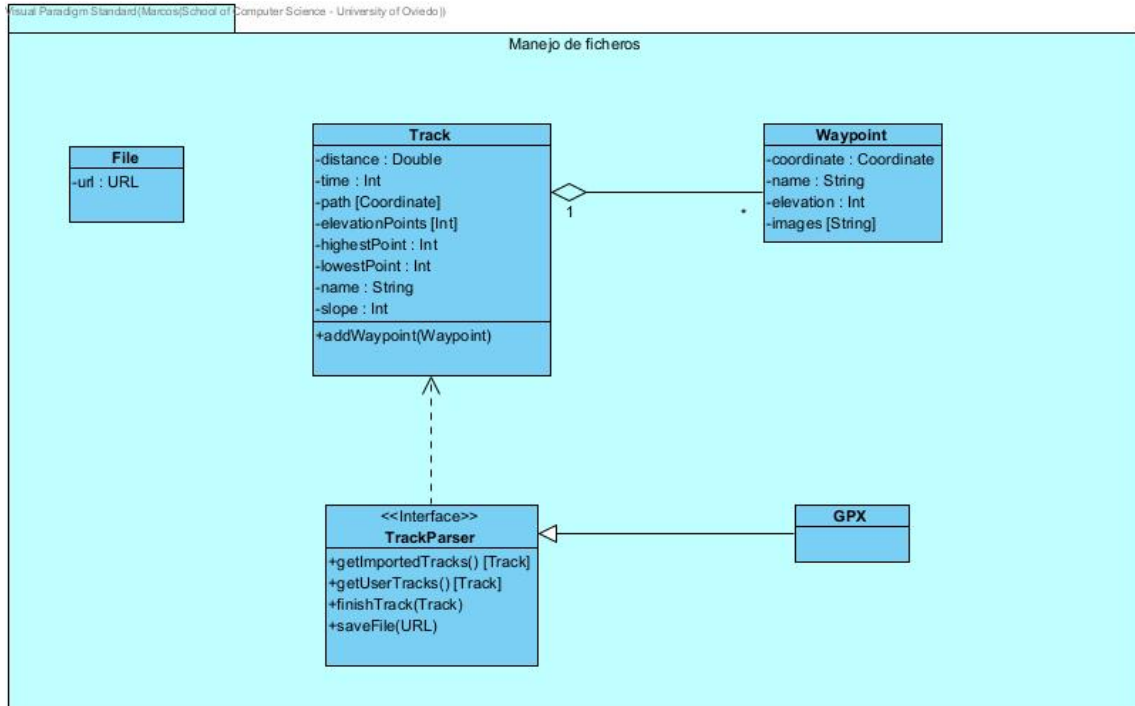


Figura 7.6: Paquete de clases relacionadas con el manejo de ficheros

En este diagrama podemos observar las entidades relacionadas con el manejo de ficheros GPX y su representación en el dominio de la aplicación.

Un archivo GPX, se corresponderá con una instancia de *Track* la cual, como se puede observar, se obtendrá a través de las diferentes operaciones que nos ofrecerá el parseador.

A pesar de que este proyecto se pensó desde un principio para trabajar en todo momento con GPX, parece necesario y conveniente abstraer esta lógica y no atarse únicamente a este formato, ya existen otros formatos muy conocidos y usados, como por ejemplo KML.

7.4. ESPECIFICACIÓN TÉCNICA DEL PLAN DE PRUEBAS

7.4.1. Pruebas Unitarias y del Sistema

Tabla 7.1: Diseño de pruebas: Importar ruta GPX

Importar ruta GPX		
Prueba	Pasos	Resultado esperado
Importar archivo con formato correcto	Descargar un archivo GPX en el dispositivo (en el centro de descargas del IGN se pueden descargar numerosas rutas en formato GPX. Cualquiera es válida). En el menú de navegación, pulsar en el botón de importar archivo. A continuación, seleccionar el archivo previamente descargado.	El archivo se guardará en los datos internos de la aplicación y mostrará un mensaje de éxito.
Importar archivo con formato incorrecto	La aplicación no permite seleccionar archivos con extensión incorrecta. Para forzar la situación, cambiar la extensión de un archivo de texto a .xml. En el menú de navegación, pulsar el botón de importar archivo y seleccionar al que le hemos cambiado la extensión.	El archivo no se guardará y se mostrará un mensaje de error.

Tabla 7.2: Diseño de pruebas: Ver rutas importadas

Ver rutas importadas		
Prueba	Pasos	Resultado esperado
Obtener rutas importadas existiendo cero	Acceder a través del menú de navegación a la pantalla de ver rutas importadas	Se mostrará un listado vacío.
Obtener rutas importadas existiendo una o más	En caso de no haberlo hecho, ejecutar los pasos del 8.3 caso de uso Importar ruta GPX. Acceder a través del menú de navegación a la pantalla de ver rutas importadas	Se mostrará un listado con todas las rutas.

Tabla 7.3: Diseño de pruebas: Enviar ruta importada

Enviar ruta importada		
Prueba	Pasos	Resultado esperado
Enviar ruta importada. WhatsApp	Acceder a través del menú de navegación a la pantalla de ver rutas importadas. Pulsar en el botón de compartir de la primera ruta del listado y seleccionar como medio WhatsApp. Seleccionar finalmente un contacto.	El contacto recibirá un mensaje con el archivo GPX.
Enviar ruta importada. Email	Acceder a través del menú de navegación a la pantalla de ver rutas importadas. Pulsar en el botón de compartir de la primera ruta del listado y seleccionar como medio Email. Seleccionar finalmente un contacto.	El contacto recibirá un mensaje con el archivo GPX.

Tabla 7.4: Diseño de pruebas: Seleccionar ruta importada

Seleccionar ruta importada		
Prueba	Pasos	Resultado esperado
Seleccionar ruta importada.	Acceder a través del menú de navegación a la pantalla de ver rutas importadas. Pulsar sobre la primera del listado.	El sistema mostrará el botón de inicio de la ruta además de la representación de esta sobre el mapa.

Tabla 7.5: Diseño de pruebas: Registrar actividad

Registrar actividad		
Prueba	Pasos	Resultado esperado
Registrar actividad sin ruta seleccionada. Caso completo	Acceder a través del menú de navegación a la pantalla de registrar ruta. Iniciar el cronómetro y hacer el camino deseado. Pulsar en el botón de añadir waypoint. Sacar una foto, añadir nombre al punto de interés y guardar. Pausar el cronómetro durante unos segundos y pulsar continuar. Pausar de nuevo el cronómetro y finalizar. En la siguiente pantalla introducir el nombre y guardar.	El sistema mostrará un resumen de la actividad en la pantalla de introducir el nombre. Además, habrá guardado los datos con el nombre introducido y los mostrará en el listado de actividades registradas.
Registrar actividad sin ruta seleccionada. Caso cancelado	Acceder a través del menú de navegación a la pantalla de registrar ruta. Iniciar el cronómetro y hacer el camino deseado. Pulsar en el botón de cancelar.	El sistema volverá a la pantalla de inicio sin haber registrado ninguna actividad.
Registrar actividad con ruta seleccionada. Caso completo	Seguir los pasos del caso de uso seleccionar ruta importada (8.3). Llegado a este punto, pulsar el botón de iniciar. Pulsar en el botón de añadir waypoint. Sacar una foto, añadir nombre al punto de interés y guardar. Iniciar el cronómetro y hacer el camino deseado. Pausar el cronómetro durante unos segundos y pulsar continuar. Pausar de nuevo el cronómetro y finalizar. En la siguiente pantalla introducir el nombre y guardar.	El sistema mostrará un resumen de la actividad en la pantalla de introducir el nombre. Además, habrá guardado los datos con el nombre introducido y los mostrará en el listado de actividades registradas.

Prueba	Pasos	Resultado esperado
Registrar actividad con ruta seleccionada. Caso cancelado	Seguir los pasos del caso de uso seleccionar ruta importada (8.3). Llegado a este punto, pulsar el botón de iniciar. Iniciar el cronómetro y hacer el camino deseado. Pulsar en el botón de cancelar.	El sistema volverá a la pantalla de inicio sin haber registrado ninguna actividad.
Registrar actividad con actividad registrada. Caso completo	Seguir los pasos del caso de uso seleccionar actividad registrada (8.3). Llegado a este punto, pulsar el botón de iniciar. Pulsar en el botón de añadir waypoint. Sacar una foto, añadir nombre al punto de interés y guardar. Iniciar el cronómetro y hacer el camino deseado. Pausar el cronómetro durante unos segundos y pulsar continuar. Pausar de nuevo el cronómetro y finalizar. En la siguiente pantalla introducir el nombre y guardar.	El sistema mostrará un resumen de la actividad en la pantalla de introducir el nombre. Además, habrá guardado los datos con el nombre introducido y los mostrará en el listado de actividades registradas.
Registrar actividad con ruta seleccionada. Caso cancelado	Seguir los pasos del caso de uso seleccionar actividad registrada (8.3). Llegado a este punto, pulsar el botón de iniciar. Iniciar el cronómetro y hacer el camino deseado. Pulsar en el botón de cancelar.	El sistema volverá a la pantalla de inicio sin haber registrado ninguna actividad.

Tabla 7.6: Diseño de pruebas: Seleccionar actividad registrada

Seleccionar actividad registrada		
Prueba	Pasos	Resultado esperado
Seleccionar actividad registrada.	Acceder a través del menú de navegación a la pantalla de ver actividades registradas. Pulsar sobre la primera del listado.	El sistema mostrará el botón de inicio de la ruta además de la representación de esta sobre el mapa.

Tabla 7.7: Diseño de pruebas: Ver actividades registradas

Ver actividades registradas		
Prueba	Pasos	Resultado esperado
Obtener actividades registradas existiendo cero	Acceder a través del menú de navegación a la pantalla de ver actividades registradas	Se mostrará un listado vacío.
Obtener actividades registradas existiendo una o más	En caso de no haberlo hecho, ejecutar los pasos del caso de uso registrar actividad (8.3). Acceder a través del menú de navegación a la pantalla de ver actividades registradas	Se mostrará un listado con todas las rutas.

Tabla 7.8: Diseño de pruebas: Enviar actividad registrada

Enviar actividad registrada		
Prueba	Pasos	Resultado esperado
Enviar actividad registrada. WhatsApp	Acceder a través del menú de navegación a la pantalla de ver actividades registradas. Pulsar en el botón de compartir de la primera ruta del listado y seleccionar como medio WhatsApp. Seleccionar finalmente un contacto.	El contacto recibirá un mensaje con el archivo GPX.
Enviar actividad registrada. Email	Acceder a través del menú de navegación a la pantalla de ver actividades registradas. Pulsar en el botón de compartir de la primera ruta del listado y seleccionar como medio Email. Seleccionar finalmente un contacto.	El contacto recibirá un mensaje con el archivo GPX.

Tabla 7.9: Diseño de pruebas: Añadir capa al mapa

Añadir capa al mapa		
Prueba	Pasos	Resultado esperado
Añadir capa al mapa	Pulsar en el botón de capas situado sobre el mapa. Ir cambiando entre ellas.	Se aplicará la capa seleccionada al mapa.

Tabla 7.10: Diseño de pruebas: Añadir waypoint

Añadir waypoint		
Prueba	Pasos	Resultado esperado
Añadir waypoint con una foto	Pulsar en el botón de añadir waypoint situado sobre el mapa. Sacar una foto y a continuación introducir el nombre. Pulsar en guardar.	El waypoint se guardará en los datos de la ruta con la localización del usuario en el momento en que lo guardó. Se guardarán también el nombre y la foto sacada por el usuario.
Añadir waypoint con más de una foto	Pulsar en el botón de añadir waypoint situado sobre el mapa. Sacar una foto. A continuación, pulsar en el botón de añadir otra foto. Sacar otra foto y a continuación introducir el nombre. Pulsar en guardar.	El waypoint se guardará en los datos de la ruta con la localización del usuario en el momento en que lo guardó. Se guardarán también el nombre y las fotos sacada por el usuario.

7.4.2. Pruebas de Usabilidad y Accesibilidad

Para unas pruebas de usabilidad y accesibilidad exitosas y con valor, se escogerán diferentes perfiles de usuario, de manera que la retroalimentación obtenida será diferente según varíe el perfil. Primeramente, se le realizarán una serie de preguntas al usuario para determinar su perfil. A continuación realizará una serie de actividades guiadas. Finalmente, le haremos una serie de preguntas de manera que evalúen la aplicación y su experiencia con ella.

7.4.2.1. Diseño de Cuestionarios

7.4.2.1.1 Preguntas de carácter general

Tabla 7.11: Pruebas de usabilidad: preguntas de carácter general

¿Usa el móvil frecuentemente?

1. Todos los días
2. Varias veces a la semana
3. Ocasionalmente
4. Nunca

¿Qué tipo de actividades realiza con el móvil?

1. Es parte de mi trabajo o profesión
2. Lo uso para ocio
3. Solo para comunicarme

¿Ha usado alguna vez software como el de esta prueba?

1. Sí, he usado software similar
 2. No, aunque si empleo otras aplicaciones que se le pueden asemejar
 3. No, nunca
-

7.4.2.1.2 Actividades guiadas

Estas son las actividades propuestas para que los usuarios hagan pruebas del sistema:

- Importar un archivo GPX
- Ver los datos de un archivo importado previamente
- Registrar una actividad a partir de una ruta importada previamente
- Compartir una actividad registrada

7.4.2.1.3 Preguntas cortas sobre la aplicación y observaciones

Tabla 7.12: Pruebas de usabilidad: preguntas cortas sobre la aplicación y observaciones

Facilidad de uso	Siempre	Frecuentemente	Ocasionalmente	Nunca
<i>¿Sabe donde está dentro de la aplicación?</i>				
<i>¿Necesita ayuda para utilizar la aplicación?</i>				
<i>¿Le resulta sencillo el uso de la aplicación?</i>				
Funcionalidad	Siempre	Frecuentemente	Ocasionalmente	Nunca
<i>¿Funciona cada tarea como Vd. Espera?</i>				
<i>¿El tiempo de respuesta de la aplicación es muy grande?</i>				
Calidad del interfaz				
Aspectos gráficos	Muy adecuado	Adecuado	Poco adecuado	Nada adecuado
<i>El tipo y tamaño de letra es</i>				
<i>Los iconos e imágenes usados son</i>				
<i>Los colores empleados son</i>				
<i>Diseño de la interfaz</i>				
<i>¿Le resulta fácil de usar?</i>				
<i>¿El diseño de las pantallas es claro y atractivo?</i>				
<i>¿Cree que el programa está bien estructurado?</i>				
Observaciones				

7.4.2.1.4 Cuestionario para el responsable de las pruebas

Tabla 7.13: Pruebas de usabilidad: cuestionario para el responsable de las pruebas

Aspecto observado
<i>Tiempo emepleado</i>
<i>Problemas encontrados</i>
<i>Observaciones</i>

Capítulo 8

CONSTRUCCIÓN DEL SISTEMA DE INFORMACIÓN

FASE DE DESARROLLO

CSI

8.1. PREPARACIÓN DEL ENTORNO DE GENERACIÓN Y CONSTRUCCIÓN

8.1.1. Estándares y normas seguidos

A continuación, se observan las normas y estándares seguidos en este proyecto.

8.1.1.1. SwiftUI Clean Architecture

No es realmente una norma o estándar como tal, sino que son diferentes recomendaciones para conseguir un código limpio y mantenible[21][22]. Como ya se comentó en capítulos anteriores, se recomienda la arquitectura MVVM, ya que las características del framework están preparadas para esto.

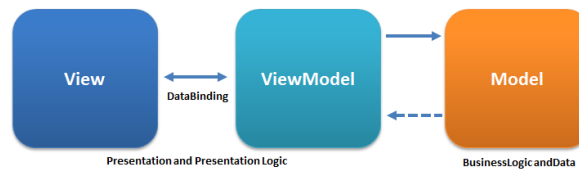


Figura 8.1: MVVM

8.1.1.2. GPX

También se comentó previamente el uso del estándar GPX. Es importante el correcto uso de este para conseguir la interoperabilidad de nuestro sistema con otros.

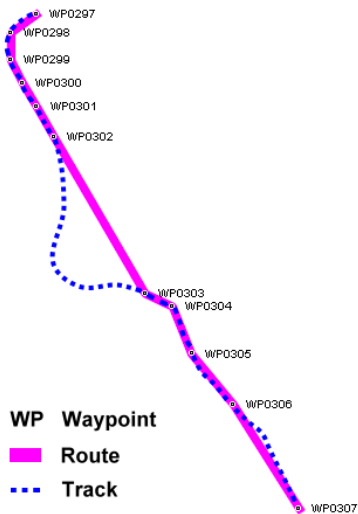


Figura 8.2: GPX

8.1.2. Lenguajes de programación

8.1.2.1. Swift

Swift es el lenguaje de programación nativo de Apple. Ofrece características y llamadas que facilitan el desarrollo de aplicaciones para sistemas Apple.



Figura 8.3: Logo de Swift

8.1.3. Librerías y frameworks

8.1.3.1. SwiftUI

Ya se ha comentado numerosas veces en el documento el uso de este framework. Fue introducido por Apple en 2019 y permite crear interfaces de usuario utilizando el lenguaje *Swift*.



Figura 8.4: Logo de SwiftUI

8.1.3.2. CoreGPX

Librería[23] utilizada para manejar de manera fácil los archivos GPX.



Figura 8.5: Logo de CoreGPX

8.1.3.3. SDK de Google Maps para iOS

Como se vio previamente en otros capítulos, el SDK de visualización de mapas seleccionado fue el de *Google*.



Figura 8.6: Logo de Google Maps

8.1.3.4. Swinject

Librería[24] para el lenguaje de programación *Swift* que facilita la inyección de dependencias.



Figura 8.7: Logo de Swinject

8.1.3.5. AlertToast

Esta librería[25] ha sido utilizada para mostrar fácilmente los clásicos mensajes *toast* de los teléfonos móviles, para mostrar mensajes de éxito y error.

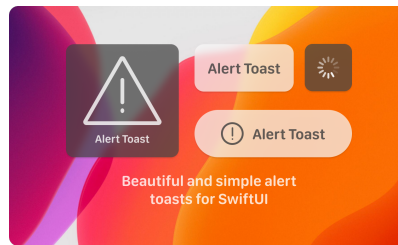


Figura 8.8: Ejemplos de AlertToast

8.1.4. Herramientas y programas usados para el desarrollo

8.1.4.1. Xcode

Xcode[26] es el entorno de desarrollo gratuito creado por Apple para sus desarrolladores. Permite crear y probar aplicaciones fácilmente.



Figura 8.9: Logo de Xcode

8.1.4.2. CocoaPods

CocoaPods[27] es un gestor de dependencias compatible con Swift. En el proyecto ha servido de ayuda para añadir las librerías fácilmente.



Figura 8.10: Logo de CocoaPods

8.1.4.3. Overleaf

Overleaf [28] es un editor y compilador online de Latex. Se ha utilizado herramienta para generar esta documentación propiamente.



Figura 8.11: Logo de Overleaf

8.1.4.4. Microsoft Project

Esta herramienta ha sido utilizada para crear los diagramas de planificación del proyecto.



Figura 8.12: Logo de MS Project

8.1.4.5. Microsoft Excel

Software utilizado para realizar los cálculos de los presupuestos.



Figura 8.13: Logo de Excel

8.1.4.6. Microsoft PowerPoint

Utilizada para crear la presentación del trabajo.



Figura 8.14: Logo de PowerPoint

8.1.4.7. Visual Paradigm

Este programa ha sido utilizado para la creación de los diagramas vistos en este documento.



Figura 8.15: Logo de Visual Paradigm

8.2. GENERACIÓN DEL CÓDIGO DE LOS COMPONENTES Y PROCEDIMIENTOS

Swift tiene junto con *Xcode* un generador de documentación automático. En el código se escribe lo que el desarrollador crea conveniente en formato Markdown[29], y *Xcode* ofrece una herramienta que autoconstruye esta documentación con el formato de Apple Developer. A continuación, se presentan las entidades más significativas.

8.2.1. Track

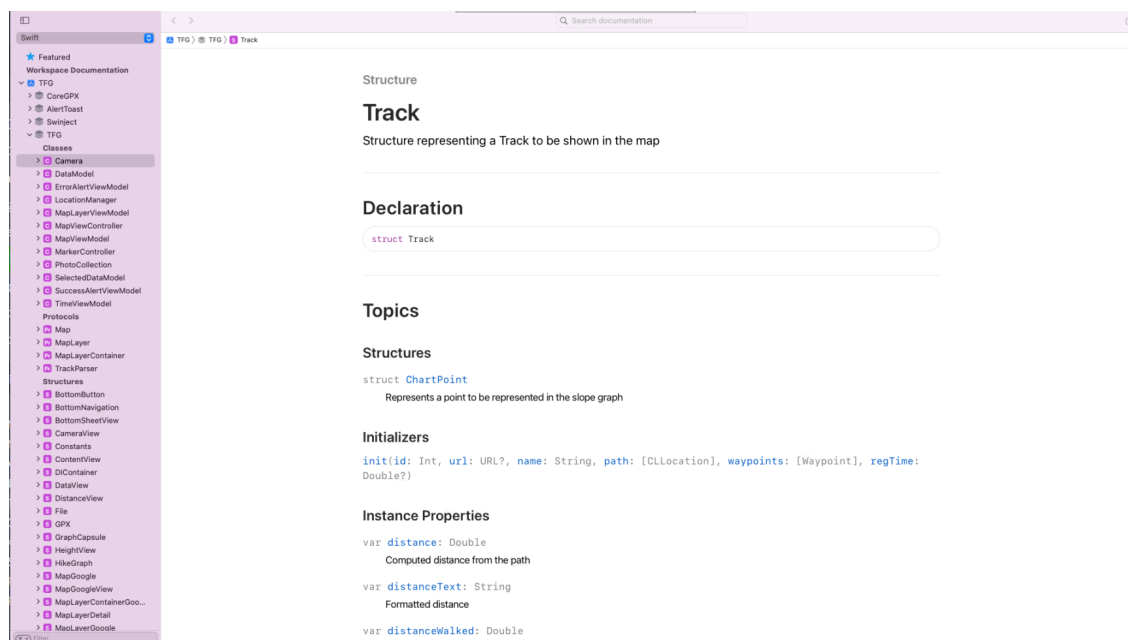


Figura 8.16: Primera parte de la documentación de la estructura Track

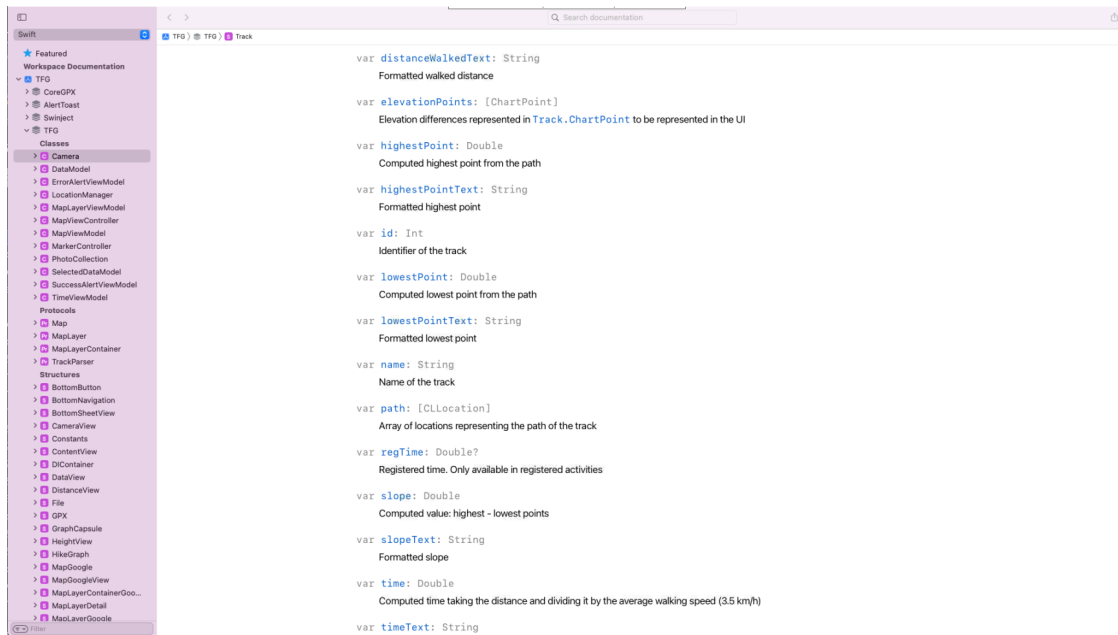


Figura 8.17: Segunda parte de la documentación de la estructura Track

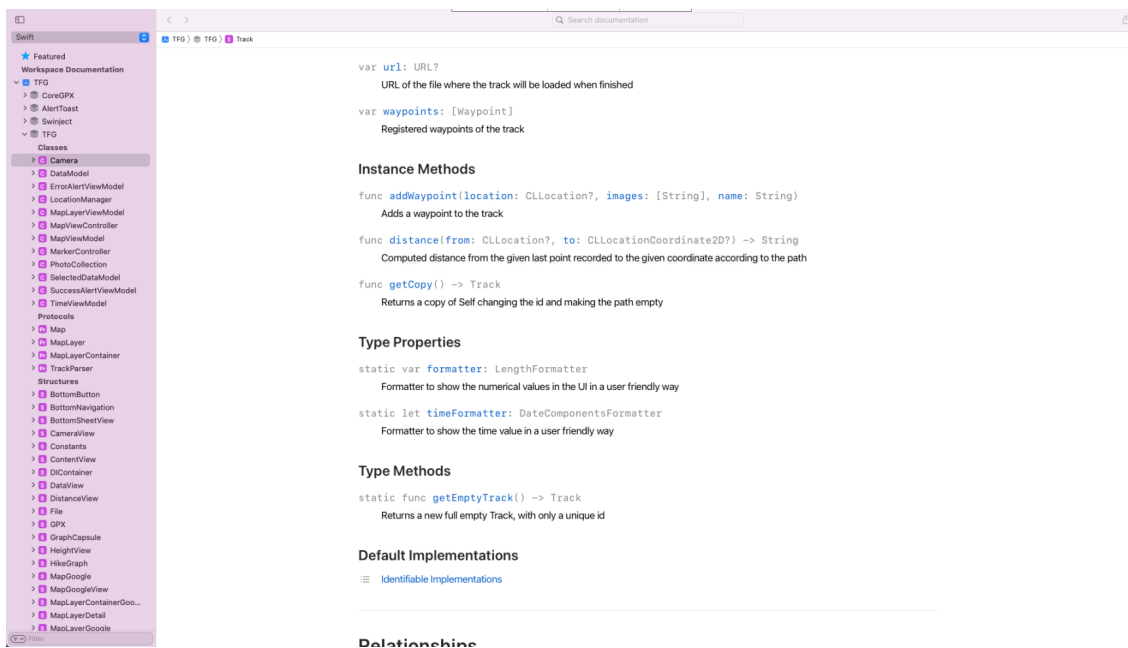


Figura 8.18: Tercera parte de la documentación de la estructura Track



Figura 8.19: Cuarta parte de la documentación de la estructura Track

8.2.2. Map

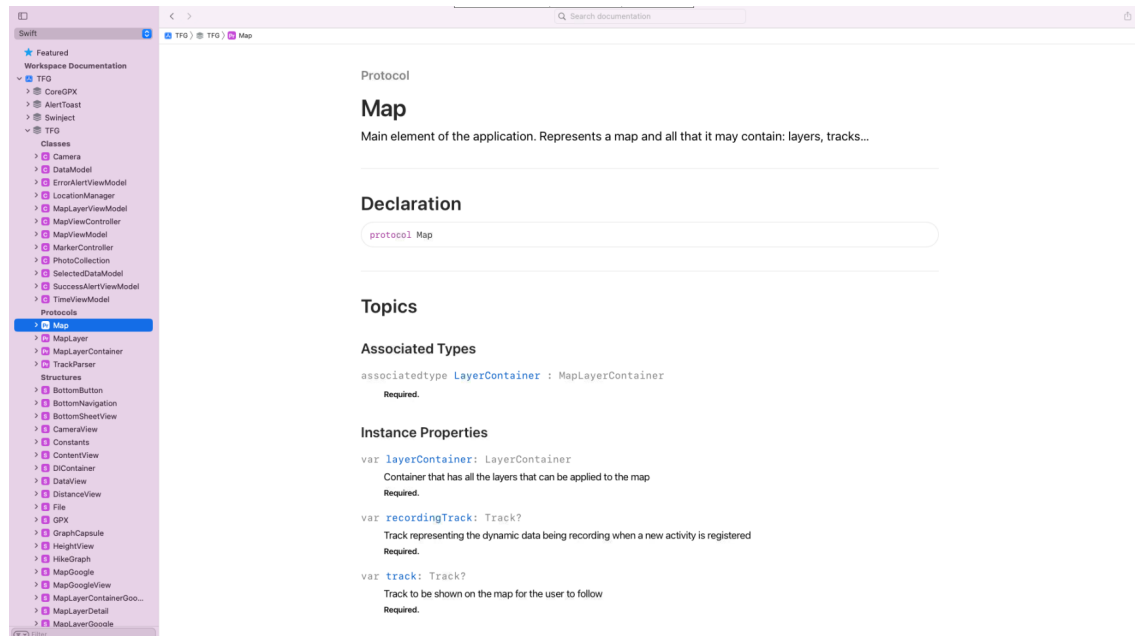


Figura 8.20: Primera parte de la documentación del protocolo Map

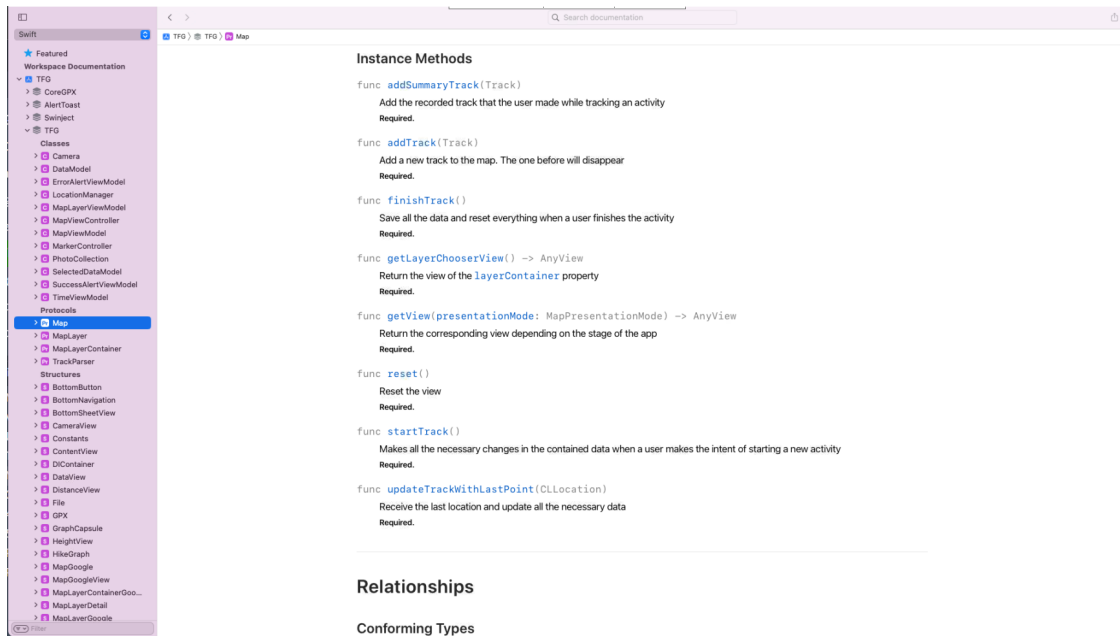


Figura 8.21: Segunda parte de la documentación del protocolo Map

8.3. EJECUCIÓN DE LAS PRUEBAS UNITARIAS

Como se comentó previamente, la intención era hacer todos los tests unitarios con el framework de XCode, llamado *XCTest*. Al estar utilizando SDKs y llamadas al sistema, este framework no nos sirve en muchos casos, ya que necesitaríamos poder acceder al código y añadirle etiquetas de accesibilidad.

Por otro lado, al ejecutar las pruebas de cero, la aplicación no tendría ningún archivo GPX importado (ni podríamos importarlo automáticamente por las razones comentadas previamente). Por suerte, al tener bien diseñado el sistema con el inyector de dependencias, creamos un *MockParser* el cual nos devolverá una ruta virtual de pruebas y podremos usarlo fácilmente en la aplicación detectando si estamos lanzando las pruebas o no.

Gracias a esto, sí se han podido ejecutar las pruebas automáticas para los casos de uso: **Ver rutas importadas** (con ruta *mock*), **Seleccionar ruta importada**, **Registrar actividad**, **Ver actividades registradas**, **Seleccionar actividad registrada** y **Añadir waypoint**.

La mayoría de estos casos de uso están relacionados, incluyéndose y extendiéndose unos a otros, por lo que, con dos tests unitarios automáticos cubrimos todos los casos. En los dos, se hacen los mismos pasos pero con diferente inicio: iniciar actividad (sin ruta a seguir, con ruta importada a seguir o, en el caso del segundo test, terminar y volver a empezar con actividad previamente registrada a seguir), iniciar el cronómetro, guardar un waypoint, terminar la ruta, ponerle nombre y guardarla y revisar que se ha guardado correctamente.

En la siguiente ilustración están los resultados de las pruebas automáticas, el resto se han hecho manualmente y se ha reflejado su resultado en las tablas.

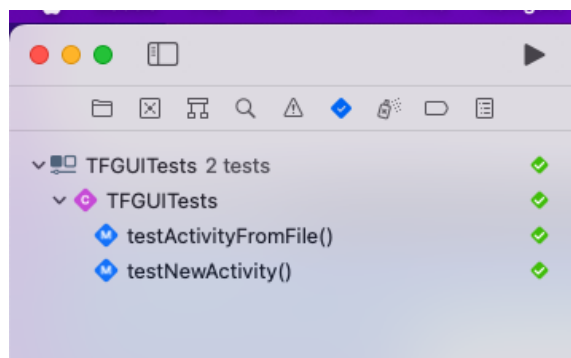


Figura 8.22: Resultado de pruebas unitarias

Tabla 8.1: Ejecución de pruebas: Importar ruta GPX

Importar ruta GPX		
Prueba	Resultado esperado	¿Superado?
Importar archivo con formato correcto	El archivo se guardará en los datos internos de la aplicación y mostrará un mensaje de éxito.	Sí
Importar archivo con formato incorrecto	El archivo no se guardará y se mostrará un mensaje de error.	Sí

Tabla 8.2: Ejecución de pruebas: Ver rutas importadas

Ver rutas importadas		
Prueba	Resultado esperado	¿Superado?
Obtener rutas importadas existiendo cero	Se mostrará un listado vacío.	Sí
Obtener rutas importadas existiendo una o más	Se mostrará un listado con todas las rutas.	Sí

Tabla 8.3: Ejecución de pruebas: Enviar ruta importada

Enviar ruta importada		
Prueba	Resultado esperado	¿Superado?
Enviar ruta importada. WhatsApp.	El contacto recibirá un mensaje con el archivo GPX.	Sí
Enviar ruta importada. Email.	El contacto recibirá un mensaje con el archivo GPX.	Sí

Tabla 8.4: Ejecución de pruebas: Seleccionar ruta importada

Seleccionar ruta importada		
Prueba	Resultado esperado	¿Superado?
Seleccionar ruta importada.	El sistema mostrará el botón de inicio de la ruta además de la representación de esta sobre el mapa.	Sí

Tabla 8.5: Ejecución de pruebas: Registrar actividad

Registrar actividad		
Prueba	Resultado esperado	¿Superado?
Registrar actividad sin ruta seleccionada. Caso completo	El sistema mostrará un resumen de la actividad en la pantalla de introducir el nombre. Además, habrá guardado los datos con el nombre introducido y los mostrará en el listado de actividades registradas.	Sí
Registrar actividad sin ruta seleccionada. Caso cancelado	El sistema volverá a la pantalla de inicio sin haber registrado ninguna actividad.	Sí
Registrar actividad con ruta seleccionada. Caso completo	El sistema mostrará un resumen de la actividad en la pantalla de introducir el nombre. Además, habrá guardado los datos con el nombre introducido y los mostrará en el listado de actividades registradas.	Sí
Registrar actividad con ruta seleccionada. Caso cancelado	El sistema volverá a la pantalla de inicio sin haber registrado ninguna actividad.	Sí
Registrar actividad con actividad registrada. Caso completo	El sistema mostrará un resumen de la actividad en la pantalla de introducir el nombre. Además, habrá guardado los datos con el nombre introducido y los mostrará en el listado de actividades registradas.	Sí
Registrar actividad con ruta seleccionada. Caso cancelado	El sistema volverá a la pantalla de inicio sin haber registrado ninguna actividad.	Sí

Tabla 8.6: Ejecución de pruebas: Seleccionar actividad registrada

Seleccionar actividad registrada		
Prueba	Resultado esperado	¿Superado?
Seleccionar actividad registrada.	El sistema mostrará el botón de inicio de la ruta además de la representación de esta sobre el mapa.	Sí

Tabla 8.7: Diseño de pruebas: Ver actividades registradas

Ver actividades registradas		
Prueba	Resultado esperado	¿Superado?
Obtener actividades registradas existiendo cero	Se mostrará un listado vacío.	Sí
Obtener actividades registradas existiendo una o más	Se mostrará un listado con todas las rutas.	Sí

Tabla 8.8: Diseño de pruebas: Enviar actividad registrada

Enviar actividad registrada		
Prueba	Resultado esperado	Sí
Enviar actividad registrada. WhatsApp	El contacto recibirá un mensaje con el archivo GPX.	Sí
Enviar actividad registrada. Email	El contacto recibirá un mensaje con el archivo GPX.	Sí

Tabla 8.9: Diseño de pruebas: Añadir capa al mapa

Añadir capa al mapa		
Prueba	Resultado esperado	¿Superado?
Añadir capa al mapa	Se aplicará la capa seleccionada al mapa.	Sí

Tabla 8.10: Diseño de pruebas: Añadir waypoint

Añadir waypoint		
Prueba	Resultado esperado	¿Superado?
Añadir waypoint con una foto	Se guardará un waypoint en la localización en el momento que se guardó junto con la foto y el nombre que el usuario haya introducido.	Sí
Añadir waypoint con más de una foto	Se guardará un waypoint en la localización en el momento que se guardó junto con las fotos y el nombre que el usuario haya introducido.	No

8.4. EJECUCIÓN DE LAS PRUEBAS DEL SISTEMA

8.4.1. Prueba de Usabilidad y accesibilidad

8.4.1.0.1 Usuario 1

Tabla 8.11: Pruebas de usabilidad: preguntas de carácter general para el usuario 1

¿Usa el móvil frecuentemente?

1. Todos los días
2. Varias veces a la semana
3. Ocasionalmente
4. Nunca

¿Qué tipo de actividades realiza con el móvil?

1. Es parte de mi trabajo o profesión
2. Lo uso para ocio
3. Solo para comunicarme

¿Ha usado alguna vez software como el de esta prueba?

1. Sí, he usado software similar
 2. No, aunque si empleo otras aplicaciones que se le pueden asemejar
 3. No, nunca
-

Tabla 8.12: Pruebas de usabilidad: preguntas cortas sobre la aplicación y observaciones, usuario 1

Facilidad de uso	Siempre	Frecuentemente	Ocasionalmente	Nunca
<i>¿Sabe donde está dentro de la aplicación?</i>		X		
<i>¿Necesita ayuda para utilizar la aplicación?</i>				X
<i>¿Le resulta sencillo el uso de la aplicación?</i>	X			
Funcionalidad	Siempre	Frecuentemente	Ocasionalmente	Nunca
<i>¿Funciona cada tarea como Vd. Espera?</i>	X			
<i>¿El tiempo de respuesta de la aplicación es muy grande?</i>				X
Calidad del interfaz				
Aspectos gráficos	Muy adecuado	Adecuado	Poco adecuado	Nada adecuado
<i>El tipo y tamaño de letra es</i>	X			
<i>Los iconos e imágenes usados son</i>		X		
<i>Los colores empleados son</i>	X			
<i>Diseño de la interfaz</i>	X			
<i>¿Le resulta fácil de usar?</i>	X			
<i>¿El diseño de las pantallas es claro y atractivo?</i>	X			
<i>¿Cree que el programa está bien estructurado?</i>		X		
Observaciones				
Poner algo más de texto				

Tabla 8.13: Pruebas de usabilidad: prueba del usuario 1 para la actividad Importar un archivo GPX

Importar un archivo GPX	
<i>Tiempo empleado</i>	18 s
<i>Problemas encontrados</i>	Ninguno
<i>Observaciones</i>	Ninguna

Tabla 8.14: Pruebas de usabilidad: prueba del usuario 1 para la actividad Ver los datos de un archivo importado previamente

Ver los datos de un archivo importado previamente	
<i>Tiempo empleado</i>	22 s
<i>Problemas encontrados</i>	Ninguno
<i>Observaciones</i>	Ninguna

Para esta siguiente actividad, se le pide al usuario que haga una ruta de 30 s, algo a tener en cuenta al ver el tiempo necesitado para llevar a cabo la tarea.

Tabla 8.15: Pruebas de usabilidad: prueba del usuario 1 para la actividad Registrar una actividad a partir de una ruta importada previamente

Registrar una actividad a partir de una ruta importada previamente	
<i>Tiempo empleado</i>	2 m 12 s
<i>Problemas encontrados</i>	Ninguno
<i>Observaciones</i>	Ninguna

Tabla 8.16: Pruebas de usabilidad: prueba del usuario 1 para la actividad Compartir una actividad registrada

Compartir una actividad registrada	
<i>Tiempo empleado</i>	32 s
<i>Problemas encontrados</i>	Ninguno
<i>Observaciones</i>	Ninguna

8.4.1.0.2 Usuario 2

Tabla 8.17: Pruebas de usabilidad: preguntas de carácter general para el usuario 2

¿Usa el móvil frecuentemente?

1. Todos los días
2. Varias veces a la semana
3. Ocasionalmente
4. Nunca

¿Qué tipo de actividades realiza con el móvil?

1. Es parte de mi trabajo o profesión
2. Lo uso para ocio
3. Solo para comunicarme

¿Ha usado alguna vez software como el de esta prueba?

1. Sí, he usado software similar
 2. No, aunque si empleo otras aplicaciones que se le pueden asemejar
 3. No, nunca
-

Tabla 8.18: Pruebas de usabilidad: preguntas cortas sobre la aplicación y observaciones, usuario 2

Facilidad de uso	Siempre	Frecuentemente	Ocasionalmente	Nunca
<i>¿Sabe donde está dentro de la aplicación?</i>	X			
<i>¿Necesita ayuda para utilizar la aplicación?</i>				X
<i>¿Le resulta sencillo el uso de la aplicación?</i>		X		
Funcionalidad	Siempre	Frecuentemente	Ocasionalmente	Nunca
<i>¿Funciona cada tarea como Vd. Espera?</i>	X			
<i>¿El tiempo de respuesta de la aplicación es muy grande?</i>				X
Calidad del interfaz				
Aspectos gráficos	Muy adecuado	Adecuado	Poco adecuado	Nada adecuado
<i>El tipo y tamaño de letra es</i>	X			
<i>Los iconos e imágenes usados son</i>	X			
<i>Los colores empleados son</i>	X			
<i>Diseño de la interfaz</i>	X			
<i>¿Le resulta fácil de usar?</i>		X		
<i>¿El diseño de las pantallas es claro y atractivo?</i>	X			
<i>¿Cree que el programa está bien estructurado?</i>	X			
Observaciones				
Hacer una pantalla de carga				

Tabla 8.19: Pruebas de usabilidad: prueba del usuario 2 para la actividad Importar un archivo GPX

Importar un archivo GPX	
<i>Tiempo empleado</i>	33 s
<i>Problemas encontrados</i>	Ninguno
<i>Observaciones</i>	Ninguna

Tabla 8.20: Pruebas de usabilidad: prueba del usuario 2 para la actividad Ver los datos de un archivo importado previamente

Ver los datos de un archivo importado previamente	
<i>Tiempo empleado</i>	20 s
<i>Problemas encontrados</i>	Ninguno
<i>Observaciones</i>	Ninguna

Para esta siguiente actividad, se le pide al usuario que haga una ruta de 30 s, algo a tener en cuenta al ver el tiempo necesitado para llevar a cabo la tarea.

Tabla 8.21: Pruebas de usabilidad: prueba del usuario 2 para la actividad Registrar una actividad a partir de una ruta importada previamente

Registrar una actividad a partir de una ruta importada previamente	
<i>Tiempo empleado</i>	2 m 44 s
<i>Problemas encontrados</i>	Ninguno
<i>Observaciones</i>	Ninguna

Tabla 8.22: Pruebas de usabilidad: prueba del usuario 2 para la actividad Compartir una actividad registrada

Compartir una actividad registrada	
<i>Tiempo empleado</i>	25 s
<i>Problemas encontrados</i>	Ninguno
<i>Observaciones</i>	Ninguna

8.4.1.0.3 Usuario 3

Tabla 8.23: Pruebas de usabilidad: preguntas de carácter general para el usuario 3

¿Usa el móvil frecuentemente?

1. [Todos los días](#)
2. Varias veces a la semana
3. Ocasionalmente
4. Nunca

¿Qué tipo de actividades realiza con el móvil?

1. Es parte de mi trabajo o profesión
2. [Lo uso para ocio](#)
3. Solo para comunicarme

¿Ha usado alguna vez software como el de esta prueba?

1. [Sí, he usado software similar](#)
 2. No, aunque si empleo otras aplicaciones que se le pueden asemejar
 3. No, nunca
-

Tabla 8.24: Pruebas de usabilidad: preguntas cortas sobre la aplicación y observaciones, usuario 3

Facilidad de uso	Siempre	Frecuentemente	Ocasionalmente	Nunca
<i>¿Sabe donde está dentro de la aplicación?</i>	X			
<i>¿Necesita ayuda para utilizar la aplicación?</i>				X
<i>¿Le resulta sencillo el uso de la aplicación?</i>	X			
Funcionalidad	Siempre	Frecuentemente	Ocasionalmente	Nunca
<i>¿Funciona cada tarea como Vd. Espera?</i>	X			
<i>¿El tiempo de respuesta de la aplicación es muy grande?</i>				X
Calidad del interfaz				
Aspectos gráficos	Muy adecuado	Adecuado	Poco adecuado	Nada adecuado
<i>El tipo y tamaño de letra es</i>	X			
<i>Los iconos e imágenes usados son</i>	X			
<i>Los colores empleados son</i>	X			
<i>Diseño de la interfaz</i>		X		
<i>¿Le resulta fácil de usar?</i>	X			
<i>¿El diseño de las pantallas es claro y atractivo?</i>		X		
<i>¿Cree que el programa está bien estructurado?</i>	X			
Observaciones				

Tabla 8.25: Pruebas de usabilidad: prueba del usuario 3 para la actividad Importar un archivo GPX

Importar un archivo GPX	
<i>Tiempo empleado</i>	12 s
<i>Problemas encontrados</i>	Ninguno
<i>Observaciones</i>	Ninguna

Tabla 8.26: Pruebas de usabilidad: prueba del usuario 3 para la actividad Ver los datos de un archivo importado previamente

Ver los datos de un archivo importado previamente	
<i>Tiempo empleado</i>	16 s
<i>Problemas encontrados</i>	Ninguno
<i>Observaciones</i>	Ninguna

Para esta siguiente actividad, se le pide al usuario que haga una ruta de 30 s, algo a tener en cuenta al ver el tiempo necesitado para llevar a cabo la tarea.

Tabla 8.27: Pruebas de usabilidad: prueba del usuario 3 para la actividad Registrar una actividad a partir de una ruta importada previamente

Registrar una actividad a partir de una ruta importada previamente	
<i>Tiempo empleado</i>	1 m 38 s
<i>Problemas encontrados</i>	Un poco lenta la cámara al sacar foto para waypoint
<i>Observaciones</i>	Ninguna

Tabla 8.28: Pruebas de usabilidad: prueba del usuario 3 para la actividad Compartir una actividad registrada

Compartir una actividad registrada	
<i>Tiempo empleado</i>	20 s
<i>Problemas encontrados</i>	Ninguno
<i>Observaciones</i>	Ninguna

8.4.1.0.4 Usuario 4

Tabla 8.29: Pruebas de usabilidad: preguntas de carácter general para el usuario 4

¿Usa el móvil frecuentemente?

1. Todos los días
2. Varias veces a la semana
3. Ocasionalmente
4. Nunca

¿Qué tipo de actividades realiza con el móvil?

1. Es parte de mi trabajo o profesión
2. Lo uso para ocio
3. Solo para comunicarme

¿Ha usado alguna vez software como el de esta prueba?

1. Sí, he usado software similar
 2. No, aunque si empleo otras aplicaciones que se le pueden asemejar
 3. No, nunca
-

Tabla 8.30: Pruebas de usabilidad: preguntas cortas sobre la aplicación y observaciones, usuario 4

Facilidad de uso	Siempre	Frecuentemente	Ocasionalmente	Nunca
<i>¿Sabe donde está dentro de la aplicación?</i>			X	
<i>¿Necesita ayuda para utilizar la aplicación?</i>			X	
<i>¿Le resulta sencillo el uso de la aplicación?</i>	X			
Funcionalidad	Siempre	Frecuentemente	Ocasionalmente	Nunca
<i>¿Funciona cada tarea como Vd. Espera?</i>	X			
<i>¿El tiempo de respuesta de la aplicación es muy grande?</i>				X
Calidad del interfaz				
Aspectos gráficos	Muy adecuado	Adecuado	Poco adecuado	Nada adecuado
<i>El tipo y tamaño de letra es</i>		X		
<i>Los iconos e imágenes usados son</i>		X		
<i>Los colores empleados son</i>	X			
<i>Diseño de la interfaz</i>	X			
<i>¿Le resulta fácil de usar?</i>		X		
<i>¿El diseño de las pantallas es claro y atractivo?</i>	X			
<i>¿Cree que el programa está bien estructurado?</i>		X		
Observaciones				

Tabla 8.31: Pruebas de usabilidad: prueba del usuario 4 para la actividad Importar un archivo GPX

Importar un archivo GPX	
<i>Tiempo empleado</i>	37 s
<i>Problemas encontrados</i>	Ninguno
<i>Observaciones</i>	Ninguna

Tabla 8.32: Pruebas de usabilidad: prueba del usuario 4 para la actividad Ver los datos de un archivo importado previamente

Ver los datos de un archivo importado previamente	
<i>Tiempo empleado</i>	31 s
<i>Problemas encontrados</i>	Ninguno
<i>Observaciones</i>	Ninguna

Para esta siguiente actividad, se le pide al usuario que haga una ruta de 30 s, algo a tener en cuenta al ver el tiempo necesitado para llevar a cabo la tarea.

Tabla 8.33: Pruebas de usabilidad: prueba del usuario 4 para la actividad Registrar una actividad a partir de una ruta importada previamente

Registrar una actividad a partir de una ruta importada previamente	
<i>Tiempo empleado</i>	3 m 40 s
<i>Problemas encontrados</i>	No se ve muy bien el campo de texto al poner el nombre de la ruta
<i>Observaciones</i>	Ninguna

Tabla 8.34: Pruebas de usabilidad: prueba del usuario 4 para la actividad Compartir una actividad registrada

Compartir una actividad registrada	
<i>Tiempo empleado</i>	45 s
<i>Problemas encontrados</i>	Ninguno
<i>Observaciones</i>	Ninguna

8.5. ELABORACIÓN DE LOS MANUALES DE USUARIO

8.5.1. Manual de Instalación

El transcurso de llevar esta aplicación a producción terminaría con ella en la AppStore, donde el usuario simplemente tendría que descargarla como cualquier otra aplicación. Para esto habría que pagar 99€ para entrar en el programa de Apple Developer, algo que queda fuera del alcance de este proyecto.

Existe la opción de instalarla a través de XCode por lo que sería necesario un Mac además del dispositivo iOS. En este caso la aplicación solo tendría una licencia de 7 días. Los pasos a realizar serían:

1. Instalar XCode en caso de no tenerlo.
2. En los archivos entregados, se encuentra el código fuente en el directorio *TFG*. Dentro de esta carpeta existe un archivo llamado *TFG.xcworkspace*. Hacer doble clic sobre este archivo.
3. Una vez haya cargado todo, conectar con un USB el teléfono móvil al ordenador.
4. En el desplegable de la siguiente imagen aparecerá el dispositivo. Seleccionarlo.

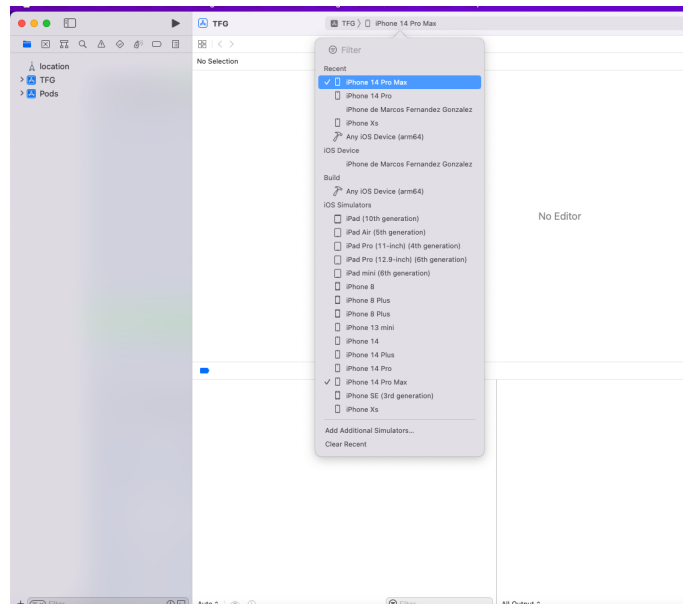


Figura 8.23: Desplegable para seleccionar el móvil donde instalar la aplicación

5. Darle al botón de *play* que se encuentra a la izquierda del desplegable.
6. El móvil no confiará en el certificado al no estar en el *Developer Program* de Apple. Para poder ejecutar la aplicación seguir los siguientes pasos en el móvil:
 - a) Ajustes
 - b) General
 - c) VPN y Dispositivos
 - d) Aparecerá mi correo electrónico personal (**marcosfernandezflemig@gmail.com**, pulsar sobre él.
 - e) Pulsar en confiar.

8.5.2. Manual de Ejecución

Una vez descargada la aplicación del AppStore o siguiendo los pasos a través de XCode, el usuario tan solo tendría que pulsar sobre ella desde el menú de inicio como haría con cualquier otra aplicación.

8.5.3. Manual de Usuario

La primero que tendrá que realizar el usuario al abrir la aplicación será aceptar el uso de la localización del teléfono pulsando en *Permitir al usar la app*.

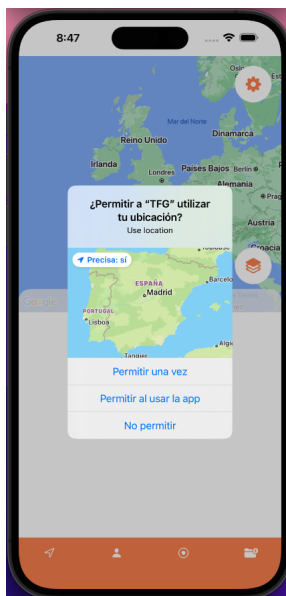


Figura 8.24: Aviso de uso localización al usuario

Esta será la primera pantalla que el usuario observe en la aplicación. Se irá detallando sección por sección, pero este sería un resumen general del sistema:

- Botón de arriba a la derecha (rueda dentada): Nos llevará a otra pantalla donde podremos seleccionar qué campos queremos ver al estar registrando una actividad
- Botón medio derecha (capas): Es un desplegable con las capas que podremos aplicar al mapa. Seleccionar una y se aplicará.
- Barra de navegación en la parte de abajo, de izquierda a derecha:
 1. Rutas importadas: Es la pestaña por defecto al abrir la aplicación. Nos mostrará un listado de las rutas importadas. Al abrir la aplicación por primera vez estará vacío.
 2. Actividades registradas: En esta sección el usuario verá todas las actividades que haya registrado.
 3. Registrar actividad de cero: Este botón nos llevará a la vista de registrar actividad sin ninguna ruta a seguir en el mapa.
 4. Importar archivo: Abrirá el sistema de archivos del móvil. En caso de tener descargado previamente un archivo GPX, se podrá seleccionar y se importará.

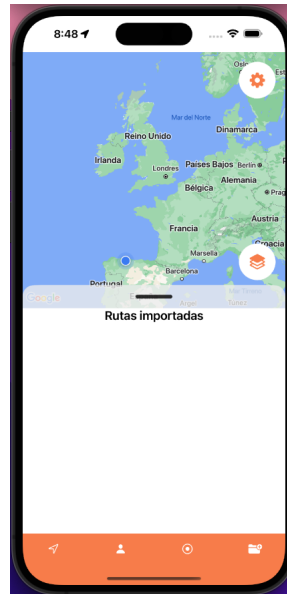


Figura 8.25: Aviso de uso localización al usuario

8.5.3.1. Añadir una capa al mapa

Pulsando sobre el botón de capas se desplegarán las opciones. Pulsando sobre cualquiera de ellas se aplicará su respectiva capa.

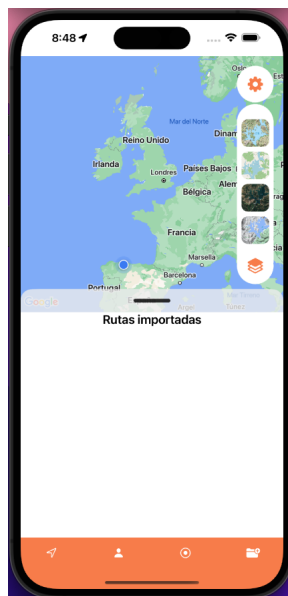


Figura 8.26: Capas aplicables al mapa desplegadas



Figura 8.27: Capa seleccionada aplicada sobre el mapa

8.5.3.2. Importar ruta

Si pulsamos sobre el botón de importar ruta veremos la siguiente pantalla. En caso de que alguno de los archivos sea formato GPX, el sistema nos permitirá pulsar encima de él. Al terminar nos dirá el resultado de la operación a través de un mensaje.

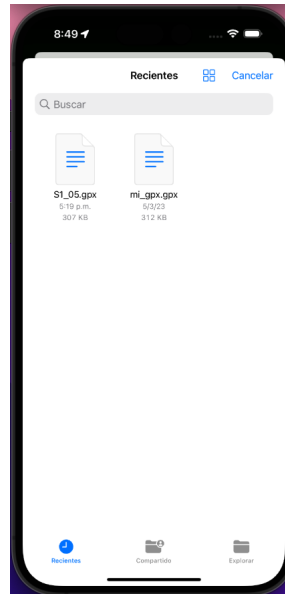


Figura 8.28: Sistema de archivos del sistema para seleccionar archivo GPX

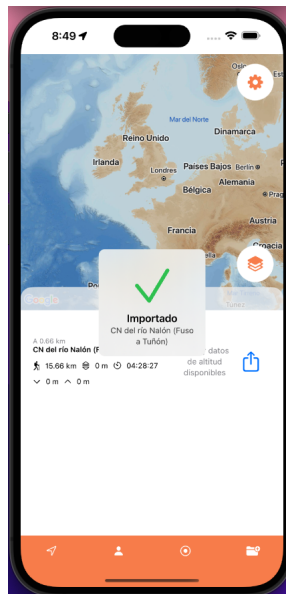


Figura 8.29: Mensaje de éxito de importación

8.5.3.3. Compartir archivo

Esto aplica tanto a las rutas previamente importadas como a las actividades registradas por el usuario. Pulsar sobre el botón azul de la caja y la flecha de la ruta que se desea enviar. Aparecerá la siguiente pantalla donde tendremos que escoger el medio por el cual queremos enviar el archivo.



Figura 8.30: Selección de medio multimedia para enviar un archivo GPX desde la aplicación

8.5.3.4. Seleccionar ruta y seguirla

Esto aplica tanto a las rutas previamente importadas como las actividades registradas por el usuario. Pulsar sobre la ruta que se desea seguir. Esta desplegará un botón de inicio y se mostrará en el mapa como en la siguiente imagen.

En caso de querer cancelar la actividad, simplemente pulsar en la cruz de arriba a la izquierda.

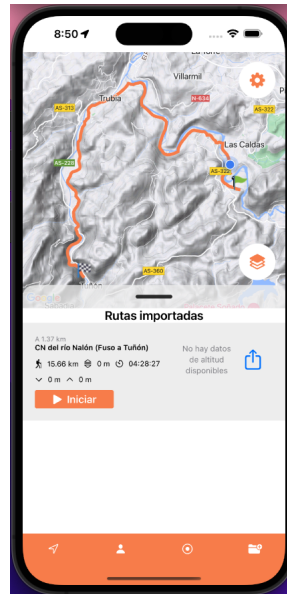


Figura 8.31: Selección de una ruta

Al pulsar sobre el botón de iniciar, este nos llevará a la pantalla de registro de actividad, donde veremos los datos sobre la actividad que estamos realizando. Pulsando el botón de *play* el cronómetro comenzará a contar el tiempo.

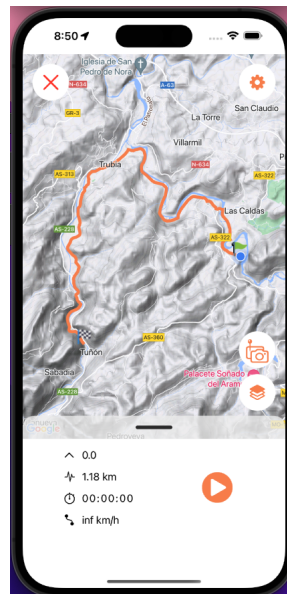


Figura 8.32: Ruta iniciada

Quando deseemos finalizar la ruta, pulsaremos sobre el botón de pausa. En este momento aparecerán dos botones, uno de *play* para seguir registrando la actividad y otro de *stop* para finalizarla. Al pulsar el de *stop* nos llevará a una pantalla donde podremos ver el resumen de nuestra actividad, introducir el nombre y guardarla.

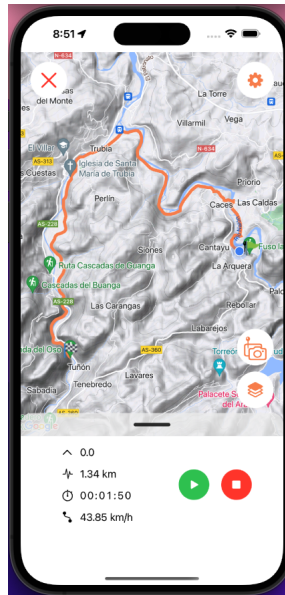


Figura 8.33: Registro de actividad pausado

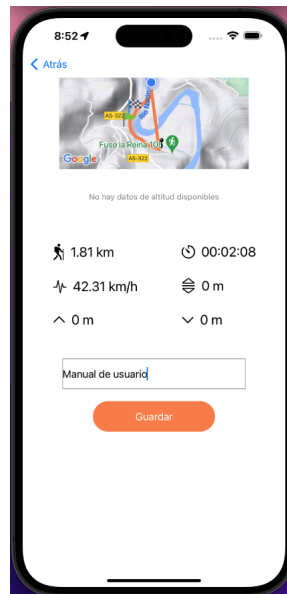


Figura 8.34: Resumen de registro de actividad con nombre introducido

Si guardamos y todo ha ido correctamente, obtendremos un mensaje de éxito y si navegamos a actividades registradas veremos nuestra ruta.

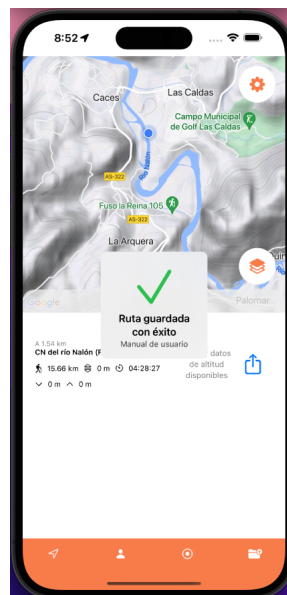


Figura 8.35: Mensaje de éxito a guarda la actividad

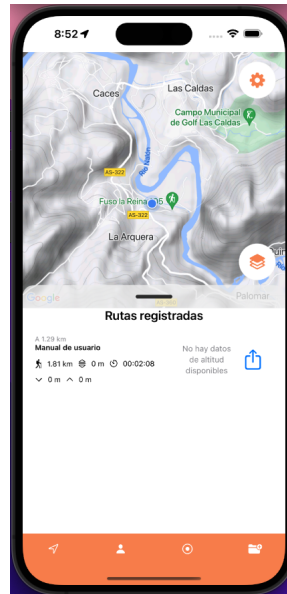


Figura 8.36: Ruta guardada en el registro de actividades

8.5.3.4.1 Waypoints

Cuando estamos registrando una actividad, tenemos la posibilidad de ir añadiendo waypoints a lo largo del camino pulsando sobre el botón del marcador y la cámara. Este nos abrirá la cámara del móvil y tendremos que aceptar el uso de esta.

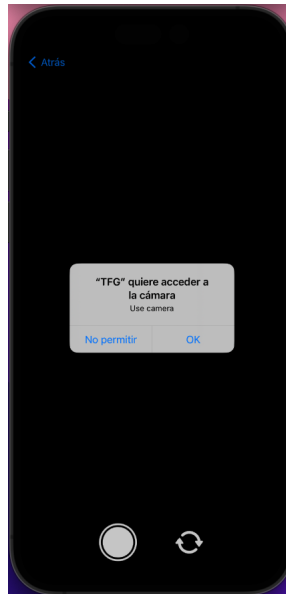


Figura 8.37: Permiso de cámara al guardar waypoint

Una vez hemos sacado una o más fotos, introducimos el nombre que queremos ponerle al waypoint y guardamos.

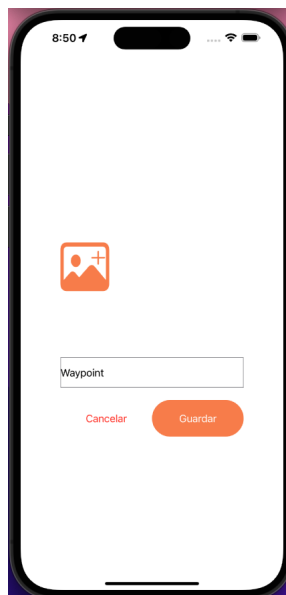


Figura 8.38: Introducir nombre waypoint y guardar

Si pulsamos sobre alguno de los waypoints, veremos información sobre este.



Figura 8.39: Información sobre un waypoint

8.5.3.5.

Selección de visualización de datos Si pulsamos el botón de la rueda dentada (podemos pulsarlo tanto registrando una actividad o en cualquier otro momento) veremos una lista de los campos que se muestran al registrar una actividad junto con un *check*. Si no queremos ver alguno de esos campos pulsar sobre él y el *check* desaparecerá.

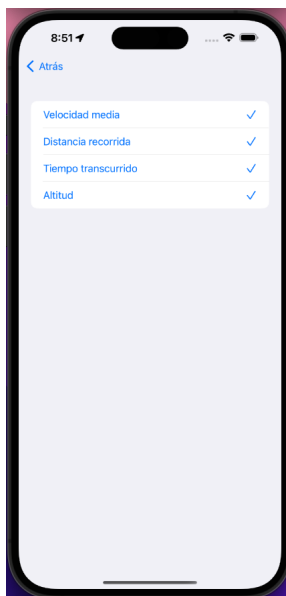


Figura 8.40: Selección de visualización de datos

8.5.3.6. Registrar una actividad sin ruta previa

Pulsar el botón de registrar actividad en el menú de navegación de la parte baja de la pantalla. A partir de este punto seguir los mismos pasos que en 8.5.3.4.

8.5.4. Manual del Programador

En esta sección se explicará los pasos que deberá seguir un desarrollador para añadir un nuevo servicio de visualización de mapas y un parser de archivos que rutas (por ejemplo, KML).

8.5.4.1. Añadir un nuevo servicio de visualización de mapas

Como vimos previamente en el diseño de clases del mapa (7.3.1) el sistema está preparado para facilitar a los desarrolladores añadir un nuevo servicio de visualización de mapas en la aplicación.

Un desarrollador deberá crear tres estructuras, cada una implementando los siguientes protocolos: *Map*, *MapLayerContainer* y *MapLayer*. Los nombres recomendados para cada una de estas estructuras son, repectivamente: *MapServicioDeMapas*, *MapLayerContainerServicioDeMapas* y *MapLayerServicioDeMapas*. Por ejemplo, si el programador deseara integrar *MapBox*, los nombres serían *MapMapBox*, *MapLayerContainerMapBox* y *MapLayerMapBox*.

En los métodos al implementar el protocolo *Map*, hay que destacar dos. Por un lado, *getView* en el cual tendremos que devolver la vista *SwiftUI* del servicio que estamos integrando. Esto será diferente para cada uno, por lo que se deberá leer la documentación de este mismo. El otro es *getLayerChooserView* el cual tendrá la misma implementación para todas las estructuras que implementen este protocolo. Se puede simplemente copiar y pegar el código de *MapGoogle* en el que

se hace una llamada a la función *getView* del *MapLayerContainer* asociado donde le pasamos por parámetro *self*. Esto, a priori, parece una mala práctica, ya que tendremos código repetido. La solución a esto sería hacer una extensión del protocolo *Map* donde se encuentre este código y de esta manera todas las estructuras que implementen el protocolo lo compartirían. Lo que ocurre, es que esta solución no compilaría ya que estamos usando la característica de *Swift* de tipos asociados, y en la función *getView* del protocolo *MapLayerContainer* estamos usando este tipo asociado y en tiempo de compilación el compilador debe saber con qué tipo de mapa concreto estamos trabajando.

En el método mencionando del protocolo *MapLayerContainer*, *getView* se deberá devolver una instancia de la vista *MapLayers* la cual servirá para cualquier tipo de contenedor de capas. Simplemente debemos pasarle el contenido que debe mostrar, que será una serie de *thumbnails* las cuales se crearán recorriendo la lista de capas que tiene el contenedor y llamando a su método *getView*. La implementación de este método en la estructura que implemente el protocolo *MapLayer* es muy sencilla, simplemente hay que devolver la vista *MapLayerDetail* pasándole los argumentos obligatorios, entre los que destaca el llamado *action*, el cual es una función, donde se encontrará la implementación de los pasos a seguir para aplicar la capa al mapa.

El sistema ahora mismo, tan solo trabaja con *GoogleMaps* y no soportaría directamente otro servicio de mapas. Habría que hacer algo más de trabajo. Por un lado, existe un inyector de dependencias, llamado *DIContainer* que ahora mismo tan solo inyecta dependencias relacionadas con *GoogleMaps*. Lo que habría que hacer es, crear una nueva interfaz de usuario donde el usuario pudiera escoger el servicio que desea utilizar, y en función de su selección cambiar la implementación con la que el inyector resuelve cada protocolo.

8.5.4.2. Añadir un nuevo parser de rutas

Para crear un nuevo parser de rutas, simplemente hay que implementar el protocolo *TrackParser*.

Una vez tenemos el protocolo implementado, nos encontramos en la misma situación que con los mapas previamente. El sistema no está preparado completamente. Habría que crear una interfaz donde el usuario pueda seleccionar con qué tipo de archivos quiere trabajar y cambiar en el inyector de dependencias la implementación con la que el sistema debe trabajar.

Capítulo 9

CONCLUSIONES Y AMPLIACIONES

9.1. CONCLUSIONES

A pesar de que el proyecto se haya retrasado bastante más de lo previsto, los objetivos se han conseguido cumplir, la aplicación realiza las funciones que se han propuesto a lo largo de la vida de este. Un usuario puede importar una ruta que haya descargado previamente en su móvil, seguirla en un mapa donde puede visualizarla de diferentes maneras, añadir waypoints junto con una foto, guardarla en el teléfono y compartirla a través de diferentes medios.

Me parece muy destacable el haber realizado este trabajo con el lenguaje de programación *Swift* y el framework *SwiftUI* ya que no tenía ningún conocimiento sobre ninguno de los dos. Este lenguaje tiene además características bastante complejas y diferentes, por lo que el tiempo de aprendizaje e investigación ha sido grande y costoso. En cuento al framework, tuve gran suerte con poder contar con un curso completamente gratuito de la Universidad de Stanford[30] que pude seguir a través de YouTube. A pesar del gran esfuerzo, creo que aprender estas dos tecnologías por mi cuenta me ha hecho crecer enormemente como ingeniero software y que me será muy útil en el futuro, no solo para tener la capacidad de crear aplicaciones móviles o en general para dispositivos de Apple, si no a la hora de comprender problemas, proponer soluciones y pensar por mí mismo. He de decir además, que bajo mi opinión personal, *Swift* es un lenguaje muy interesante, que reta y obliga al programador a hacer las cosas bien, sin atajos, creando un código mantenible y organizado.

La documentación por otro lado ha servido para aplicar muchos de los conocimientos aprendidos a lo largo del grado y que sale a relucir ahora su utilidad y aplicabilidad.

Para finalizar, creo que he de estar orgulloso y satisfecho del trabajo realizado. He tenido dificultades que podría haberme ahorrado haciendo algo diferente, pero creo que estas me han hecho crecer. Lógicamente, existe una inmensa cantidad de mejoras, algunas de las cuales repasaré en la siguiente sección. Algunas de estas tienen una alta posibilidad de ser implementadas en el futuro, ya que he disfrutado este proceso y realizar un proyecto que de verdad tiene el nombre y apellidos de uno mismo es algo único y altamente satisfactorio.

9.2. AMPLIACIONES

Estas son algunas de las ampliaciones que cuadrarían en la aplicación y que la mejorarían.

9.2.1. Usuarios

En este momento la aplicación es de usuario único y de uso completamente local. Algo a mejorar es añadir un registro de usuarios que se identifiquen, y que además, sus actividades estén identificadas.

9.2.2. Base de datos

La ampliación anterior necesitaría esta previamente para poder guardar los registros. Añadir una base de datos donde se guarden todos los datos en lugar de en el almacenamiento interno del dispositivo. Esto además mejoraría algo que podría suponer un problema, una ocupación muy alta de espacio por parte de la aplicación.

9.2.3. Trabajar con otro tipo de archivos (KML, GeoJSON...)

Ya se ha comentado a lo largo del documento que el sistema está preparado para soportar nuevos parsers de archivos que contengan rutas. KML y GeoJSON, por ejemplo, son dos tipos de archivos muy utilizados y que sería interesante añadir su integración dentro del sistema.

9.2.4. Otros sistemas de visualización de mapas

Esto también se ha visto previamente. El sistema ya facilita la integración con otros sistemas de visualización de mapas, ya que enriquecería la aplicación y mejoraría la experiencia de usuario. Además, no parece que esto sea algo que ninguna aplicación ofrece, por lo que le añade un gran atractivo.

ANEXOS

PLAN DE GESTIÓN DE RIESGOS

Siguiendo el PMBOK[31] y la metodología Boehm, la gestión de riesgos consiste en los siguientes pasos:

- Planificar la Gestión de riesgos
- Valoración de riesgos
- Gestión de riesgos

El primer paso dentro de todo el conjunto es identificar estos riesgos, algo que se ha realizado en el punto Identificación de Riesgos.

9.2.5. Registro y valoración de los riesgos

Lo siguiente que haremos será el registro de riesgos, donde para cada riesgo identificado lo clasificaremos. Para ello usaremos diferentes parámetros que la metodología Boehm propone.

Cada uno de los riesgos tendrá una categoría asignada, la cual se obtiene de la estructura de desglose de riesgos.

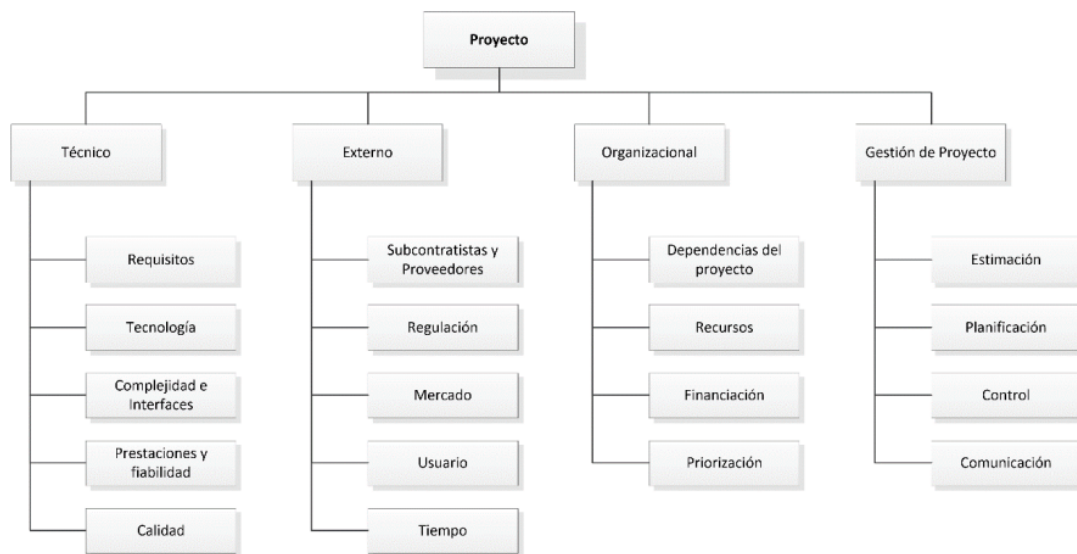


Figura 9.1: Desglose de las categorías de riesgos de la metodología Boehm

A los riesgos se les asignarán dos datos. Por un lado la probabilidad de que ocurran y por otro el impacto sobre los objetivos principales del proyecto. Para ello utilizamos los siguientes datos.

Etiqueta	Rango	Valor usado para los cálculos (y en la Matriz de Probabilidad/Impacto)
Muy bajo	[0%...20%]	10%
Bajo	(20%...40%]	30%
Medio	(40%...60%]	50%
Alto	(60%...80%]	70%
Muy alto	(80%...100%]	90%

Figura 9.2: Probabilidad de que ocurra un riesgo

Condiciones definidas para las escalas de impacto de un riesgo sobre los objetivos principales del proyecto (Se muestran ejemplos para impactos negativos únicamente)					
Objetivos de proyecto	Escalas relativas o numéricas				
	Muy bajo / 5%	Bajo / 10%	Moderado / 20%	Alto / 40%	Muy alto / 80%
Coste	Incremento del coste insignificante	Incremento del coste <10%	Incremento del coste entre el 10-20%	Incremento del coste entre el 20-40%	Incremento del coste >40%
Tiempo	Incremento de tiempo insignificante	Incremento de tiempo <5%	Incremento de tiempo entre el 5-10%	Incremento de tiempo entre el 10-20%	Incremento de tiempo >20%
Alcance	Reducciones del alcance inapreciables	Afectadas áreas poco importantes del alcance	Afectadas áreas importantes del alcance	Reducciones del alcance inaceptables para el cliente	El resultado final del proyecto no es realmente útil
Calidad	La degradación de la calidad es inapreciable	Sólo las aplicaciones muy exigentes se ven afectadas	La reducción de la calidad requiere la aceptación del cliente	Reducción de la calidad inaceptable para el cliente	El resultado final del proyecto no es realmente útil

Esta tabla presenta ejemplos de definiciones de impacto para cuatro objetivos de proyecto diferentes. Debe ser ajustada en el proceso de elaboración del Plan de Riesgos a cada proyecto concreto y a los umbrales de riesgo de la organización. Las definiciones del impacto deben ser desarrolladas para los riesgos positivos (oportunidades) de una manera similar.

Figura 9.3: Escala de impacto relativo de cada riesgo

Con estos dos datos, podremos asignar un valor numérico a cada riesgo de manera que podamos priorizarlos. Para ello usamos la matriz de probabilidad e impacto.

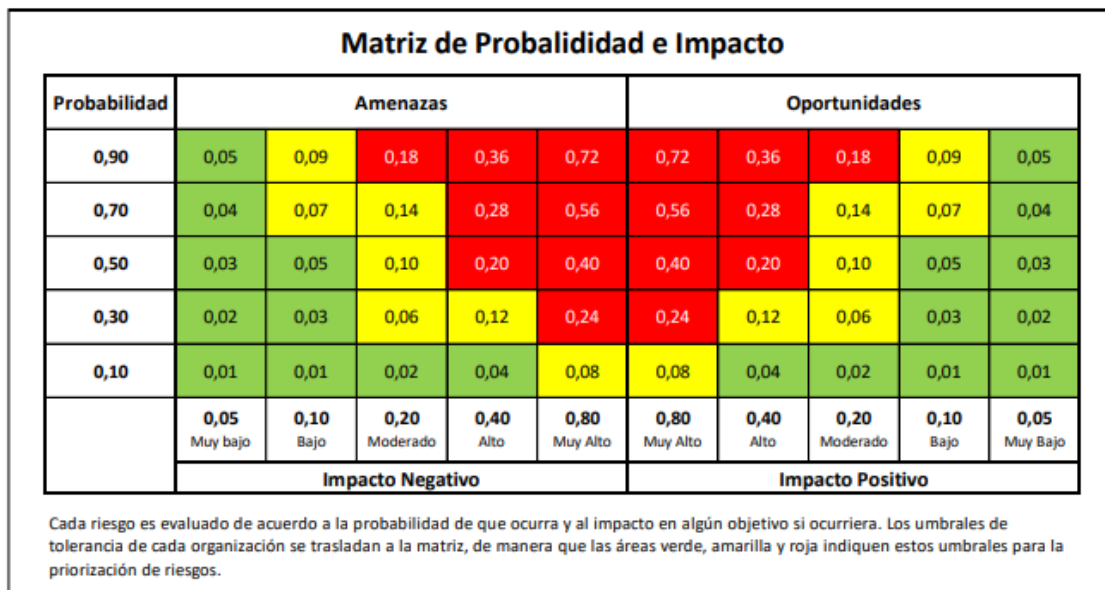


Figura 9.4: Matriz de probabilidad e impacto

Finalmente, estos son los riesgos clasificados y priorizados registrados aplicando todo lo que acabamos de ver.

ID	Nombre del Riesgo	Breve Descripción	Categoría	Probabilidad	Impacto				Impacto
					Presup.	Planific.	Alcance	Calidad	
1	Fallo en la estimación de tiempo	Se planifica el proyecto con un tiempo que finalmente no se cumple	Gestión del proyecto	Muy Alta	Alto	Crítico	Bajo	Medio	0,81
2	Falta de formación del desarrollador	El programador software no tiene suficientes conocimientos sobre las tecnologías que se van a usar	Técnico	Muy Alta	Alto	Alto	Bajo	Medio	0,50
3	Falta de usabilidad	Los usuarios pueden no guiarse y situarse a través de la aplicación como se desearía	Externo	Media	Inapreciable	Inapreciable	Bajo	Alto	0,28
4	Pérdida de código	Se pierde el código de la aplicación	Técnico	Muy Baja	Bajo	Crítico	Alto	Alto	0,09
5	Cambios en el framework	El framework se actualiza y varias de las características utilizadas dejan de funcionar	Externo	Muy Baja	Inapreciable	Bajo	Inapreciable	Medio	0,03

Figura 9.5: Registro y valoración de riesgos

9.2.6. Gestión de los riesgos

Finalmente, debemos plantear cómo gestionaremos cada riesgo en caso de que pudiera ocurrir. Estas son las estrategias a seguir:

- Eliminación del riesgo
- Transferencia del riesgo
- Mitigación del riesgo
- Asumir el riesgo

Estas son las soluciones propuestas para cada riesgo:

1. **Fallo en la estimación de tiempo - Asumir el riesgo:** Tratar de manejar el nuevo tiempo disponible aprendiendo de los errores que han llevado a esta situación.
2. **Falta de formación del desarrollador - Asumir el riesgo:** Se intentará manejar la situación de manera que la planificación se vea muy afectada.
3. **Falta de usabilidad - Mitigar el riesgo:** Se tratarán de seguir las recomendaciones y estándares de usabilidad en dispositivos móviles.
4. **Pérdida de código - Asumir el riesgo:** Se investigará la manera de recuperar los datos.
5. **Cambios en el framework - Mitigar el riesgo:** Se invertirá el tiempo necesario para sustituir los elementos deprecados.

CONTENIDO ENTREGADO EN LOS ANEXOS

Esta es la estructura de directorios y los contenidos que tiene cada uno de todo lo entregado:

1. Directorio raíz: contiene esta misma documentación, la carpeta anexos y la carpeta TFG_UO270803_MarcosFernandezGonzalez
 - a) Anexos: contiene los presupuestos, planificaciones y registro de riesgos
 - b) TFG_UO270803_MarcosFernandezGonzalez: contiene el código fuente del proyecto. Dentro hay otra carpeta llamada TFG_MarcosFernandezGonzalez y en esta otra más con el nombre de TFG, hay que entrar para llegar hasta el código. Una vez nos encontramos en el directorio TFG_UO270803_MarcosFernandezGonzalez/TFG_MarcosFernandezGonzalez/TFG, están los siguientes elementos:
 - 1) TFGUITests: los tests unitarios automatizados
 - 2) TFG.xworkspace: el archivo a ejecutar en caso de querer ver todo el código en XCode y ejecutar
 - 3) Pods: librerías
 - 4) TFG: donde realmente está el código fuente. Una vez dentro, se divide en los siguientes módulos:
 - a' Core: Contiene los elementos más independientes del proyecto y que aparecerían en casi cualquiera: constantes, archivos de keys... y el inyector de dependencias.
 - b' Domain: todo lo relacionada con el modelo y lógica de negocio.
 - c' Presentation: todas las vistas y lo relacionado directamente con ellas.

Bibliografía

- [1] Jose Manuel Redondo, “Documentos-modelo para Trabajos de Fin de Grado/Master de la Escuela de Informática de Oviedo.” https://www.researchgate.net/publication/327882831_Plantilla_de_Proyectos_de_Fin_de_Carrera_de_la_Escuela_de_Informatica_de_Oviedo, 2019. Online; accessed 20 Jan 2023.
- [2] Jose Manuel Redondo, “Creación y evaluación de plantillas para trabajos de fin de grado como buena práctica docente,” *Revista de Innovación y Buenas Prácticas Docentes*, vol. pp, no. pp, p. pp, 2020.
- [3] Topografix, “GPX 1.1 Schema Documentation.” <https://www.topografix.com/GPX/1/1/>, 2023. Online; accessed 10 Jan 2023.
- [4] Google, “Google Maps Platform.” <https://developers.google.com/maps?hl=es-419>, 2023. Online; accessed 12 Jan 2023.
- [5] Open Street Map, “Cartografía con Open Street Map.” <https://www.openstreetmap.org/#map=6/40.007/-2.488>, 2023. Online; accessed 11 Jan 2023.
- [6] Instituto Geográfico Nacional, “API Rest del Instituto Geográfico Nacional.” <https://github.com/IGN-CNIG/API-CNIG/wiki>, 2023. Online; accessed 10 Jan 2023.
- [7] Apple Developer, “SwiftUI Overview.” <https://developer.apple.com/xcode/swiftui/>, 2023. Online; accessed 08 Dec 2022.
- [8] Meta, “React Native. Learn once, write anywhere.” <https://reactnative.dev>, 2023. Online; accessed 18 Feb 2023.
- [9] Flutter Dev, “Flutter. Build apps for any screen.” <https://flutter.dev>, 2023. Online; accessed 18 Feb 2023.
- [10] Apple, “Swift.” <https://www.apple.com/es/swift/>, 2022. Online; accessed 08 Dec 2022.
- [11] Apple, “Apple.” <https://www.apple.com>, 2022. Online; accessed 15 Dec 2022.
- [12] Google, “Google Maps SDK for iOS.” <https://developers.google.com/maps/documentation/ios-sdk?hl=es-419>, 2022. Online; accessed 08 Dec 2022.
- [13] MapBox, “MapBox SDK for iOS.” <https://docs.mapbox.com/ios/navigation/guides/>, 2023. Online; accessed 15 Jan 2023.

-
- [14] Wikiloc, “Rutas del mundo.” <https://es.wikiloc.com>, 2023. Online; accessed 28 Feb 2023.
- [15] Strava, “Registra + Suda + Comparte = Kudos.” <https://www.strava.com>, 2023. Online; accessed 28 Feb 2023.
- [16] AllTrails, “Encuentra tu aventura.” <https://www.alltrails.com/es/>, 2023. Online; accessed 28 Feb 2023.
- [17] Apple, “XCTest.” <https://developer.apple.com/documentation/xctest>, 2023. Online; accessed 13 Mar 2023.
- [18] Hacking With Swift, “Introducing MVVM into your SwiftUI project.” <https://www.hackingwithswift.com/books/ios-swiftui/introducing-mvvm-into-your-swiftui-project>, 2023. Online; accessed 12 Feb 2023.
- [19] On My Way 133, “How to use any vs some in Swift.” <https://onmyway133.com/posts/how-to-use-any-vs-some-in-swift/>, 2023. Online; accessed 25 Mar 2023.
- [20] Donny Walls, “What’s the difference between any and some in Swift 5.7?.” <https://www.donnywals.com/whats-the-difference-between-any-and-some-in-swift-5-7/>, 2023. Online; accessed 25 Mar 2023.
- [21] NalexN, “Clean Architecture for SwiftUI.” <https://nalexn.github.io/clean-architecture-swiftui/>, 2023. Online; accessed 15 Feb 2023.
- [22] Paul Allies. Medium.com, “Clean Architecture in SwiftUI 5.5.” <https://paulallies.medium.com/clean-architecture-in-the-flavour-of-swiftui-5-5-8430786a83>, 2023. Online; accessed 15 Feb 2023.
- [23] Vincent Neo, “Core GPX.” <https://github.com/vincentneo/CoreGPX>, 2023. Online; accessed 18 Mar 2023.
- [24] Swinject, “Swinject.” <https://github.com/Swinject/Swinject>, 2023. Online; accessed 20 Apr 2023.
- [25] Elai Zuberma. Medium.com, “How to create toast in SwiftUI.” <https://medium.com/swlh/presenting-apples-music-alerts-in-swiftui-7f5c32cebed6>, 2023. Online; accessed 30 Apr 2023.
- [26] Apple, “XCode.” <https://developer.apple.com/xcode/>, 2022. Online; accessed 28 Oct 2022.
- [27] Cocoapods, “What is Cocoapods.” <https://cocoapods.org>, 2023. Online; accessed 10 Feb 2023.
- [28] Overleaf, “Overleaf.” <https://www.overleaf.com/>, 2023. Online; accessed 15 Feb 2023.
- [29] Hacking With Swift, “How to add Markdown comments to your code.” <https://www.hackingwithswift.com/example-code/language/how-to-add-markdown-comments-to-your-code>, 2023. Online; accessed 20 May 2023.

- [30] Stanford University, “CS193p - Developing Apps for iOS.” <https://cs193p.sites.stanford.edu>, 2022. Online; accessed 10 Nov 2022.
- [31] PMI, “PMBOK Guide — Project Management Institute.” https://www.pmi.org/pmbok-guide-standards/foundational/pmbok?sc_camp=D750AAC10C2F4378CE6D51F8D987F49D, 2023. Online; accessed 15 Apr 2023.