



Universidad de Valladolid

ESCUELA DE INGENIERÍA INFORMÁTICA
Trabajo Fin de Grado

Grado en Ingeniería Informática
(Mención en Ingeniería de Software)

**Mejoras en la usabilidad de la aplicación móvil OrientaTree para
orientación deportiva aplicada a la educación**

Alumno:
Mario Llorente Pérez

Tutores:
Alejandra Martínez Monés
Juan Alberto Muñoz Cristóbal

Agradecimientos

A mis tutores, Alejandra y Juan, por su inestimable ayuda y dedicación a lo largo de este proyecto.

A mi familia, por ayudarme y apoyarme durante toda mi etapa universitaria y en especial, recordar a los que ya no están con nosotros.

A mis amigos, por estar siempre a mi lado y llenar mis días de alegría.

Resumen

Este proyecto está enfocado en el ámbito de las actividades físicas en el medio natural. Más concretamente, en las carreras de orientación con un enfoque educativo. En este documento, se describe el proceso por el que se ha analizado una solución ya existente, y a partir de ella, se ha realizado un análisis de usuarios y requisitos, del que surgieron nuevas necesidades de los usuarios respecto de la aplicación, resaltando la necesidad de una funcionalidad sin conexión a internet y una mejora de la interfaz de seguimiento de los participantes.

Debido a las características del proyecto, se ha empleado una metodología de diseño centrado en el usuario, donde contamos con la inestimable ayuda de miembros de la Facultad de Educación y Trabajo Social. En este documento se recogen las técnicas que han sido empleadas, así como las pruebas realizadas con usuarios de las cuales se ha extraído una retroalimentación muy valiosa con la que mejorar nuestra aplicación.

Una de las necesidades identificadas en el proceso de diseño centrado en el usuario fue la de añadir una funcionalidad sin conexión a Internet, que funciona de manera transparente para los usuarios, facilitando así la posibilidad de realizar carreras de orientación sin conexión a internet y que posteriormente se sincronicen los datos para su valoración.

Palabras Clave: Orientación, Android, modo sin conexión, interfaz, usabilidad, diseño centrado en el usuario.

Abstract

This project is focused on physical activities in natural environments, specifically on orienteering with an educational approach. This document describes the process of analysing an existing solution and, based on that, conducting a user and requirements analysis, which revealed new user needs regarding the application. These needs highlighted the demand for offline functionality and an improved participant tracking interface.

Due to the nature of the project, a user centered design methodology was employed, with invaluable assistance from members of the Education and Social Work Faculty. This document outlines the techniques used, as well as the user testing conducted, which provided highly valuable feedback for improving the application.

One of the needs identified during the user centered design process was the addition of offline functionality, which works seamlessly for users. This enables orienteering activities to be conducted without an internet connection, with data syncing later for evaluation purposes.

Keywords: Orienteering, Android, offline mode, interface, usability, user centered design.

Índice general

Agradecimientos	1
Resumen	3
Abstract	5
Índice	6
Lista de figuras	10
Lista de Tablas	13
1. Introducción	1
1.1. Contexto	1
1.2. Orientación deportiva	2
1.3. Investigación de herramientas ya existentes	5
1.4. Motivación	5
1.5. Objetivos	6
1.6. Organización del documento	6
2. Planificación	9
2.1. Metodología	9
2.1.1. Diseño centrado en el usuario	9
2.1.2. Desarrollo incremental	10
2.2. Planificación	11

2.2.1.	Fase de análisis	12
2.2.2.	Fase de diseño y desarrollo	12
2.2.3.	Análisis de riesgos	14
2.2.4.	Planificación de riesgos	15
2.2.5.	Monitorización de riesgos	17
2.2.6.	Comparación respecto a la planificación inicial	18
2.2.7.	Estimación de costes	18
3.	Análisis	21
3.1.	Análisis de usuarios	21
3.1.1.	Proceso de investigación con usuarios	21
3.1.2.	Actores principales del sistema: Definición de "personas"	21
3.2.	Tecnologías para el funcionamiento sin conexión	22
3.2.1.	WorkManager	23
3.2.2.	Bluetooth 5.1	23
3.2.3.	Wi-Fi Hotspot	23
3.2.4.	Método empleado en el desarrollo de la funcionalidad <i>offline</i>	24
3.3.	Requisitos	24
3.3.1.	Requisitos funcionales	24
3.3.2.	Requisitos de información	27
3.3.3.	Requisitos no funcionales	28
3.4.	Casos de uso	29
3.5.	Modelo de dominio	36
3.5.1.	Realización en análisis de los casos de uso	38
3.5.2.	Modelo de dominio detallado	41
4.	Diseño	43
4.1.	Guías de diseño aplicadas	43
4.1.1.	Heurísticas de Nielsen	43
4.1.2.	Material Design	44

4.2. Proceso de diseño iterativo	45
4.2.1. Interfaz ya existente	45
4.2.2. Diseño inicial	47
4.2.3. Primera prueba de usabilidad	51
4.2.4. Segunda prueba de usabilidad	57
4.2.5. Tercera prueba de usabilidad: prueba presencial con usuarios	62
4.3. Arquitectura de la aplicación	65
4.3.1. Estructura del front-end	65
4.3.2. Estructura del back-end	68
4.3.3. Diagrama de despliegue	68
4.4. Diseño de la funcionalidad sin conexión	69
4.4.1. Capa de persistencia	69
4.4.2. WorkManager	70
4.4.3. Localización	70
4.5. Patrones de diseño	70
4.5.1. Patrón DAO/DTO	71
4.5.2. Patrón Singleton	71
4.5.3. Patrón Abstract Factory	72
4.5.4. Patrón Facade	72
4.5.5. Patrón Builder	72
4.5.6. Patrón Adapter	73
5. Implementación	77
5.1. Tecnologías usadas	77
5.2. Pruebas de la aplicación	79
6. Conclusiones y trabajo futuro	85
6.1. Conclusiones	85
6.1.1. Objetivos conseguidos	85
6.2. Trabajo futuro	86
6.3. Valoración personal	88

A. Manual de usuario	91
A.1. Inicio de sesión y registro	91
A.2. Pantalla de inicio	92
A.3. Configuración de perfil	92
A.4. Organizar actividad	93
A.5. Inscribirse en una actividad	95
A.6. Participar en una actividad	96
A.7. Supervisión de una actividad	97
A.8. Revisión de una actividad	98
A.9. Revisión de una actividad como participante	99
B. Enlaces de interés	101
Bibliografía	103

Índice de figuras

1.1. Mapa de orientación deportiva [7]	3
1.2. Brújula de orientación deportiva [8]	4
1.3. Baliza de orientación deportiva	4
2.1. Proceso de desarrollo incremental mediante iteraciones	11
2.2. Diagrama de Gantt general del proyecto	12
2.3. Gantt de la fase de análisis	12
2.4. Gantt de la primera iteración	13
2.5. Gantt de la segunda iteración	13
2.6. Gantt de la tercera iteración	13
2.7. Gantt de la cuarta iteración	14
2.8. Gantt de la quinta iteración	14
2.9. Gantt de la duración real del proyecto	18
3.1. Arquetipo de Organizador. Sara Márquez, profesora de Educación Física	22
3.2. Arquetipo de Participante. Jorge Ramírez, alumno de 5º de primaria	22
3.3. Casos de uso comunes a todos los usuarios	30
3.4. Casos de uso del organizador	30
3.5. Casos de uso del participante	31
3.6. Casos de uso abordados en este TFG	31
3.7. Modelo de dominio, indicándose en azul la clase en la que se han introducido modificaciones en el presente TFG con respecto a la aplicación original	37
3.8. Diagrama Secuencia CU1: Realizar el seguimiento de una actividad	39
3.9. Diagrama Secuencia CU2: Consultar el progreso de un participante	39

3.10. Diagrama Secuencia CU3: Consultar el recorrido de un participante	40
3.11. Diagrama Secuencia CU4: Consultar la localización de un participante	40
3.12. Diagrama Secuencia CU5: Revisar los resultados de una actividad	40
3.13. Diagrama Secuencia CU6: Comparar los recorridos de varios participantes	41
3.14. Modelo de dominio extendido	41
4.1. Portada de la actividad, lista de participantes y lista de balizas alcanzadas por un participante	46
4.2. Pregunta de una baliza	46
4.3. Vista de tarjetas	47
4.4. Vista de lista	48
4.5. Vistas de mapa del organizador y del usuario	49
4.6. Vistas de balizas del organizador y el usuario	50
4.7. Prototipos con información para el organizador	50
4.8. Prototipo de vista de balizas y recorrido para el organizador	51
4.9. Prototipo de participación en actividad	52
4.10. Prototipo de actividad de orientación. Selección de actividad	52
4.11. Prototipo de actividad de orientación. Información de los participantes	53
4.12. Prototipo de actividad de orientación. Preguntas	53
4.13. Prototipo de revisión de actividad	54
4.14. Segundo prototipo de la interfaz. Vista completa	58
4.15. Parte de <i>Log-in</i> del segundo prototipo de la interfaz	58
4.16. Parte de información de los participantes del segundo prototipo de la interfaz . . .	59
4.17. Parte de balizas del segundo prototipo de la interfaz	59
4.18. Vistas de selección de actividad y de información de la actividad	62
4.19. Vistas de información de los participantes	63
4.20. Vistas de información de las balizas	63
4.21. Diagrama de paquetes de la aplicación cliente	66
4.22. Detalle del paquete ui	66
4.23. Detalle del paquete model	67

4.24. Detalle del paquete persistence	67
4.25. Detalle del paquete adapters	67
4.26. Diagrama de despliegue de la aplicación	69
4.27. Patrón DAO/DTO	71
4.28. Patrón Singleton	72
4.29. Patrón Abstract Factory	73
4.30. Patrón Facade	74
4.31. Patrón Builder	74
4.32. Patrón Adapter	75
A.1. Pantallas de inicio de sesión y registro	91
A.2. Pantallas con la información de las actividades y menú lateral	92
A.3. Pantallas con menú lateral y configuración del usuario	93
A.4. Pantallas con menú lateral y plantillas de actividad	94
A.5. Pantallas con configuración de la actividad	94
A.6. Pantallas con información de una actividad	95
A.7. Pantallas de inscripción en una actividad	95
A.8. Pantallas de inicio de participación	96
A.9. Pantallas de participación y balizas	96
A.10. Pantallas de actividad del organizador	97
A.11. Pantallas con información de los participantes	98
A.12. Pantallas con información de las balizas y el recorrido de un participante	98
A.13. Pantallas con información de los participantes y las balizas	99
A.14. Pantallas con selección de participantes e información del recorrido	99
A.15. Pantallas con información del desempeño propio del participante, su recorrido y las respuestas de las balizas	100

Índice de cuadros

2.1. Riesgo 1: Estimaciones de tiempo erróneas	15
2.2. Riesgo 2: Modificar código ya existente	15
2.3. Riesgo 3: Errores durante el desarrollo	15
2.4. Riesgo 4: Especificaciones Genéricas	16
2.5. Riesgo 5: Cambios en la interfaz de usuario	16
2.6. Riesgo 6: Añadido de características innecesarias	16
2.7. Riesgo 7: Falta de experiencia con alguna de las tecnologías	16
2.8. Riesgo 8: Indisposición del desarrollador	16
2.9. Riesgo 9: Parálisis por análisis	17
2.10. Riesgo 10: Problemas con el entorno de desarrollo	17
2.11. Costes totales del proyecto	19
2.12. Costes reales del proyecto	20
3.1. Caso de uso: Realizar el seguimiento de una actividad	32
3.2. Caso de uso: Consultar el progreso de un participante	33
3.3. Caso de uso: Consultar el recorrido de un participante	33
3.4. Caso de uso: Consultar la localización de un participante	34
3.5. Caso de uso: Revisar los resultados de una actividad	35
3.6. Caso de uso: Comparar los recorridos de varios participantes	36
4.1. % de completitud de las tareas	56
4.2. % de completitud de las tareas	61
5.1. CP1. Posición de un participante que no ha comenzado la actividad	80

5.2. CP2. Posición de un participante que ha finalizado la actividad	80
5.3. CP3. Un participante comienza una actividad sin conexión a internet	80
5.4. CP4. Un participante finaliza una actividad sin conexión a internet	81
5.5. CP5. Un participante se inscribe en una actividad	81
5.6. CP6. Comprobar la posición de los participantes	81
5.7. CP7. Comprobar el progreso de los participantes	82
5.8. CP8. Ver el detalle de las balizas de un participante durante una actividad	82
5.9. CP9. Ver la respuesta a una baliza de un participante durante una actividad . . .	82
5.10. CP10. Ver el recorrido de un participante durante una actividad	83
5.11. CP11. Comprobar el desempeño de los participantes tras la actividad	83
5.12. CP12. Ver el recorrido de varios participantes tras una actividad	83
5.13. CP13. Responder a la pregunta de una baliza sin conexión	84

Capítulo 1

Introducción

1.1. Contexto

La orientación deportiva es tanto una actividad física en el medio natural [1] como un deporte, llevado a cabo por uno o varios participantes. Este consiste en recorrer una serie de puntos de control con la única ayuda de una brújula y un mapa. Este es un mapa específico de este deporte, ya que incluye curvas de nivel, vegetación, accidentes naturales e información topológica mediante una leyenda bien conocida. Estos puntos de control o balizas son desconocidos para los participantes previamente al inicio de la carrera, así que los participantes deberán conocer su localización mediante los puntos marcados en el mapa, y planificar cómo llegar a las mismas y el recorrido a seguir. Además, los participantes suelen poseer una hoja de seguimiento en la que marcarán con unas pinzas perforadoras ubicadas en cada baliza su paso por ellas. Estas pinzas poseen un patrón único, por lo cual los organizadores pueden validar si el recorrido ha sido correcto [2].

En el ámbito educativo, se busca utilizar la orientación con fines educacionales buscando enseñar a los alumnos tanto el deporte de la orientación, como conocer el funcionamiento y características de los elementos de este deporte, como son el mapa y la brújula, muy útiles en la vida cotidiana. Una variación de la orientación tradicional plantea un recorrido algo diferente en el cual se añade una pequeña actividad en cada baliza, ya sea esta una adivinanza, una pregunta relacionada con el entorno, una pequeña prueba o aportar cierta información que ayuda a los alumnos a conocer su medio y alimentar su interés por el mismo [3]. En este sentido, las TIC son de gran valor, ya que aportan soporte y enriquecen estas actividades.

La orientación educativa¹ se enmarca en los criterios establecidos por el Real Decreto 1631/2006, también conocido como la *Ley Orgánica de Educación* (LOE) [3] en la que se busca fomentar la actividad física y el descubrimiento del entorno a los jóvenes. Actualmente, muchos profesores emplean medios informáticos para simplificar la tarea de organizar y coordinar una actividad de orientación educativa. Se emplean plataformas digitales, aplicaciones de gestión y herramientas de comunicación en línea que facilitan la planificación de sesiones, el seguimiento del progreso de los estudiantes y la colaboración entre docentes y alumnos. Estas tecnologías permiten una mayor flexibilidad y accesibilidad, posibilitando la realización de actividades interactivas que se adaptan a las necesidades individuales de los estudiantes. Además, ofrecen la oportunidad de integrar materiales multimedia y recursos interactivos, enriqueciendo así el proceso de enseñanza y

¹Orientación deportiva enfocada en el ámbito educativo

aprendizaje [4].

1.2. Orientación deportiva

Las actividades físicas en la naturaleza están actualmente en auge, ya que tanto el turismo como el deporte han impulsado la realización de las mismas, debido principalmente a la creciente preocupación de la población por la práctica física y el esparcimiento al aire libre. También se ha visto impulsado por el desgaste de los deportes tradicionales y el ritmo de vida de grandes núcleos de población [1].

La naturaleza nos ofrece un marco increíble para la práctica de deportes al aire libre con su variedad de escenarios y situaciones, en las que estas actividades se realizan sirviéndose principalmente de los recursos que ofrece la naturaleza en el medio en el que se desarrollan.

Una de estas actividades es la orientación, un deporte centrado en conocer el medio natural mediante la búsqueda de una serie de localizaciones, ayudados solamente por un mapa y una brújula. Este deporte es interesante, ya que permite conectar con la naturaleza y el entorno local, favorece la curiosidad y es sencillo de adaptar dependiendo de las necesidades de los participantes [3].

La orientación como deporte consiste en una carrera que se realiza sobre un terreno variado con una serie de puntos de control conocidos como balizas, en los que el participante ha de alcanzarlos con la ayuda de un mapa y una brújula. Este recorrido a realizar no es explícito, sino que queda a juicio del corredor el orden de las balizas y el recorrido a seguir para llegar a cada una de ellas. El objetivo es llegar a todas las balizas en el menor tiempo posible [5]. En nuestro caso, nos centramos en la orientación educativa, una modalidad de la orientación utilizada principalmente en el ámbito educativo, en la que se busca tanto enseñar a los más pequeños el uso del mapa y la brújula como descubrir el medio a su alrededor. Una de las modificaciones más empleadas para esto último es sustituir el marcado de la baliza por una pregunta sencilla ubicada en la baliza acerca del entorno en el que está situada. Con esto se anima a los más pequeños a conocer sus alrededores estimulando la curiosidad y haciendo un deporte ligero y adaptable según sus capacidades.

El equipamiento necesario para la orientación consiste en:

■ Mapa

Es un mapa topográfico especial (ver Figura 1.1), caracterizado por contener todos los pequeños detalles del terreno. Este es de gran ayuda, ya que nos permite valorar aspectos como la vegetación, los accidentes del terreno, obstáculos a encontrar o zonas cuya entrada está prohibida. Además, ha de contar con las siguientes características [6]:

- Escala: Relación entre el tamaño del mapa y el tamaño del terreno. La más común es 1:15000 (1 cm equivale a 150 metros).
- Curvas de nivel: Estas son las encargadas de representar la altura del terreno. Una curva de nivel representa todos los puntos del terreno contiguos a la misma altura.
- Leyenda: Esta nos ayuda a comprender el significado de los posibles símbolos y notaciones que puedan aparecer en el mapa.
- Dirección del norte magnético: Una serie de líneas que cruzan el mapa señalando el norte magnético para facilitar el uso de la brújula.

- Líneas de meridianos y paralelos: Ocasionalmente, se pueden incluir estas líneas que marcan la longitud y la latitud y facilitan la labor de orientación.



Figura 1.1: Mapa de orientación deportiva [7]

■ Brújula

Este es un instrumento de navegación (ver Figura 1.2) que siempre apunta al norte magnético. Generalmente constan de una aguja montada sobre un pivote que le permite girar libremente por una carcasa marcada con los puntos cardinales: Norte, Sur, Este y Oeste. Las brújulas de orientación tienen un formato particular, ya que cuentan con una serie de características [6]:

- Aguja imantada de color rojo.
- Limbo giratorio graduado en 360 grados indicando los cuatro puntos cardinales.
- Líneas Norte-Sur en el limbo.
- Base rectangular transparente, generalmente de plástico.
- Flecha de dirección en la base de la brújula.
- Regla graduada.



Figura 1.2: Brújula de orientación deportiva [8]

■ Balizas y tarjetas de verificación

Las balizas (ver Figura 1.3) consisten en prismas triangulares de tela divididos en dos triángulos, uno blanco y uno naranja, y están señaladas por un círculo magenta y una codificación en el mapa. Permiten verificar el paso por las mismas usando un dispositivo de marcado ubicado en su interior sobre la tarjeta de verificación o apuntando el código indicado en las mismas. Recientemente se han empezado a sustituir por chips que recogen la información de paso al acercarse o aplicaciones informáticas geolocalizadas.



Figura 1.3: Baliza de orientación deportiva

1.3. Investigación de herramientas ya existentes

La web de la FEDO (Federación Española de Orientación)² posee un apartado con distintas aplicaciones relacionadas con el mundo de la orientación [9]. A continuación, citamos algunas de las más importantes:

- **iOrienteering:** Esta aplicación de orientación es una de las más populares actualmente, ya que nos permite crear recorridos de orientación, eventos de orientación, participar en los mismos y hacer comparación de los resultados después. También dispone de funciones de cuentas secundarias destinadas a su uso educativo, grupal o familiar. Los resultados se pueden exportar para analizarlos en un dispositivo distinto y la aplicación también dispone de un modo *offline* de funcionalidad limitada. Posee aplicaciones para iOS, Android y Windows [10].
- **COMPASS Orienteering:** Esta aplicación, desarrollada por la Universidad de Viena, es una aplicación de orientación con funcionalidades de orientación educativa. Permite a los usuarios crear y participar en actividades de orientación. Está destinada a su uso educativo y a clubs profesionales de orientación, aunque se puede utilizar para otros fines. Posee una aplicación para dispositivos Android [11].

Además de estas, en este proyecto vamos a trabajar especialmente con otra herramienta desarrollada en la Universidad de Valladolid:

- **OrientaTree** Esta aplicación de orientación educativa es la base sobre la que se construye Orientatree 2.0. Esta aplicación, desarrollada por Gabriel Rodríguez González, permite realizar recorridos de orientación en el parque de Campo Grande en Valladolid. La aplicación permite a los usuarios organizar una actividad de orientación a partir de unas plantillas ya establecidas y realizar un seguimiento de la actividad. También permite a los usuarios inscribirse en actividades de orientación y participar en ellas. Posee varias modalidades de actividades y también permite realizar una revisión de las actividades tras su realización. Es una aplicación para dispositivos Android [12].

1.4. Motivación

Este proyecto surge a raíz del proyecto de OrientaTree [12] ya mencionado en la sección anterior. Con este, se buscaba ofrecer una aplicación útil en el contexto escolar y de aprendizaje formal, especialmente centrada en un ámbito de educación física en primaria. A partir de este proyecto ya existente, se pretende continuar el trabajo realizado en el mismo y recoger algunas de las propuestas abiertas en el proceso de diseño iterativo planteado en dicho TFG. En concreto, se busca una mejora de la aplicación, centrada en dos puntos principales:

- Mejorar la monitorización ofrecida por la aplicación en cuanto al seguimiento y análisis del recorrido y actividades de los participantes, ya que la aplicación está limitada tanto en las opciones del docente en la información que le ofrece, como en las opciones que posee el docente para analizar y comparar dicha información.

²<https://www.fedo.org/web/>

- Ofrecer una funcionalidad *offline* de la aplicación, ya que no siempre se cuenta con conexión a internet en el medio natural y los niños de primaria no suelen tener acceso a una conexión de datos en su móvil, si es que acaso poseen un terminal personal.

En definitiva, se pretende retomar un proyecto ya existente y enriquecerlo de cara a un uso educativo y permitir su uso en mayor variedad de escenarios.

1.5. Objetivos

El principal objetivo de este proyecto es:

Diseñar una nueva versión de la aplicación OrientaTree, incorporando aspectos necesarios para mejorar la experiencia de uso por parte del alumnado y el profesorado. Estos aspectos son:

- *Posibilidad de realizar carreras sin acceso a internet en los dispositivos móviles.*
- *Facilitar el seguimiento de la actividad de los participantes por parte de los docentes.*

Debido a esto, se plantean importantes retos en cuanto a la capacidad de conectividad de los dispositivos móviles, unido a la necesidad de mejorar la interfaz de usuario, añade dos objetivos secundarios:

- *Aplicar técnicas del diseño centrado en el usuario, con el fin de mejorar el diseño de la aplicación.*
- *Identificar y explorar distintas alternativas para realizar el seguimiento y análisis del recorrido de los usuarios en caso de no disponer de conexión a internet de manera constante.*

En este trabajo de fin de grado, se parte de una aplicación ya existente, y se cuenta con acceso tanto a la memoria del trabajo realizado, como al código fuente de la aplicación ya existente. A partir de esto, se busca identificar posibles mejoras de la aplicación, centrándonos en las posibilidades de monitorización de los recorridos y actividades realizadas por los estudiantes, así como alternativas en el funcionamiento de la aplicación para adaptar su funcionamiento a un modo sin conexión a internet. Durante este proyecto, se contará con el apoyo de profesores de la facultad de Educación y Trabajo Social de la Universidad de Valladolid para valorar y probar las posibles mejoras y cambios a introducir en la aplicación.

1.6. Organización del documento

El resto del documento se estructura de la siguiente manera:

- **Capítulo 2.** En este capítulo se recogen las metodologías seguidas durante la realización del proyecto, así como la planificación establecida para el mismo.

- **Capítulo 3.** En el tercer capítulo se describe la investigación de los usuarios y la investigación de tecnologías para el funcionamiento sin conexión, así como otros elementos propios de la fase de análisis de un proyecto software como requisitos, casos de uso, modelo de dominio y realización en análisis de los casos de uso.
- **Capítulo 4.** En este capítulo se recogen las guías de diseño aplicadas, el proceso de diseño iterativo apoyado por las pruebas con usuarios, la arquitectura de la aplicación, el diseño de la funcionalidad sin conexión y los patrones de diseño empleados .
- **Capítulo 5.** En el quinto capítulo se exponen las tecnologías y herramientas empleadas en el desarrollo de la aplicación junto con las pruebas realizadas para verificar el correcto funcionamiento de la aplicación.
- **Capítulo 6.** En el último capítulo se muestran las conclusiones obtenidas tras la realización de este proyecto, así como posibles vías de trabajo futuro y la valoración personal.
- **Anexos.**
- **Bibliografía.**

Capítulo 2

Planificación

Este capítulo recoge la metodología, plazos, actividades y riesgos a tener en cuenta para poder alcanzar un resultado satisfactorio al final del proyecto. Estableceremos varias etapas en las que se irá reuniendo información y completando objetivos de manera progresiva.

2.1. Metodología

Debido al contexto en el que se desarrolla esta aplicación, es importante considerar cuáles serán las mejores metodologías a seguir. En este caso, hay dos perfiles de usuario claros, los principales participantes, público infantil, donde es importante hacer la aplicación sencilla de manejar y entender; y por otro lado los profesores, donde la aplicación ha de ser eficiente para realizar el seguimiento de múltiples participantes de manera simultánea, siendo esto último uno de los focos de este proyecto. Además, podemos disponer de la colaboración de una serie de profesores que nos pueden ayudar a identificar puntos clave durante el desarrollo para poder obtener retroalimentación de usuarios finales. Debido a esto, he considerado que el diseño centrado en el usuario y el desarrollo incremental son las mejores metodologías a seguir.

2.1.1. Diseño centrado en el usuario

El diseño centrado en el usuario (DCU) es tanto una filosofía como un conjunto de procedimientos que se centran en la observación y comprensión de las necesidades reales de las personas para el diseño de productos. Se hace énfasis en el diseño iterativo y las pruebas con usuarios para adaptar y mejorar los productos [13]. La meta de esta filosofía es crear productos que se adapten a las necesidades de los usuarios y tengan una alta usabilidad. Esto incluye la eficacia, la eficiencia y la satisfacción. El proceso basado en esta filosofía se puede agrupar en cuatro fases generales [14]:

- **Entender el contexto de uso**

- Identificar quiénes van a ser los principales usuarios del producto, cuáles son sus requisitos

y dónde lo van a utilizar. En este caso, fueron mis tutores los que realizaron esta primera fase previa a la realización de este TFG.

- **Especificar los requisitos del usuario**

Es imperativo especificar cuáles van a ser los principales requisitos que van a guiar el desarrollo del producto. El punto de entrada para la formulación de los requisitos fue un documento, proporcionado por mis tutores y obtenido en una investigación de usuarios anterior, que contenía una serie de preguntas que se hacía un usuario e ilustraban lo que el usuario necesitaba de la interfaz [15]. El presente TFG debe diseñar una interfaz que proporcione respuestas a dichas preguntas. Con esta información, se idean una serie de escenarios en los que se presenten esas preguntas. Con estos escenarios ya estructurados, se procede a establecer una serie de requisitos y objetivos a cumplimentar.

- **Diseñar soluciones**

En base a los requisitos y objetivos, iniciar un desarrollo iterativo del producto. Con los escenarios a resolver y los requisitos definidos, se crea un primer prototipo de interfaz con posibles cambios y mejoras a realizar sobre la existente. Con esto, se procede a construir una segunda iteración de este prototipo de baja fidelidad con capacidad interactiva limitada para realizar una prueba de las ideas propuestas con los usuarios.

- **Evaluar el producto**

Realizar test de usabilidad frecuentes para obtener retroalimentación del usuario es clave en un desarrollo centrado en el usuario satisfactorio. Con el prototipo, se realizan varias pruebas con usuarios, en las que los usuarios desarrollan distintas tareas de manera individual para comprobar la usabilidad de la aplicación complementada por una pequeña entrevista informal. Tras esto, se realiza una puesta en común de las experiencias de los usuarios, en la que obtener más información acerca de las impresiones de los usuarios.

2.1.2. Desarrollo incremental

El desarrollo incremental se presenta como una estrategia para abordar proyectos de elevada complejidad, y se basa en dividir el desarrollo en etapas manejables y entregables parciales. Debido a esto, el desarrollo es más flexible y es capaz de adaptarse a los cambios propuestos por los usuarios derivados del diseño centrado en el usuario. Esta estrategia se adapta mejor a los cambios que el desarrollo clásico en cascada, dónde se busca definir todos los requisitos en un primer momento para después desarrollar el producto [16].

Durante el desarrollo de este proyecto se buscó seguir una metodología de diseño incremental mediante el uso de iteraciones, que permiten dividir en partes pequeñas y manejables de manera sencilla un conjunto específico de tareas a realizar y objetivos a cumplir. Esto permite ir abordando requisitos de los usuarios de manera más manejable junto con los cambios propuestos por los usuarios en las pruebas de usabilidad.

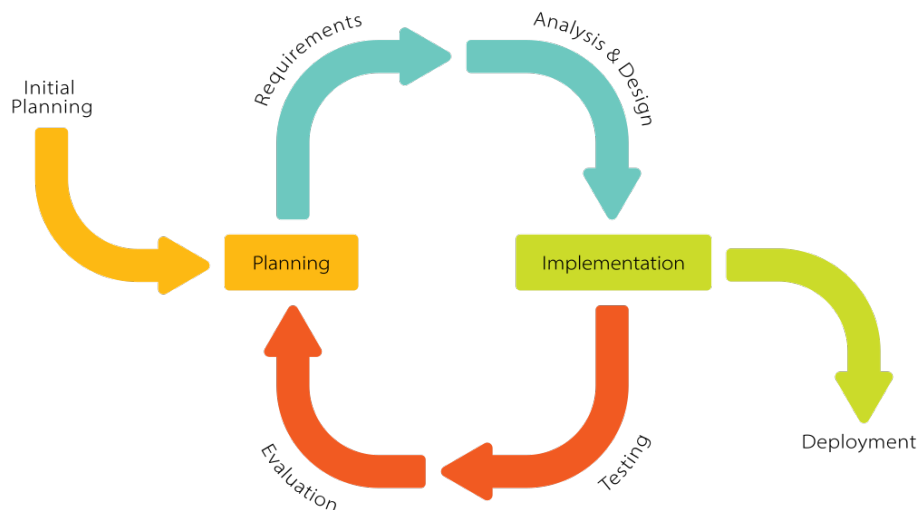


Figura 2.1: Proceso de desarrollo incremental mediante iteraciones
[17]

Al integrar principios ágiles como la colaboración con los usuarios finales y su retroalimentación, el desarrollo incremental cumple con las necesidades de este proyecto al ofrecer una solución centrada en los usuarios [18]. Además, esta metodología permite un sistema de entregas tempranas que puedan ser probadas por los usuarios a cada iteración del desarrollo para aportar retroalimentación que permita realizar ajustes en las siguientes etapas.

2.2. Planificación

Este proyecto consta de dos partes, cuya planificación se describe a continuación: La primera parte del proyecto, está destinada a analizar el material ya existente referente al desarrollo de la anterior aplicación, identificar posibles áreas de mejora en las que hacer que la experiencia del usuario sea más satisfactoria, de acuerdo con los requisitos. Luego, con la ayuda de prototipos de baja fidelidad, se realizará una prueba preliminar con usuarios reales, de la que se extraerá información acerca de los cambios propuestos y de posibles mejoras propuestas por los usuarios. Este proceso proporcionará una base para continuar desarrollando la aplicación. En la segunda parte, los esfuerzos se destinarán principalmente en la parte de desarrollo software, en la que me centraré en dos puntos principales: La mejora de la interfaz de usuario y el añadido de un modo de funcionamiento sin conexión a internet. Este proceso se dividirá en iteraciones en las que se intercalen partes tanto del desarrollo software como de documentación y bibliografía. También se llevarán a cabo pruebas de distintos tipos con usuarios para validar que el desarrollo de la aplicación esté siendo correcto. Se adjunta una visión general de la planificación (Figura 2.2).

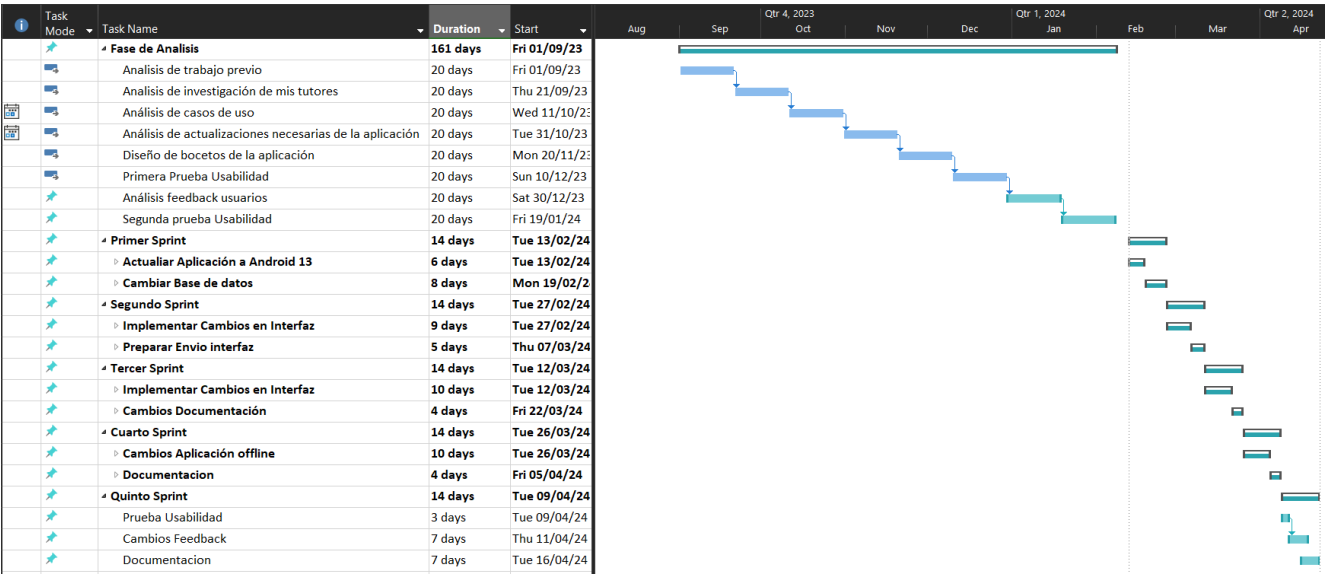


Figura 2.2: Diagrama de Gantt general del proyecto

2.2.1. Fase de análisis

En esta primera fase (Figura 2.3) se inicia con tareas de análisis del trabajo previo realizado por los tutores del proyecto y la extracción de requisitos. A continuación, se realizan las tareas de definición de los casos de uso, los diagramas de secuencia, y modelo de dominio. También se planean los cambios previos al desarrollo de la nueva funcionalidad.

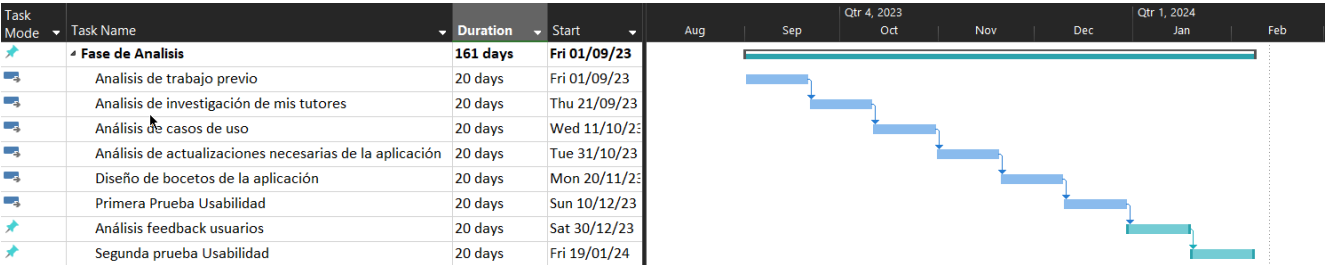


Figura 2.3: Gantt de la fase de análisis

2.2.2. Fase de diseño y desarrollo

Cada iteración realizada debe producir una mejora en la funcionalidad de la aplicación, y cada una de ellas constará de cada una de las fases descritas en la figura 2.1:

- Planificación
- Análisis
- Diseño

- Implementación
- Evaluación

Primera iteración

En esta iteración (figura 2.4) los objetivos a cumplimentar son actualizar la aplicación Android a una versión más reciente que permita subirla a la Playstore para hacer más sencilla su distribución, así como actualizar la base de datos a una nueva para preservar la antigua.

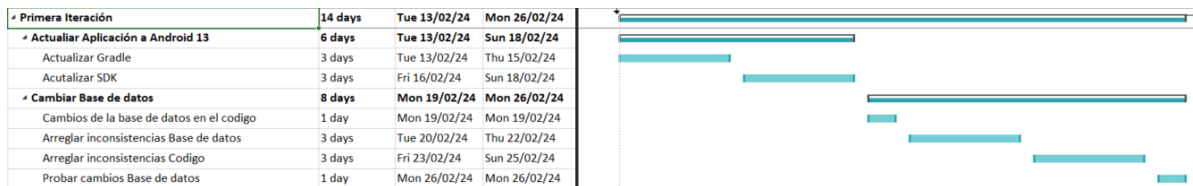


Figura 2.4: Gantt de la primera iteración

Segunda y tercera iteración

En estas iteraciones (figuras 2.5 y 2.6) consisten en actualizar el mapa de orientación a uno con mayor resolución, preparar el prototipo para la primera prueba de usabilidad, adaptar la revisión para poder visualizar varios recorridos y realizar la primera prueba de usabilidad.

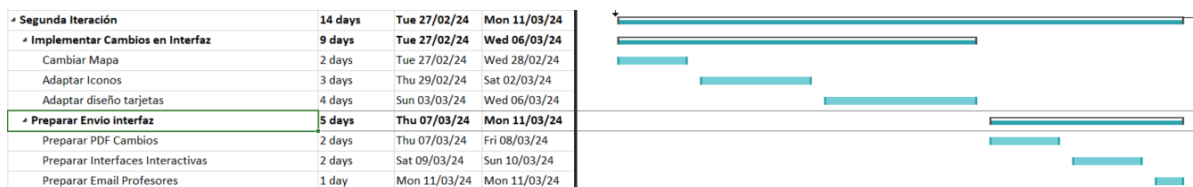


Figura 2.5: Gantt de la segunda iteración

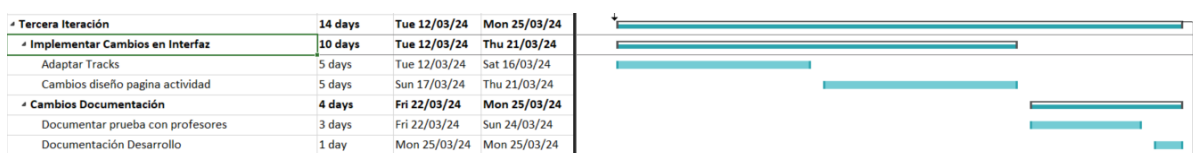


Figura 2.6: Gantt de la tercera iteración

Cuarta iteración

En esta iteración (figura 2.7) el principal objetivo es actualizar la aplicación Android para ofrecer una funcionalidad *offline* limitada, así como una funcionalidad asíncrona para poder sincronizar la información con la base de datos cuando se disponga de conexión a internet y corregir los errores de posicionamiento GPS.

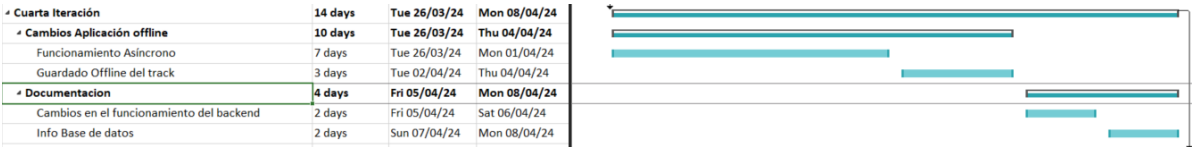


Figura 2.7: Gantt de la cuarta iteración

Quinta iteración

En esta iteración (figura 2.8) consiste en mejorar la aplicación mediante la integración de la retroalimentación obtenida durante las distintas pruebas realizadas con usuarios.

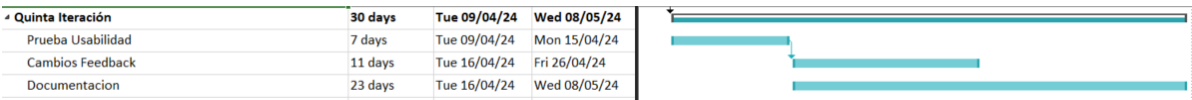


Figura 2.8: Gantt de la quinta iteración

2.2.3. Análisis de riesgos

Una parte importante de la planificación de un proyecto de desarrollo software es realizar un análisis de riesgos. Un riesgo es un evento incierto que, en el caso de ocurrir, afecta positiva o negativamente en los objetivos del proyecto [19]. Es importante identificar y analizar los riesgos para asegurar la completitud de los objetivos del proyecto. Algunas de las técnicas empleadas pueden ser *checklists*, basadas en riesgos recurrentes de proyectos de desarrollo software, como por ejemplo la elaborada por B.W. Boehm [20]; o *brainstorming*, donde los *stakeholders* identifican posibles riesgos basándose en su experiencia previa. Los riesgos identificados serán priorizados mediante la técnica de análisis de exposición al riesgo (Risk Exposure, RE) [21], donde se calcula el riesgo mediante un valor numérico que representa cómo de posible es que se dé el riesgo durante el desarrollo del proyecto. Se calcula de la siguiente manera:

$$RE = probabilidad \times impacto$$

La variable de probabilidad representa cómo de posible es que el riesgo se manifieste en nuestro proyecto, mientras que impacto representa cuánto afectaría a nuestros objetivos en caso de hacerlo. En la tabla siguiente aparecen los riesgos que hemos identificado en el proyecto.

Riesgo	Probabilidad	Impacto	Prioridad
1.Estimaciones de tiempo erróneas	9	7	63
2.Modificar código ya existente	5	5	25
3.Errores durante el desarrollo	7	7	49
4.Especificaciones genéricas	6	6	36
5.Cambios en la interfaz de usuario	8	6	48
6.Añadido de características innecesarias	6	5	30
7.Falta de experiencia con alguna de las tecnologías	8	6	48
8.Indisposición del desarrollador	2	3	6
9.Parálisis por análisis	5	4	20
10.Problemas con el entorno de desarrollo	5	2	10

2.2.4. Planificación de riesgos

Una vez identificados los riesgos, lo siguiente es decidir las medidas a tomar para reducir la probabilidad de que el riesgo ocurra, medidas de reducción o medidas de contingencia, destinadas a reducir el impacto de este una vez ocurra.

Riesgo	Estimaciones de tiempo erróneas
Reducción	Planificar las tareas con margen por si la estimación ha sido inferior al tiempo necesario
Contingencia	Cambiar la planificación, reducir el alcance del proyecto

Cuadro 2.1: Riesgo 1: Estimaciones de tiempo erróneas

Riesgo	Modificar código ya existente
Reducción	Reunir toda la documentación disponible
Contingencia	Reemplazar el código problemático

Cuadro 2.2: Riesgo 2: Modificar código ya existente

Riesgo	Errores durante el desarrollo
Reducción	Verificar que la funcionalidad está implementada correctamente antes de dar por finalizado el incremento
Contingencia	Volver al último punto testado completamente y continuar desde ahí

Cuadro 2.3: Riesgo 3: Errores durante el desarrollo

Riesgo	Especificaciones Genéricas
Reducción	Reunir toda la información posible antes de abordar una tarea
Contingencia	Preguntar a los <i>stakeholders</i> y prototipar en caso de ser necesario

Cuadro 2.4: Riesgo 4: Especificaciones Genéricas

Riesgo	Cambios en la interfaz de usuario
Reducción	El diseño centrado en el usuario previene que esto ocurra y el diseño incremental mitiga el impacto
Contingencia	Prototipar nuevas interfaces que aprobar por los <i>stakeholders</i>

Cuadro 2.5: Riesgo 5: Cambios en la interfaz de usuario

Riesgo	Añadido de características innecesarias
Reducción	Verificar que las mejoras a implementar sean realmente importantes mediante la comunicación con los usuarios
Contingencia	Realizar una reunión con los usuarios para verificar si lo que estamos implementando es realmente útil

Cuadro 2.6: Riesgo 6: Añadido de características innecesarias

Riesgo	Falta de experiencia con alguna de las tecnologías
Reducción	Formarse en las tecnologías clave antes de proceder al desarrollo
Contingencia	Aprovechar la experiencia de los tutores para solventar los posibles problemas

Cuadro 2.7: Riesgo 7: Falta de experiencia con alguna de las tecnologías

Riesgo	Indisposición del desarrollador
Reducción	Planificar las tareas con margen por si es necesario extender el tiempo en completar el incremento
Contingencia	Realizar horas extra en caso de ser necesario

Cuadro 2.8: Riesgo 8: Indisposición del desarrollador

Riesgo	Parálisis por análisis
Reducción	El DCU simplifica la tarea de identificar cuáles son las prioridades clave para los usuarios y centrarse en ellas primero
Contingencia	Preguntar a los usuarios cuáles son las características clave a desarrollar

Cuadro 2.9: Riesgo 9: Parálisis por análisis

Riesgo	Problemas con el entorno de desarrollo
Reducción	Utilizar un repositorio remoto en el cual ir almacenando copias de seguridad
Contingencia	Utilizar una máquina de respaldo

Cuadro 2.10: Riesgo 10: Problemas con el entorno de desarrollo

2.2.5. Monitorización de riesgos

Durante el desarrollo de este proyecto, se dieron varios de los riesgos descritos en 2.2.4. A continuación se exponen cuales fueron, el impacto que tuvieron, y que medidas se tomaron para subsanarlos.

- **Errores al estimar la duración de las tareas**

Este fue el principal riesgo que se presentó durante el desarrollo de este proyecto. La planificación inicial se vio lastrada debido a que las tareas llevaron más tiempo del esperado y se tuvo que retrasar el proyecto. Una de las medidas que se tomaron a mayores fue realizar iteraciones más cortas con algo menos de contenido para avanzar más rápido.

- **Indisposición del desarrollador**

Este fue otro de los riesgos que se manifestaron durante el proyecto. Debido a mi incorporación al mundo laboral, se vio reducido el tiempo que podía dedicarle al proyecto, lo que provocó retrasos. una de las medidas que se llevó a cabo fue desarrollar después de mi horario laboral y durante los fines de semana.

- **Falta de experiencia con alguna de las tecnologías**

Este riesgo se manifestó durante el proyecto y fue una de las causas que llevó al retraso del proyecto. Una de las medidas de contingencia que se tomó fue realizar pequeños prototipos que implementen la funcionalidad de la aplicación en un mini-proyecto rápido donde se pueda probar la funcionalidad sin los problemas relacionados con la integración con el resto de la aplicación.

- **Problemas con el entorno de desarrollo**

Debido a varios cambios en las API de Google y los cambios en varias de las dependencias de la aplicación, se necesitó tiempo adicional para adaptar la aplicación a estos.

■ Modificar código ya existente

Debido tanto a los cambios *offline*, como a la necesidad de actualizar a versiones más recientes de Android por problemas de compatibilidad, se debió de modificar varias veces el código existente, tanto del trabajo previo, como del trabajo propio.

2.2.6. Comparación respecto a la planificación inicial

El proyecto ha sufrido un retraso debido a la aparición de uno de los riesgos identificados, en concreto, mi incorporación al mundo laboral. Esto provocó una pausa de 6 meses en el proyecto y un ligero aumento de horas dedicadas. Se adjunta un gráfico con la duración real del proyecto (figura 2.9).

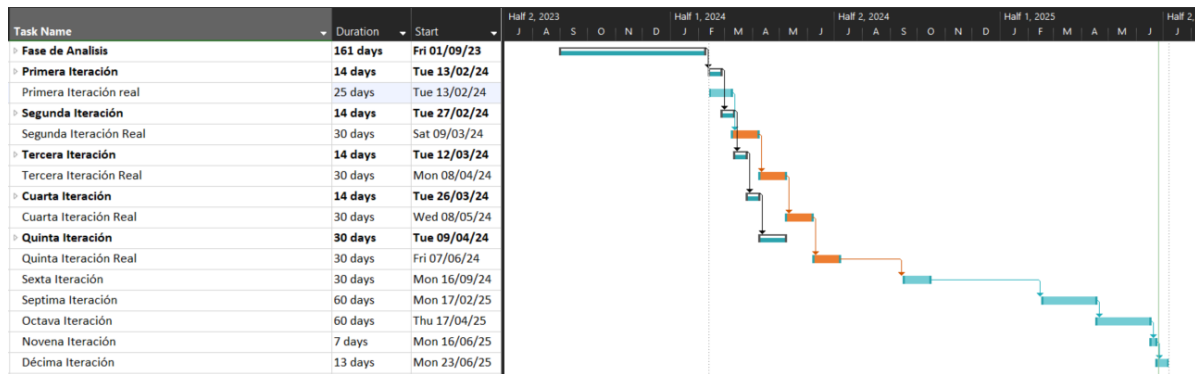


Figura 2.9: Gantt de la duración real del proyecto

2.2.7. Estimación de costes

A continuación, se describe el presupuesto del proyecto, centrándonos en las partes del desarrollador, el equipo y los recursos tecnológicos.

Costes de personal

En un equipo de desarrollo software, suele haber distintos profesionales con distintas habilidades y generalmente, centrados en un área concreta. En este caso, solo hay un único desarrollador que lleva todo el peso del proyecto.

Para un desarrollador de Android, el sueldo medio en 2025 es de unos 27.000€ anuales para un desarrollador junior de entre 1 y 3 años de experiencia [22], lo que supone un coste de unos 35.000€ a una empresa sumando los costes asociados a contingencias comunes, contingencias profesionales, prestaciones por desempleo, aportaciones al Fondo de Garantía Salarial (FOGASA) y formación. Esto conlleva un coste de unos 19€ la hora, lo que resultaría en un coste total de 5700€ para las 300 horas estimadas de duración del proyecto.

Costes de equipo

En cuanto el equipo a utilizar, se ha empleado un ordenador portátil valorado en unos 1000€ con una duración estimada de unos 5 años, con lo que el precio por mes es de unos 16.66€ al mes, con la estimación de 9 meses de duración del proyecto, da un coste de 149.4€ totales.

Costes de recursos tecnológicos

Respecto del software empleado para el desarrollo del proyecto, se adjuntan unas estimaciones del coste total:

- Android Studio: 0€
- Overleaf: 0€ en el caso del plan gratuito
- Microsoft Project: 11.28€/mes
- Figma: 0€ en el caso del plan gratuito
- Google Firebase: 0€ en el caso de no superar los límites de uso gratuito, coste variable en función del uso en caso de superarlo.
- Github: 0€ en el caso del plan gratuito

Coste total

Con los precios anteriores, se va a calcular el precio del proyecto para una duración de 300 horas en 9 meses en la siguiente tabla 2.11.

Debido al contexto de que este proyecto es un trabajo de fin de grado, los costes de personal se descartan y debido a que la universidad me proporciona una licencia de Microsoft Project de forma gratuita, vamos a calcular los costes reales del proyecto en la tabla 2.12 .

Elemento	Coste
Personal	5700,00€
Equipo	149,40€
Software	101,52€
Total	5950,92€

Cuadro 2.11: Costes totales del proyecto

Elemento	Coste
Personal	0,0 €
Equipo	149,4 €
Software	0,0 €
Total	149,40€

Cuadro 2.12: Costes reales del proyecto

Capítulo 3

Análisis

En este capítulo veremos el análisis acerca de las mejoras que hemos introducido en la aplicación, tanto de funcionamiento como de interfaz, además de las necesidades y propuestas de los usuarios.

3.1. Análisis de usuarios

Como hemos expuesto antes en el apartado de 2.1.1, emplearemos técnicas del desarrollo centrado en el usuario para identificar las principales necesidades de los usuarios e idear soluciones para las mismas. Con esto intentaremos establecer las bases a partir de las cuales diseñaremos los cambios en la aplicación para hacerla más intuitiva y agradable para los usuarios.

3.1.1. Proceso de investigación con usuarios

La investigación con usuarios fue realizada por mis tutores del TFG siguiendo el método de diseño centrado en el humano ideado por Pozdniakov [15]. De este proceso, yo recibí una serie de preguntas que ilustraban las necesidades que tenían los usuarios de la interfaz transformé en requisitos.

3.1.2. Actores principales del sistema: Definición de "personas"

Para este apartado, estableceremos un procedimiento mediante el cual, caracterizaremos a los usuarios como un arquetipo concreto atendiendo a sus objetivos y características. Para ello aplicamos la técnica de la descripción de "personas"[23], descripciones de usuarios ficticias, englobando sus habilidades, objetivos y patrones de comportamiento, así como el contexto y el entorno en el que el usuario opera. Esto tiene como fin facilitar que, en la fase de diseño, se permita concebir un producto que sirva a las necesidades de los usuarios en vez de a las del desarrollador.

3.2. TECNOLOGÍAS PARA EL FUNCIONAMIENTO SIN CONEXIÓN

En este sistema distinguimos dos tipos de usuarios claros, los organizadores y los participantes. Nuestro trabajo se centra principalmente en los primeros, así que son aquellos en los que nos centraremos.



SARA MÁRQUEZ

EDAD: 28
UBICACIÓN: VALLADOLID
PROFESORA DE ED. FÍSICA

SOBRE SARA MÁRQUEZ
Sara es profesora de Educación Física en un colegio de primaria de Valladolid. Está interesada en incorporar la tecnología en su docencia, pero no encuentra una aplicación que le permita integrarla de manera sencilla y aplicada a su campo. Además, cuando realiza actividades al aire libre, tiene que estar muy atenta para evitar que algún alumno se pierda, lo que complica la organización de estas.

OBJETIVOS

- Realizar AFMNs con sus alumnos
- Aprovechar su entorno cercano para realizar AFMNs
- Integrar la tecnología en su docencia

FRUSTRACIONES

- No dispone de una aplicación que usar con sus alumnos
- Le gustaría poder monitorizar a los alumnos en tiempo real
- Necesita una aplicación que permita el funcionamiento offline, ya que la mayoría de sus alumnos no disponen de un móvil con datos

CITAS

“Cuando realizo actividades al aire libre, siempre es un riesgo que le pase algo a un alumno, lo cual es una carga de responsabilidad importante”

“No encuentro una aplicación que me permita realizar actividades de manera sencilla”

Figura 3.1: Arquipo de Organizadora. Sara Márquez, profesora de Educación Física



JORGE RAMÍREZ

EDAD: 11
UBICACIÓN: VALLADOLID
ESTUDIANTE DE 5º DE PRIMARIA

SOBRE JORGE RAMÍREZ
Jorge es un estudiante de primaria y alumno de Sara. Le gusta jugar al fútbol con sus amigos y la clase de educación física. Le gusta especialmente cuando salen del centro a otro lugar para dar la clase. Ha realizado una actividad de orientación en el patio del colegio y le ha gustado, aunque se le ha hecho un poco simple. Utiliza ocasionalmente un móvil viejo de sus padres para jugar al Roblox y ver videos en Youtube.

OBJETIVOS

- Hacer más actividades fuera del centro
- Divertirse jugando con sus amigos

FRUSTRACIONES

- No dispone de un móvil con datos para jugar fuera de casa
- Le gustaría pasar más tiempo con sus amigos
- Le gusta más colaborar que competir

CITAS

“Cuando salimos del cole para hacer una actividad me lo paso muy bien con mis compañeros”

“Me gusta mucho la clase de educación física porque la profe es muy maja”

Figura 3.2: Arquipo de Participante. Jorge Ramírez, alumno de 5º de primaria

3.2. Tecnologías para el funcionamiento sin conexión

Uno de los principales objetivos del proyecto consiste en analizar y estudiar las distintas posibilidades existentes a la hora de transformar una aplicación pensada para funcionar con internet

de manera constante a una que permita un funcionamiento sin conexión a internet de manera continua (comúnmente denominado *offline* en inglés), a pesar de no contar con todas las funcionalidades que sí tiene cuando dispone de conexión a internet.

A continuación, se recoge una serie de tecnologías que se han investigado para implementar la funcionalidad sin conexión a internet, con las ventajas e inconvenientes que tiene cada una de ellas.

3.2.1. WorkManager

WorkManager es una API de Android que permite programar tareas asíncronas y diferibles que se ejecutan incluso si la aplicación se cierra o el dispositivo se reinicia [24]. Es el método recomendado para todo tipo de procesamiento en segundo plano. Permite la creación de tareas que se ejecuten en base a una serie de condiciones de manera asíncrona y sin la necesidad de actividad por parte del usuario de la aplicación. Tiene una amplia compatibilidad con versiones antiguas de Android y permite al desarrollador observar el estado y progreso de las tareas para poder crear flujos de trabajo que encadenen varias tareas de manera conjunta. Se emplearía de manera que se realizase la aplicación con datos guardados de manera local y se emplease esta funcionalidad para sincronizar los datos con la base de datos cuando se recupere la conexión.

3.2.2. Bluetooth 5.1

Bluetooth es una tecnología de conectividad *wireless* que permite comunicar varios dispositivos de manera remota. Esta tecnología está en todos los móviles modernos y su última especificación es Bluetooth 6.1. En este caso, nos interesa su versión 5.1 [25], introducida en enero de 2019, que introduce las funcionalidades de *Angle of Arrival* y *Angle of Departure*, que junto a la funcionalidad de *Low Energy Long Range* introducida en la versión 5.0, permite la localización de otros dispositivos de manera precisa en un rango amplio (unos 600 metros en condiciones normales) [26]. Esta tecnología ha sido empleada por Apple o Samsung en la creación de las Apple Tags o las Samsung Smart Tags respectivamente, que permiten localizar los objetos a los que están unidas. Con esta tecnología sería posible crear una red de dispositivos con internet que localizasen mediante el uso de bluetooth 5.1 al participante y fuesen actualizando su ubicación en tiempo real. La principal desventaja de esta tecnología es que sería necesario adquirir y desplegar los dispositivos, así como su mantenimiento de manera periódica, ya que mínimo es necesario recargar sus baterías.

3.2.3. Wi-Fi Hotspot

El Wi-Fi hotspot o punto de acceso Wi-Fi es una tecnología que permite la conexión entre varios dispositivos de tal manera que se pueda compartir información entre ellos, como fotos o videos [27], además, permite compartir una conexión a internet mediante conexión telefónica, comúnmente conocido como datos móviles o simplemente datos [28]. La capacidad de crear un punto de acceso Wi-Fi para compartir archivos se implementó en Android 8.0 Oreo, lanzado en agosto de 2017. Se podría emplear para que los dispositivos sin conexión de datos móviles se conecten a un dispositivo portátil dedicado a proporcionar este punto de conexión wifi de manera

puntual, o en su defecto un dispositivo móvil que proporcione esta conexión. Las principales desventajas de esta solución son la necesidad de adquirir un dispositivo portátil y una SIM con datos móviles que emplear para hacerlo funcionar, el límite de conexiones simultáneas (20 en la mayoría de los dispositivos dedicados, 10 en el caso de los dispositivos móviles), el rango limitado y la atenuación de la señal Wi-Fi con los problemas derivados de ambos y la necesidad de colocar el dispositivo en un lugar estratégico para proporcionar conexión en el mayor rango posible. Se podría crear una red de estos dispositivos para asegurar la cobertura de la red, pero aumentaría el costo de la solución, requeriría la configuración de los dispositivos para funcionar de manera integrada y sin necesidad de conectarse a cada uno de manera individual y el mantenimiento y colocación de los dispositivos.

3.2.4. Método empleado en el desarrollo de la funcionalidad *offline*

En conclusión, optando por la solución que no requiere ni un despliegue previo ni mantenimiento, se ha decidido emplear la API de WorkManager, debido a que no necesita un dispositivo externo y permite configurar la aplicación para funcionar de manera *offline* sin necesidad de que el usuario realice ninguna acción específica. Esto elimina una posible capa de complejidad derivada de la necesidad de realizar una configuración previa, ya sea conectándose a un dispositivo externo o realizando algún tipo de configuración previa a la realización de la actividad para poder realizarla sin conexión. Además, la capacidad de WorkManager para sincronizar los datos sin necesidad de tener la aplicación funcionando o si el móvil se reinicia, facilita que los datos lleguen al servidor ya que el usuario no necesita realizar ninguna acción.

3.3. Requisitos

3.3.1. Requisitos funcionales

Los requisitos funcionales se centran en lo que el sistema debe hacer y cómo debe comportarse en ciertas situaciones. Estos requisitos están diseñados para guiar el diseño, desarrollo y pruebas del sistema, y son esenciales para asegurar que el producto final cumpla con las expectativas y requisitos de los usuarios y las partes interesadas.

En nuestro caso, muchos de los requisitos funcionales se heredan de la versión previa de la aplicación [12] y se recogen aquí debido a que la nueva versión también habrá de cumplimentarlos.

Estos son los nuevos requisitos que la aplicación deberá de cumplimentar

- **RF-31:** El sistema deberá permitir a los organizadores conocer la posición de los participantes siempre y cuando estos dispongan de conexión a internet
- **RF-32:** El sistema deberá permitir a los organizadores comparar distintos recorridos durante la revisión de la actividad

Estos son los requisitos heredados de la versión anterior y que hemos de mantener

- **RF-01** El sistema deberá permitir a los usuarios registrarse en la aplicación mediante email y contraseña
- **RF-02:** El sistema deberá permitir a un usuario eliminar su perfil.
- **RF-03:** El sistema deberá permitir a un usuario editar sus datos de perfil (nombre, apellidos y foto).
- **RF-04:** El sistema deberá permitir a los usuarios restablecer su contraseña en caso de que la olviden.
- **RF-05:** El sistema deberá ser capaz de verificar que el correo de los usuarios es real.
- **RF-06:** El sistema deberá permitir a los usuarios programar actividades de orientación, escogiendo de entre un catálogo de actividades existentes.
- **RF-07:** Al programar una nueva actividad, el sistema permitirá al usuario organizador de la misma elegir si la actividad será score o lineal, y si estará permitido que los participantes soliciten a la aplicación ayuda con la ubicación.
- **RF-08:** El sistema deberá permitir a los usuarios inscribirse como participantes de una actividad, mediante unas claves proporcionadas por el organizador de la actividad.
- **RF-09:** El sistema deberá permitir al organizador de una actividad conocer los participantes inscritos en la misma.
- **RF-10:** El sistema deberá permitir a los organizadores de una actividad eliminar dicha actividad, de tal forma que la información deje de estar disponible también para los participantes de esta.
- **RF-11:** El sistema deberá contar con un catálogo de plantillas de actividad (4.4.1) a partir de las cuales el organizador pueda programar una actividad de ese tipo.
- **RF-12:** El sistema deberá ser capaz de detectar el paso de los participantes de las actividades por las balizas.
- **RF-13:** El sistema deberá ser capaz de determinar si una baliza tiene contenido educativo asociado, y en caso afirmativo, mostrárselo al usuario.
- **RF-14:** El sistema deberá ser capaz de notificar a los participantes de una actividad su paso por una baliza.
- **RF-15:** El sistema deberá permitir a los participantes de las actividades responder a los retos planteados en las balizas de una actividad.
- **RF-16:** El sistema deberá ser capaz de mostrar un feedback positivo al usuario en caso de que su respuesta a un reto sea correcta, o una corrección en caso de que la respuesta sea incorrecta.
- **RF-17:** El sistema deberá ser capaz de comprobar si un usuario se encuentra en el lugar de salida a la hora de poder iniciar su participación en una actividad, así como de detectar cuándo éste llega a la meta.

- **RF-18:** El sistema deberá mostrar a los participantes de las actividades un cronómetro que indique el tiempo que llevan participando.
- **RF-19:** Una vez acabada la actividad, el sistema deberá permitir a los participantes revisar sus respuestas dadas a los retos de las balizas, así como su recorrido seguido indicado sobre el mapa de orientación (track). El organizador de la actividad también podrá ver estos datos de cada participante.
- **RF-20:** Habiendo conexión a internet, el sistema deberá permitir al organizador de una actividad observar en tiempo real las balizas alcanzadas por cada participante, así como sus respuestas dadas a los retos asociados a las mismas.
- **RF-21:** El sistema deberá permitir a los participantes de una actividad visualizar el mapa de orientación de esa actividad y manipularlo (zoom-in, zoom-out y rotar).
- **RF-22:** El sistema deberá ser capaz de comprobar que un usuario ha comenzado su participación en una actividad como requisito para permitirle ver el mapa de orientación.
- **RF-23:** Una vez que una actividad haya terminado, el sistema deberá seguir permitiendo a los participantes ver el mapa de orientación si así lo desean.
- **RF-24:** El sistema deberá ser capaz de mostrar a los usuarios una descripción y las normas de aquellas actividades en las que participan.
- **RF-25:** El sistema deberá mostrar a los participantes de una actividad un resumen de su participación, en el que se incluyan la hora de inicio, la hora de finalización, el tiempo total y las balizas alcanzadas.
- **RF-26:** Mientras la aplicación esté realizando el seguimiento de la ubicación de un participante en una actividad, el sistema deberá ser capaz de mostrar una notificación que permita al usuario saber que la actividad está en curso y se está monitorizando su ubicación.
- **RF-27:** Si un participante abandona una actividad en curso (por ejemplo, porque cierra sin querer la aplicación), el sistema deberá permitirle retomarla siempre y cuando ésta no haya terminado.
- **RF-28:** El sistema deberá ser capaz de continuar con la monitorización de la ubicación del participante de una actividad, aunque éste se mueva por las distintas pantallas de la aplicación, minimice la aplicación o bloquee el dispositivo.
- **RF-29:** El sistema deberá permitir a un participante de una actividad, durante el transcurso de esta, conocer cuántas balizas ya ha alcanzado y cuántas le quedan por alcanzar.
- **RF-30:** El sistema deberá permitir usar un mapa de orientación para la actividad, sobre el que se indicará la ubicación de las balizas, y la posición del usuario si se solicita y está permitido por el organizador.

3.3.2. Requisitos de información

Los requisitos de información establecen qué datos, información y contenido son necesarios para el proyecto. Estos requisitos ayudan a definir la estructura y la gestión de la información. De manera similar, estos son heredados de la versión previa [12]. A continuación, se describen los requisitos de información identificados para este proyecto:

- **RI-01:** El sistema almacenará la siguiente información sobre los usuarios de la aplicación:
 - Nombre
 - Apellido/s
 - Correo electrónico
 - Contraseña
 - Imagen de usuario
- **RI-02:** Sobre la participación de un usuario en una actividad, el sistema almacenará la siguiente información:
 - Hora de comienzo
 - Hora de finalización
 - Balizas alcanzadas
 - Hora de llegada a cada baliza
 - Respuesta proporcionada al reto asociado a cada baliza
 - Historial de ubicaciones durante la actividad (track)
- **RI-03:** El sistema almacenará la siguiente información sobre las actividades:
 - Título de la actividad
 - Identificador para que los usuarios puedan encontrarla
 - Clave para que los usuarios puedan unirse a ella
 - Hora de comienzo
 - Hora de fin
 - Usuario organizador
 - Participantes
 - Plantilla a partir de la cual se ha programado
 - Si es score o lineal
 - Si permite solicitar ayuda con la ubicación o no
- **RI-04:** El sistema almacenará la siguiente información sobre las plantillas a partir de las cuales se programan las actividades:
 - Nombre de la plantilla
 - Tipo
 - Mapa

- Descripción
- Normas
- Balizas

- **RI-05:** El sistema almacenará la siguiente información sobre las balizas:

- Nombre
- Coordenadas geográficas
- Número relativo dentro de la actividad
- Imagen

En caso de que la baliza tenga asociada algún contenido educativo (reto) también almacenará la siguiente información:

- Texto
- Pregunta
- Respuesta correcta
- Posibles respuestas (solo si la pregunta es tipo test)

3.3.3. Requisitos no funcionales

Los requisitos no funcionales complementan los requisitos funcionales al describir características y restricciones que afectan la calidad, el rendimiento y otros aspectos del sistema. Estos requisitos se centran en *cómo* debe funcionar el sistema en lugar de *qué* debe hacer. Estos requisitos vienen heredados de la versión anterior de la aplicación y es importante mantenerlos. Estos son los requisitos no funcionales del proyecto:

- **RNF-01:** La aplicación se comunicará con un sistema de back-end, de donde obtendrá y a donde enviará los datos de las actividades.
- **RNF-02:** El sistema de back-end deberá permitir la integración con otra aplicación cliente en un futuro, que servirá para crear y configurar nuevas plantillas de actividades. Para ello, la información almacenada en su base de datos debe ser accesible a través de peticiones HTTP, mediante una REST API.
- **RNF-03:** El sistema de back-end deberá implementar reglas de seguridad que permitan establecer qué usuarios pueden acceder a cada tipo de información en la base de datos.
- **RNF-04:** El sistema deberá permitir ser generalizado a otros escenarios que no sean el Campo Grande de Valladolid (es donde se desarrollan todas las actividades diseñadas en este proyecto).

Requisitos de usabilidad

Los requisitos de usabilidad de un proyecto se centran en asegurar que el sistema o producto sea fácil de usar, eficiente y satisfactorio para los usuarios finales. Estos requisitos son fundamentales para garantizar que la experiencia del usuario sea positiva y que el sistema cumpla con las expectativas de usabilidad. Similar al resto de requisitos, muchos de estos son heredados de la versión anterior. Estas medidas se concretarán a la hora de realizar las pruebas de usabilidad teniendo en cuenta las características del prototipo.

- **RNF-11:** Los organizadores deben ser capaces de identificar de manera sencilla el progreso de los participantes en la actividad
- **RNF-12:** Los organizadores deben ser capaces de identificar inequívocamente a los participantes de la actividad
- **RNF-13:** El sistema deberá permitir a los participantes realizar la actividad sin conexión a internet y posteriormente enviar los resultados al servidor

Estos son los requisitos no funcionales heredados y que hemos de mantener

- **RNF-05:** Un usuario no experto debe ser capaz de programar con éxito una actividad por sí mismo/a.
- **RNF-06:** Un usuario que ya conoce la aplicación debe ser capaz de programar una actividad en poco tiempo.
- **RNF-07:** Un usuario no experto debe ser capaz de encontrar una actividad e inscribirse en ella exitosamente por sí mismo/a.
- **RNF-08:** Los usuarios deberán ser capaces de encontrar actividades e inscribirse en ellas en poco tiempo.
- **RNF-09:** Los participantes de una actividad deben ser capaces de reconocer los elementos del entorno que actúan como baliza la mayor parte de las veces.
- **RNF-10:** Los usuarios deben ser capaces de encontrar por sí mismos la pantalla de la aplicación que les permite dar respuesta a los retos de las balizas durante las actividades.

3.4. Casos de uso

Los casos de uso son los escenarios detallados que describen cómo se interactúa con la aplicación desde la navegación hasta las actividades educativas. Debido a que los usuarios se pueden comportar tanto como organizadores como participantes, es importante diferenciar los casos de uso entre ambos. Para ello he dividido los mismos en tres gráficos (ver figuras 3.3, 3.4, 3.5), uno para cada rol, así como uno común a ambos. Similar a los requisitos, estos son heredados de la versión anterior [12].

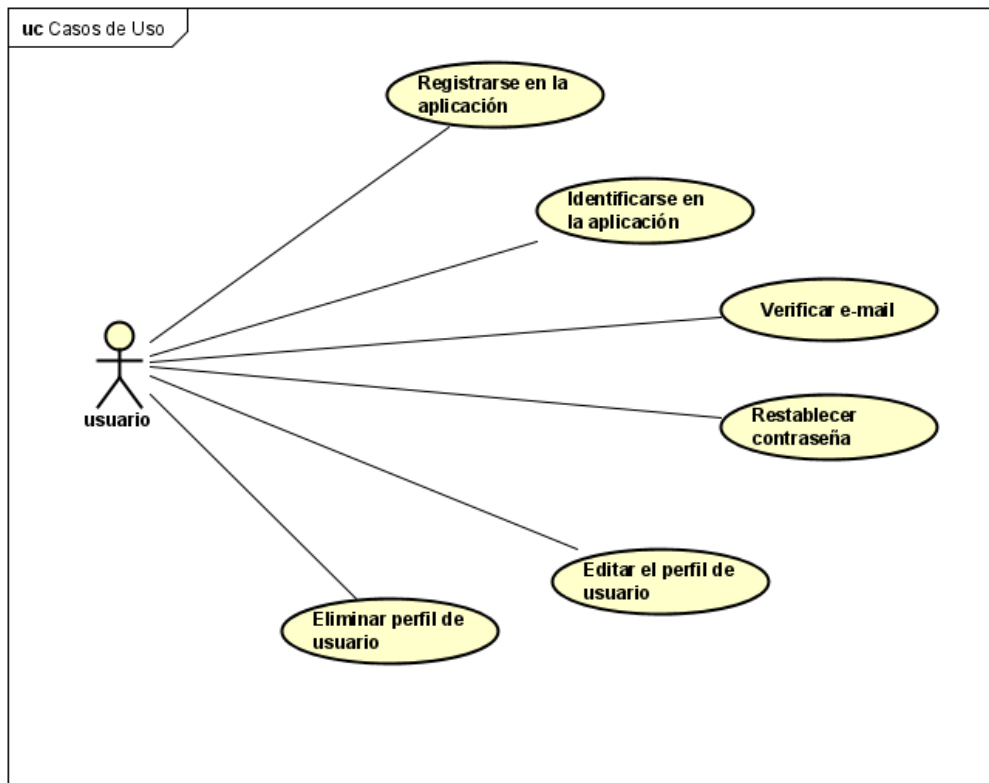


Figura 3.3: Casos de uso comunes a todos los usuarios

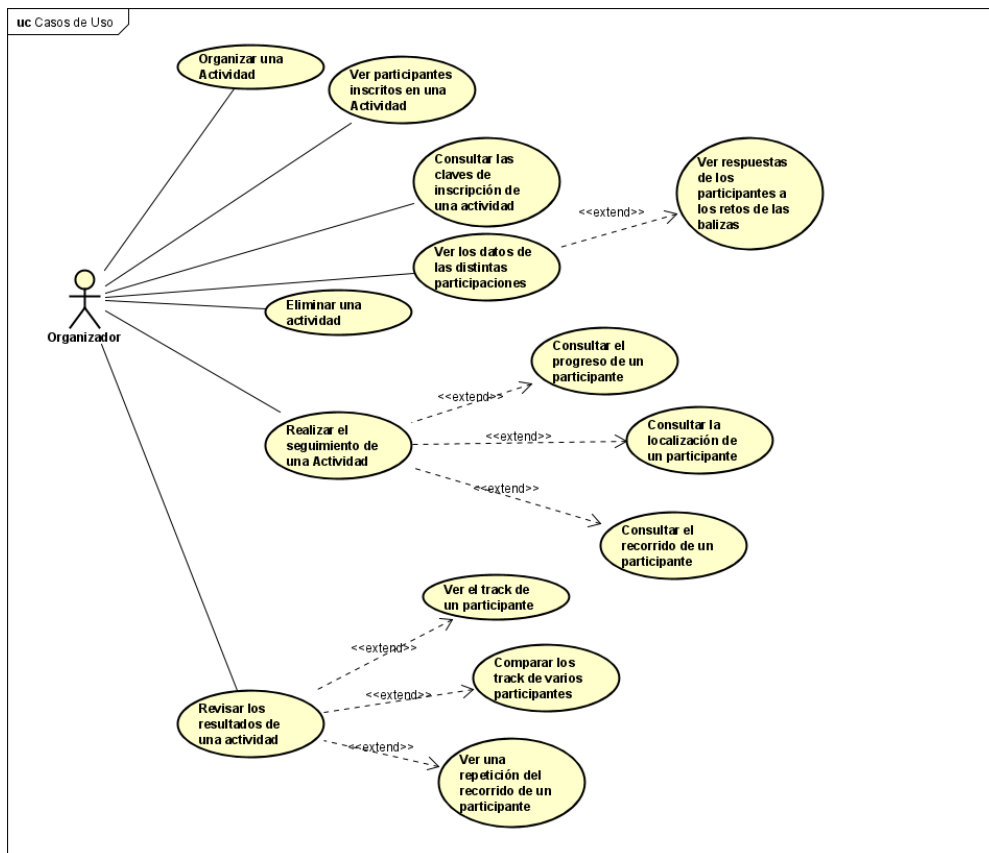


Figura 3.4: Casos de uso del organizador

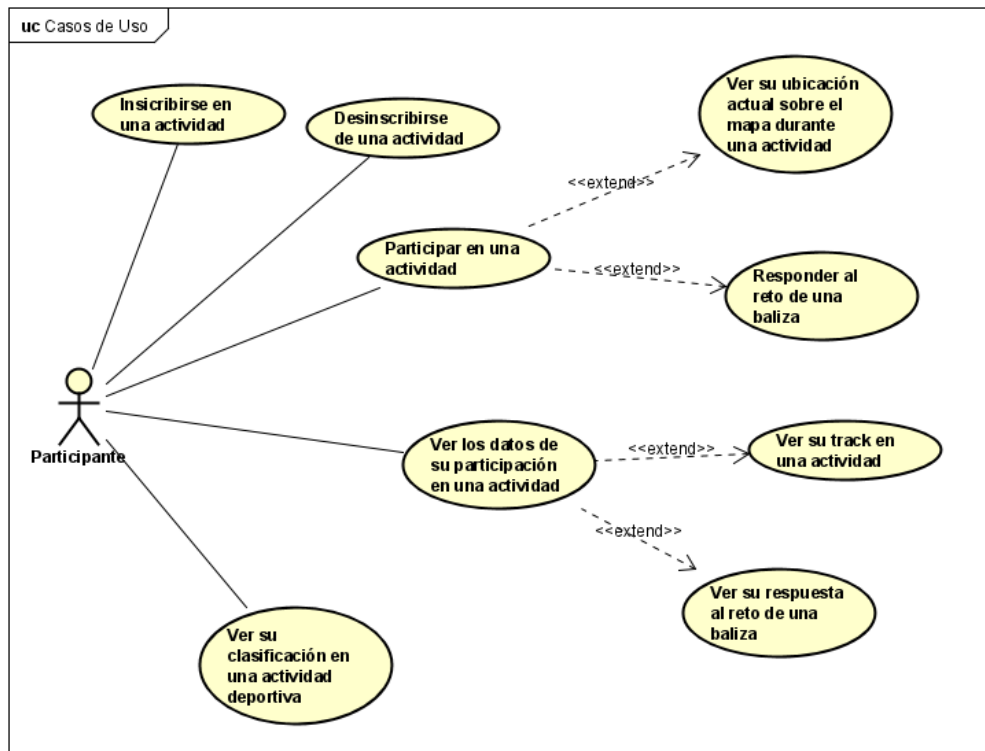


Figura 3.5: Casos de uso del participante

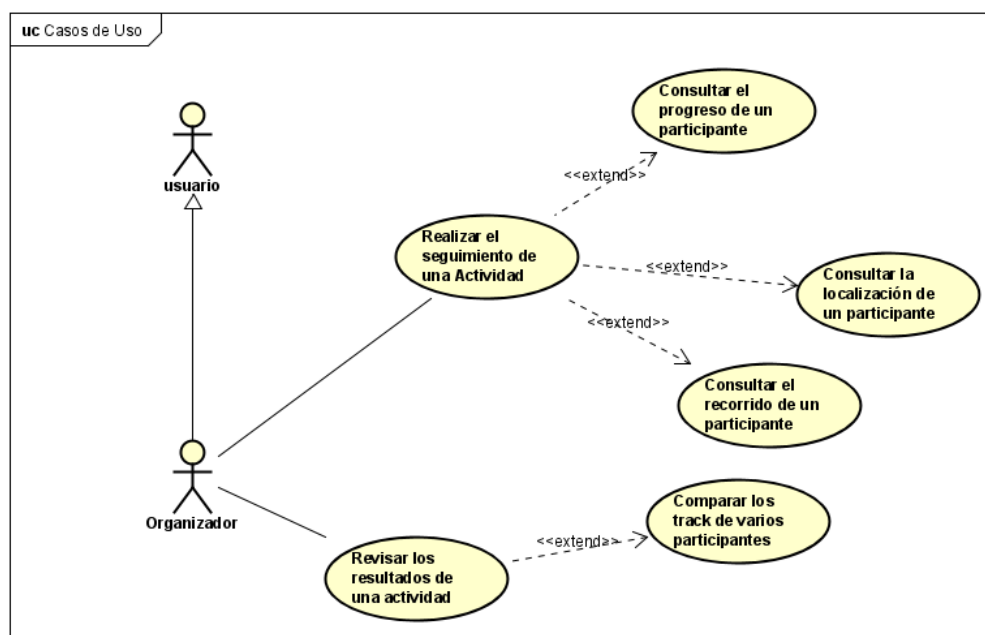


Figura 3.6: Casos de uso abordados en este TFG

En este caso es importante describir los casos de uso cuyo desarrollo es relevante para las modificaciones a implementar por esta versión referenciados en la Figura 3.6. Estos se harán teniendo en cuenta los requisitos anteriormente descritos. Los casos de uso a modificar se recogen en las tablas siguientes, donde se exponen el caso de uso, los actores que lo realizan, si hay alguna condición previa a iniciar el caso de uso, si debe de cumplirse alguna condición al completarse

3.4. CASOS DE USO

el caso de uso para considerar este completo, una secuencia base, secuencias alternativas que se pueden dar mientras se realiza el caso de uso y casos de uso secundarios que están relacionados con el caso de uso principal.

Caso de uso: Realizar el seguimiento de una actividad	
Resumen	El organizador quiere conocer algún tipo de información acerca del estado de los participantes.
Actor	Organizador
Precondición	El organizador deberá estar identificado y organizar una actividad en curso con algún participante realizando la actividad.
Secuencia Base	
[1] El organizador solicita la lista de actividades [2] El sistema muestra la lista de actividades [3] El organizador selecciona la actividad [4] El sistema mostrará la información de la actividad, como hora de inicio, hora de finalización, la lista de participantes, una breve descripción e información adicional en caso de existir [5] El sistema queda a la espera de qué información quiere consultar el organizador [6] El organizador selecciona el participante que quiere consultar	
Secuencia Alternativa	
[6.a.1] El organizador solicita consultar el progreso de un participante [6.a.2] El sistema inicia el caso de uso consultar el progreso de un participante [6.b.1] El organizador solicita consultar la localización de un participante [6.b.2] El sistema inicia el caso de uso consultar la localización de un participante [6.c.1] El organizador solicita consultar el recorrido de un participante [6.c.2] El sistema inicia el caso de uso consultar el recorrido de un participante [6.d.1] El organizador cancela, el caso de uso queda sin efecto	
Secuencia en caso de excepción	
No se contemplan excepciones	
Casos de uso secundarios	Consultar el progreso de un participante Consultar el recorrido de un participante Consultar la localización de un participante
Notas	

Cuadro 3.1: Caso de uso: Realizar el seguimiento de una actividad

Caso de uso: Consultar el progreso de un participante	
Resumen	El organizador quiere conocer cuál es el progreso de un participante en la actividad. Esto incluye su estado general, el número de balizas encontradas y el resultado de las preguntas de estas.
Actor	Organizador
Precondición	El organizador ha solicitado consultar el progreso de dicho participante. El participante está realizando la actividad.
Postcondición	
Secuencia Base	
[1] El sistema mostrará la información del participante, como estado de su participación (sin iniciar, en progreso, o finalizado), progreso en las balizas (sin encontrar, encontrada, respondida correcta o respondida incorrecta) y resultado de las mismas [2] El sistema solicita al usuario la siguiente acción	
Secuencia Alternativa	
[2.a.1] Si el organizador solicita conocer más información sobre las balizas [2.a.2] El sistema mostrará más información acerca de las balizas, como la respuesta concreta [2.b.1] El organizador cancela, el caso de uso termina	
Secuencia en caso de excepción	
No se contemplan excepciones	
Notas	

Cuadro 3.2: Caso de uso: Consultar el progreso de un participante

Caso de uso: Consultar el recorrido de un participante	
Resumen	El organizador quiere conocer cuál es el recorrido de un participante en la actividad.
Actor	Organizador
Precondición	El organizador ha solicitado consultar el recorrido de dicho participante. El participante está realizando la actividad.
Postcondición	
Secuencia Base	
[1] El sistema mostrará el recorrido del participante [2] El sistema solicita al usuario la siguiente acción	
Secuencia Alternativa	
[2.a.1] El organizador cancela, el caso de uso termina	
Secuencia en caso de excepción	
No se contemplan excepciones	
Notas	

Cuadro 3.3: Caso de uso: Consultar el recorrido de un participante

Caso de uso: Consultar la localización de un participante	
Resumen	El organizador solicita conocer cuál es la localización de un participante en la actividad.
Actor	Organizador
Precondición	El organizador ha solicitado consultar la localización de dicho participante. El participante está realizando la actividad.
Postcondición	
Secuencia Base	
[1] El sistema mostrará el mapa y la localización de los participantes	
[2] El sistema solicita al usuario la siguiente acción	
Secuencia Alternativa	
[2.a.1] El organizador cancela, el caso de uso termina	
Secuencia en caso de excepción	
No se contemplan excepciones	
Notas	

Cuadro 3.4: Caso de uso: Consultar la localización de un participante

Caso de uso: Revisar los resultados de una actividad	
Resumen	El organizador quiere conocer algún tipo de información acerca del resultado de una actividad.
Actor	Organizador
Precondición	El organizador deberá estar identificado y tener una actividad finalizada.
Secuencia Base	
[1] El organizador solicita la lista de actividades [2] El sistema muestra la lista de actividades [3] El organizador selecciona la actividad [4] El sistema mostrará la información de la actividad, como hora de inicio, hora de finalización, la lista de participantes, una breve descripción e información adicional en caso de existir [5] El sistema queda a la espera de qué información quiere consultar el organizador [6] El organizador selecciona la información que quiere consultar	
Secuencia Alternativa	
[6.a.1] 2] El organizador selecciona los participantes deseados [6.a.2] El organizador solicita comparar el recorrido de varios participantes [6.a.3] El sistema inicia el caso de uso comparar los recorridos de varios participantes [6.b.1] El organizador cancela, el caso de uso queda sin efecto	
Secuencia en caso de excepción	
No se contemplan excepciones	
Casos de uso secundarios	Comparar los recorridos de varios participantes
Notas	

Cuadro 3.5: Caso de uso: Revisar los resultados de una actividad

Caso de uso: Comparar los recorridos de varios participantes	
Resumen	El organizador solicita comparar los recorridos de varios participantes.
Actor	Organizador
Precondición	El organizador ha solicitado comparar la localización de distintos participantes en una actividad finalizada
Postcondición	
Secuencia Base	
[1] El sistema mostrará el mapa con los distintos recorridos de cada uno de los participantes	
[2] El sistema solicita al usuario la siguiente acción	
Secuencia Alternativa	
[2.a.1] El organizador cancela, el caso de uso termina	
Secuencia en caso de excepción	
No se contemplan excepciones	
Notas	

Cuadro 3.6: Caso de uso: Comparar los recorridos de varios participantes

3.5. Modelo de dominio

En esta sección se detalla el diseño conceptual del sistema, incluyendo las entidades clave y sus relaciones. El modelo refleja la estructura lógica del negocio, garantizando una representación fiel de sus procesos y reglas. Este modelo de dominio es heredado de la versión anterior. Se explican brevemente las abstracciones incluidas.

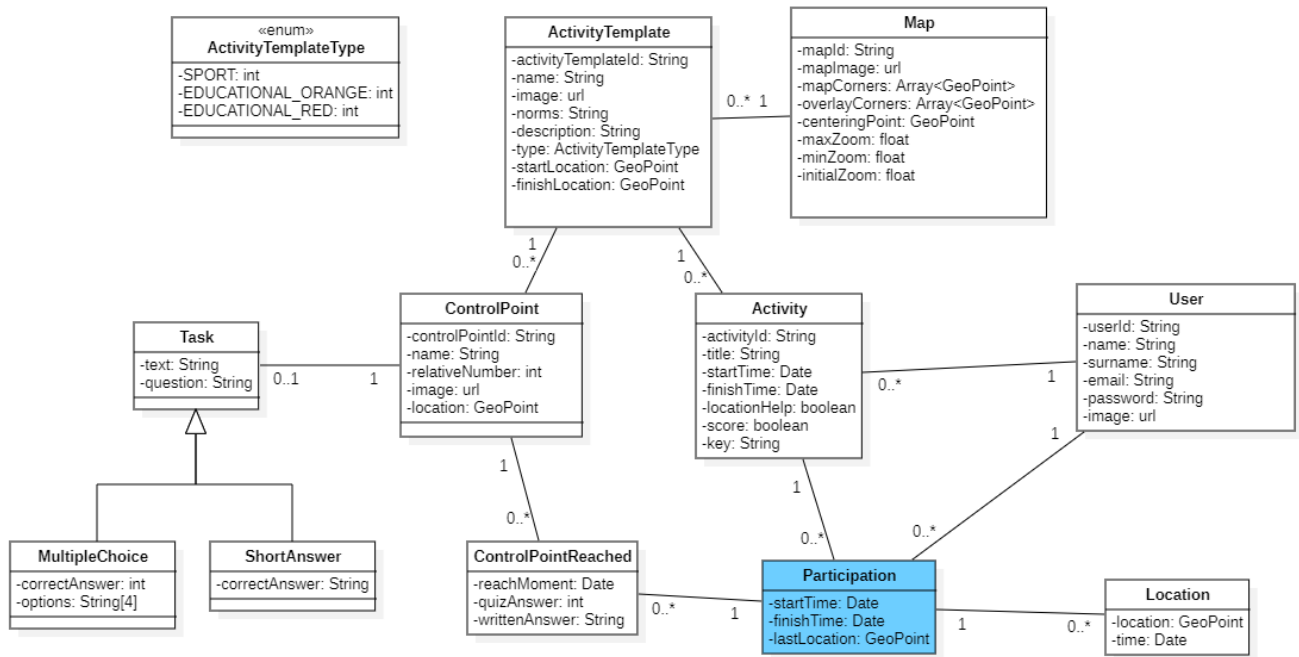


Figura 3.7: Modelo de dominio, indicándose en azul la clase en la que se han introducido modificaciones en el presente TFG con respecto a la aplicación original

- **ActivityTemplate.** Esta clase representa la plantilla de la actividad a partir de la cual los organizadores crean las suyas propias. Encapsulan la información general de la actividad y permite crear muchas actividades a partir de las mismas. Se relaciona con la clase Map para referenciar el mapa a usar en la actividad, con la clase ControlPoint para recoger las balizas de la actividad y con las múltiples actividades creadas a partir de la template.
- **Map.** La clase Map del modelo contiene toda la información relativa al mapa de la actividad. Contiene información acerca del mapa, como el punto de centrado y las esquinas del mismo, la imagen superpuesta con el mapa de orientación, y la posición de la misma.
- **ControlPoint.** Los ControlPoint representan las balizas (puntos por los que tienen que pasar los participantes). Contienen información como su localización, su nombre o la imagen de las mismas. Se relacionan con las tareas de la baliza (En el caso de que esta tenga tarea) y con la clase ControlPointReached, que representa las balizas que ha alcanzado el participante.
- **Task.** Esta clase representa la posible prueba que se tenga que realizar al llegar a una baliza. Contiene el texto introductorio a la pregunta y la pregunta en sí. Es una generalización de dos clases. Una representa las preguntas de respuesta múltiple (*MultipleChoice*) y otra las preguntas de respuesta corta *ShortAnswer*.
- **ControlPointReached.** Esta clase define el hecho de pasar por una baliza. Recoge el momento temporal en el que se llega a la baliza, así como la respuesta del usuario en caso de que esta tuviese una prueba complementaria.
- **Activity.** Esta clase simboliza la instancia de una actividad a partir de una plantilla. Recoge información como el título, cuando inicia y finaliza, si se puede pedir ayuda con la localización

o si la modalidad es *SCORE*. Se relaciona con la clase *Participation* para representar los participantes de la actividad.

- **User.** Esta clase representa toda la información del usuario. Se incluye información como el nombre, el email o la imagen del usuario. Se relaciona con la clase *Activity* para representar las actividades creadas como organizador y con la clase *Participation* para representar las actividades en las que eres un participante.

- **Participation.** Esta clase es una abstracción de la participación de un usuario en una actividad. Recoge el tiempo de inicio y de fin, así como la última localización conocida del usuario. Este es el único cambio del modelo de dominio y se utiliza para representar la posición del participante en el mapa del organizador. Se relaciona con la clase *location* para representar todas las ubicaciones que forman el recorrido del participante.

- **Location.** Esta clase recoge todas las localizaciones del usuario durante la actividad y se emplea para representar el recorrido del usuario a posteriori durante la revisión.

3.5.1. Realización en análisis de los casos de uso

En este apartado se mostrarán los diagramas de secuencia obtenidos del análisis a partir de los casos de uso mencionados anteriormente.

Se han excluido de esta sección las secuencias alternativas de cancelar a fin de simplificar los diagramas.

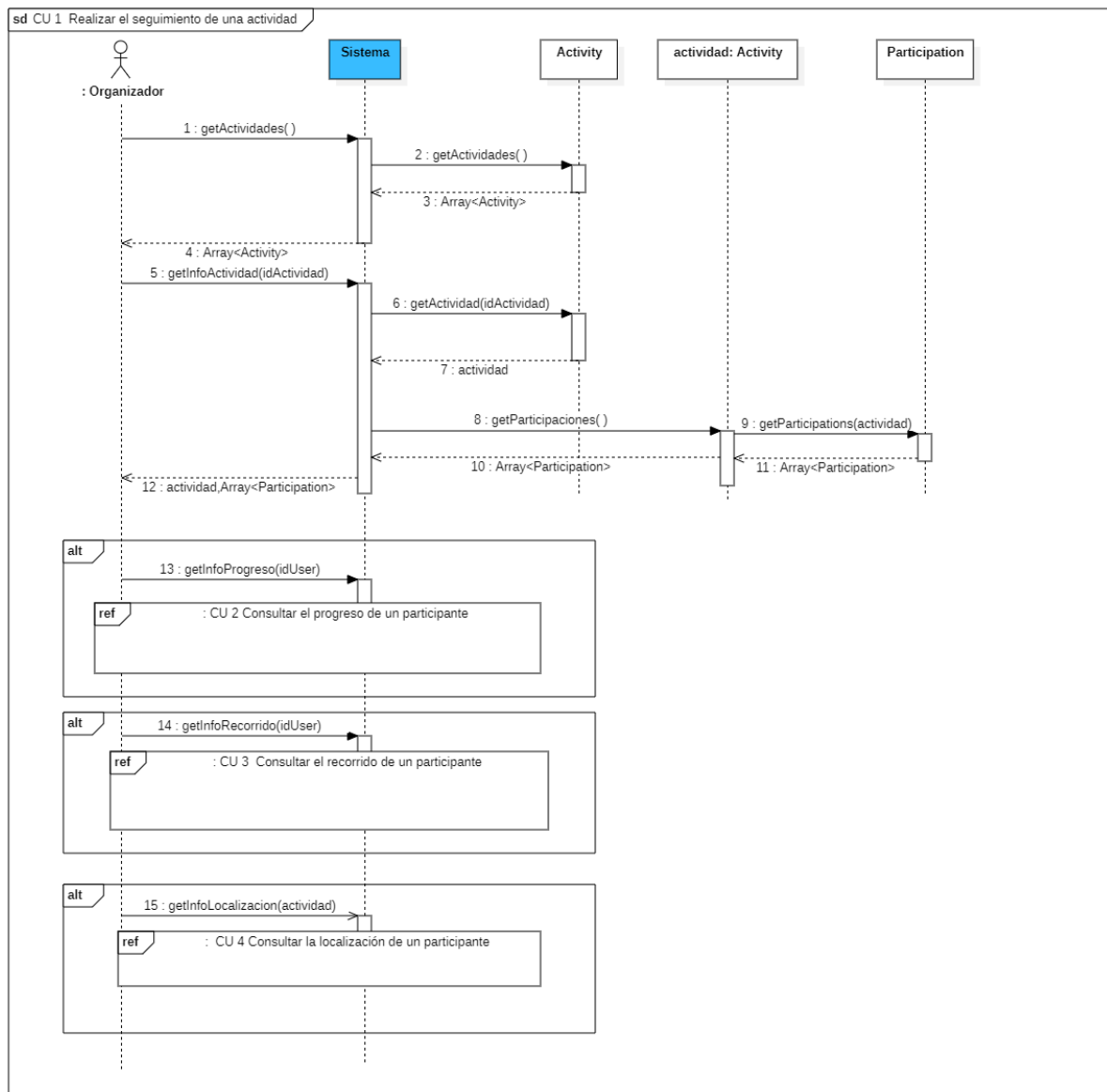


Figura 3.8: Diagrama Secuencia CU1: Realizar el seguimiento de una actividad

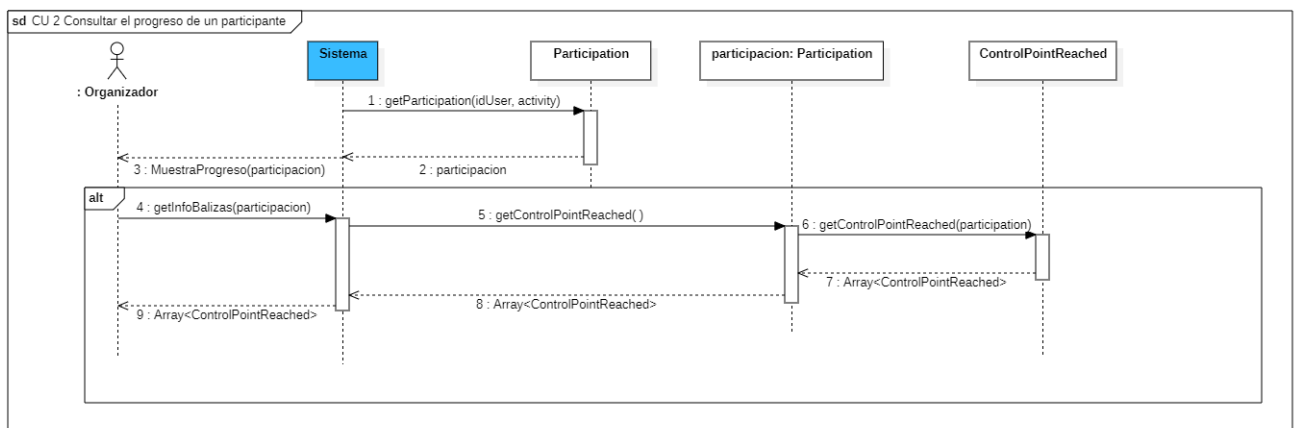


Figura 3.9: Diagrama Secuencia CU2: Consultar el progreso de un participante

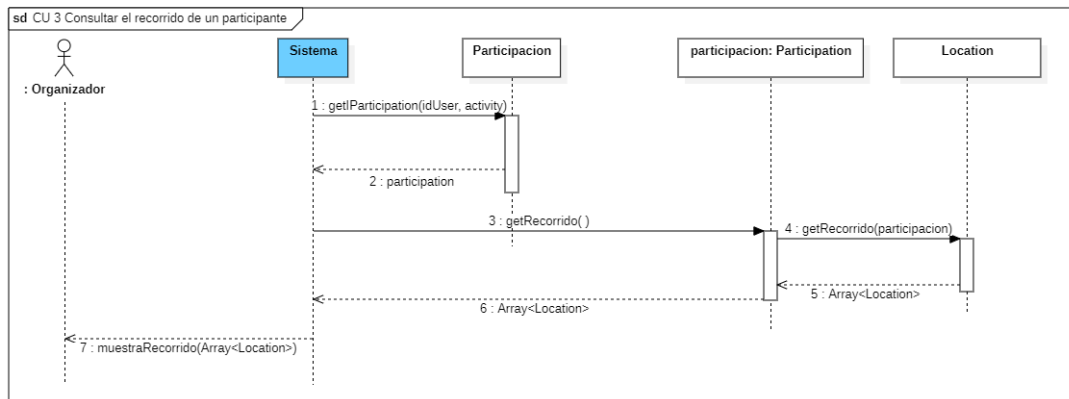


Figura 3.10: Diagrama Secuencia CU3: Consultar el recorrido de un participante

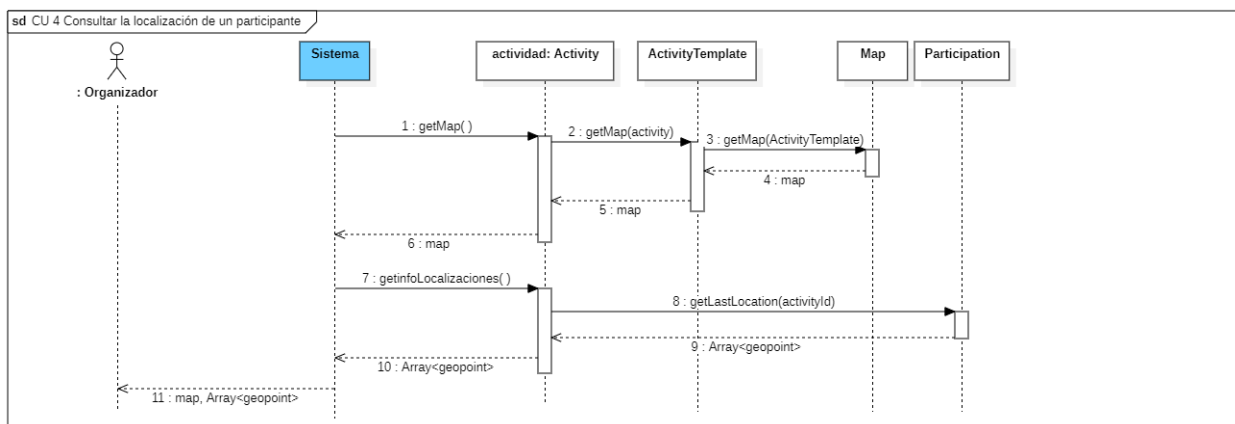


Figura 3.11: Diagrama Secuencia CU4: Consultar la localización de un participante

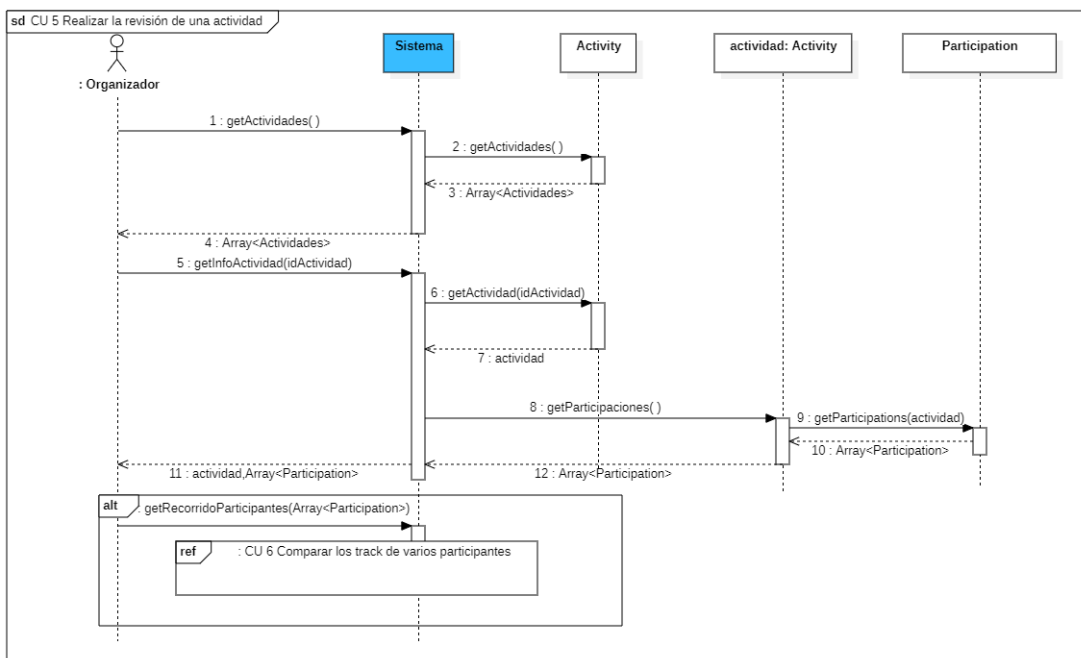


Figura 3.12: Diagrama Secuencia CU5: Revisar los resultados de una actividad

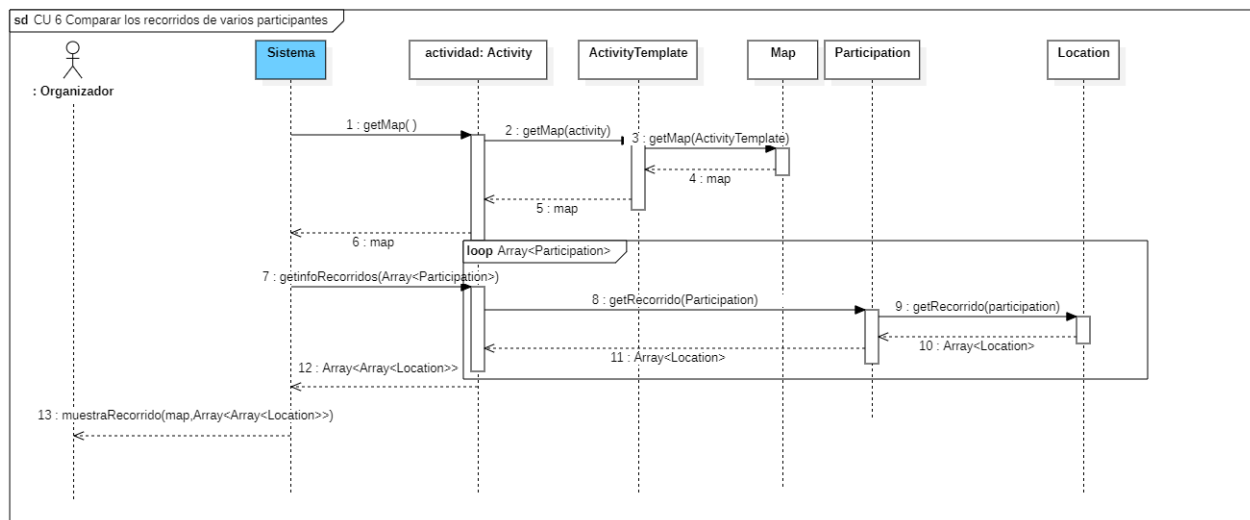


Figura 3.13: Diagrama Secuencia CU6: Comparar los recorridos de varios participantes

3.5.2. Modelo de dominio detallado

Con el modelo de dominio y los diagramas de secuencia anteriores, construimos un modelo de dominio extendido (Figura 3.14) con los métodos a implementar.

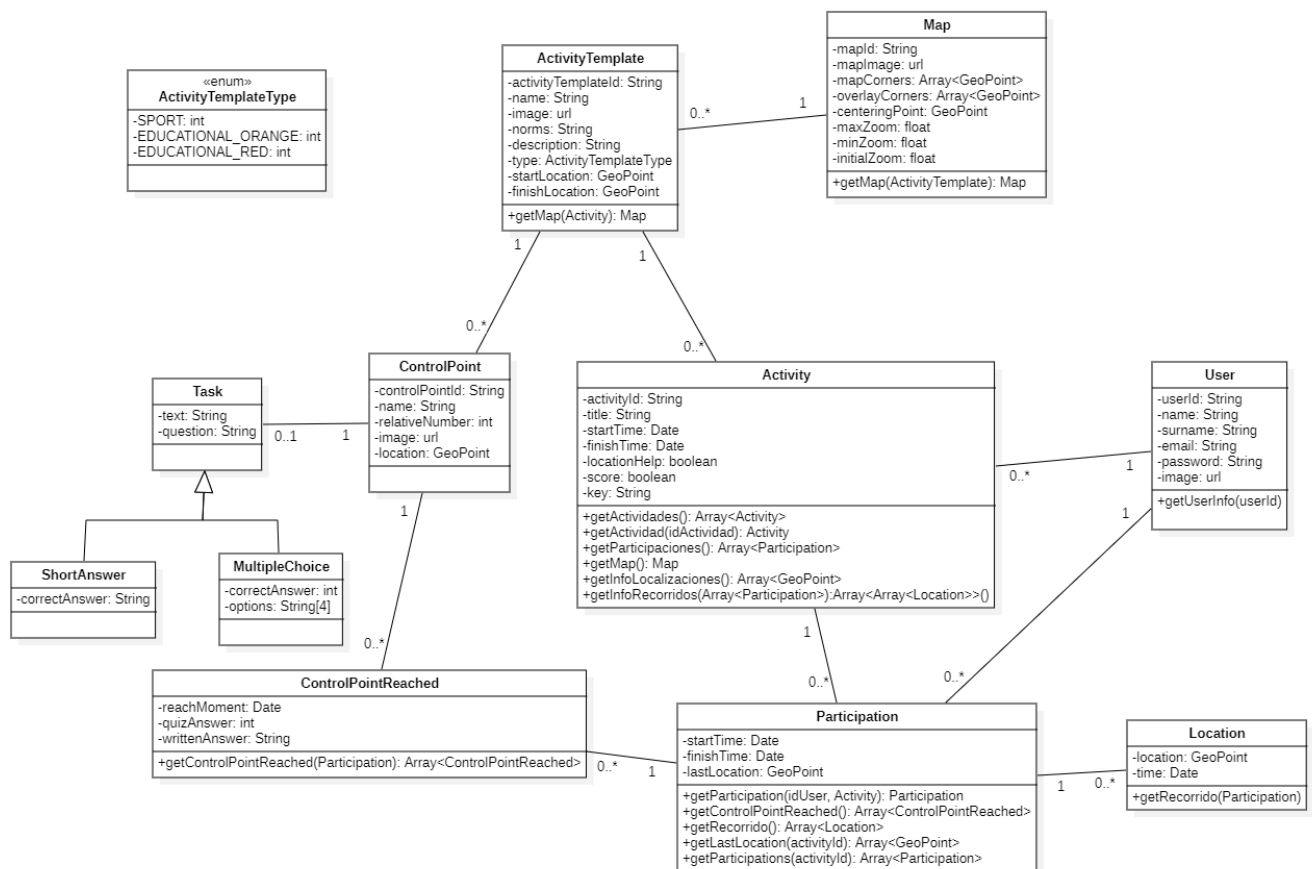


Figura 3.14: Modelo de dominio extendido

Capítulo 4

Diseño

En este apartado describiremos las principales guías de diseño empleadas, las pruebas realizadas con usuarios y las distintas iteraciones de la aplicación.

4.1. Guías de diseño aplicadas

Aquí se detallan las distintas guías de diseño empleadas

4.1.1. Heurísticas de Nielsen

Las heurísticas de Nielsen [29] son un conjunto de principios heurísticos utilizados para evaluar la usabilidad de interfaces de usuario y sistemas interactivos. Estas heurísticas proporcionan una lista de pautas generales que los diseñadores y evaluadores pueden utilizar para identificar problemas comunes que afectan a la experiencia del usuario. Las heurísticas de Nielsen son diez en total:

1. **Visibilidad del estado del sistema:** El sistema debe mantener informados a los usuarios sobre lo que está sucediendo a través de retroalimentación adecuada en tiempo real. Los usuarios deben estar al tanto del estado actual del sistema y de cualquier acción que hayan realizado.
2. **Coincidencia entre el sistema y el mundo real:** El lenguaje, los términos y las convenciones utilizadas en la interfaz deben reflejar el mundo real del usuario, lo que facilita la comprensión y el uso intuitivo del sistema.
3. **Control y libertad del usuario:** Los usuarios deben tener la posibilidad de deshacer acciones no deseadas y de salir de situaciones no deseadas sin dificultad o penalizaciones.
4. **Consistencia y estándares:** Los elementos de la interfaz y las acciones deben seguir patrones y convenciones consistentes, tanto dentro del sistema como con respecto a estándares ampliamente aceptados.

5. **Prevención de errores:** El diseño debe anticipar y evitar errores en la medida de lo posible. Si se producen errores, el sistema debe proporcionar mensajes claros y descriptivos para guiar a los usuarios hacia una solución.
6. **Reconocimiento en lugar de recuerdo:** Las opciones y acciones importantes deben ser visibles y estar disponibles para los usuarios, evitando que los usuarios tengan que recordar información o rutas específicas.
7. **Flexibilidad y eficiencia de uso:** El diseño debe permitir tanto a los usuarios novatos como a los usuarios experimentados realizar tareas de manera rápida y eficiente, brindando atajos y métodos para agilizar el flujo de trabajo.
8. **Diseño estético y minimalista:** La interfaz debe ser estéticamente agradable, pero sin exceso de elementos innecesarios que puedan distraer o confundir a los usuarios.
9. **Ayuda y documentación:** Cuando sea necesario, el sistema debe proporcionar ayuda y documentación para guiar a los usuarios a través de tareas complejas o desconocidas.
10. **Ayudar a los usuarios a reconocer, diagnosticar y recuperarse de los errores:** Los mensajes de error deben estar en lenguaje sencillo (sin códigos), indicar precisamente el problema y sugerir una solución constructiva.

4.1.2. Material Design

Material Design 2 es la segunda versión de la filosofía de diseño desarrollada por Google, conocida como Material Design [30]. Esta filosofía de diseño fue presentada inicialmente en 2014 y se basa en principios visuales y de interacción que buscan crear una experiencia de usuario coherente y atractiva en diversas plataformas y dispositivos. Algunas de las características y enfoques clave de Material Design 2 son los siguientes:

1. **Mayor énfasis en la personalización:** Material Design 2 se centra más en permitir que los diseñadores adapten y personalicen el diseño para satisfacer las necesidades específicas de sus productos y audiencias. Proporciona una mayor flexibilidad para la creación de diseños únicos y distintivos.
2. **Animaciones y transiciones mejoradas:** El diseño original de Material Design ya enfatizaba las animaciones y transiciones fluidas para mejorar la experiencia del usuario. Material Design 2 lleva esto un paso más allá, con una mayor atención a los detalles y la suavidad de las animaciones, lo que proporciona una sensación más natural y agradable en las interacciones.
3. **Enfoque en la accesibilidad:** Material Design 2 pone una mayor importancia en la accesibilidad y la inclusión. Se proporcionan directrices para garantizar que los productos y servicios creados sigan pautas de accesibilidad y que sean utilizables para personas con discapacidades.
4. **Consistencia multiplataforma:** Material Design 2 está diseñado para ser coherente en múltiples plataformas, incluidas aplicaciones móviles, sitios web y aplicaciones de escritorio. Esto permite que la experiencia del usuario sea más uniforme independientemente, del dispositivo que estén utilizando.

5. **Integración con nuevas tecnologías:** Material Design 2 se adapta para aprovechar las nuevas tecnologías emergentes, como interfaces de voz, realidad virtual y aumentada, y dispositivos con pantallas plegables, brindando una experiencia de usuario mejorada y adaptada a estas nuevas posibilidades.
6. **Actualizaciones continuas:** Al igual que la primera versión, Material Design 2 es una filosofía de diseño en constante evolución. Google sigue mejorando y actualizando las directrices y componentes para reflejar los avances en diseño, tecnología y retroalimentación de los usuarios.

4.2. Proceso de diseño iterativo

Dentro de un proceso de diseño iterativo con la metodología del diseño centrado en el usuario, los prototipos y las pruebas de usabilidad son muy importantes para validar las ideas iniciales. Estos nos permiten que los usuarios interactúen con una versión temprana del producto y nos proporcionen retroalimentación acerca de nuestras decisiones de diseño. A continuación, se detallan las etapas del proceso:

4.2.1. Interfaz ya existente

La versión anterior de la aplicación ya cumplía con sus funciones básicas, como crear actividades y participar en ellas. Los usuarios ya estaban familiarizados con ella, pero les gustaría añadir algunas funcionalidades extra, así como mejoras en la interfaz. Previamente a explicar el proceso de mejora de la aplicación, voy a exponer cómo se encontraba la interfaz de la aplicación. Para empezar, esta es la portada de la aplicación (Figura 4.1.a), que nos permite ver las claves de la misma, visualizar el mapa, nos ofrece un resumen de la información de la actividad, así como una breve descripción de la actividad y las normas de la misma. Para finalizar, tenemos un botón que nos permite acceder a la lista de participantes. Tras pulsar el botón, accedemos a la lista de participantes (Figura 4.1.b) donde encontramos una lista con todos los participantes, su nombre, la hora de salida y la de llegada en caso de que se hayan dado estos eventos. Al pulsar en un participante, accedemos a la lista de balizas alcanzadas (Figura 4.1.c). Estas se colorean según si la respuesta es correcta o errónea, manteniéndose grises en caso de que no haya contestado aun a la pregunta de la baliza. También disponemos de un botón para consultar el recorrido de un participante.

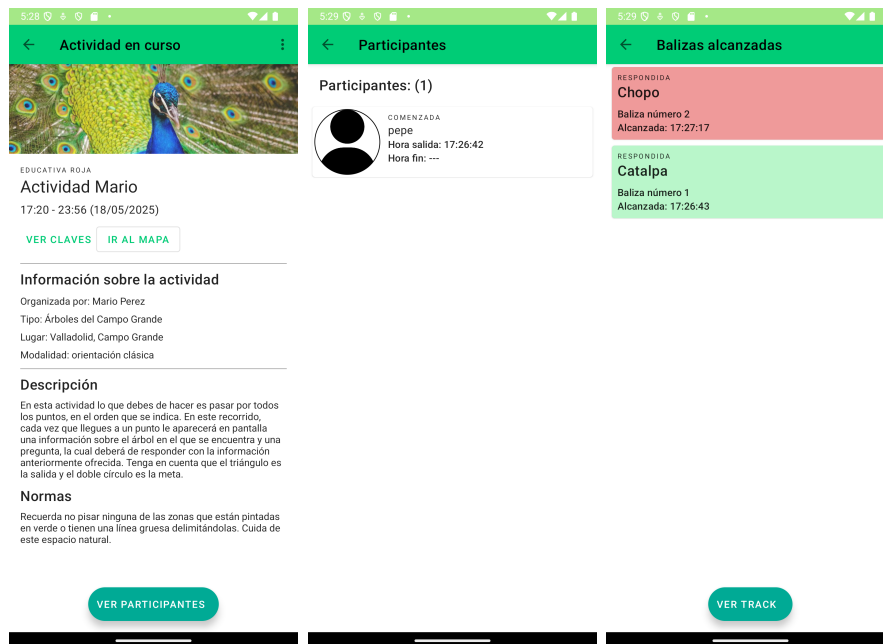


Figura 4.1: Portada de la actividad, lista de participantes y lista de balizas alcanzadas por un participante

Si seleccionamos una de las balizas, nos mostrarán la pregunta propuesta en la misma (Figura 4.2.a), así como la respuesta del participante, mostrando si es correcta o no. Si en la vista del participante pulsamos el botón de "Ver Track" nos llevará a una pantalla con el mapa de la actividad (Figura 4.2.b) en el que podremos deslizar la línea de tiempo para ir viendo su localización en cada momento, así como un interruptor para mostrar todo su recorrido completo.

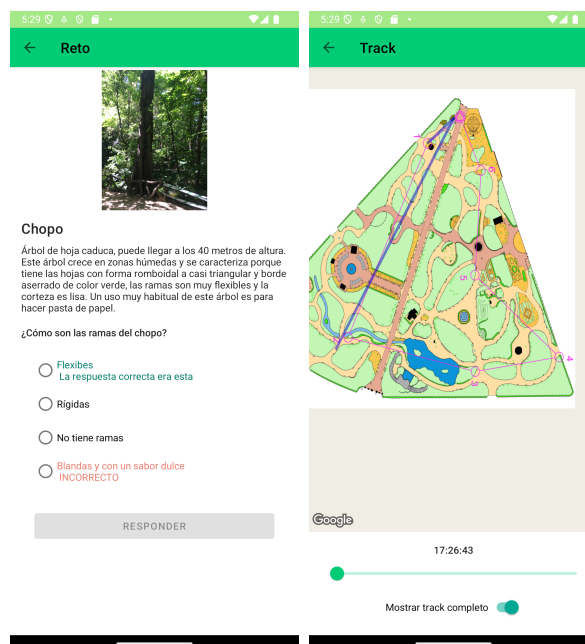


Figura 4.2: Pregunta de una baliza

4.2.2. Diseño inicial

Primeros bocetos de la interfaz

Partiendo de los casos de uso descritos en el apartado 3.6, se comenzó a prototipar una interfaz con Balsamiq Wireframes¹ creando una serie de *mock-ups* destinada a implementar los mismos como un prototipo de baja fidelidad en Uizard. También han de ajustarse a los requisitos no funcionales y de información planteados en el apartado 3.3.

Vistas de información de los participantes

Para las vistas con información de los participantes, es importante que sean muy eficiente en mostrar toda la información posible. También es importante que presente la información de manera clara y ordenada sin llegar a saturar al organizador. Si bien existía ya una noción de este concepto en forma de una lista con las fotos de los participantes y una breve información acerca de su participación en la actividad (ver 4.1.b), los usuarios nos han pedido rediseñar y ampliar esta parte con el objetivo de facilitar el seguimiento de los participantes por parte del organizador durante la monitorización de la actividad. Debido a esto, he diseñado dos vistas para mostrar la información de los participantes, una con un formato de tarjetas (ver Figura 4.3) y otra de lista (ver Figura 4.4).

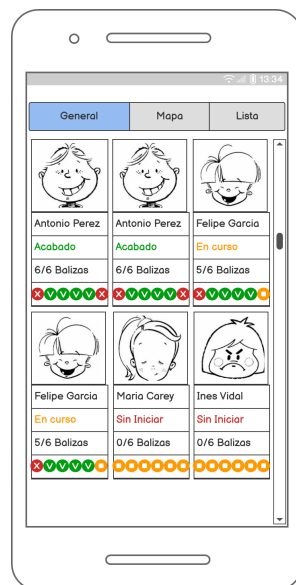


Figura 4.3: Vista de tarjetas

Para la vista de tarjetas (Figura 4.3), he seguido la guía de material design [31] que propone que, para volúmenes de datos con jerarquías y distintos tipos de información, como imágenes y texto, utilizar la agrupación de datos en sistemas de “carta”, de tal manera que cada carta se identifique como una unidad individual que contiene toda su información, además de hacer sencillo el escaneo ya que permite identificar el contenido por algo visible.

¹<https://balsamiq.com/product/desktop/>

■ Ventajas

- Facilidad de identificar al usuario
- Adaptable a dispositivos pequeños
- Información jerarquizada
- Se pueden añadir acciones de manera sencilla
- Se puede añadir más información de manera sencilla

■ Desventajas

- En dispositivos pequeños puede que haya que bajar mucho para encontrar lo que deseamos
- Es difícil ordenar y filtrar
- Puede hacerse compleja para usuarios poco familiarizados
- Los usuarios no suelen cambiarse la foto de perfil, lo que puede disminuir la sencillez con la que se identifica al usuario



Figura 4.4: Vista de lista

Para la lista (Figura 4.4), se ha seguido un formato de tabla, ya que la guía de material design propone que, para grandes volúmenes de datos con información similar, es recomendable darle un formato de tabla a los datos, ya que hace que sea sencillo escanear la información, y ordenar y filtrar los datos de manera que sea rápido encontrar lo que buscamos [32].

■ Ventajas

- Similar a las listas clásicas de alumnos (familiaridad)
- Sencillez
- Facilidad para ordenar y filtrar

■ Desventajas

- Puede ser difícil de ver los datos en dispositivos pequeños
- Puede ser difícil encontrar el participante en grandes grupos de participantes
- No se pueden añadir muchas columnas ya que habría que tener un movimiento lateral, lo que añade complejidad al uso de la aplicación

Para los mapas no hay tantos documentos de diseño, pero la mayoría coinciden en que lo mejor es dejar el mapa en el centro y añadir información en los bordes. [33] [34] [35] [36] Por ello, creo que la disposición actual sería válida para cumplir con una funcionalidad básica. Para esta interfaz, he ideado dos vistas, una para el organizador (Figura 4.5.a) y otra para el participante (Figura 4.5.b).

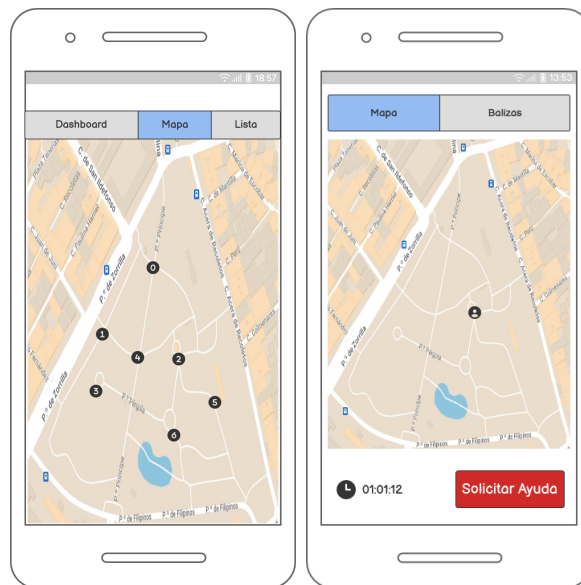


Figura 4.5: Vistas de mapa del organizador y del usuario

Además, para revisar el progreso en las balizas para el organizador (Figura 4.6.a) y el participante (Figura 4.6.b), se ha ideado un par de vistas nuevas. Estas vistas representan en formato de tabla los datos de las balizas, siguiendo los principios de Material design [30] que permite escanear la información de manera sencilla, fundamental en el caso de las balizas, ya que se necesita poder conocer el progreso de un vistazo.



Figura 4.6: Vistas de balizas del organizador y el usuario

Acomodación a los casos de uso

En el caso de uso *Realizar el seguimiento de una actividad* (ver tabla 3.1) , las interfaces destinadas a resolver el escenario son las correspondientes a la vista de tarjetas, la vista de mapa y la vista de lista (ver Figura 4.7). También se incluyen los casos de uso *Consultar el progreso de los participantes* (ver tabla 3.2) y *Consultar la localización de los participantes* (ver tabla 3.4).

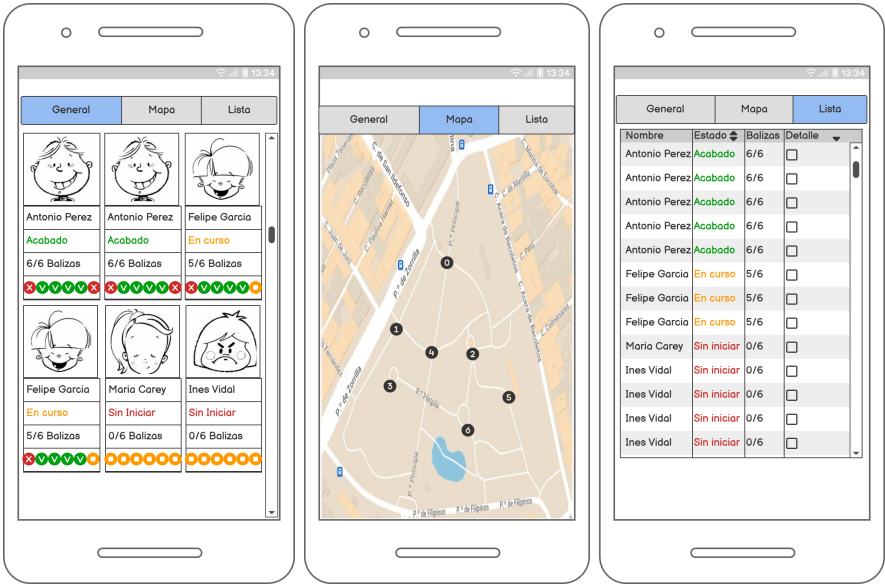


Figura 4.7: Prototipos con información para el organizador

En el caso de uso *Consultar el recorrido de los participantes* (ver tabla 3.3), las interfaces destinadas a resolver el escenario son las vistas de balizas y de recorrido sobre el mapa (ver Figura 4.8).



Figura 4.8: Prototipo de vista de balizas y recorrido para el organizador

4.2.3. Primera prueba de usabilidad

Primer prototipo de la interfaz

Para crear el primer prototipo de baja fidelidad, se han empleado las interfaces anteriores junto con algunas páginas de apoyo (*Login*, selector de actividad, pruebas de balizas, etc) para crear una serie de prototipos en Uizard². Debido a las limitaciones de la herramienta, se han realizado pequeños cambios en las interfaces para adaptarlas a esta (tamaño y elementos periféricos) y se ha desechado implementar la funcionalidad completa de la aplicación en un solo prototipo. Estos se emplearán en las pruebas de usabilidad con usuarios posteriores.

- **Prototipo de realización de actividad como participante.**

Este prototipo (ver Figura 4.9) contempla la participación en una actividad de orientación como participante. Este prototipo comprende la selección de actividad, así como orientarse con el mapa y la resolución de un reto de baliza. Se incluye para comprobar si hay alguna necesidad de mejorar la interfaz del usuario que no haya surgido en el análisis previo.

- **Prototipo de realización de actividad como organizador.**

Este prototipo contempla el proceso de monitorizar una actividad de orientación como organizador, haciendo referencia a los casos de uso relacionados con la monitorización de la actividad (ver tablas 3.1, 3.4, 3.2 y 3.3). Este comprende toda la parte de inicio en la que seleccionaremos la actividad que queremos monitorizar (Figura 4.10); la parte que compete a la información de los participantes, tanto la vista general como la vista de lista y el mapa (Figura 4.11); y la parte de verificar las respuestas de los usuarios a las balizas (Figura 4.12).

²<https://uizard.io/>

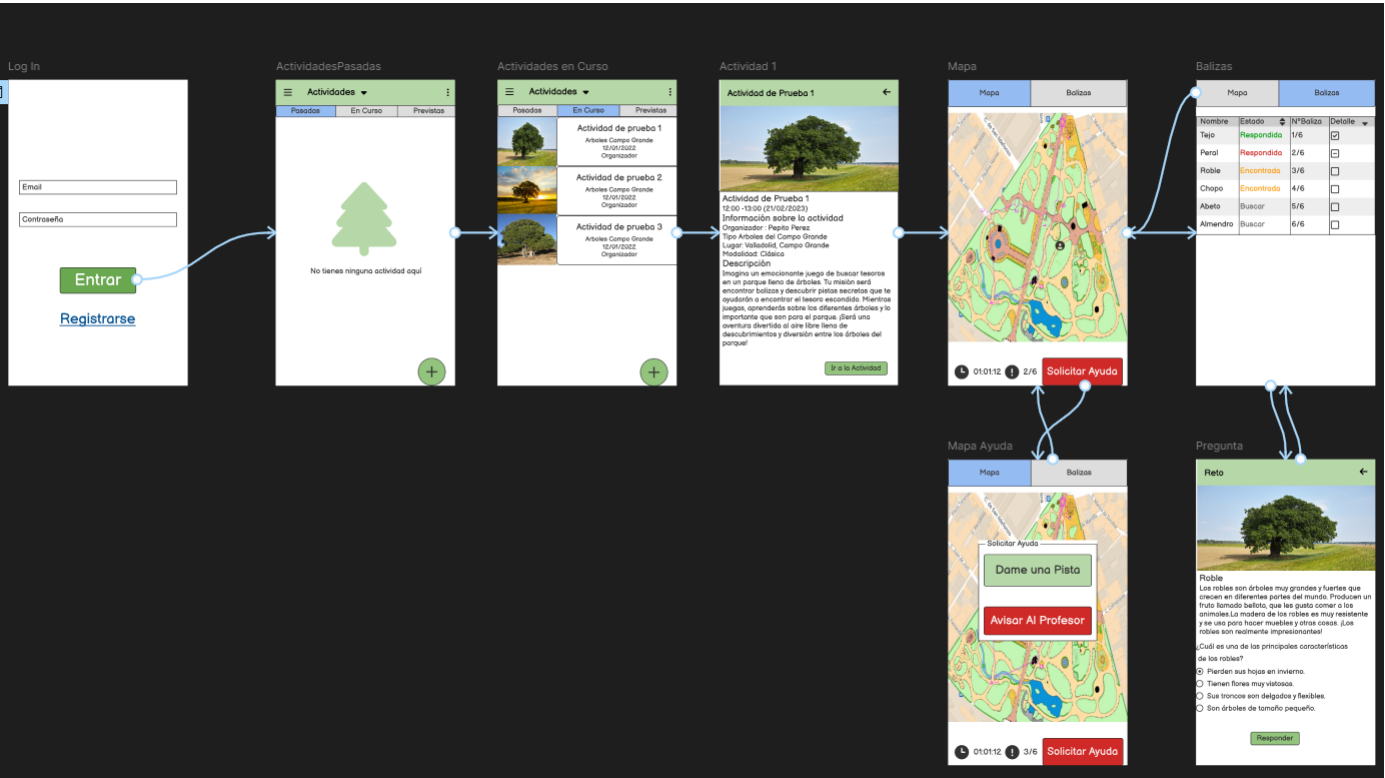


Figura 4.9: Prototipo de participación en actividad

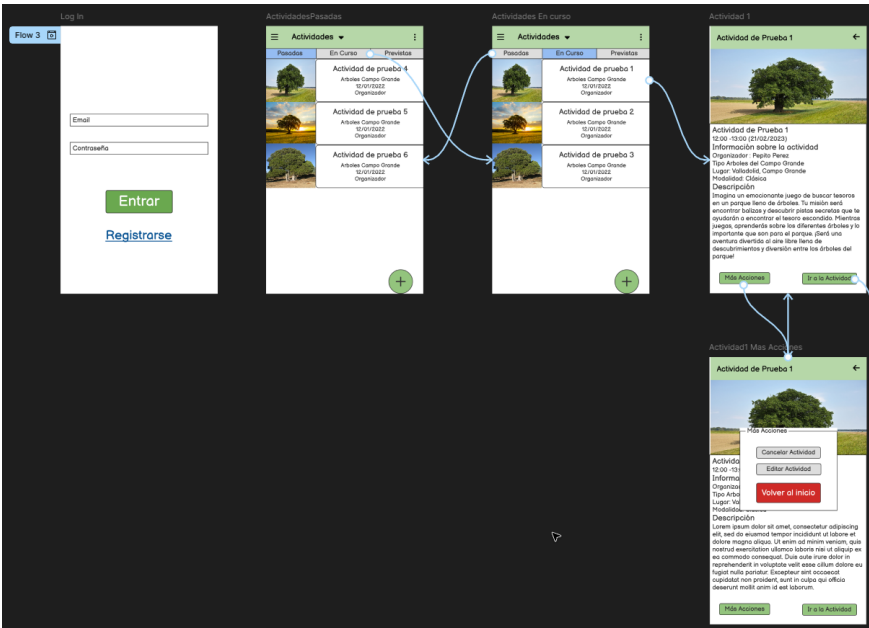


Figura 4.10: Prototipo de actividad de orientación. Selección de actividad

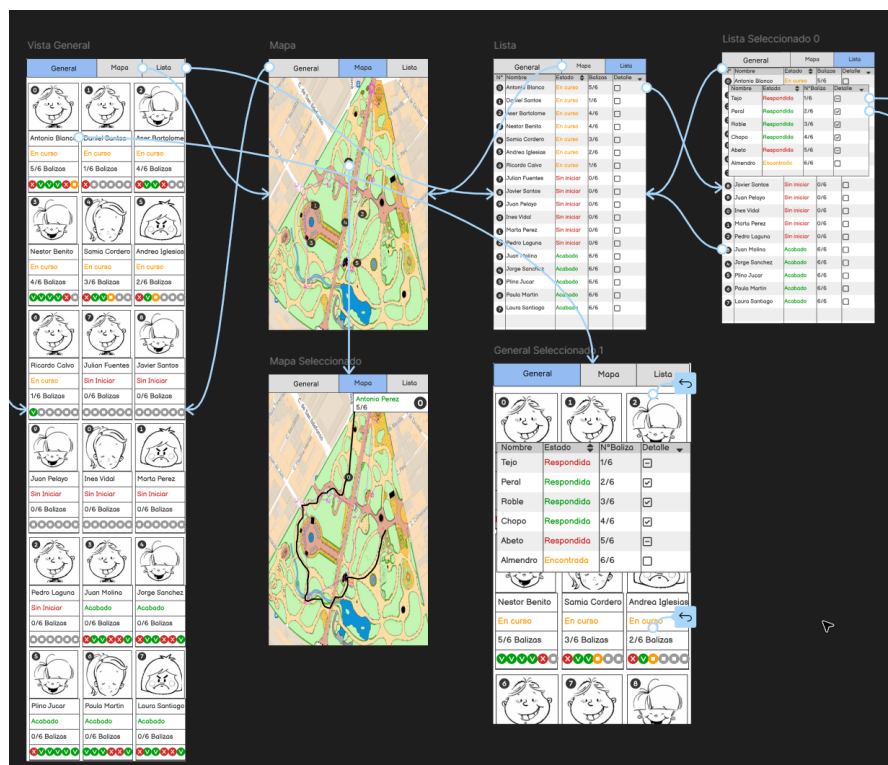


Figura 4.11: Prototipo de actividad de orientación. Información de los participantes

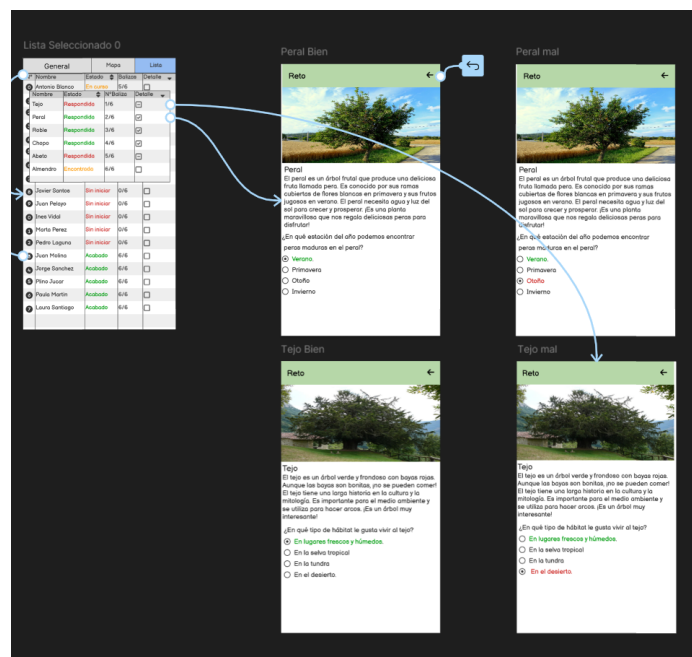


Figura 4.12: Prototipo de actividad de orientación. Preguntas

■ Prototipo de revisión de actividad

Este prototipo (ver Figura 4.13) contempla la revisión de una actividad de orientación ya completada, haciendo referencia a los casos de uso relacionados con la revisión (ver tablas 3.5 y 3.6). Este prototipo comprende la parte de selección de actividad, revisar el recorrido de los participantes y la revisión de las respuestas de un participante.

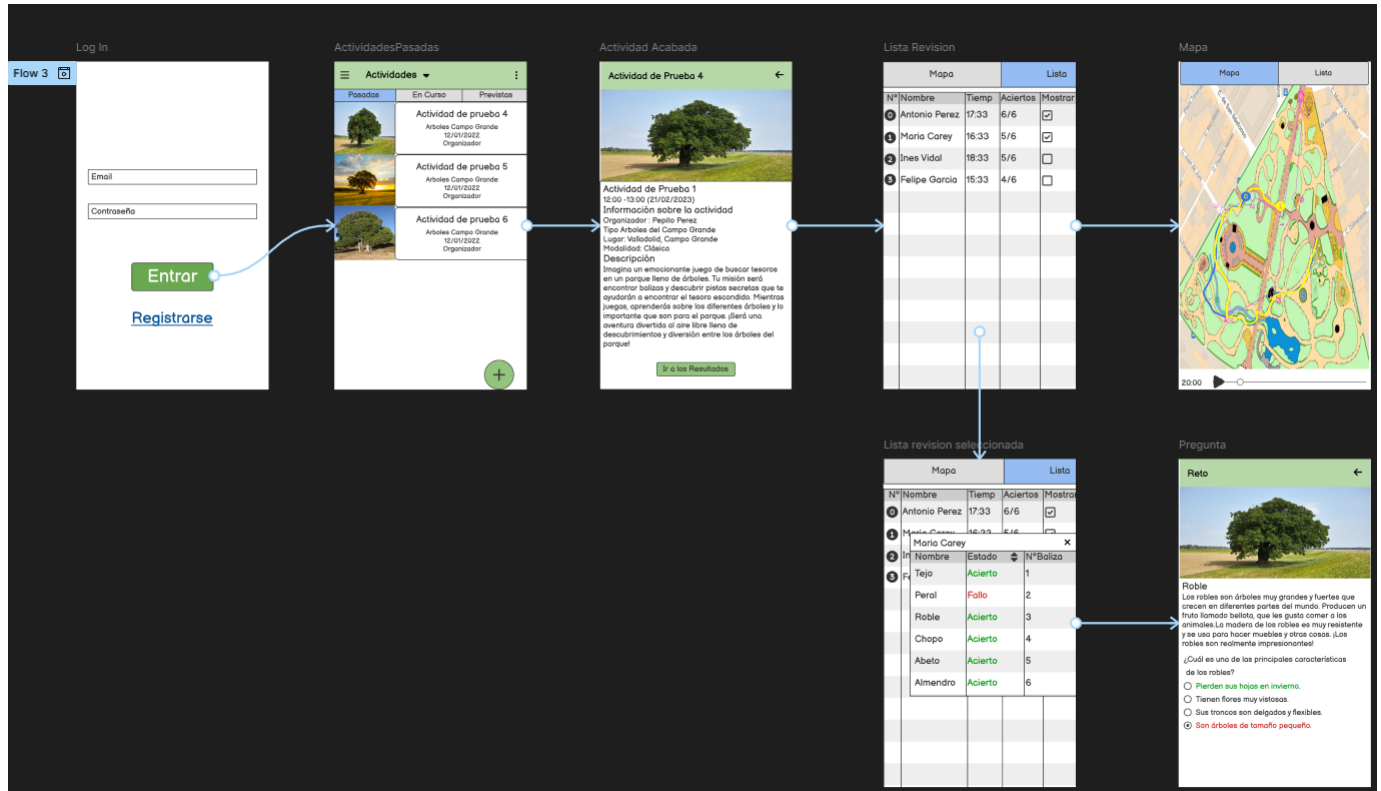


Figura 4.13: Prototipo de revisión de actividad

Diseño de la primera prueba de usabilidad

Con este primer prototipo, se preparó una primera prueba de usabilidad destinada a probar los cambios en la interfaz con usuarios finales. Esta prueba consistió en solicitar a los participantes que realizaran una serie de tareas en el prototipo con unas actividades de prueba preconfiguradas para realizar las tareas anteriormente mencionadas.

Objetivos

- Comprobar que la interacción con la aplicación es fácil de aprender para los usuarios.
Para comprobar esto, tendremos en cuenta tanto la tasa de éxito en la completitud de las tareas propuestas como los posibles inconvenientes que pueda tener el participante durante las pruebas.
- Comprobar posibles problemas en la navegación de la interfaz
- Validar el diseño de la interfaz

- Evaluar la satisfacción del usuario

Sesiones

Una sesión de 90 minutos será suficiente para realizar la prueba. Esta sesión se dividió en tres partes:

- Un pequeño resumen del proyecto a los participantes y descripción de los principales cambios
- Una parte de prueba donde los participantes realizarán las tareas propuestas con ayuda de un facilitador
- Una charla retrospectiva con todos los participantes a la vez para tener retroalimentación tanto de los cambios de la interfaz como de la prueba

Participantes

Para esta prueba, los participantes serán cuatro profesores voluntarios. Estos profesores utilizan actualmente la orientación educativa en su docencia y han empleado Orientatree en algunas de sus actividades de orientación.

Equipamiento

Para esta prueba se empleó un dispositivo móvil con un prototipo sencillo que utilizaron los usuarios.

Tareas a realizar

1. Consultar el progreso de la actividad (como participante)
En esta tarea, el usuario deberá de ser capaz de localizarse en el mapa, comprobar su progreso en la actividad y responder a un reto de una baliza. Estas son las preguntas que deberán de contestar para comprobar si progresan en la tarea adecuadamente.
 - Imagina que estás participando en una actividad de orientación
 - ¿Podrías decirme dónde estás en el mapa?
 - ¿Podrías decirme cuántas balizas has encontrado?
 - ¿Puedes decirme cuántas has respondido?
 - ¿Puedes contestar a la pregunta Roble?
2. Comprobar el progreso de los participantes (como organizador)
En esta tarea, el usuario deberá de ser capaz de identificar un comportamiento extraño por parte de un participante y comprobar la localización y el progreso de otro participante. Estas son las preguntas que deberán de contestar para comprobar si progresan en la tarea adecuadamente.

- Imagina que eres el organizador de una actividad de orientación en curso y necesitas hacerte una idea del progreso general de los participantes. La actividad en cuestión será la actividad de prueba 1.

- ¿Hay algún alumno que esté teniendo un desempeño extraño?
- ¿Dónde se encuentra Antonio Blanco?
- ¿Cuántas balizas ha encontrado?
- ¿Cuántas balizas ha respondido?

3. Comprobar el recorrido de un participante tras la actividad (como organizador)

En esta tarea, el usuario deberá de ser capaz de identificar a los participantes que hayan realizado la actividad, comprobar el desempeño de un participante e identificar a los participantes cuyo recorrido está en pantalla. Estas son las preguntas que deberán de contestar para comprobar si progresan en la tarea adecuadamente.

- Imagina que eres el organizador de una actividad de orientación (actividad de prueba 4) y quieres revisar cómo han desarrollado la prueba los alumnos.

- ¿Cuántos alumnos han realizado la prueba?
- ¿Qué respuesta ha fallado María Carey?
- ¿Que personas estás viendo en el mapa?

Desarrollo de la prueba

La prueba se desarrolló sin ningún problema técnico ni de desarrollo reseñables. Todos los participantes pudieron realizar la prueba y todos participaron en la revisión grupal final. Toda la prueba se realizó de manera remota a través de una llamada de Microsoft Teams.

Resultados de la prueba

Los resultados de la prueba se expresan como el % de completitud de las subtareas (ver Tabla 4.1) comprendida en las tareas, siendo el 100 % la completitud exitosa de todas las subtareas.

	Participante 1	Participante 2	Participante 3	Participante 4	Total
Tarea 1	50 %	100 %	100 %	100 %	87.5 %
Tarea 2	75 %	75 %	100 %	100 %	87.5 %
Tarea 3	100 %	100 %	100 %	100 %	100 %

Cuadro 4.1: % de completitud de las tareas

Errores cometidos durante la prueba

■ Tarea 1

- Participante 1: No es capaz de localizar su posición en el mapa

■ **Tarea 2**

- Participante 1: No identifica al alumno con un desempeño incorrecto
- Participante 2: No es capaz de encontrar al alumno especificado

■ **Tarea 3**

- Sin errores a reseñar

Principal retroalimentación de los usuarios

Tras la realización de la prueba, se comprobó que los conceptos planteados eran exitosos, pero necesitan unos pequeños cambios sugeridos por los usuarios en la charla posterior a la prueba:

- Eliminar la columna de detalle.
La columna de detalle, pensada para dirigir a los usuarios a la información de las balizas, no les ha sido útil a los usuarios.
- Emplear números para identificar a los participantes
Había dudas entre emplear iniciales, las fotos de perfil de usuario y números para identificar a los participantes. Los usuarios han preferido emplear números.
- Aumentar la calidad del mapa
- La aplicación es sencilla de aprender y la información dentro de la misma está bien estructurada

4.2.4. Segunda prueba de usabilidad

Segundo prototipo de la interfaz

En la segunda prueba de usabilidad se revisó el prototipo y se eliminó la columna de detalle, de acuerdo a los resultados obtenidos en la primera prueba de usabilidad. Con ello, se creó un segundo prototipo de la interfaz, esta vez en figma³, ya que la herramienta de Uizard quedó sin uso útil tras un cambio en sus condiciones de uso. Con este prototipo, se procedió a realizar una segunda prueba de usabilidad.

³<https://www.figma.com/prototyping/>

4.2. PROCESO DE DISEÑO ITERATIVO

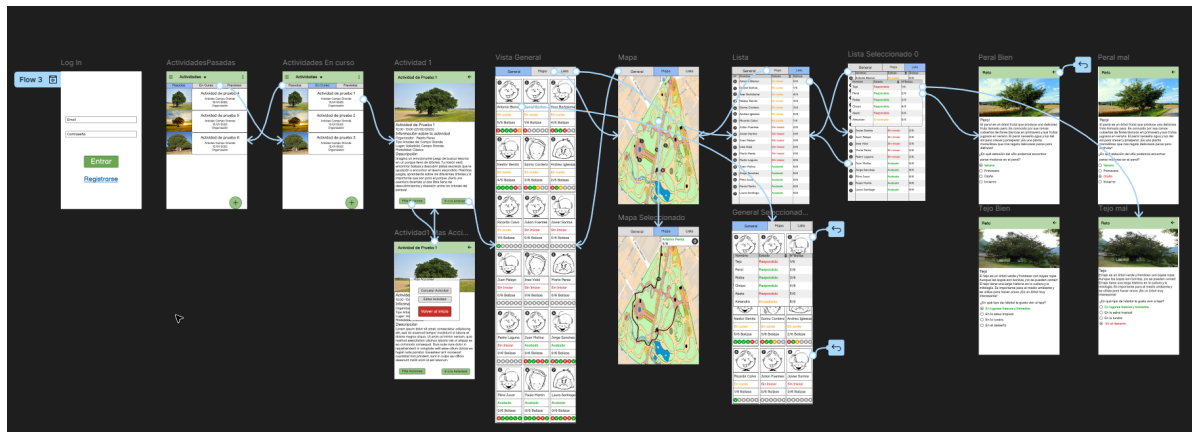


Figura 4.14: Segundo prototipo de la interfaz. Vista completa

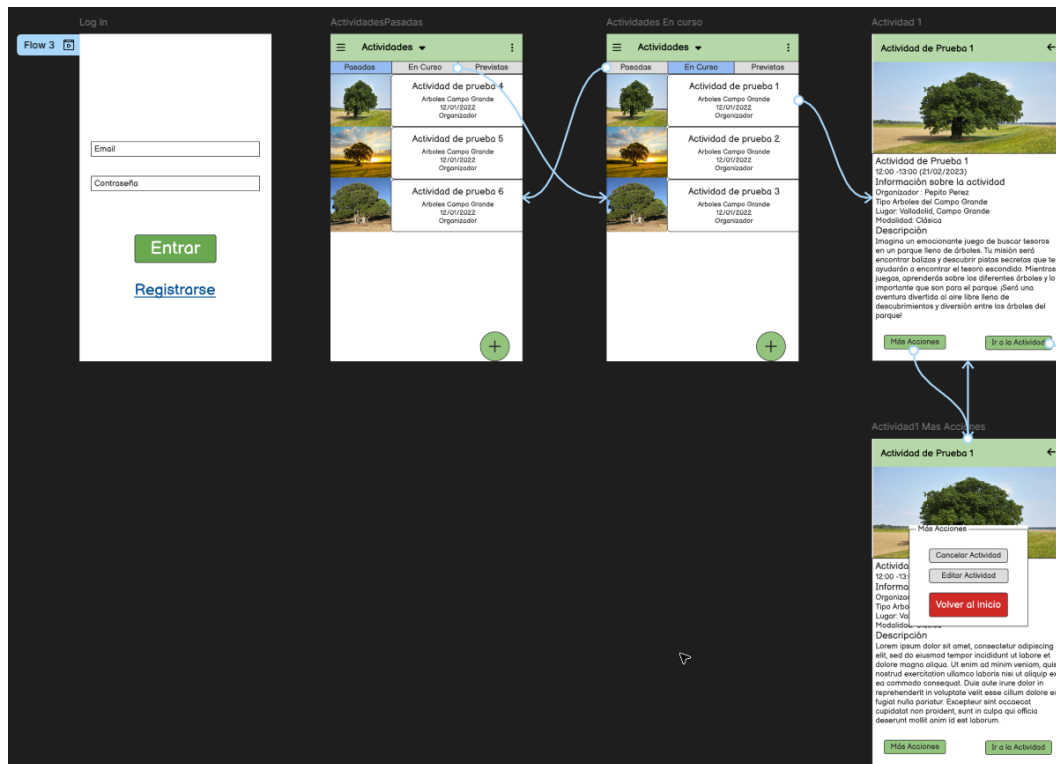


Figura 4.15: Parte de Log-in del segundo prototipo de la interfaz

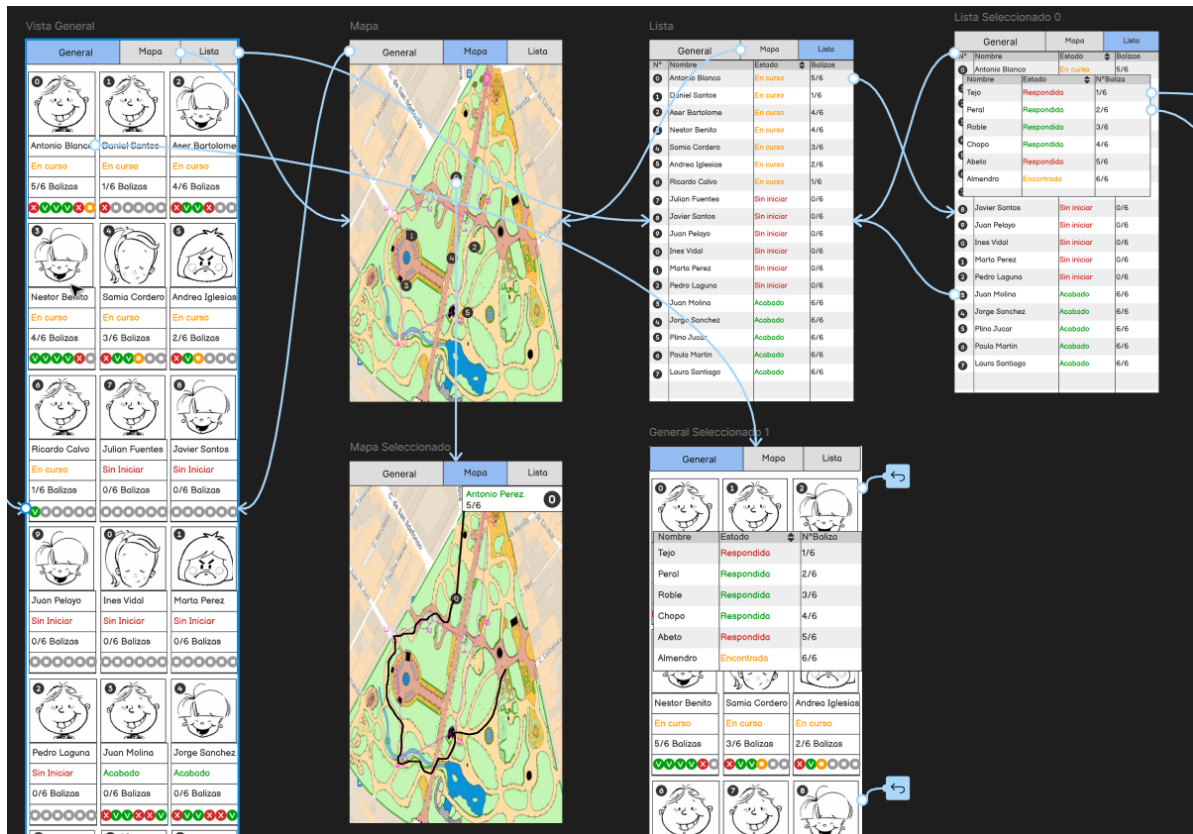


Figura 4.16: Parte de información de los participantes del segundo prototipo de la interfaz

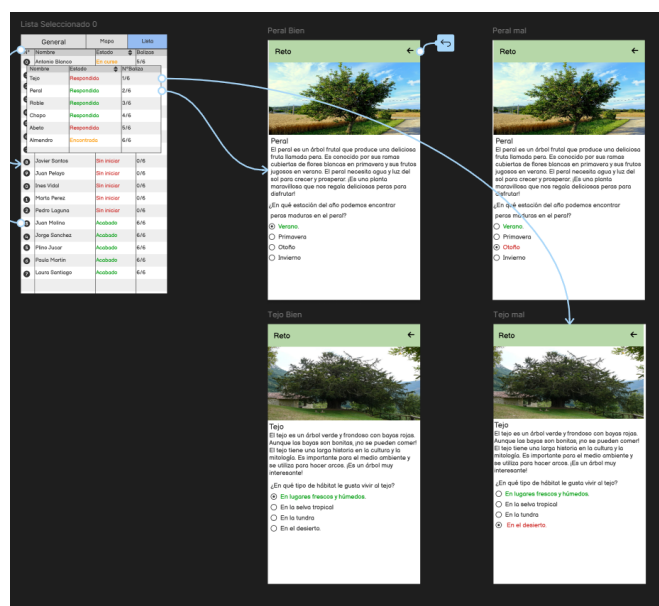


Figura 4.17: Parte de balizas del segundo prototipo de la interfaz

Diseño de la segunda prueba de usabilidad

Con este segundo prototipo centrado en la vista del organizador, se decidió realizar una breve prueba para comprobar si los cambios en la interfaz la seguían haciendo usable y si era realista plantear una prueba de usabilidad con una versión temprana de la aplicación.

Objetivos

Aquí enumeraremos los puntos a verificar con esta prueba:

- Comprobar que los cambios en la interfaz son efectivos.

Sesiones

Para la realización de la prueba se emplea una sesión de 30 minutos de manera remota que cada profesor realiza de manera asíncrona, compuesta de los siguientes elementos:

- Un pequeño resumen a los participantes de dónde se han centrado los cambios mediante un documento explicando los mismos.
- Una parte de prueba donde los participantes realizarán una tarea sencilla sin ayuda.
- Una serie de preguntas de respuesta abierta donde los participantes puedan expresar sus opiniones

Participantes

Para esta prueba, los participantes son tres de los profesores que ya realizaron la primera prueba de usabilidad.

Equipamiento

Para esta prueba se emplea el prototipo antes mencionado que los usuarios prueban en el dispositivo de su elección.

Tareas a realizar

Estas tareas están destinadas a probar la interfaz del prototipo con una serie de preguntas sencillas a contestar mediante el empleo de este.

1. Comprobar el progreso de una prueba de orientación
En esta tarea, los usuarios deben de entrar a la actividad de Prueba 1, en curso, e identificar el progreso de varios de los participantes.

- Suponiendo que estás comprobando el progreso de la Prueba 1, intenta obtener la siguiente información:
 - ¿Cuántas balizas ha encontrado Samia Cordero?
 - ¿Qué ha respondido Aser Bartolomé en la pregunta Roble?
 - ¿Qué ha respondido Andrea Iglesias en la pregunta Peral?
 - ¿Cuántas preguntas ha acertado Néstor Benito?

Cuestionario tras la prueba

Para verificar el correcto funcionamiento de la aplicación, se incluye un breve cuestionario a cumplimentar para que los profesores expresen sus opiniones acerca del prototipo y de la interfaz.

- ¿Os ha parecido sencillo conseguir la información que necesitabais para contestar a las preguntas?
- ¿Hay alguna información que haya sido más difícil de obtener?
- ¿Se puede obtener la información de manera clara e inequívoca?
- Ruegos, preguntas y sugerencias

Desarrollo de la prueba

La prueba se desarrolló sin ningún problema, ya que todos los participantes pudieron acabar la prueba sin incidentes reseñables.

Resultados de la prueba

Los resultados de la prueba se expresan como el % de completitud de las subtareas (ver Tabla 4.2) comprendida en las tareas, siendo el 100 % la completitud exitosa de todas las subtareas.

	Participante 1	Participante 2	Participante 3	Total
Tarea	100 %	100 %	100 %	100 %

Cuadro 4.2: % de completitud de las tareas

Como podemos ver, el porcentaje de éxito es total, lo que indica que los cambios propuestos son efectivos y no hay problema con la interfaz.

Errores cometidos durante la prueba

No hay errores a reseñar

Principal retroalimentación de los usuarios

Tras la realización de la prueba, hemos visto que los cambios propuestos son acertados ya que la retroalimentación de los usuarios es favorable, aunque también sugieren algunas mejoras.

- El número de balizas de los participantes tendría que hacer referencia a las balizas encontradas, no las contestadas.
- La ventana con la información de las balizas superpuesta sobre la pantalla de los usuarios al hacer click sobre uno de ellos es una buena idea.
- Esta ventana superpuesta debería de tener el nombre del alumno indicado en algún lugar.

4.2.5. Tercera prueba de usabilidad: prueba presencial con usuarios

Interfaz de la tercera prueba de usabilidad

Debido a que esta prueba se realizó con una versión temprana de la aplicación, se adjuntan capturas de las interfaces relevantes para la prueba. Estas interfaces se organizan en tres grupos que se describen a continuación.

En el primer grupo encontramos las interfaces de selección de actividad y la vista previa a la monitorización de la actividad en la que se muestra información relevante para la misma (Figura 4.18).

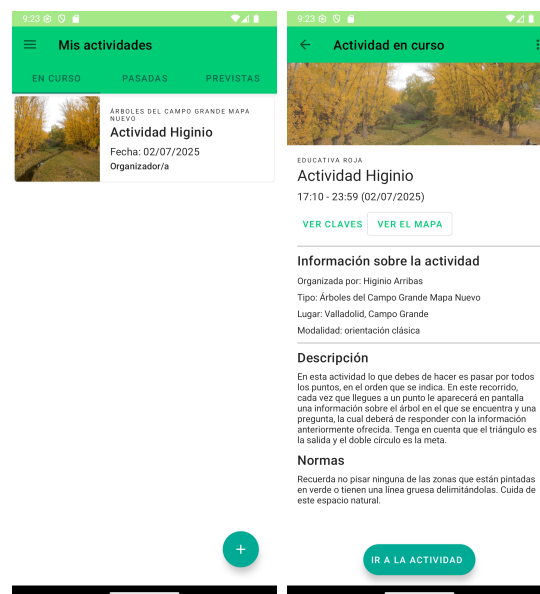


Figura 4.18: Vistas de selección de actividad y de información de la actividad

En este segundo grupo se recogen las interfaces que recogen la información general de los participantes (Figura 4.19). En la primera se muestra el mapa con las localizaciones de los participantes, en la segunda se muestra la vista de tarjetas y en la última la vista de lista.

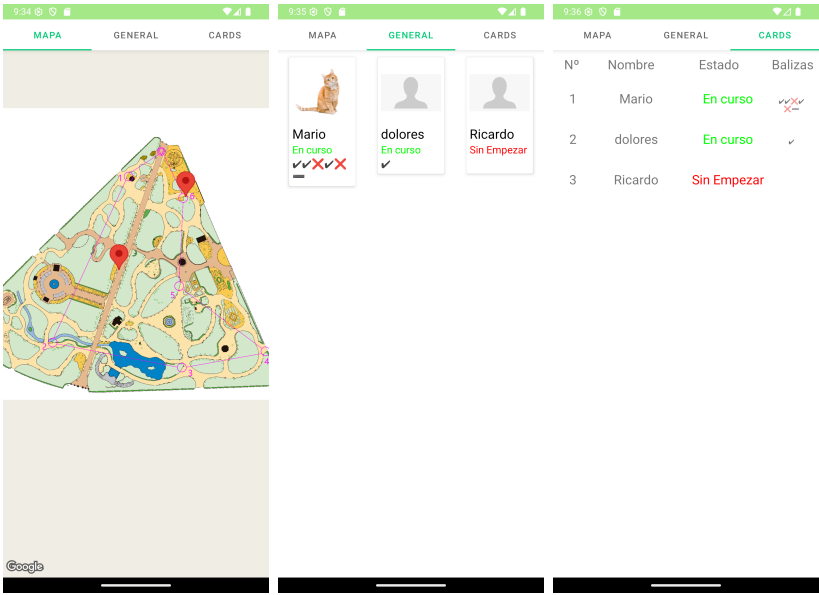


Figura 4.19: Vistas de información de los participantes

En este último grupo se recogen las interfaces que recogen la información de las balizas, tanto en la vista de tarjetas como en la vista de lista, así como la vista de la respuesta del participante a la pregunta planteada en una de ellas (Figura 4.20).

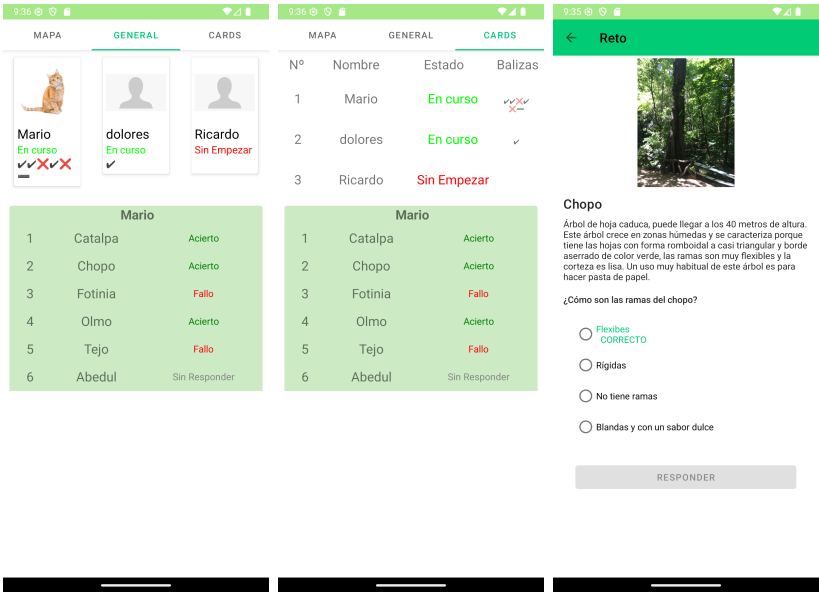


Figura 4.20: Vistas de información de las balizas

Diseño de la tercera prueba de usabilidad

Con la versión temprana de la aplicación basada en el prototipo de la prueba anterior, se preparó una prueba presencial en Campo Grande simulando una actividad real. Esto se realizó mediante la simulación de una actividad de orientación educativa. Por motivos de simplicidad y tiempo, se decidió realizar una sola actividad de orientación en la que uno de los participantes

realizase el rol de organizador y el resto realizasen las funciones de los participantes, sin evitar que los otros usuarios pudiesen observar cómo se desarrollaba la prueba desde la vista del orientador para dar su opinión.

Objetivos

- Comprobar que la aplicación permite a los usuarios realizar una actividad de orientación educativa de manera satisfactoria
- Comprobar posibles errores en el funcionamiento de la aplicación
- Validar el diseño de la interfaz
- Evaluar la satisfacción de los usuarios con la nueva aplicación

Sesiones

Se realizó una sesión de 90 minutos para realizar la prueba.

Participantes

Los participantes fueron tres de los profesores que ya habían realizado las pruebas de usabilidad anteriores, un corredor de carreras de orientación experimentado, uno de mis tutores y yo.

Equipamiento

Para esta prueba se emplearon varios dispositivos móviles Android que proporcionó cada participante.

Desarrollo de la prueba

La prueba se pudo completar de manera correcta, ya que se pudo completar por completo, excepto un participante que tuvo que abandonarla pronto debido a una falta de tiempo ya prevista.

Conclusiones de la tercera prueba con usuarios

Tras la realización de la prueba, se comprobó que la aplicación evoluciona de manera satisfactoria, aunque los usuarios propusieron varias mejoras a incorporar:

- Eliminar la vista de lista.
Los usuarios notaban que la vista con información de los participantes en forma de lista no estaba mal, pero no aportaba nada al organizador que no tuviese ya la vista de tarjetas. Sugirieron eliminar esta vista para facilitar la navegación por la aplicación.

- Añadir tiempos parciales a las balizas.
En el caso de una carrera de orientación, es importante conocer el momento en el que se alcanzan las balizas. Además, es interesante ver el parcial de tiempo en caso de una actividad educativa para comprobar el progreso de los participantes.
- Arreglar pequeños fallos en la interfaz:
 - Parpadeo en las imágenes
 - Cambios en la nomenclatura de distintos elementos
 - Tamaño del texto en algunas vistas

4.3. Arquitectura de la aplicación

La arquitectura de esta aplicación sigue el modelo **Cliente-Servidor**, separando claramente las responsabilidades entre el front-end y el back-end. Por un lado, el cliente se encarga de la interfaz de usuario y de la lógica de negocio. Aquí se gestionan todas las interacciones del usuario, como la navegación, las acciones en pantalla y la ejecución de funciones específicas de la aplicación. Por otro lado, el servidor, implementado con Firebase, actúa como el núcleo de gestión de datos. Su función principal es almacenar, recuperar y sincronizar la información generada por los usuarios. Cada vez que la aplicación cliente produce datos como registros de actividades o interacciones con balizas, estos se envían al back-end, donde Firebase los organiza y los mantiene actualizados para todos los dispositivos conectados. Además, el servidor centraliza tareas críticas como la autenticación de usuarios y el almacenamiento seguro de sus perfiles.

La comunicación entre ambas partes se realiza mediante consultas desde la aplicación Android hacia Firebase. Por ejemplo, cuando el cliente necesita mostrar información sobre una actividad o una baliza, solicita estos datos al back-end, que responde con la información actualizada. Este flujo bidireccional garantiza que los usuarios siempre accedan a contenido en tiempo real, con consistencia y sin redundancias.

Entre las ventajas clave de esta arquitectura destacan la sincronización instantánea de datos, la escalabilidad automática que ofrece Firebase y la seguridad integrada en la gestión de usuarios. Al dividir claramente las capas de la aplicación, se optimiza tanto el rendimiento como el mantenimiento del sistema, asegurando una experiencia fluida y confiable para los usuarios finales.

4.3.1. Estructura del front-end

A continuación, se representa la estructura de la parte del cliente. Se adjunta una vista general (Figura 4.21), y una vista detallada de cada uno de los componentes (Figuras 4.22, 4.23, 4.24 y 4.25). Se ha añadido toda una capa de persistencia y se han añadido varias vistas y adaptadores para elementos de las mismas. Además, se han modificado varias de las vistas y el `LocationService` para adaptar las nuevas funcionalidades. Se ha añadido la clase `mainApplication` para englobar la información relevante a toda la aplicación, como la inicialización de la base de datos de Firestore o los `WorkManager`.

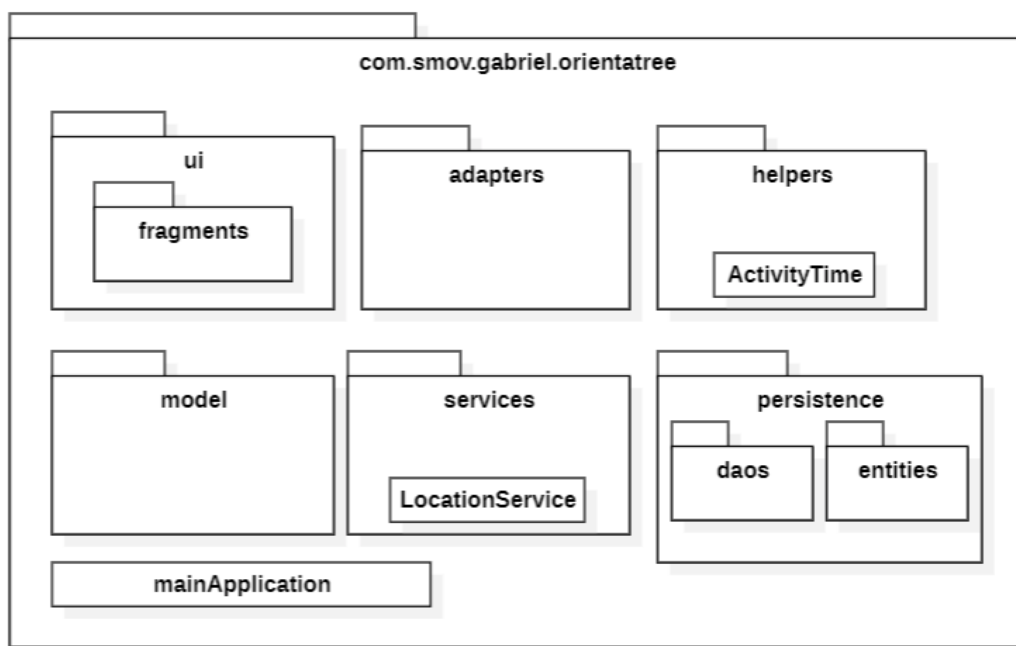


Figura 4.21: Diagrama de paquetes de la aplicación cliente

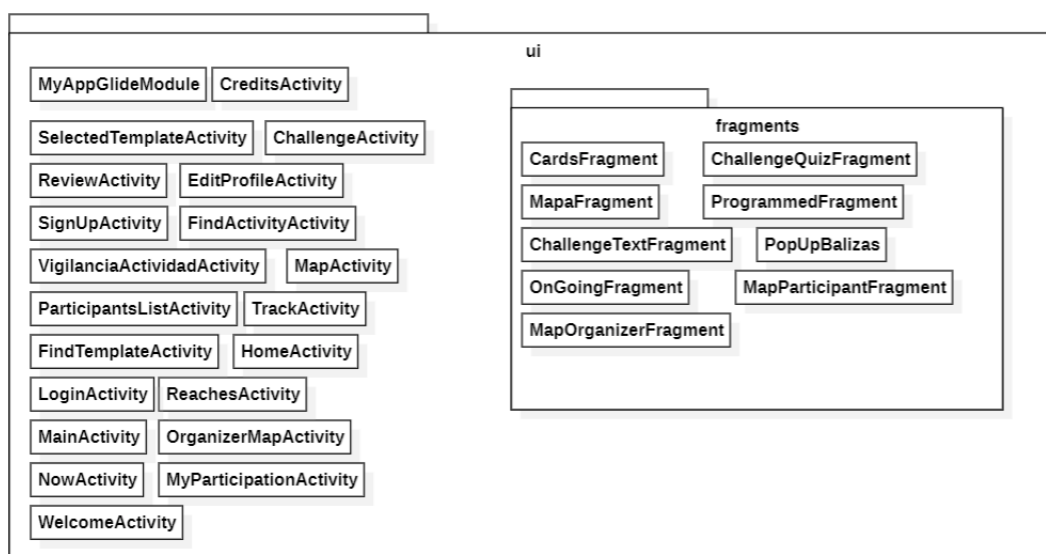


Figura 4.22: Detalle del paquete ui

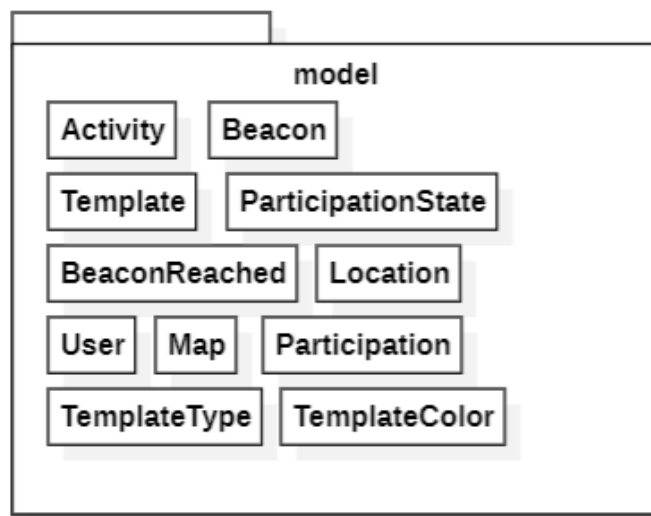


Figura 4.23: Detalle del paquete model

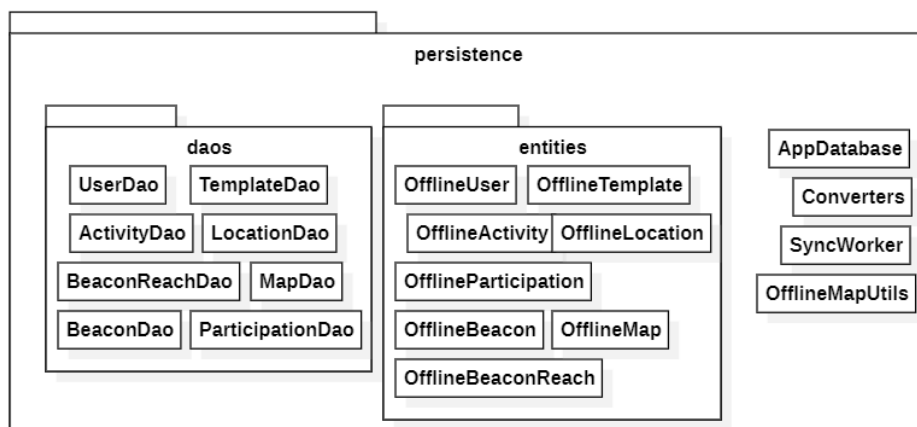


Figura 4.24: Detalle del paquete persistence

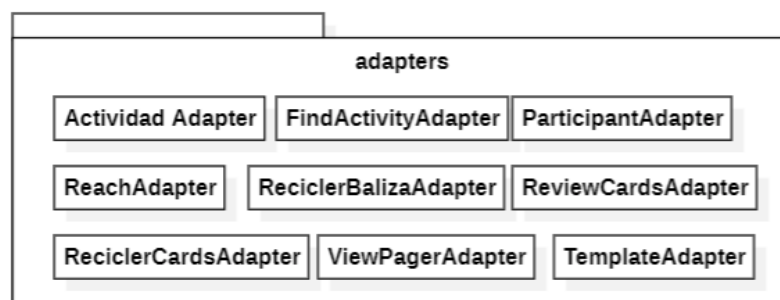


Figura 4.25: Detalle del paquete adapters

4.3.2. Estructura del back-end

En esta versión de la aplicación se ha mantenido en gran medida la estructura del back-end de la versión anterior, por lo que en esta sección se detallaran los componentes y cambios realizados en la estructura del back-end.

Cloud Firestore

Como base de datos, se ha mantenido Cloud Firestore como base de datos. Esta es una base de datos NoSQL en la nube desarrollada por Google. Está diseñada para almacenar, sincronizar y consultar datos en aplicaciones web, móvil y de servidor. A diferencia de las bases de datos tradicionales, utiliza un modelo estructurado en documentos y colecciones, siendo los documentos conjuntos de distintos tipos de datos (Cadenas de texto, números, fechas, etc) y las colecciones agrupaciones de estos documentos. Una de las ventajas de Cloud Firestore es su capacidad de escalabilidad automática, lo que significa que puede manejar grandes volúmenes de datos sin requerir configuración manual, además de actualizaciones en tiempo real, permitiendo que los clientes reciban cambios de manera instantánea cuando se modifican los datos.

Uno de los cambios realizados durante el desarrollo de esta versión de la aplicación, fue habilitar el plan *Blaze*, debido a los cambios en los términos y condiciones de uso de Google. Se sigue manteniendo una funcionalidad gratuita, pero en el caso de sobrepasar los límites, se producirá un cobro en función del uso.

Firestore Authentication

Este servicio de Firestore está diseñado para simplificar el proceso de autenticación de los usuarios en las aplicaciones móviles o web. Esto nos permite añadir un sistema de registro y verificación de usuarios de manera sencilla, así como la posibilidad de que recuperen la contraseña en caso de olvidarla. También permite a los usuarios mantener su sesión iniciada con un token de autenticación, facilitando así el acceso a la aplicación.

Firestore Storage

Este servicio de almacenamiento de Firestore está diseñado para almacenar imágenes, audio y vídeo. Se emplea para almacenar las imágenes de las plantillas, de las balizas, los mapas y las fotos de perfil de los usuarios. Se han añadido nuevos mapas con mayor resolución y nuevas fotos para las plantillas con dicho mapa.

4.3.3. Diagrama de despliegue

Para el despliegue de la aplicación, se emplea una base de datos Firestore con tres funcionalidades: Cloud Firestore, Firestore Authentication y Firestore Storage. Además, se emplea una base de datos Room para la capa de persistencia local, y se emplea el almacenamiento del dispositivo para guardar las imágenes necesarias para el funcionamiento de la aplicación.

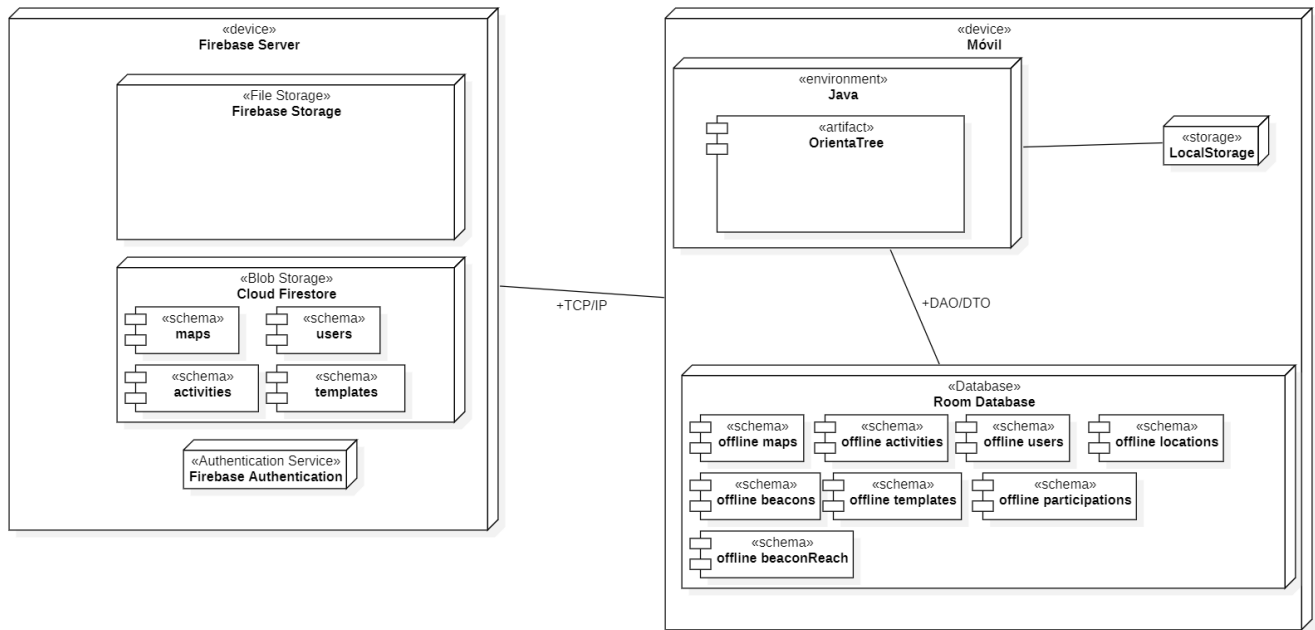


Figura 4.26: Diagrama de despliegue de la aplicación

4.4. Diseño de la funcionalidad sin conexión

A continuación, detallamos cómo se ha diseñado la funcionalidad sin conexión de la aplicación con WorkManager en base al análisis de alternativas previo 3.2.

4.4.1. Capa de persistencia

Previamente a la implementación de la sincronización de los datos con WorkManager, ha sido necesario implementar una capa de persistencia *offline* para guardar los datos de manera local antes de subirlos al servidor. En un principio se planteó sólo para la parte de la realización de la actividad, pero se expandió para soportar toda la parte de selección e inicio de actividad, ya que es altamente improbable que los usuarios tengan la actividad ya seleccionada previamente a la pérdida de conexión (al salir de casa o del centro probablemente), además de simplificar el proceso.

La capa de persistencia sigue un patrón DAO con DTOs (ver 4.27) y una base de datos Room basada en SQLite. Esta capa incluye varios DAOs con sus correspondientes DTOs para dar soporte a toda la información que es necesaria almacenar en *offline* para la realización de la actividad y la sincronización con el servidor una vez se recupera la conexión. Una vez el participante se inscribe en la aplicación, se descarga toda la información necesaria para la realización de la actividad de manera local. Cuando el usuario llega al punto de inicio, la aplicación carga todas las actividades guardadas de manera local para que el participante pueda seleccionar la deseada y realizarla.

4.4.2. WorkManager

Workmanager está diseñado para gestionar tareas en segundo plano de manera eficiente. Su principal ventaja es que garantiza la ejecución incluso si la aplicación se cierra o el dispositivo se reinicia. Cuando creas una tarea con WorkManager, esta sigue un ciclo de vida definido. Primero, se define la tarea especificando que se debe hacer y cuáles son las condiciones bajo las que puede ejecutarse. Una vez configurada, se encola y WorkManager se encarga de almacenarla de manera persistente, en nuestro caso, en la base de datos Room. Esto asegura que no se pierda si el dispositivo se apaga o la aplicación se cierra de manera inesperada. WorkManager decide cómo y cuándo ejecutar la tarea según las condiciones establecidas y los recursos disponibles. Cuando la tarea se ejecuta en segundo plano, WorkManager gestiona automáticamente elementos como los reintentos en caso de fallo. Finalmente, una vez completada, la tarea puede notificar su resultado para que la aplicación reaccione en consecuencia en caso de ser necesario.

Cuando el participante realiza una acción o cambia su estado, se crea una tarea con WorkManager, que se almacena hasta que se dispone de conexión a internet, momento en el que WorkManager intenta sincronizarlo con la base de datos para actualizar la información en el servidor.

4.4.3. Localización

Una vez se ha implementado la actualización de los datos de la participación de manera asíncrona, cabe plantearse la duda de ¿Cómo se obtiene la localización del participante sin conexión a internet? En el caso de nuestra aplicación, se emplea la API **Fused Location Provider**, que es parte de la API de Location de Google y permite la localización de dispositivos en base a varias fuentes [37]. Fused Location Provider obtiene los datos de distintas fuentes, siendo las principales el GPS, las Redes móviles, el Wi-Fi y distintos sensores del dispositivo como el acelerómetro o el giroscopio. Combinando esta información, proporciona una localización precisa del dispositivo y en este caso que nos atañe, del portador que se encuentra realizando una actividad de orientación. En el caso de la ausencia de conexión a internet, se emplea la triangulación clásica por GPS (o Galileo en el caso de la UE) para la localización del participante. Si bien esto aumenta el margen de error de la localización del participante, se mantiene en torno a los 5 metros de error, suficiente para hacerse una idea lo suficientemente precisa del recorrido del participante durante la actividad.

4.5. Patrones de diseño

En esta sección se recogen los patrones de diseño empleados en la aplicación. Se han empleado los siguientes patrones de diseño:

- Patrón DAO/DTO
- Patrón Singleton
- Patrón AbstractFactory
- Patrón Facade
- Patrón Builder

- Patrón Adapter

4.5.1. Patrón DAO/DTO

El patrón DAO/DTO (figura 4.27) hace referencia a dos patrones individuales que se emplean de forma conjunta:

- **Patrón DAO** (Data Access Object): Este patrón de diseño software se emplea para separar la lógica de acceso a datos de la lógica de negocio, empleando una clase DAO que implementa la interfaz de operaciones que se pueden realizar con la base de datos. Este patrón favorece la transparencia, facilita la migración de los componentes, reduce la complejidad del código y centraliza el acceso a los datos.
- **Patrón DTO** (Data Transfer Object): Este patrón proporciona un objeto DTO que permite transmitir datos entre distintas capas. Su objetivo principal es encapsular y transportar datos de manera eficiente, limitando el número de llamadas y evitando exponer detalles internos del modelo de dominio.

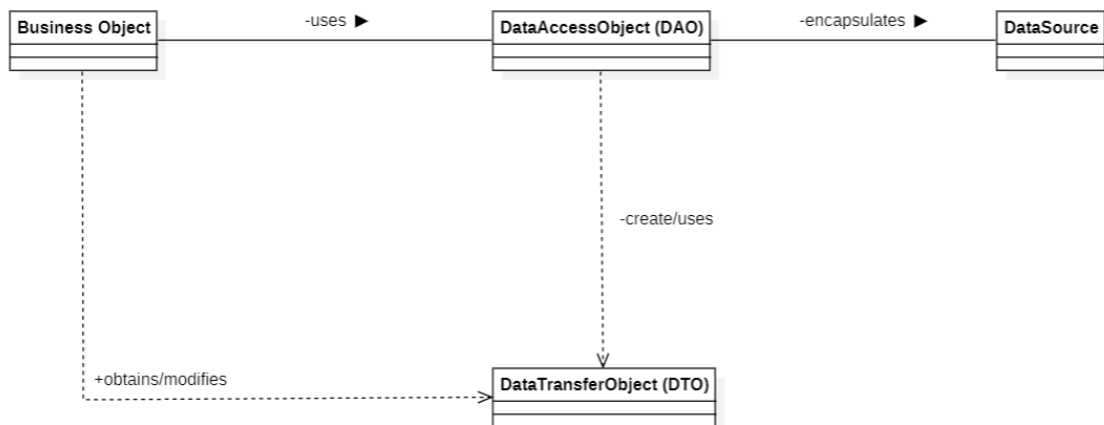


Figura 4.27: Patrón DAO/DTO

Este patrón se emplea para el acceso a datos en la parte *offline* (ver 4.4.1).

4.5.2. Patrón Singleton

El patrón singleton (figura 4.28) es un patrón de diseño que garantiza que una clase tenga una única instancia y proporciona un punto de acceso global a ella. Permite asegurar que solo exista una instancia en toda la aplicación, así como un acceso unificado a dicha instancia.

Este patrón se emplea en la base de datos *offline* para asegurar el acceso único a la misma.

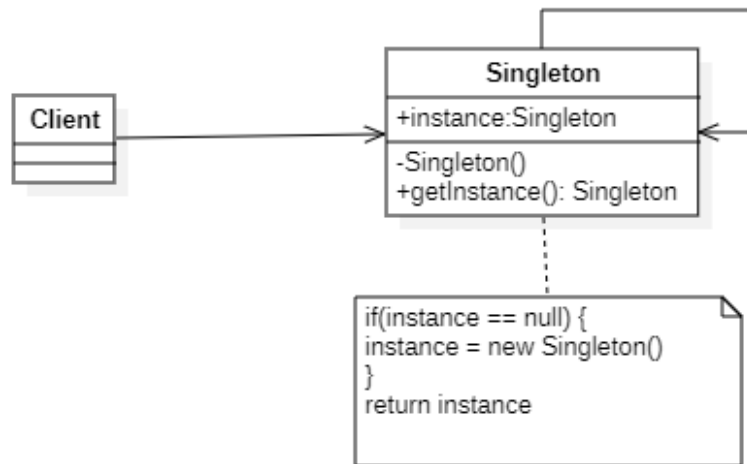


Figura 4.28: Patrón Singleton

4.5.3. Patrón Abstract Factory

Este patrón de diseño (figura 4.29) permite desacoplar la creación de objetos de su uso, mejorando así la consistencia, reduciendo el acoplamiento y mejorando la capacidad de cambio.

Este patrón se emplea durante la creación de la base de datos *offline* para permitir cambiar la implementación de la base de datos en caso de ser necesario.

4.5.4. Patrón Facade

Este patrón de diseño estructural (figura 4.30) proporciona una interfaz simplificada para facilitar el empleo de sistemas complejos, ocultando así su complejidad, reduciendo el acoplamiento y mejorando la legibilidad del código.

Este patrón se emplea en la base de datos *offline* para proporcionar una interfaz simplificada a las diferentes partes de la base de datos (DAOs).

4.5.5. Patrón Builder

El patrón de diseño builder (figura 4.31) es un patrón de diseño creacional que permite construir objetos complejos paso a paso separando la construcción de la representación. Algunas de las ventajas son la creación de objetos con distintas variaciones sin emplear constructores telescópicos, simplifica el uso de parámetros opcionales y oculta los detalles de construcción del objeto final.

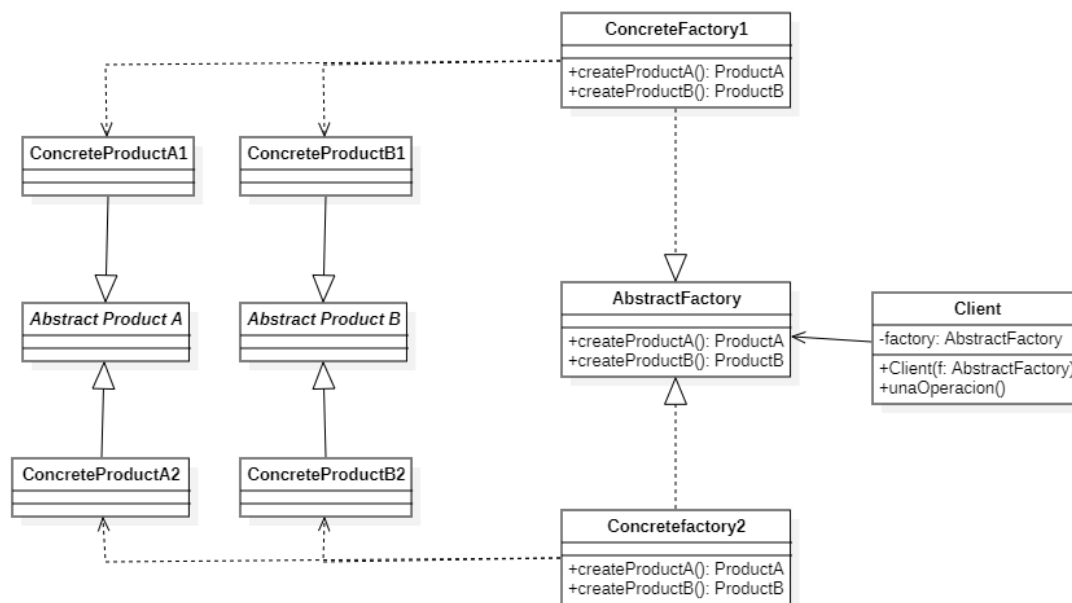


Figura 4.29: Patrón Abstract Factory

Este patrón se emplea en distintos componentes Android, como por ejemplo las notificaciones o los *pop-up* para simplificar su creación debido a su gran número de opciones.

4.5.6. Patrón Adapter

Este patrón (figura 4.32) de diseño estructural permite que objetos con interfaces incompatibles trabajen juntos. Actúa como un puente entre las dos clases, convirtiendo una clase en la que otra espera. Algunas de las ventajas son la capacidad de reutilizar clases, encapsular los cambios o la integración de sistemas legados.

En el caso de nuestra aplicación, se emplea con *RecyclerView* en Android para adaptar las listas a la vista en la que se muestran.

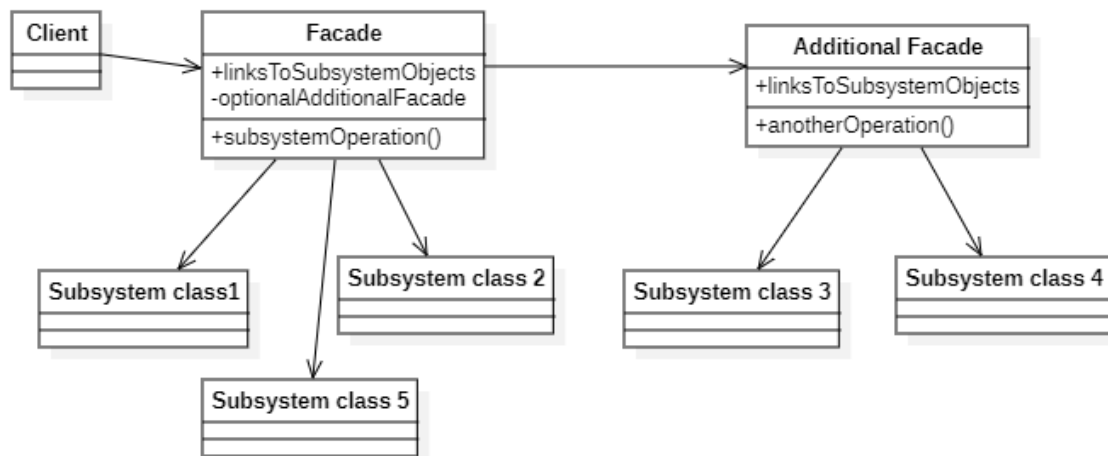


Figura 4.30: Patrón Facade

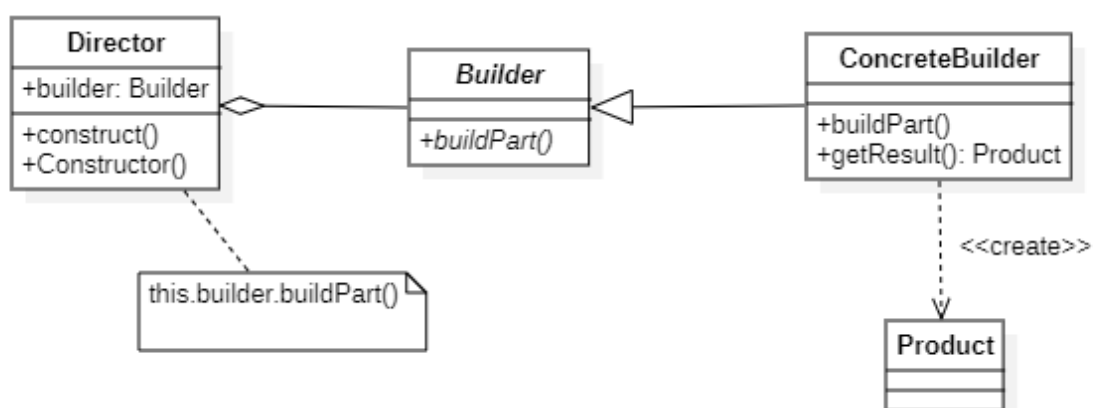


Figura 4.31: Patrón Builder

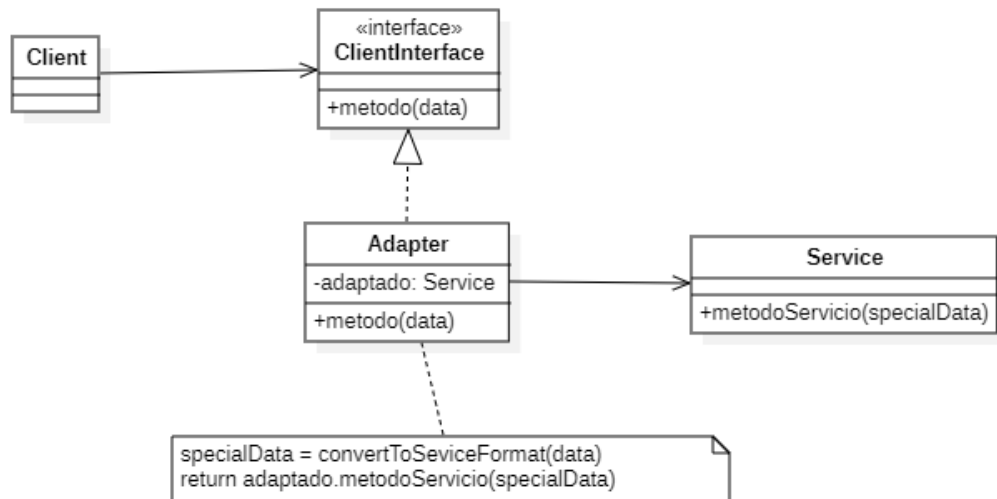


Figura 4.32: Patrón Adapter

Capítulo 5

Implementación

En este capítulo se describen las herramientas empleadas en el desarrollo de la aplicación, así como las pruebas que se han realizado en la aplicación.

5.1. Tecnologías usadas

- **Java**

La funcionalidad de la aplicación Android se ha desarrollado utilizando Java. Se ha utilizado Java en vez de Kotlin debido a la familiaridad con Java y que la aplicación previa ya estaba programada en Java.

- **XML**

XML es un el lenguaje empleado para diseñar interfaces en Android. Permite definir la estructura y apariencia de las pantallas sin necesidad de escribir código directamente. Estos archivos describen la jerarquía de vistas (Views) y grupos de ellas (ViewGroups) mediante el uso de etiquetas. Este enfoque separa el diseño de la vista de la lógica de la aplicación, facilitando el mantenimiento y la adaptación a distintos tipos de pantalla. Además, Android Studio proporciona un editor de diseño con una interfaz que permite añadir y gestionar elementos mediante una interfaz gráfica sin necesidad de escribir código.

- **CEL**

CEL es un lenguaje semántico para la evaluación de expresiones. Se utiliza en el diseño de las reglas de la base de datos Firebase. Está optimizado para evaluar permisos de forma eficiente y segura en entornos distribuidos y su sintaxis clara lo hacen una herramienta muy útil para proteger la información sin necesidad de lógica compleja en el código de la aplicación.

- **Android Studio**

Android Studio es el entorno de desarrollo integrado oficial para la plataforma Android. Está mantenido por Google y por JetBrains. Algunas de las herramientas que ofrece son:

- Compilación con Gradle
- Refactorización a medida para desarrollo en Android
- Chequeo de errores en el código (*lint tools*)

- Integración con Git y GitHub
- Soporte nativo de Firebase y Google Cloud Platform
- Emulador de dispositivos Android integrado

■ **Firebase**

Firebase es un conjunto de servicios en la nube y plataformas de desarrollo ofrecido por Google. En nuestro caso, se ha empleado principalmente Firestore, que es una base de datos NoSQL para almacenar los datos de la aplicación y los recursos necesarios para que esta funcione. Una de las ventajas de Firebase es su integración sencilla con Android, permitiendo implementar características complejas sin necesidad de configurar servidores complejos. Incluye una herramienta de análisis de uso (Firebase Analytics) que permite monitorizar el uso de la base de datos en tiempo real.

■ **Glide**

El framework Glide es una herramienta de gestión de imágenes para Android que facilita la carga y visualización eficiente de imágenes en aplicaciones Android. Ofrece características como carga asíncrona de imágenes; almacenamiento en caché inteligente, decodificación y manipulación de imágenes; y soporte para cargar imágenes desde diferentes fuentes, como recursos locales, URL de internet, y más.

■ **Github**

Github es una plataforma que permite ir llevando un control de versiones mediante el uso de Git. En esta se puede almacenar el control de versiones, además de control de acceso y trabajo colaborativo; seguimiento de errores; gestión de tareas; o integración continua entre otros.

■ **Firebase Emulator Suite**

Firebase Emulator Suite es una herramienta proporcionada por Google que proporciona una serie de herramientas para simular una base de datos Firebase de manera local, permitiendo así a los desarrolladores probar cambios en la base de datos de manera local antes de desplegar los cambios en producción.

■ **Google Maps SDK**

Para la geolocalización de la actividad se ha empleado el SDK de Google Maps. Esta permite la gestión de geolocalización de dispositivos, el uso de mapas ya existentes, seguimiento de rutas de usuarios, cálculo de distancias, vista aérea entre otros. Simplifica el trabajo de desarrollo de la parte de la geolocalización y permite su uso de manera sencilla con cualquier dispositivo Android.

■ **Ground Overlays**

Ground Overlays es una funcionalidad del SDK de Google Maps que permite la superposición de una imagen sobre el mapa de una zona y utilizar toda la funcionalidad del SDK tras la imagen, permitiéndonos de manera sencilla el utilizar el mapa de orientación junto con la geolocalización para realizar las actividades de orientación.

■ **Polylines**

Polylines es una funcionalidad del SDK de Google Maps que permite dibujar líneas y figuras sobre un mapa en base a una serie de coordenadas. En este desarrollo se utiliza para mostrar los recorridos de los participantes durante el proceso de revisión de una actividad.

- **Base de datos Room**

La base de datos Room es una capa de abstracción sobre SQLite que forma parte la biblioteca de persistencia de Android Jetpack. Su principal tarea es simplificar el trabajo con las bases de datos en aplicaciones Android, reduciendo la repetición de código y aportando una solución robusta y sencilla de mantener. Una de las ventajas de Room es que comprueba las consultas en tiempo de compilación, lo que ayuda a detectar errores antes de que se ejecute la aplicación, al contrario que con una base SQLite tradicional, donde los errores solamente se manifiestan en tiempo de ejecución. Otra ventaja es que soporta aplicaciones asíncronas mediante corrutinas o RxJava, facilitando la gestión de hilos y evitando bloqueos en el hilo principal.

- **WorkManager**

WorkManager es una API de Android que opera mediante un sistema que gestiona el ciclo de vida completo de las tareas en segundo plano. Se divide en tres capas, una capa de API, siendo esta una interfaz pública que emplean los desarrolladores, una capa de control, que gestiona la programación y ejecución, y una capa de persistencia, en nuestro caso, la base de datos Room. Una vez se crea la tarea, se serializa y se almacena en la base de datos; se programa empleando el mecanismo óptimo según la versión de Android; una vez el sistema verifica que se cumplen todas las restricciones de la tarea, WorkManager instancia una clase Worker que ejecuta el método *doWork()* en un hilo administrativo. Una vez acaba todo este proceso, se gestionan los resultados, ya sea verificar si la tarea ha tenido éxito, o disparar la siguiente tarea en el caso de un flujo de trabajo.

En nuestra aplicación, una vez el participante se desplaza, se crea una tarea de WorkManager que una vez se verifica que hay conexión a internet, actualiza el estado del participante en la base de datos. Esto también permite la funcionalidad en caso de que se pierda la conexión de manera momentánea y luego se recupere.

5.2. Pruebas de la aplicación

En esta sección se exponen las pruebas realizadas para validar distintos escenarios de la aplicación, asegurando que cumple correctamente con los escenarios. También se han realizado las pruebas realizadas durante el trabajo anterior [12]. Estas pruebas tenían el fin de comprobar que todas las funciones existentes seguían siendo soportadas de forma correcta. En todas estas pruebas se obtuvieron los resultados esperados.

CP01	Mostrar la posición de un participante que no ha empezado la actividad
Descripción	El organizador quiere ver donde se encuentra un participante que no ha empezado la actividad.
Acción	El organizador accede a la vista de Mapa en una actividad iniciada.
Resultado Esperado	La aplicación no muestra la posición del participante.
Resultado Obtenido	Correcto.

Cuadro 5.1: CP1. Posición de un participante que no ha comenzado la actividad

CP02	Posición de un participante que ha acabado la actividad
Descripción	Un participante se aproxima a la meta tras alcanzar todas las balizas.
Acción	Durante el transcurso de la actividad, un participante finaliza el recorrido.
Resultado Esperado	La aplicación oculta la posición del participante.
Resultado Obtenido	Correcto.

Cuadro 5.2: CP2. Posición de un participante que ha finalizado la actividad

CP03	Participante sin conexión a internet inicia una actividad
Descripción	Un participante que no dispone de conexión a internet inicia la actividad.
Acción	Iniciar una actividad de orientación ya registrada sin conexión a internet.
Resultado Esperado	La aplicación da comienzo a la actividad. Actualiza el estado de la participación y la hora de salida. Programa la actualización de los datos en la base de datos cuando se disponga de conexión a internet.
Resultado Obtenido	Correcto.

Cuadro 5.3: CP3. Un participante comienza una actividad sin conexión a internet

CP04	Participante sin conexión a internet finaliza una actividad
Descripción	Un participante que no dispone de conexión a internet finaliza la actividad.
Acción	Completar una actividad de orientación sin conexión a internet.
Resultado Esperado	La aplicación da por finalizada la actividad. Actualiza el estado de la participación y la hora de finalización. Programa la actualización de los datos de la base de datos cuando se disponga de conexión a internet.
Resultado Obtenido	Correcto.

Cuadro 5.4: CP4. Un participante finaliza una actividad sin conexión a internet

CP05	Inscribirse en una actividad y descargar los datos de la misma
Descripción	El usuario, tras haber localizado una actividad a partir de su identificador, se inscribe en ella.
Acción	El usuario pulsa sobre el botón inscribirse de la actividad e introduce la clave correcta.
Resultado Esperado	El usuario queda inscrito en la actividad. Se crea un nuevo documento de <i>Participación</i> en Cloud Firestore. Se actualiza la lista de participantes en la actividad. Se descargan los datos de la actividad, de la <i>template</i> y de la participación en local. Se informa al usuario de que la inscripción se ha completado y los datos se han descargado.
Resultado Obtenido	Correcto.

Cuadro 5.5: CP5. Un participante se inscribe en una actividad

CP06	Comprobar la posición de los participantes durante una actividad como organizador
Descripción	El organizador accede a la vista de mapa para comprobar las localizaciones de los participantes.
Acción	El organizador selecciona la vista de mapa durante la supervisión de la actividad.
Resultado Esperado	La aplicación muestra el mapa y las localizaciones de los participantes que están realizando la actividad. Esta localización se actualiza cuando los participantes se desplazan.
Resultado Obtenido	Correcto.

Cuadro 5.6: CP6. Comprobar la posición de los participantes

CP07	Comprobar el progreso de los participantes durante una actividad como organizador
Descripción	El organizador accede a la vista general para comprobar el progreso de los participantes.
Acción	El organizador selecciona la vista general durante la supervisión de la actividad.
Resultado Esperado	La aplicación muestra la información del progreso de los participantes (estado, tiempo transcurrido, balizas alcanzadas y respondidas). Esta información se actualiza cuando los participantes progresan en la actividad.
Resultado Obtenido	Correcto.

Cuadro 5.7: CP7. Comprobar el progreso de los participantes

CP08	Ver el detalle de las balizas de un participante durante una actividad
Descripción	El organizador comprueba el detalle de las balizas de un participante.
Acción	El organizador selecciona a un participante en la vista general.
Resultado Esperado	La aplicación muestra la información de las balizas del participante (Nº, nombre, estado de la respuesta y momento de alcance).
Resultado Obtenido	Correcto.

Cuadro 5.8: CP8. Ver el detalle de las balizas de un participante durante una actividad

CP09	Ver la respuesta a una baliza de un participante durante una actividad
Descripción	El organizador comprueba la respuesta a una baliza de un participante.
Acción	El organizador selecciona una baliza en el detalle de las balizas de un participante.
Resultado Esperado	La aplicación muestra la información de la baliza junto con la respuesta del usuario.
Resultado Obtenido	Correcto.

Cuadro 5.9: CP9. Ver la respuesta a una baliza de un participante durante una actividad

CP10	Ver el recorrido de un participante durante una actividad
Descripción	El organizador comprueba el recorrido de un participante durante una actividad.
Acción	El organizador pulsa el botón de "Ver Track" en el detalle de balizas de un participante.
Resultado Esperado	La aplicación muestra la pantalla con el recorrido del participante, donde el organizador puede ver el recorrido del participante a lo largo del tiempo.
Resultado Obtenido	Correcto.

Cuadro 5.10: CP10. Ver el recorrido de un participante durante una actividad

CP11	Comprobar el desempeño de los participantes tras la actividad
Descripción	El organizador comprueba el desempeño de varios participantes tras una actividad.
Acción	El organizador selecciona la actividad deseada y pulsa el botón de "Ir a la actividad".
Resultado Esperado	La aplicación muestra la pantalla de revisión con la información del desempeño de los participantes (estado, tiempo transcurrido, balizas alcanzadas y respondidas).
Resultado Obtenido	Correcto.

Cuadro 5.11: CP11. Comprobar el desempeño de los participantes tras la actividad

CP12	Ver el recorrido de varios participantes tras una actividad
Descripción	El organizador comprueba el recorrido de varios participantes.
Acción	El organizador selecciona varios participantes y pulsa el botón de "Ver Track" en la pantalla de la revisión.
Resultado Esperado	La aplicación muestra la pantalla con el recorrido de los participantes, donde el organizador puede ver el recorrido de los participantes a lo largo del tiempo.
Resultado Obtenido	Correcto.

Cuadro 5.12: CP12. Ver el recorrido de varios participantes tras una actividad

CP13	Responder a la pregunta de una baliza sin conexión
Descripción	Un participante contesta a la pregunta de una baliza sin conexión a internet.
Acción	El participante selecciona una respuesta a la pregunta de una baliza y pulsa el botón de "Responder".
Resultado Esperado	La aplicación registra la respuesta y programa una subida a la base de datos en el momento que se restaure la conexión a internet.
Resultado Obtenido	Correcto.

Cuadro 5.13: CP13. Responder a la pregunta de una baliza sin conexión

Capítulo 6

Conclusiones y trabajo futuro

6.1. Conclusiones

En este apartado se expondrán las conclusiones obtenidas del desarrollo del proyecto junto con posibles líneas de trabajo futuras

6.1.1. Objetivos conseguidos

Durante el desarrollo de este trabajo de fin de grado se han conseguido los siguientes objetivos:

- **Aplicar técnicas del diseño centrado en el usuario, con el fin de mejorar el diseño de la aplicación.**

Mediante el uso de principios, heurísticas y guías de usabilidad junto con tres pruebas de usabilidad con usuarios, se ha mejorado el diseño de la interfaz de usuario de los profesores, además de añadir varias funcionalidades y cambios solicitados por los usuarios para hacer esta más acorde a sus necesidades.

- **Identificar y explorar distintas alternativas para realizar el seguimiento y análisis del recorrido de los usuarios en caso de no disponer de conexión a internet de manera constante**

Se han estudiado distintas alternativas para realizar el seguimiento y análisis del recorrido de los usuarios cuando estos no disponen de conexión a internet. Se ha optado por emplear WorkManager junto con una capa de persistencia debido a que solamente requiere cambios en la parte de la aplicación y no implica cambios por parte del usuario, además de no necesitar comprar, desplegar y mantener una serie de dispositivos para emplear otras técnicas alternativas de seguimiento de los participantes.

- **Posibilidad de realizar carreras sin acceso a internet en los dispositivos móviles.**

Empleando la API de WorkManager e implementando una capa de persistencia en la aplicación, se ha conseguido implementar una funcionalidad que permite realizar las carreras de orientación sin acceso a internet para posteriormente sincronizar los datos de la carrera con

la base de datos. Esta funcionalidad permite que una vez el participante se registra en la actividad, pueda realizar todo el proceso de participar en una actividad de orientación de manera transparente para él.

- **Facilitar el seguimiento de la actividad de los participantes por parte de los docentes.**

Mediante el diseño centrado en el usuario, se han empleado técnicas de análisis de usuario y pruebas de usabilidad para construir una nueva interfaz que muestre mejor la información a los organizadores y les permita realizar el seguimiento de las actividades de manera más efectiva. Además, se ha añadido la posibilidad de conocer la localización actual de los participantes mientras estos realizan una actividad.

6.2. Trabajo futuro

Durante las pruebas con usuarios, estos expusieron una serie de nuevas necesidades que, debido al alcance del Trabajo de Fin de Grado, no se han implementado:

- **Posibilidad de añadir varios organizadores.**

Debido a que las actividades con alumnos de primaria suelen necesitar varios profesores que monitoricen la actividad para asegurar el bienestar de los niños, se solicitó que se pudiesen añadir varios organizadores a la actividad para poder monitorizarla desde sus dispositivos. Esto se puede hacer actualmente con una solución ad-hoc, registrándose en varios dispositivos con el mismo usuario organizador para poder realizar el seguimiento de la actividad. Para añadir varios organizadores con diferentes usuarios, sería necesario cambiar el modelo de dominio y el diseño de la base de datos, registrando a todos los organizadores en una lista para posteriormente poder acceder a la actividad.

- **Añadir opciones de QR o de URL para unirse a una actividad.**

Los usuarios nos comentaron que surgen habitualmente problemas a la hora de que los participantes se registren en la actividad, por lo que sería interesante añadir alguna manera de registrarse que se pudiese compartir de manera sencilla. Actualmente, las aplicaciones de Android pueden abrir URLs de manera automática si se les da permiso, así que una de las maneras de hacerlo sería mapear un prefijo de una URL para que se abriese de manera automática con la aplicación y nos llevase a la pantalla de registro o incluso se registrase directamente.

- **Añadir avisos en tiempo real tanto al organizador como a los participantes.**

Una de las principales preocupaciones de los usuarios al realizar una actividad de orientación es que ocurra algún percance y haga falta avisar a los participantes para que regresen a la meta. De la misma manera, se podría avisar a los participantes automáticamente si salen del área de la actividad o los participantes podrían avisar al profesor en caso de que tengan un problema y requieran de ayuda inmediata. Los avisos a participantes se podrían implementar con un publish-subscribe que notifique a los participantes en caso de que el organizador lo necesite. Esto requiere de un análisis previo para especificar cuáles son los requisitos más específicos de esta funcionalidad. En cuanto a los avisos si un participante se sale de la zona de la actividad, se podría añadir que cuando se actualiza la posición del participante, si esta está fuera de los límites del mapa de la actividad, se le notifique para que regrese al área

de la actividad. Los avisos por parte del participante al organizador se podrían implementar como una cola de mensajes que se envían del participante al organizador y se recogen en una vista específica para los mismos, lanzando una notificación en caso de ser urgentes.

- **Exportar los datos de las actividades y los recorridos a plataformas educativas para realizar el seguimiento en ellas.**

Se puede crear una funcionalidad para exportar los datos de las actividades en algún formato de datos específico que permita su tratamiento en plataformas externas. Se debería estudiar el formato que admiten otras aplicaciones y plataformas para adaptar la extracción y transformación de datos de la actividad para su posterior carga en las mismas.

A continuación, se añaden algunas funcionalidades que surgieron en el análisis de usuarios previo y que serían interesantes de implementar para expandir la funcionalidad de la aplicación, pero que se descartó su implementación debido a los límites del Trabajo de Fin de Grado:

- **Posibilidad de añadir nuevas plantillas de actividad en otras ubicaciones dentro de la aplicación.**

Esta funcionalidad no debería de requerir muchos cambios a nivel de la base de datos, ya que está construida de tal manera que se puedan añadir otros mapas y actividades en otras localizaciones de manera sencilla. La dificultad radicaría en construir una interfaz que permita realizar esta tarea de manera sencilla, ya que actualmente la localización de las balizas está impresa de manera permanente en el mapa de orientación y los cambios requerirían un cambio en la imagen del mapa. Esto se podría implementar mediante el uso de marcadores en el mapa con una imagen personalizada que señalasen las balizas. Otro de los problemas que podrían surgir nace de la dificultad de colocar de manera precisa el mapa, ya que un pequeño error de posicionamiento se añade a otros errores en la localización de los participantes como el error en la localización por GPS y puede conllevar a grandes errores a la hora de realizar el seguimiento y análisis posterior de una carrera de orientación.

- **Añadir una ayuda que muestre al participante el recorrido hacia la siguiente baliza.**

Mediante el uso de *polylines* de manera similar a cómo se emplean en la revisión del recorrido, se podría implementar un tipo de ayuda para los participantes que les muestre el camino hacia la siguiente baliza, la baliza más cercana o incluso el inicio o fin de la prueba de orientación en caso de ser necesario. La dificultad de esto radica en programar el recorrido de tal manera que tenga en cuenta los accidentes de la localización para evitar posibles problemas de cara a dirigirse a la siguiente baliza.

- **Añadir intentos a las pruebas.**

Se podría añadir una funcionalidad que permita reintentos en las pruebas de las balizas y luego que la información de estos se le muestre también al organizador.

- **Añadir nuevos tipos de prueba.**

Se pueden añadir nuevos tipos de prueba, como el hacerle una foto a un elemento del entorno, escanear un QR o unir varias opciones para crear descripciones que se ajusten a uno o varios elementos del entorno. Algunas de estas pruebas requerirían revisión por parte del organizador, que podría realizarlo de manera inmediata o posterior a la prueba. De la misma manera, se podría ocultar el resultado de las respuestas a las pruebas para que lo conozcan una vez ha finalizado la actividad o cuando lo desee el organizador.

A lo largo del proceso de diseño, debido a los límites de tiempo y alcance del Trabajo de Fin de Grado, hay una serie de limitaciones que quedan como trabajo futuro:

- La funcionalidad sin conexión a internet no tiene en cuenta la posibilidad de que un participante realice la actividad sin conexión, y antes de subir los datos, realice la actividad con conexión en otro dispositivo y no actualice los datos con la participación sin conexión (por ejemplo, desinstalando la aplicación). Se ha tomado esto como un caso de uso altamente improbable, pero es uno de los posibles ejemplos de problemas que conlleva la implementación de una funcionalidad sin conexión. Sería interesante plantear un sistema de resolución de conflictos por parte del servidor para decidir qué hacer con los datos en caso de que lleguen de dos fuentes distintas.
- Por limitaciones de tiempo y organizativas, no se ha realizado una prueba de usabilidad final con usuarios reales. Sería interesante haber realizado una prueba sobre el terreno con una clase de primaria con los que verificar de manera correcta que la aplicación es usable para ellos. Además, debido a que el desarrollo centrado en el usuario se ha centrado más en la parte de la evaluación formativa de las propuestas por encima de su evaluación cualitativa, se han enfocado las pruebas con usuarios a la valoración cualitativa de las propuestas para asegurarnos de que los cambios en la aplicación son correctos y mejoran la experiencia de los usuarios. Una de las posibles vías de trabajo futuro sería realizar un análisis cuantitativo de las mejoras en la aplicación, como el número de clicks o el tiempo que han tardado en realizar una tarea
- Se ha incrementado el número de acciones que se realizan sobre la base de datos. Si bien las pruebas realizadas no han sobrepasado el límite gratuito de Firebase, es posible que en situaciones con muchos participantes realizando la actividad al mismo tiempo, pueda resultar en un gasto por uso de la base de datos. Se podría optimizar más el número de acciones que se realizan sobre la base de datos, pero por limitaciones de tiempo no se ha podido profundizar en este aspecto. Algunos ejemplos consisten en actualizar la localización de los participantes no en tiempo real, sino según un intervalo de tiempo, o agrupar las subidas de recorrido de los usuarios en un tiempo determinado para subirlas todas de una vez, teniendo en cuenta la distancia que un usuario puede desplazarse en dicho intervalo.
- Se han mantenido las reglas de seguridad implementadas en el trabajo anterior. Si bien esto es suficiente para un uso cotidiano de la aplicación, sería interesante explorar posibles mejoras de este apartado para asegurar la seguridad de los participantes de cara al acceso de su información personal.

6.3. Valoración personal

Este proyecto ha supuesto todo un reto en lo personal y una gran oportunidad para aprender sobre nuevas metodologías y tecnologías de gran relevancia en el desarrollo de software.

En cuanto a lo personal, debido a varios factores, entre ellos la incorporación al mundo laboral ha supuesto un gran reto que me ha enseñado a gestionar mejor el tiempo, priorizar tareas y a compaginar trabajo y vida personal.

En cuanto a la parte técnica, la posibilidad de aprender acerca de una tecnología tan extendida como es el desarrollo móvil en Android, el uso de una arquitectura Cliente-Servidor, una base de

datos NoSQL como Cloud Firestore, el uso de la API de Google Maps y la implementación de una funcionalidad sin conexión ha sido muy interesante. Además, el emplear técnicas de desarrollo centrado en el usuario y el empleo de guías de diseño y conceptos de usabilidad han enriquecido mucho mi visión acerca del desarrollo software y de la importancia de crear aplicaciones y funcionalidades que le sean útiles a los usuarios.

Apéndice A

Manual de usuario

A continuación, se expone el funcionamiento de la aplicación desde el punto de vista de los usuarios. Mediante capturas de pantalla, se mostrarán las principales características de la aplicación y las acciones que los usuarios deben seguir para completar las tareas principales.

A.1. Inicio de sesión y registro

La identificación en esta aplicación funciona mediante e-mail y contraseña. Para registrarse en esta aplicación, habrá de cumplimentar el formulario descrito en la figura A.1.2. Además, tendrá que verificar su usuario accediendo al correo introducido y haciendo click en el enlace en el correo de verificación. Además de iniciar sesión, también se podrá recuperar la contraseña en caso de ser necesario.

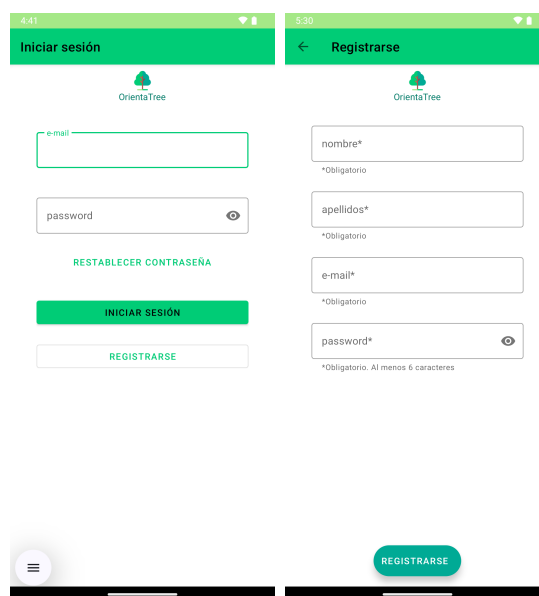


Figura A.1: Pantallas de inicio de sesión y registro

A.2. Pantalla de inicio

A continuación, se mostrará la pantalla con las actividades del usuario. Estas vienen separadas en función de si están en curso (Figura A.2.1), han finalizado (Figura A.2.2) o están previstas para el futuro (Figura A.2.3). Además, se puede acceder a un menú auxiliar con varias funciones que nos permiten navegar a otras funcionalidades de la aplicación (Figura A.2.4).

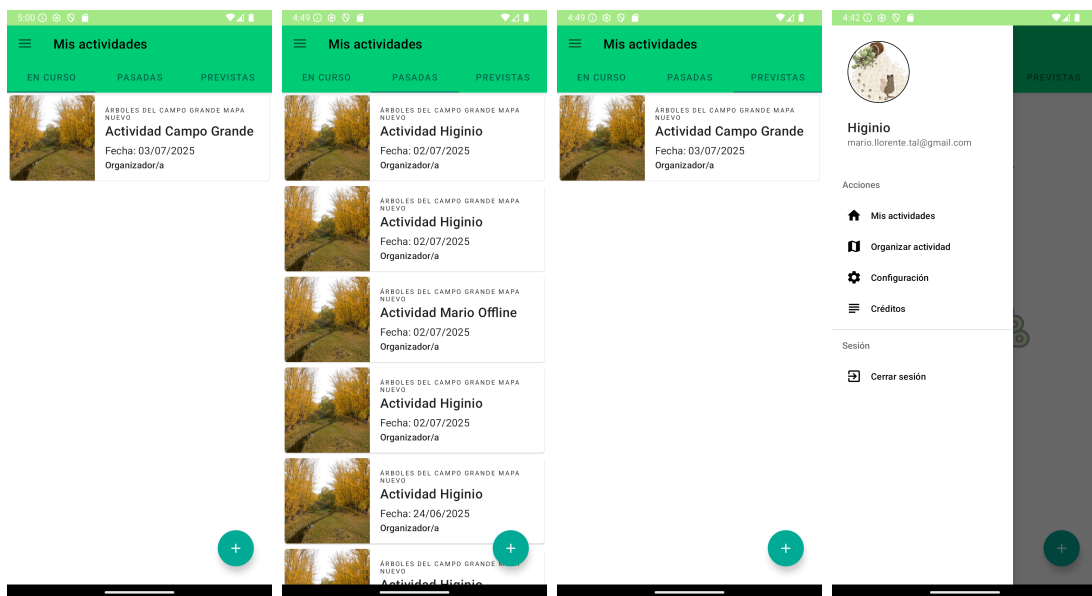


Figura A.2: Pantallas con la información de las actividades y menú lateral

A.3. Configuración de perfil

Si seleccionamos el botón de Configuración del menú lateral (Figura A.3.1), accederemos a una pantalla donde se nos permitirá editar nuestro perfil. Se podrá cambiar la foto de perfil, el nombre y los apellidos. Además, si hacemos click en la esquina superior derecha (Figura A.3.3), tendremos opciones como cerrar sesión o eliminar cuenta.

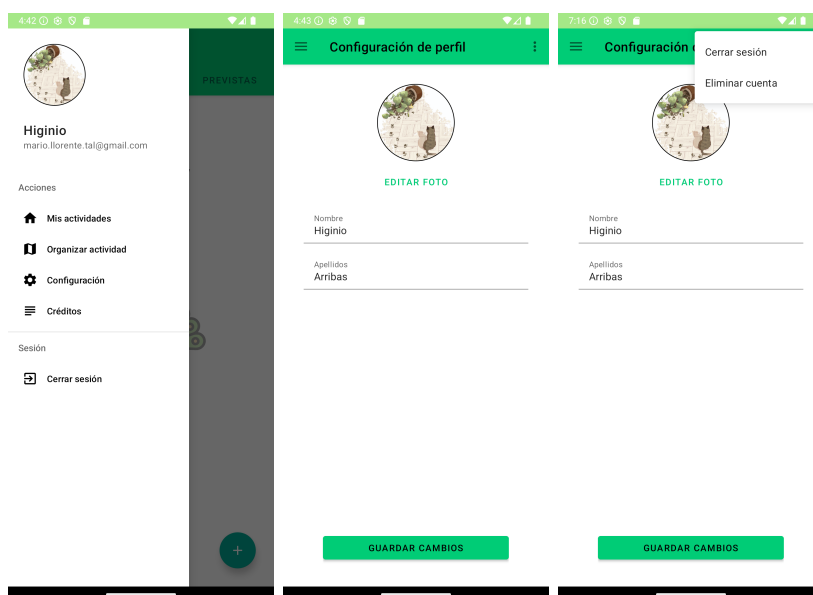


Figura A.3: Pantallas con menú lateral y configuración del usuario

A.4. Organizar actividad

Si desde el menú lateral seleccionamos la opción de Organizar Actividad, se nos llevará a una pantalla desde la que se podrá programar una actividad a partir de una plantilla existente como se puede ver en la figura A.4.2 . Estas plantillas están protegidas por contraseña para evitar que los participantes puedan acceder previamente a las actividades y trivializar la realización de las mismas. una vez introducida la contraseña, se podrá programar la fecha y hora de la actividad, cambiar la modalidad, o activar la ayuda con la ubicación para los participantes como se muestra en la figura A.5.1 .Una vez introducida esta información, se podrá introducir un nombre de la actividad, seleccionando uno por defecto en caso de no introducirse. Cuando confirmemos el nombre de la actividad, se creará y se nos mostrará el ID de la actividad y la contraseña para unirse. Esta información también puede consultarse si accedemos a la actividad desde la lista y seleccionamos el botón de ver claves.

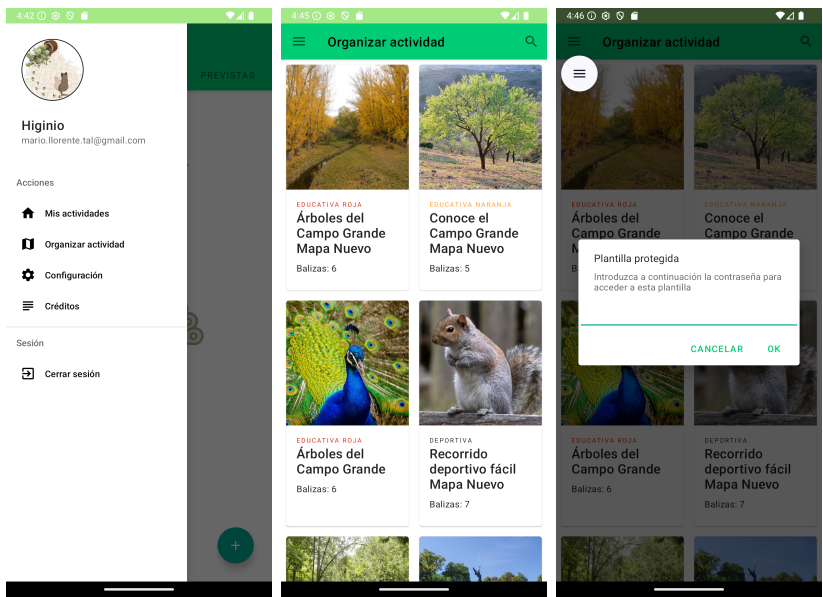


Figura A.4: Pantallas con menú lateral y plantillas de actividad

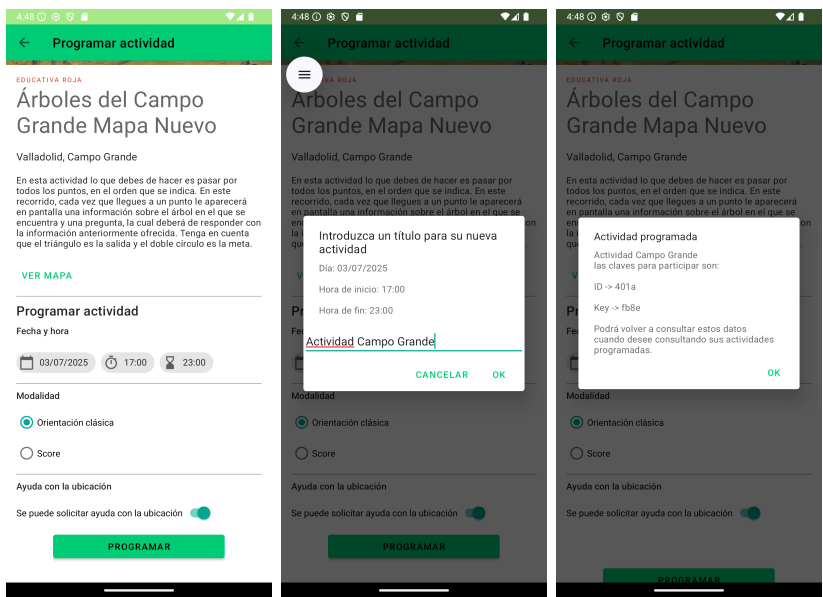


Figura A.5: Pantallas con configuración de la actividad

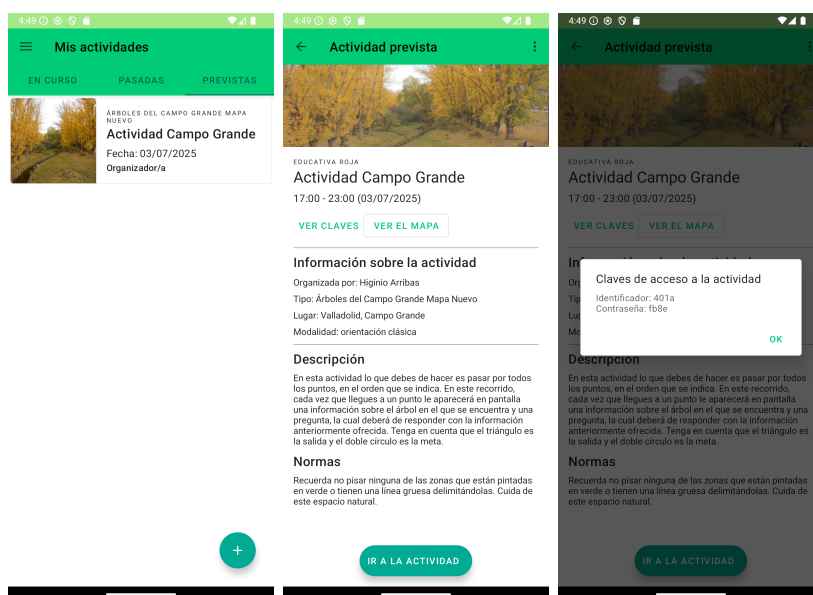


Figura A.6: Pantallas con información de una actividad

A.5. Inscribirse en una actividad

Para inscribirse en una actividad, el organizador debe proporcionar al participante (o participantes) el ID y la clave de la actividad para poder unirse. El participante deberá seleccionar el botón flotante con un $+$ en la pantalla de actividades para buscar una nueva actividad. Una vez en la pantalla de búsqueda, deberá introducir el ID proporcionado para buscar la actividad. Si la actividad existe, pulsando el botón de inscribirse se mostrará una ventana en la que introducir la contraseña proporcionada. Si la contraseña es correcta, se inscribirá en la actividad y se descargarán los datos necesarios para su realización sin conexión en caso de ser necesario.

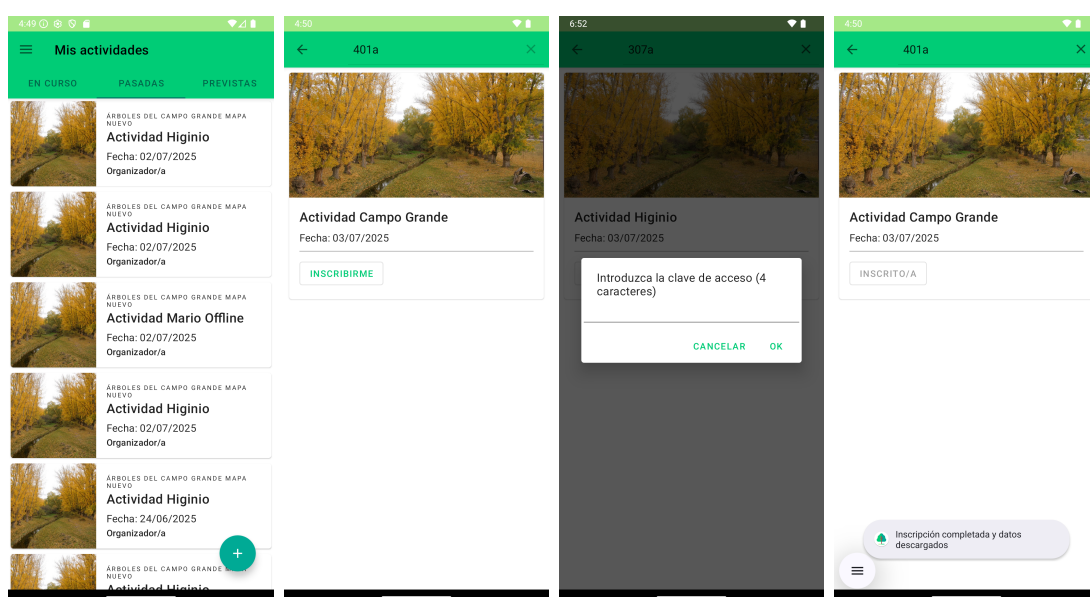


Figura A.7: Pantallas de inscripción en una actividad

A.6. Participar en una actividad

Una vez el participante se ha registrado en una actividad y ha llegado a su hora de inicio, esta aparecerá en la pestaña de *en curso*. una vez se accede a ella, se deberá de pulsar el botón comenzar, que iniciará la participación y llevará al usuario al mapa, donde podrá comprobar su progreso.

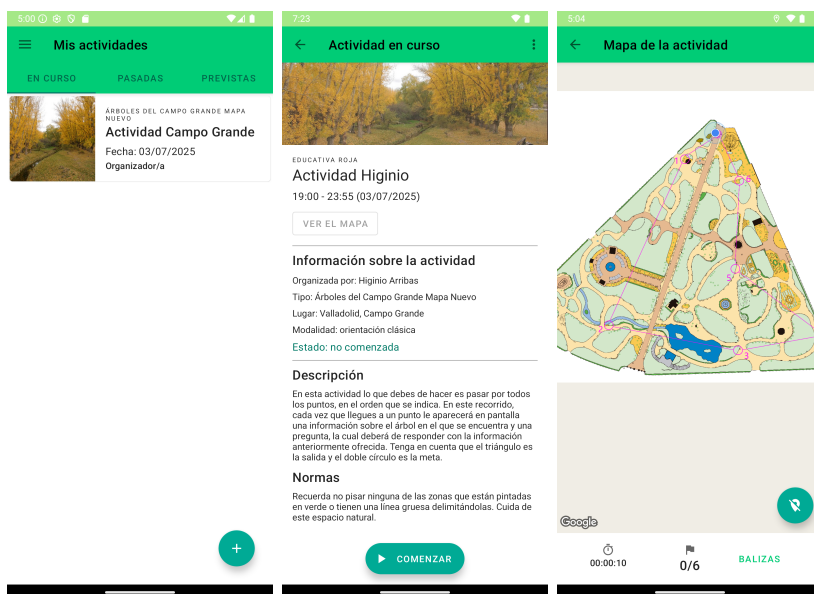


Figura A.8: Pantallas de inicio de participación

Si se selecciona el botón de balizas situado en la esquina inferior derecha, el participante podrá acceder a una vista donde se muestran todas las balizas que ha alcanzado, actualizándose según las vaya alcanzando. Si selecciona una de ellas, podrá contestar a la pregunta planteada.

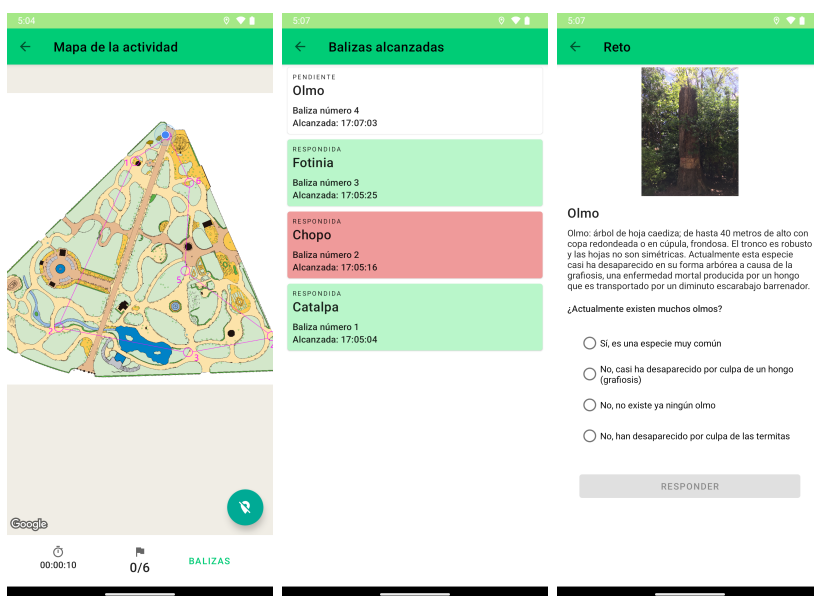


Figura A.9: Pantallas de participación y balizas

A.7. Supervisión de una actividad

Si el organizador desea supervisar una actividad, debe dirigirse a la actividad en curso y seleccionar el botón de ir a la actividad. Se desplegará una pantalla con tres ventanas con información acerca de los participantes. En la primera (Figura A.11.1) se mostrará el mapa con las localizaciones actuales de los participantes con conexión a internet. En la segunda (Figura A.11.2) se mostrará la información actual del progreso de los participantes. Se puede ver el nombre, el estado de la participación, el tiempo que llevan en la actividad y el progreso en las balizas. Si se selecciona uno de los participantes, se desplegará un *pop-up* con información acerca de su progreso en las balizas (Figura A.11.1). Desde allí, se podrá comprobar el resultado de las pruebas de las balizas (Figura A.11.2) o revisar el recorrido del participante (Figura A.11.3). Para volver, basta con tocar fuera del *pop-up*.

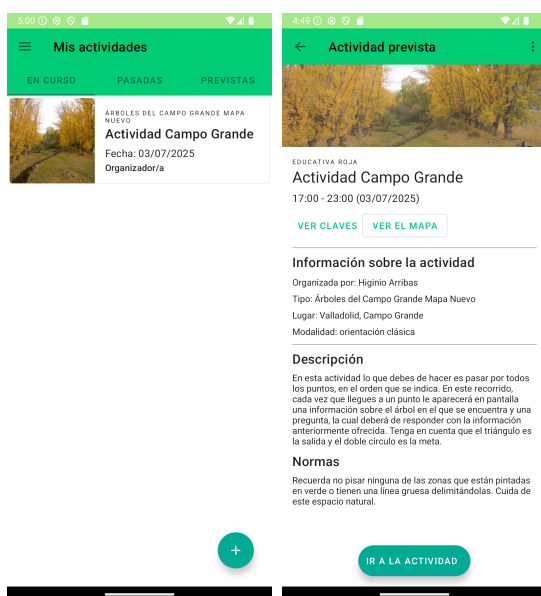


Figura A.10: Pantallas de actividad del organizador

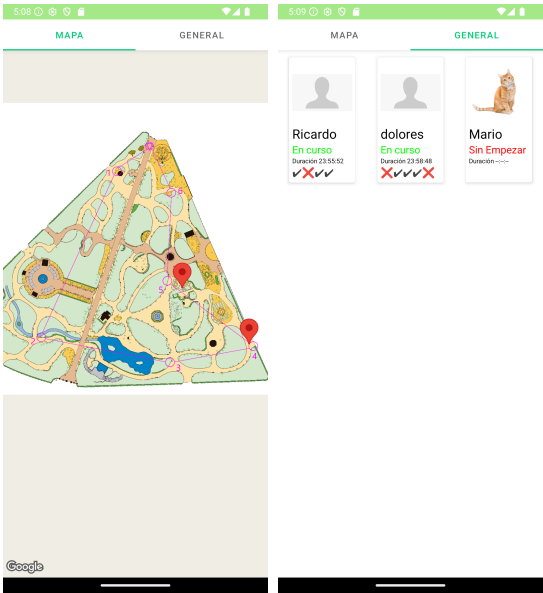


Figura A.11: Pantallas con información de los participantes

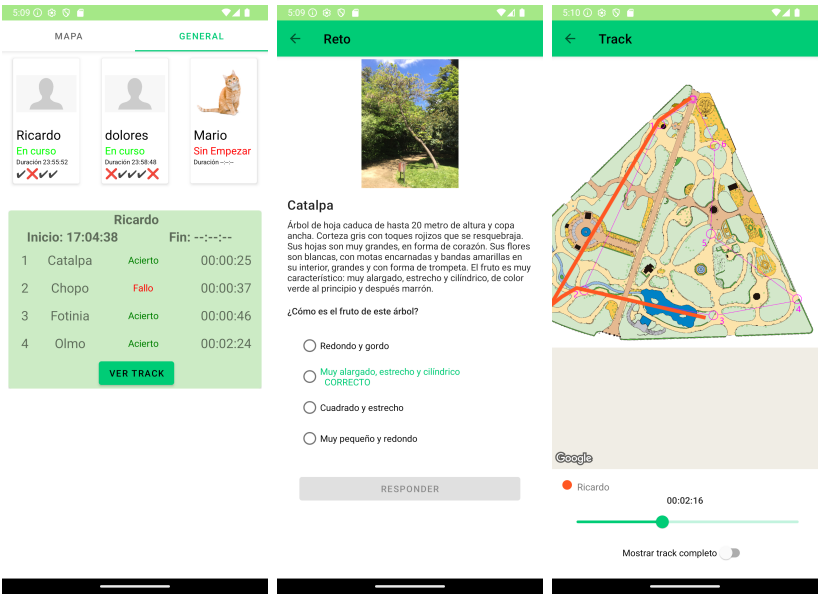


Figura A.12: Pantallas con información de las balizas y el recorrido de un participante

A.8. Revisión de una actividad

Si el organizador desea revisar una actividad, debe dirigirse a la actividad ya finalizada y pulsar el botón de ir a la actividad. Se desplegará una pantalla con información de los participantes (Figura A.13.2) similar a la ya existente en la supervisión de la actividad. La funcionalidad añadida durante la revisión es la capacidad de seleccionar varios participantes (Figura A.14.1) y comparar sus recorridos a la vez, tal y como se muestra en la figura A.14.2 .

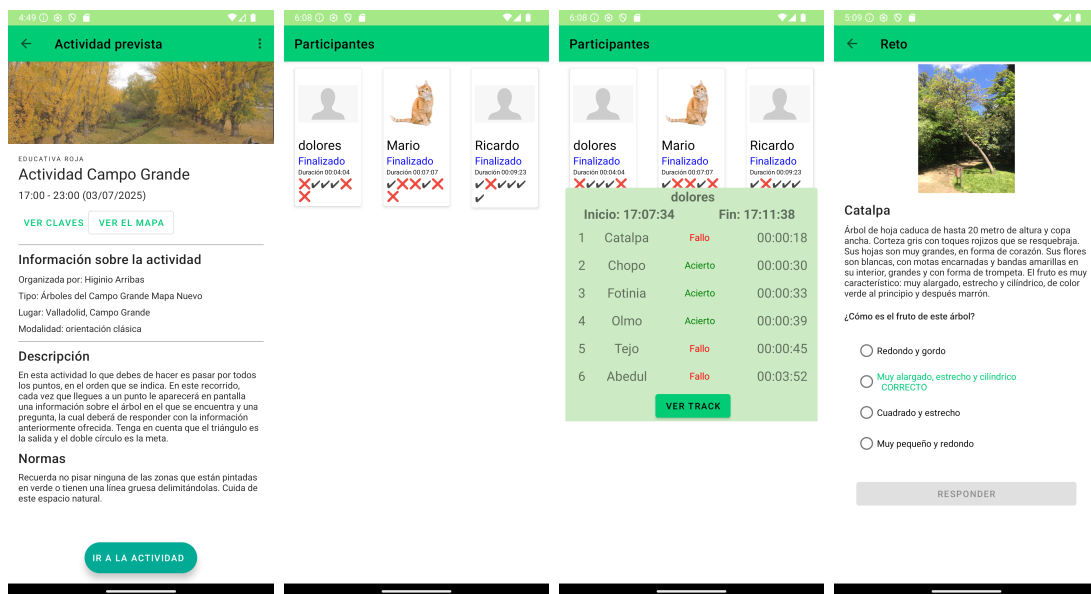


Figura A.13: Pantallas con información de los participantes y las balizas

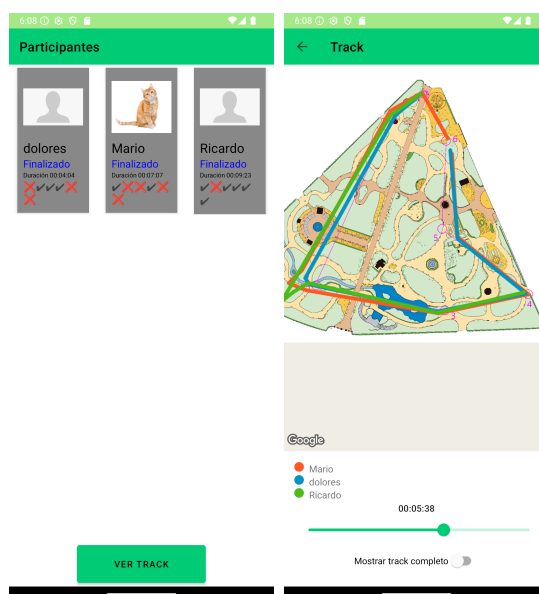


Figura A.14: Pantallas con selección de participantes e información del recorrido

A.9. Revisión de una actividad como participante

Si un participante desea revisar su desempeño en una actividad, deberá dirigirse a la actividad y desplegar el menú situado en la esquina superior derecha y seleccionar la opción de "Mi participación"(Figura A.15.1). Esto le llevará a una pantalla donde podrá consultar sus resultados generales (Figura A.15.2), ver su recorrido pulsando el botón de "Ver track"(Figura A.15.3) o comprobar sus respuestas a las pruebas de las balizas si selecciona el botón de balizas (Figura A.15.4).

A.9. REVISIÓN DE UNA ACTIVIDAD COMO PARTICIPANTE

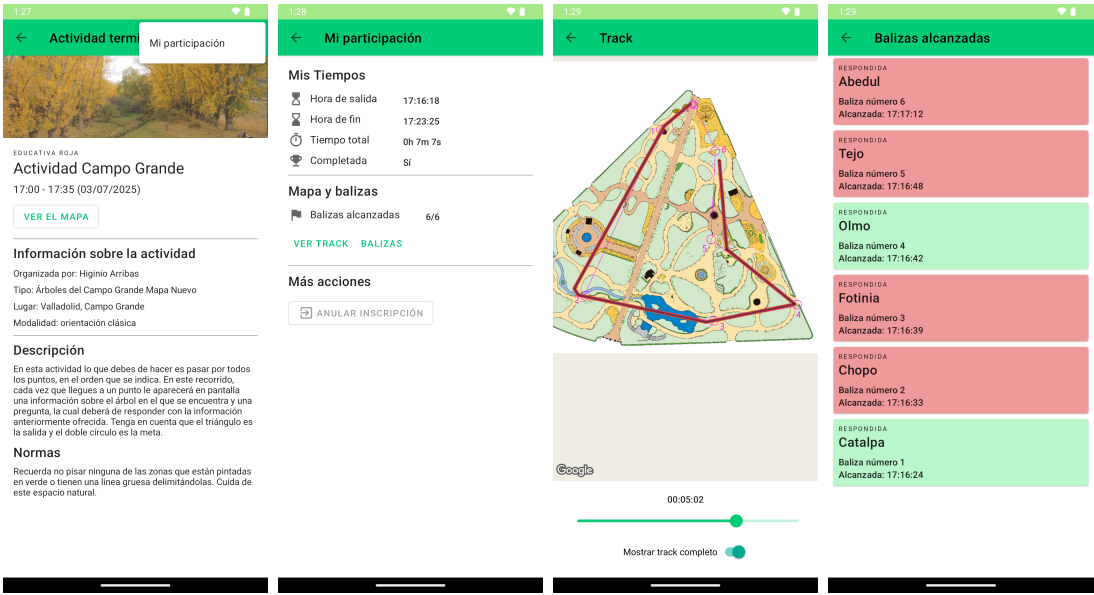


Figura A.15: Pantallas con información del desempeño propio del participante, su recorrido y las respuestas de las balizas

Apéndice B

Enlaces de interés

- Repositorio Github con el código de la aplicación:
<https://github.com/Subid99/TFGOrientatree2.0>

Bibliografía

- [1] C. Abelairas Gómez, S. López García y D. Rodríguez Río, «Actividades físicas organizadas en la naturaleza,» 12 de mayo de 2012. dirección: <https://altorendimiento.com/actividades-fisicas-organizadas-en-la-naturaleza/>.
- [2] F. E. de Orientación, «Historia de la Federación Española De Orientación,» 2023. dirección: <https://www.fedo.org/web/ficheros/federacion/orientacion/Historia-FED0.pdf>.
- [3] C. Contreras Rodríguez, «Unidad didáctica de orientación: ‘No te pierdas’. Propuesta didáctica para 4º ESO,» EFDeportes, inf. téc., 1 de mayo de 2011. dirección: <https://www.efdeportes.com/efd156/unidad-didactica-de-orientacion-para-eso.htm>.
- [4] J. A. Muñoz-Cristóbal, I. M. Jorrín-Abellán, J. I. Asensio-Pérez, A. Martínez-Monés, L. P. Prieto e Y. Dimitriadis, «Supporting Teacher Orchestration in Ubiquitous Learning Environments: A Study in Primary Education,» *IEEE Transactions on Learning Technologies*, vol. 8, n.º 1, págs. 83-97, 2015. DOI: 10.1109/TLT.2014.2370634.
- [5] S. García Delgado, «La orientación deportiva en la educación física escolar,» Trabajo de Fin de Grado, Universidad de Valladolid, 1 de dic. de 2017. dirección: <https://uvadoc.uva.es/bitstream/handle/10324/29323/TFG-L1916.pdf;jsessionid=4BF8157380668FD3132F689C60881sequence=1>.
- [6] A. D. G. montaña Pegaso, «Guía de orientación,» A. D. Grupo montaña Pegaso, inf. téc., 1 de mayo de 2015. dirección: https://www.montanapegaso.es/carteles/2015/orientacion2015_info.pdf.
- [7] F. E. de Orientación, *Mapa Orientación La Quitranera*, 6 de jun. de 2020. dirección: <https://www.fedo.org/web/ultimas-noticias/4666-x-edicion-del-concurso-de-trazados>.
- [8] F. A. de Montañismo para Montaña Segura, *Brujula de orientación*. dirección: <https://www.catedu.github.io/cartografia-para-montaneros/31-la-brujula.html>.
- [9] FEDO. «Lista de software para orientación.» FEDO, ed. (22 de nov. de 2022), dirección: <https://www.fedo.org/web/seccion-tic/software>.
- [10] iOrienteering. «iOrienteering.» iOrienteering, ed. (), dirección: <https://www.iorienteering.com/>.

- [11] U. of Vienna. «COMPASS Orienteering.» (), dirección: <https://compass.schmelz.univie.ac.at/>.
- [12] G. Rodríguez González, «OrientaTree. Aplicación móvil para la realización de actividades educativas geolocalizadas en el medio natural,» UVA, inf. téc., 2021. dirección: <https://uvadoc.uva.es/handle/10324/50459>.
- [13] D. A. Norman, *The design of everyday things*. [New York]: Basic Books, 2002, ISBN: 0465067107 9780465067107. dirección: http://www.amazon.de/The-Design-Everyday-Things-Norman/dp/0465067107/ref=wl_it_dp_o_pC_S_nC?ie=UTF8&colid=151193SNGKJT9&coliid=I262V9ZRW8HR2C.
- [14] Interaction Design Foundation - IxDF, «What is User Centered Design (UCD)?» En *Interaction Design Foundation - IxDF*. 5 de jun. de 2016. dirección: <https://www.interaction-design.org/literature/topics/user-centered-design>.
- [15] S. Pozdniakov, R. Martinez-Maldonado, Y.-S. Tsai y col., «The Question-driven Dashboard: How Can We Design Analytics Interfaces Aligned to Teachers' Inquiry?» En *LAK22: 12th International Learning Analytics and Knowledge Conference*, ép. LAK22, New York, NY, USA: Association for Computing Machinery, 2022, 175–185, ISBN: 9781450395731. DOI: 10.1145/3506860.3506885. dirección: <https://doi.org/10.1145/3506860.3506885>.
- [16] C. Larman, *Agile and Iterative Development: A Manager's Guide*, ép. Agile software development series. Addison-Wesley, 2004, ISBN: 9780131111554. dirección: <https://books.google.es/books?id=76rnV5Exs50C>.
- [17] Krupadeluxe, *Iterative development model*. dirección: https://en.wikipedia.org/wiki/Iterative_and_incremental_development#/media/File:Iterative_Process_Diagram.svg.
- [18] T. A. Alliance, «The Agile Manifesto,» The Agile Alliance, 13 de feb. de 2001. dirección: <https://agilemanifesto.org/>.
- [19] C. Bentley, «Chapter 15 - Risk,» en *Prince2™ (Third Edition)*, C. Bentley, ed., Third Edition, Oxford: Butterworth-Heinemann, 2010, págs. 205-219, ISBN: 978-1-85617-822-8. DOI: <https://doi.org/10.1016/B978-1-85617-822-8.00015-9>. dirección: <https://www.sciencedirect.com/science/article/pii/B9781856178228000159>.
- [20] B. Boehm, «Software risk management: principles and practices,» *IEEE Software*, vol. 8, n.º 1, págs. 32-41, 1991. DOI: 10.1109/52.62930.
- [21] B. Hughes y M. Cotterell, *Software Project Management*. McGraw-Hill Higher Education, 2009, ISBN: 9780077122799. dirección: https://highered.mheducation.com/sites/9387067181/information_center_view0/.
- [22] G. LLC. «Sueldos para el puesto de Android Developer en España.» (19 de jun. de 2025), dirección: https://www.glassdoor.es/Sueldos/android-developer-sueldo-SRCH_K00,17.htm.

- [23] D. Travis. «How to create personas your design team will believe in,» Userfocus. (3 de ago. de 2009), dirección: <https://www.userfocus.co.uk/articles/personas.html>.
- [24] Google. «Persistent work.» (24 de ene. de 2024), dirección: <https://developer.android.com/develop/background-work/background-tasks/persistent>.
- [25] M. Woolley, «Bluetooth® Core 5.1, Feature Overview,» Bluetooth SIG, Inc., inf. téc., 13 de ene. de 2025. dirección: <https://www.bluetooth.com/bluetooth-resources/bluetooth-core-5-1-feature-overview/>.
- [26] I. Bluetooth SIG, «Enhancing BluetoothLocation Services withDirection Finding,» Bluetooth SIG, Inc, inf. téc., 28 de ene. de 2019. dirección: <https://www.bluetooth.com/bluetooth-resources/enhancing-bluetooth-location-services-with-direction-finding/>.
- [27] Google. «Use a local-only Wi-Fi hotspot.» (24 de jun. de 2025), dirección: <https://developer.android.com/develop/connectivity/wifi/localonlyhotspot>.
- [28] —, «Share a mobile connection by hotspot or tethering on Android.» (2025), dirección: <https://support.google.com/android/answer/9059108?hl=en>.
- [29] J. Nielsen, «10 Usability Heuristics for User Interface Design,» *Nielsen 1994a*, 24 de abr. de 1994. dirección: <https://www.nngroup.com/articles/ten-usability-heuristics/>.
- [30] Google. «Material Design 2.» (10 de jun. de 2024), dirección: <https://m2.material.io/design>.
- [31] —, «Cards.» (10 de jun. de 2024), dirección: <https://m2.material.io/components/cards>.
- [32] —, «Data tables.» (10 de jun. de 2024), dirección: <https://m2.material.io/components/data-tables>.
- [33] N. Slaughter, «Designing maps for mobile devices,» Mapbox, inf. téc., 27 de jun. de 2017. dirección: <https://blog.mapbox.com/designing-maps-for-mobile-devices-32d2e49d2096>.
- [34] B. Frost, «adaptive maps,» inf. téc., 15 de nov. de 2012. dirección: <https://bradfrost.com/blog/post/adaptive-maps/>.
- [35] B. Ricker y R. Roth, «Mobile Maps and Responsive Design,» *Geographic Information Science; Technology Body of Knowledge*, vol. 2018, n.º Q2, J. P. Wilson, ed., 1 de abr. de 2018. DOI: 10.22224/gistbok/2018.2.5.
- [36] Interaction Design Foundation - IxDF, «Getting Lost and Found – Maps and the Mobile User Experience,» en *Getting Lost and Found - Maps and the Mobile User Experience” Interaction Design Foundation - IxDF*. 16 de jun. de 2024. dirección: <https://www.interaction-design.org/literature/article/getting-lost-and-found-maps-and-the-mobile-user-experience>.
- [37] Google. «Simple, battery-efficient location API for Android.» (24 de jul. de 2024), dirección: <https://developers.google.com/location-context/fused-location-provider>.