

Desarrollo de una aplicación que utiliza la gamificación para mejorar la salud, fomentando el hábito de caminar en los desplazamientos urbanos

Carlos Jiménez Díaz

Máster Universitario de Desarrollo de aplicaciones para dispositivos móviles
Desarrollo de aplicación para dispositivos Android

Francesc D'Assís Giralt Queralt
Carles Garrigues Olivella

03/01/2020



Esta obra está sujeta a una licencia de Reconocimiento-
NoComercial-SinObraDerivada [3.0 España de Creative Commons](https://creativecommons.org/licenses/by-nc-nd/3.0/es/)

FICHA DEL TRABAJO FINAL

Título del trabajo:	<i>Desarrollo de una aplicación que utiliza la gamificación para mejorar la salud, fomentando el hábito de caminar en los desplazamientos urbanos</i>
Nombre del autor:	<i>Carlos Jiménez Díaz</i>
Nombre del consultor/a:	<i>Francesc D'Assís Giralte Queralt</i>
Nombre del PRA:	<i>Carles Garrigues Olivella</i>
Fecha de entrega (mm/aaaa):	01/2020
Titulación:	<i>Máster Universitario de Desarrollo de aplicaciones para dispositivos móviles</i>
Área del Trabajo Final:	<i>Desarrollo de aplicación para dispositivos Android</i>
Idioma del trabajo:	<i>Castellano</i>
Palabras clave	<i>Android, aplicación móvil, salud</i>

Resumen del Trabajo (máximo 250 palabras): *Con la finalidad, contexto de aplicación, metodología, resultados i conclusiones del trabajo.*

El sedentarismo no para de aumentar en las sociedades occidentales. Impacta de forma muy negativa en la salud, y puede acarrear enfermedades graves. Este TFM comprende el **diseño y desarrollo de una aplicación para teléfonos móviles Android**, orientada a **incrementar la actividad física diaria de sus usuarios** y, así, contribuir a mejorar su salud. Para ello utiliza **técnicas de gamificación** para combinar el uso del Metro en la ciudad de Barcelona con trayectos a pie, y maximizar estos últimos en los desplazamientos urbanos.

Para el diseño de las funcionalidades, interfaces y flujos de la aplicación se han utilizado metodologías de **diseño centrado en el usuario (DCU)**. Se han analizado en detalle a los usuarios potenciales, sus necesidades y contextos de uso, para obtener los requerimientos funcionales. Tras el diseño conceptual se ha realizado el prototipado de la aplicación y se ha evaluado. Para el resto de las etapas se ha seguido un **modelo en cascada**, combinado con técnicas de **desarrollo rápido de aplicaciones**.

Para la implementación se ha usado la **API pública de TMB (Open Data)** y las API **Google Maps** y **Google Places**. Como servicio de *backend* se ha utilizado **Firestore**, y para almacenar las fotos **Cloud Storage** (ambas de Google).

El resultado final es una *app* plenamente funcional, que implementa todas las funcionalidades y casos de uso proyectados. Algunas líneas futuras podrían ser la extensión a otros medios de transporte y a otras ciudades, así como la inclusión de más técnicas de gamificación y motivación.

Abstract (in English, 250 words or less):

Sedentary lifestyle keeps increasing in Western societies. Impact on population's health is very negative and can lead to serious diseases. This Master Thesis includes the **design and development** of an **app for Android mobile phones**, aimed at **increasing the daily physical activity of users** and, thus, contributing to improve their health. To achieve this goal, it uses **gamification techniques** to combine the use of Metro in the city of Barcelona with walking routes, maximizing the latter in urban journeys.

For the design of the functionalities, interfaces and flows of the application, **user-centered design** methodologies (UCD) have been used. Potential users, their needs and usage contexts have been analyzed in detail to obtain the functional requirements. After the conceptual design, the prototyping of the application has been carried out and evaluated. For the rest of the stages, a **cascade model** has been followed, combined with **rapid application development techniques**.

For the implementation, the **TMB public API** (Open Data), the **Google Maps API**, and the **Google Places API** have been used. **Firestore** was used as a backend service, and **Cloud Storage** to store photos (both from Google).

The result is a fully functional app, which implements all the projected functionalities and use cases. Some future lines could be the extension to more public transport and other cities, as well as the inclusion of more gamification and motivation techniques.

Índice

1	Introducción	1
1.1	Contexto y justificación del Trabajo	1
1.1.1	Riesgos del sedentarismo para la salud	1
1.1.2	El sedentarismo en España y sus consecuencias	2
1.1.3	Beneficios de la actividad física para la salud.....	4
1.1.4	Camina: incorpora caminatas en tus desplazamientos urbanos	4
1.1.5	Comparativa con otras aplicaciones similares	5
1.1.5.1	Aplicaciones de transporte público.....	6
1.1.5.2	Aplicaciones de salud y bienestar	8
1.2	Objetivos del Trabajo	9
1.2.1	Objetivos de la aplicación	10
1.2.2	Requisitos funcionales de la aplicación	10
1.2.3	Requisitos no funcionales de la aplicación	12
1.3	Enfoque y método seguido	12
1.3.1	Estrategia de desarrollo	12
1.3.2	Tipo de dispositivo y plataforma de desarrollo de la aplicación cliente	13
1.3.3	Elección de la plataforma de Backend	13
1.4	Planificación del Trabajo	14
1.4.1	Recursos necesarios.....	14
1.4.2	Planificación de las tareas a realizar	14
1.5	Análisis de riesgos y medidas de contingencia	17
1.6	Breve resumen de productos obtenidos	18
1.7	Breve descripción de los otros capítulos de la memoria	18
2	Diseño conceptual.....	19
2.1	Enfoque del producto y usuarios potenciales	19
2.2	Necesidades de los usuarios.....	19
2.3	Contextos de uso.....	20

2.4	Fichas de usuario y escenarios	21
2.5	Árbol de navegación.....	23
2.6	Prototipado	24
2.6.1	Sketching.....	24
2.6.2	Prototipado horizontal de alta fidelidad	26
2.7	Evaluación y revisión del diseño.....	31
3	Diseño técnico.....	34
3.1	Definición de los casos de uso.....	34
3.1.1	Identificación de actores y casos de uso.....	34
3.1.1.1	Actores.....	34
3.1.1.2	Casos de uso identificados.....	34
3.1.2	Diagrama UML de los casos de uso.....	35
3.1.3	Detalle de los casos de uso	36
3.2	Diseño de la Arquitectura.....	41
3.2.1	Diseño de los datos	42
3.2.2	Patrón de diseño del Software.....	43
4	Implementación	45
4.1	Decisiones de diseño.....	45
4.1.1	El proyecto Android	45
4.2	Principales librerías utilizadas	45
4.3	APIs utilizadas.....	46
4.3.1	API Google Maps.....	46
4.3.2	API pública TMB Transit	47
4.3.3	API Google Places.....	48
4.3.4	API TMB Planner	48
4.3.5	API FireBase	52
4.4	Layouts.....	53
5	Pruebas.....	57
6	Revisión de la planificación	58
6.1	Punto de control PEC2	58

6.2	Punto de control PEC3	59
6.2.1	Estado de la implementación.....	59
6.2.2	Medidas a adoptar	60
6.3	Punto de control PEC4	60
7	CONCLUSIONES	61
7.1	Resultados obtenidos y lecciones aprendidas.....	61
7.2	Líneas futuras.....	62
8	Glosario	63
9	Bibliografía	64
10	Créditos	66
	Anexo A: Diagrama de Gantt	I

Lista de figuras

Figura 1 - Prevalencia de actividad física insuficiente por países en hombres en 2016.....	2
Figura 2 - Prevalencia de actividad física insuficiente por países en mujeres en 2016.....	2
Figura 3 - Diabetes, hipertensión y colesterol 1993-2017	3
Figura 4 - Obesidad y sobrepeso en adultos 1987-2017 (mayores de 18 años)	3
Figura 5 - Detalle del diagrama de Gantt, resaltando la cinta que simboliza los días (diferenciando fines de semana).....	16
Figura 6 - Árbol de navegación de la aplicación.....	23
Figura 7 - Sketching (parte 1).....	24
Figura 8 - Sketching (parte 2).....	25
Figura 9 – Prototipo de alta fidelidad. Pantallas de splash e inicio (log-in).....	26
Figura 10 - Prototipo de alta fidelidad. Pantallas de registro (elección de método, introducción de datos, confirmación).....	27
Figura 11- Prototipo de alta fidelidad. Menú principal lateral y Pantallas de Mapa y Plano de Metro BCN	27
Figura 12 - Prototipo de alta fidelidad. Pantallas de Certificar último trayecto, mensaje de éxito, y notificación de confirmación para borrado.	28
Figura 13- Prototipo de alta fidelidad. Pantallas de Nuevo trayecto, Selección de trayecto y Selección de parada.....	28
Figura 14- Prototipo de alta fidelidad. Pantallas de Confirmación de selección de parada, Iniciar trayecto, y notificación de confirmación de iniciar trayecto.	29
Figura 15 - Prototipo de alta fidelidad. Pantalla de Estadísticas diaria, mensual y total.	29
Figura 16 - Prototipo de alta fidelidad. Pantallas de Logros, pestañas Récor ds y Distinciones....	30
Figura 17- Prototipo de alta fidelidad. Pantalla de Clasificación mensual y global.	30
Figura 18- Detalle de la nueva opción introducida en el menú (Nuevo trayecto), y del botón flotante de Nuevo trayecto presente en la pantalla Mapa.....	31
Figura 19 - Versión inicial (izquierda) y final (derecha) de la pantalla Selección de trayecto.	32
Figura 20 - Versión inicial (izquierda) y final (derecha) de la pantalla Selección de parada.....	33
Figura 21 - Versión inicial (izquierda) y final (derecha) de la pantalla Certificar último trayecto.	33
Figura 22 - Diagrama UML de los casos de uso.....	35
Figura 23 - Interrelación de la aplicación cliente con el servicio de backend y las API's de TMB y Google Maps.....	41

Figura 24 - Esquema del paradigma MVC, extraído de la Wikipedia [9]	44
Figura 25 – Asimilación del paradigma Modelo Vista Controlador a una aplicación Android [10].....	44
Figura 26 - Distribución de las diferentes versiones de la API de Android.....	45
Figura 27 - Captura de pantalla Mapa - Geoposicionamiento de usuario con icono de parada e información.....	47
Figura 28 - Documentación API TMB - Parámetros de consulta	51
Figura 29 - Pantallas de Selección de trayecto y Selección de parada	53
Figura 30 - Detalle del Diseño del layout de la pantalla Selección de parada.....	54
Figura 31 - Pantalla de Iniciar Trayecto - Detalle de Scrollview horizontal	55
Figura 32- Árbol de componentes y diseño de layout de la pantalla Iniciar trayecto	56
Figura 33 - Diagrama de Gantt de la planificación del TFM	I

1 Introducción

1.1 Contexto y justificación del Trabajo

Los criterios básicos que se siguieron para escoger el ámbito concreto sobre el que diseñar la aplicación del Trabajo Final de Máster (TFM) fueron la utilidad (se buscaba una app que ayudara a resolver un problema y que aportara un valor tangible), la simplicidad (que tuviera pocas funcionalidades y que fueran fáciles de usar), y que pudiera ser utilizada por el mayor número posible de usuarios.

Con los criterios anteriores en mente, y atendiendo también al interés personal, se decidió desarrollar una aplicación que ayudara a las personas a mejorar su salud incrementando de manera sencilla y sin esfuerzo su actividad física diaria.

Como se argumentará en los apartados siguientes, el sedentarismo impacta de forma muy negativa en la salud, y puede acarrear enfermedades graves. Así que diseñar una aplicación que ayude a aumentar la actividad física diaria y, por tanto, mejore la salud de las personas, resulta un tema interesante y útil.

1.1.1 Riesgos del sedentarismo para la salud

La Organización Mundial de la Salud (OMS) define la **actividad física** como “cualquier movimiento corporal producido por los músculos esqueléticos, con el consiguiente consumo de energía” [1]. Debe diferenciarse por tanto del **ejercicio físico**, entendido como una “subcategoría de actividad física que se planea, está estructurada, es repetitiva y tiene como objetivo mejorar o mantener uno o más componentes del estado físico” [1].

En los últimos años diversos estudios científicos han profundizado en la relación existente entre la actividad física y la salud, determinando tanto los efectos positivos de su práctica habitual como el papel de su ausencia en el desarrollo de determinadas **enfermedades no transmisibles** (ENT).

La **inactividad física** es el cuarto factor de riesgo en importancia de mortalidad a nivel mundial (se le atribuyen el 6% del total de muertes en el mundo). Por su parte, la **obesidad** y el **sobrepeso** son responsables del 5% de la mortalidad mundial [2].

Los últimos estudios estiman que la inactividad física es la principal responsable de entre el 21-25% de los casos de cáncer de mama y de colon, del 27% de la diabetes y de un 30% de las cardiopatías isquémicas [2].

El estilo de vida sedentario se extiende cada vez más en muchos países, y la salud general de la población mundial se ve resentida por ello. La prevalencia de las ENT (como son la diabetes, las enfermedades cardiovasculares o el cáncer) aumenta, y también sus factores de riesgo (hiperglucemia, hipertensión, sobrepeso, etc.) [2].

En un estudio realizado por la OMS en 2016 y publicado por *The Lancet* en 2018 [3], en el que se entrevistó a 1,9 millones de participantes de 168 países, se concluyó que **la prevalencia global de actividad física insuficiente en el mundo era del 27,5%**, con una diferencia de 8 puntos porcentuales entre sexos (23,4% en hombre y 31,7% en mujeres). Se puso de manifiesto, además, que la prevalencia era más del doble en países ricos (36,8%) que en países pobres (16,2%), y que

la actividad física insuficiente en países ricos había ido incrementándose con el tiempo (era del 31,6% en 2001).

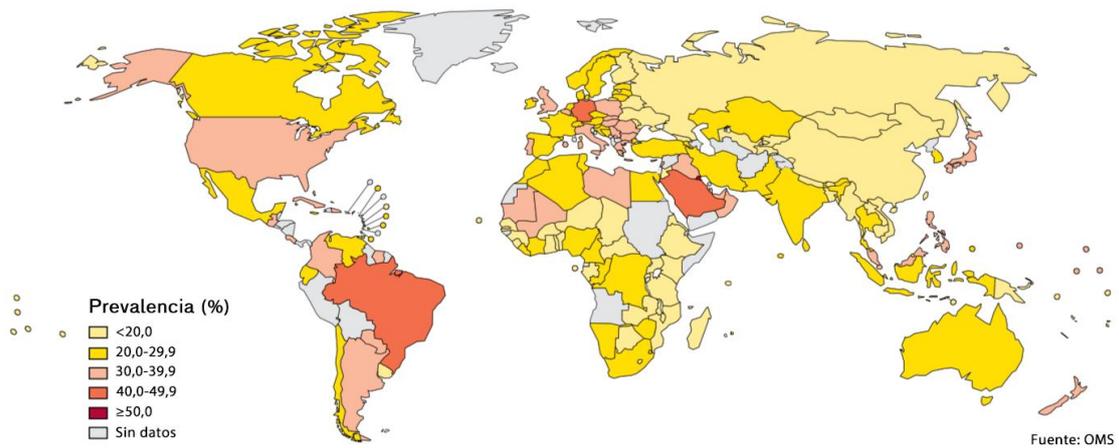


Figura 1 - Prevalencia de actividad física insuficiente por países en hombres en 2016

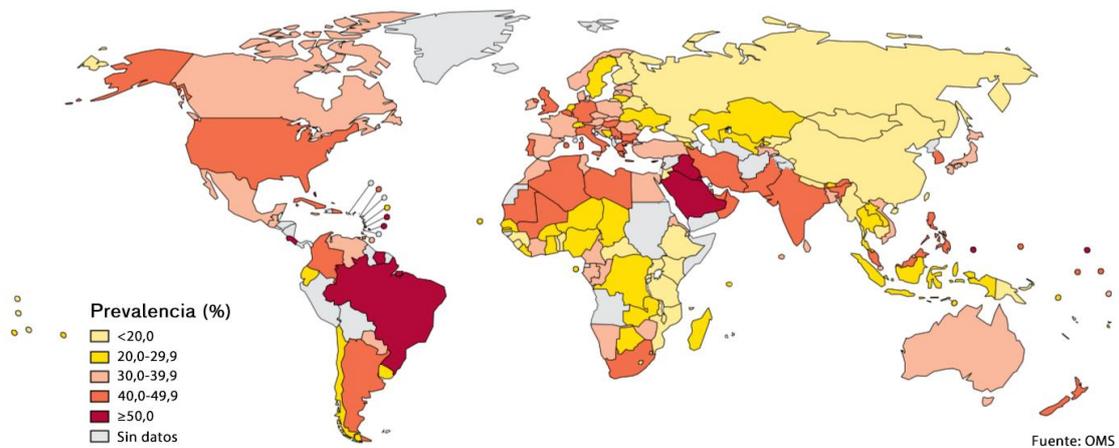
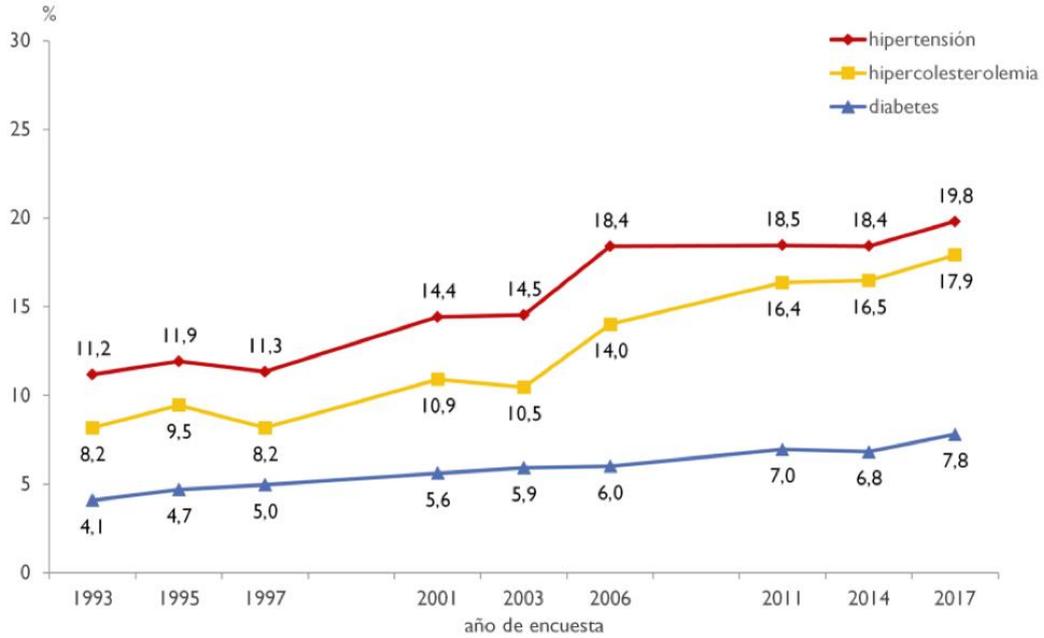


Figura 2 - Prevalencia de actividad física insuficiente por países en mujeres en 2016

1.1.2 El sedentarismo en España y sus consecuencias

La tasa de sedentarismo de España es ligeramente inferior a la media mundial. Pese a todo, los datos son alarmantes. Un 26,8% de la población adulta tiene una actividad física insuficiente: el 22,9% de los hombres frente al 30,5% de las mujeres [4].

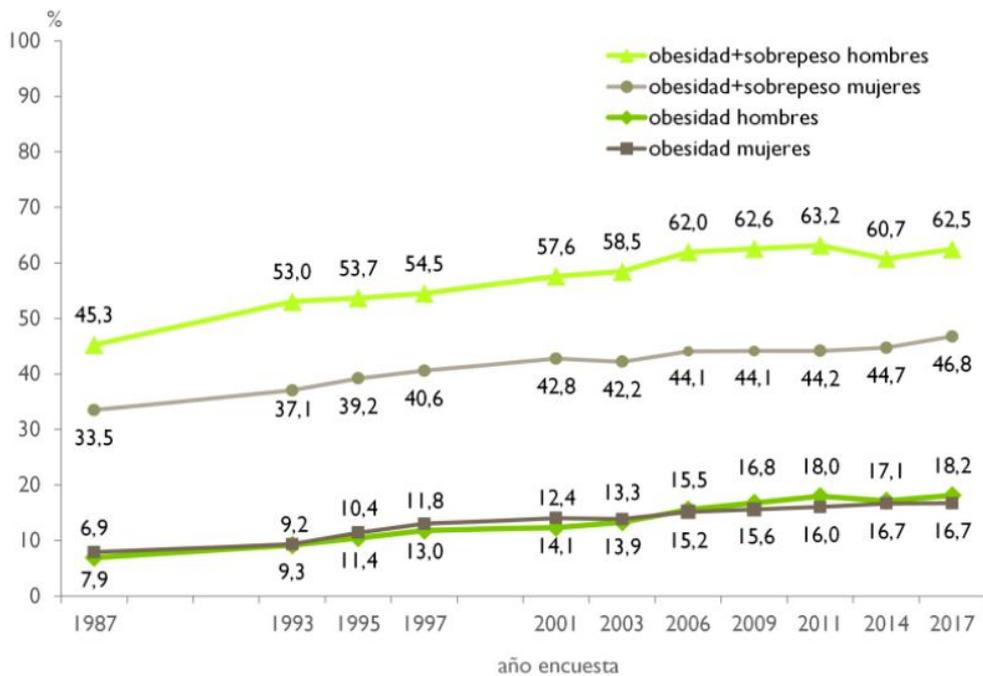
Según los datos de la Encuesta Nacional de Salud de 2017, la prevalencia de las ENT y los factores de riesgo cardiovascular ligados a actividad física insuficiente mantienen una tendencia ascendente desde el año 1993 [5]:



Fuente: Encuesta Nacional de Salud España 2017
Ministerio de Sanidad, Consumo y Bienestar Social

Figura 3 - Diabetes, hipertensión y colesterol 1993-2017

También son preocupantes las ratios de población adulta (mayores de 18 años) que sufren obesidad y los que tienen exceso de peso (que incluye el sobrepeso y la obesidad) [5]:



Fuente: Encuesta Nacional de Salud España 2017
Ministerio de Sanidad, Consumo y Bienestar Social

Figura 4 - Obesidad y sobrepeso en adultos 1987-2017 (mayores de 18 años)

Los Vehículos Eléctricos de Movilidad Personal (VEMP), y en especial los patinetes eléctricos, han irrumpido recientemente en todas las ciudades de nuestro país para empeorar la situación. Cada vez es más frecuente ver a personas que los utilizan para desplazarse desde sus casas o puestos de trabajo hasta las estaciones de transporte público. Con la adopción de este nuevo hábito, especialmente entre los más jóvenes, están renunciando al mínimo paseo que antes realizaban para llegar a pie a las estaciones de metro, tren o autobuses. Todo parece indicar, por tanto, que las nada halagüeñas cifras presentadas en las figuras anteriores serán incluso peores en los próximos años.

1.1.3 Beneficios de la actividad física para la salud

Numerosos estudios publicados en las últimas décadas por la comunidad científica confirman que **la actividad física comporta importantes beneficios para la salud y contribuye a prevenir las ENT.**

Según las recomendaciones mundiales sobre actividad física para la salud de la OMS, está demostrado que la actividad física regular y en niveles adecuados [6]:

- Mejora la salud ósea y funcional.
- Mejora el estado muscular y cardiorrespiratorio.
- Reduce el riesgo de hipertensión.
- Reduce el riesgo de cardiopatía coronaria.
- Disminuye el riesgo de accidentes cerebrovasculares.
- Disminuye el riesgo de diabetes tipo II.
- Tiene un efecto protector frente a diferentes tipos de cáncer (como el de mama y el de colon) y frente a la depresión.
- Disminuye el riesgo de caídas y de fracturas vertebrales o de cadera.
- Tiene un papel primordial en el consumo de energía, por lo que es fundamental para el equilibrio energético y el control de peso.

Para adultos de 18 años o más, la OMS recomienda [1][6]:

- Realizar un mínimo de 150 minutos a la semana de **actividad física aeróbica moderada** (como caminar a paso vivo o montar en bicicleta), o al menos 75 minutos semanales de **actividad física aeróbica intensa** (correr, nadar, etc.).
- Si se desea conseguir mayores beneficios para la salud, se debería llegar a 300 minutos semanales de actividad física aeróbica moderada (o su equivalente).
- Para que la actividad física impacte beneficiosamente en la salud cardiorrespiratoria, debe realizarse en periodos de como mínimo 10 minutos de duración.

1.1.4 Camina: incorpora caminatas en tus desplazamientos urbanos

Como se ha expuesto anteriormente, la actividad física tiene un impacto muy importante en la salud de la población, y tener unos niveles insuficientes puede acarrear enfermedades serias.

La disminución de la actividad física en las sociedades modernas se debe parcialmente al sedentarismo en el hogar y en el puesto de trabajo, y también **al mayor uso de modos de transporte pasivos** [1], como el coche, el transporte público y los VEMP.

Teniendo claro que en ciudades grandes la movilidad pasa casi inevitablemente por el uso del transporte pasivo, este TFM busca una manera de **incrementar la actividad física en los desplazamientos urbanos combinando el uso del transporte público con caminatas lo más largas posibles**. El objetivo es contribuir a alcanzar los niveles de actividad física recomendados por la OMS.

Concretamente se diseñará y desarrollará una aplicación para dispositivos móviles, **Camina**, orientada a los usuarios de la red de Metro de Barcelona que, mediante técnicas de gamificación, fomente la incorporación de trayectos a pie lo más largos e intensos posibles en sus desplazamientos por la ciudad.

Cada vez que el usuario necesite ir desde el punto A al B podrá usar esta aplicación para conocer la mejor ruta en metro. Una vez mostrada, se ofrecerá al usuario la opción de saltarse una o varias estaciones al inicio del trayecto, o incluso saltarse la primera línea de metro entera y cubrir el trayecto caminando, si la distancia no es excesiva. Cada vez que el usuario elija saltarse una o varias estaciones se contabilizarán los kilómetros andados, que se añadirán a su historial personal, y será recompensado mediante un sistema de puntos/trofeos/ranking. Además, se le informará de los beneficios conseguidos para su salud por esa elección.

1.1.5 Comparativa con otras aplicaciones similares

En este apartado se presenta un pequeño estudio de mercado en el que se analizan aplicaciones similares o relacionadas con la que se va a diseñar. Se ha centrado el estudio en la plataforma Android porque, como se explicará más adelante, ha sido la plataforma escogida para el desarrollo.

En la mayor plataforma de distribución digital de aplicaciones Android, **Google Play** (<https://play.google.com/store/apps>), se pueden encontrar varias aplicaciones que permiten consultar el mapa de estaciones de Metro de Barcelona y planificar trayectos desde un determinado origen hasta el destino deseado.

También hay disponibles muchas aplicaciones para monitorizar el ejercicio físico realizado, llevar un registro de pasos realizados y distancia recorrida mientras se camina, etc. Incluso algunas de ellas utilizan la gamificación para motivar al usuario a incrementar su actividad física.

Sin embargo, no se ha encontrado ninguna aplicación en el mercado que aúne ambas funcionalidades, es decir, que permita al usuario combinar de manera ágil y eficiente caminatas a pie con sus desplazamientos habituales en Metro por la ciudad, e incrementar así su actividad física diaria.

La aplicación objeto de este TFM pretende proveer al usuario de una herramienta que le permita no solo conocer la ruta más adecuada para llegar en Metro a su destino, sino que le anime a realizar parte de ese trayecto caminando para aumentar así su actividad física diaria. Para ello utilizará estrategias motivacionales (ofreciendo datos cuantitativos y cualitativos del beneficio que le reportará para su salud) y técnicas de gamificación (logros, trofeos, rankings, etc.) para aumentar el *engagement*.

Como se ha indicado en los párrafos precedentes, no se ha identificado ninguna aplicación realmente similar a la propuesta. No obstante, resulta relevante analizar aquellas *apps* que cubren parte de los objetivos que persigue la aplicación que se desarrollará. Se han escogido dos aplicaciones de cada uno de los ámbitos de interés: medios de transporte público de la ciudad de

Barcelona, y bienestar y salud. El criterio utilizado para la elección ha sido la valoración media de los usuarios y el interés de las funcionalidades proporcionadas.

En los subapartados siguientes se detalla, para cada una de las aplicaciones revisadas, sus puntos fuertes y sus debilidades respecto a sus competidores, y también en relación con la aplicación propuesta en este trabajo.

1.1.5.1 Aplicaciones de transporte público

 <p>TMBAPP (Metro Bus Barcelona) Transports Metropolitans de Barcelona Estilo de vida ★★★★★ 3.233 PEGI 3 Esta aplicación es compatible con todos tus dispositivos. Añadir a la lista de deseos Instalar</p>	<p>Nombre: TMBAPP (Metro Bus Barcelona) Desarrollador: Transports Metropolitans de Barcelona Categoría en Google Play: Estilo de vida Valoración: 3,7 ★ Enlace: https://play.google.com/store/apps/details?id=com.geomobile.tmbmobile</p>
Puntos fuertes	Puntos débiles
<ul style="list-style-type: none"> ○ Buscador de rutas combinadas (metro, bus i FFGG). ○ Trayectos recomendados con duración total. ○ Ubicación de medios de transporte cercanos ○ Visualización de mapas. ○ Información sobre líneas, paradas, tiempos de paso, horarios y alteraciones del servicio. ○ Asistente de viaje durante el trayecto, con indicaciones en tiempo real. Te guía en los trayectos paso a paso mediante geolocalización y te alerta con notificaciones <i>push</i> y avisos acústicos cuando te acercas a la parada de bus o metro en la que debes bajar. ○ Permite crear cuenta de usuario, y guardar lugares favoritos, trayectos, etc. ○ Recepción de alertas de servicio. ○ Widget. ○ Permite comprar títulos de transporte desde la app. ○ Acceso a promociones exclusivas. ○ Interfaz de usuario clara, intuitiva y fácil de usar. ○ Completo tutorial de uso al iniciar la app (<i>landing</i>). 	<p><u>Con respecto a sus competidores:</u></p> <ul style="list-style-type: none"> ○ Solo puede utilizarse en la zona metropolitana de Barcelona. ○ Únicamente muestras los transportes de TMB. ○ Sólo muestra los mapas de la red de Metro y de las líneas de autobús. ○ Requiere conexión a internet para funcionar. No muestra los mapas de bus estando offline. <p><u>Con respecto a Camina:</u></p> <ul style="list-style-type: none"> ○ No permite alternar trayectos a pie con los desplazamientos en transporte público para incrementar la actividad física.

	<p>Nombre: Citymapper – Rutas en transporte público Desarrollador: Citymapper Limited Categoría en Google Play: Mapas y navegación Valoración: 4,5 ★ Enlace: https://play.google.com/store/apps/details?id=com.citymapper.app.release</p>
<p style="text-align: center;">Puntos fuertes</p>	<p style="text-align: center;">Puntos débiles</p>
<ul style="list-style-type: none"> ○ Buscador de rutas combinadas de todos los medios de transporte disponibles (metro, bus, FGC, Renfe, autobuses, Uber / taxi, Car2go y bicicletas). ○ Trayectos recomendados con duración total. ○ Muchas ciudades de todo el mundo. ○ Ubicación de los distintos medios de transporte cercanos (en diferentes capas). ○ Visualización de gran variedad de mapas (en el caso de Barcelona Metro, FGC, Rodalies y Bus). ○ Información sobre líneas, paradas, tiempos de paso, horarios y alteraciones del servicio. ○ Acompañamiento durante el trayecto, con indicaciones en tiempo real. ○ Posibilidad de compartir el viaje en directo (y así informar de tu ubicación en tiempo real y cuándo llegarás al destino). ○ Permite crear cuenta de usuario, y guardar lugares favoritos, trayectos, etc. ○ Muestra todos los mapas estando offline, y permite guardar las rutas para acceder a ellas offline. ○ Da información sobre el trayecto a pie: cuántos kilómetros serían y cuántas calorías se quemarían al cubrir la distancia. 	<p><u>Con respecto a sus competidores:</u></p> <ul style="list-style-type: none"> ○ Interfaz de usuario algo confusa. ○ Requiere conexión a internet para funcionamiento completo. ○ No dispone de tutorial de uso al iniciar la app (<i>landing</i>). <p><u>Con respecto a Camina:</u></p> <ul style="list-style-type: none"> ○ Pese a que proporciona información del trayecto a pie, solo lo hace de manera global (es decir, todo el viaje a pie). No permite combinar el trayecto a pie con medios de transporte pasivos. ○ No incentiva el trayecto a pie frente a los medios de transporte pasivo, por lo que no fomenta la actividad física para mejorar la salud.

1.1.5.2 Aplicaciones de salud y bienestar

	<p>Nombre: Samsung Health Desarrollador: Samsung Electronics Co., Ltd. Categoría en Google Play: Salud y Bienestar Valoración: 4,4 ★ Enlace: https://play.google.com/store/apps/details?id=com.sec.android.app.shealth</p>
<p>Puntos fuertes</p>	<p>Puntos débiles</p>
<ul style="list-style-type: none"> ○ Registra actividades diarias y hábitos para ayudar a conseguir un estilo de vida saludable. ○ Registra automáticamente los pasos con el podómetro integrado. ○ Grabación de actividades sobre mapa en tiempo real con el GPS (si se activa). ○ Monitor de sueño. ○ Registros de salud como la frecuencia cardiaca, tensión arterial, glucosa en sangre, estrés y SpO₂, con los sensores integrados en el móvil o a través de dispositivos de terceros. ○ Registro (manual) de ingesta diaria de alimentos, agua, cafeína y de peso. ○ Permite fijar objetivos de actividad, comidas, sueño y peso. ○ Fuerte componente social. Permite competir con amigos, consultar la clasificación y compartir actividades con la Comunidad. Retos mensuales con todos los usuarios. ○ Componentes de gamificación. Implementa un sistema de logros, recompensas y trofeos. ○ Programas gratuitos de ejercicios. 	<p><u>Con respecto a sus competidores:</u></p> <ul style="list-style-type: none"> ○ No dispone de tutorial de uso al iniciar la app (<i>landing</i>). ○ Interfaz de usuario algo complicada. <p><u>Con respecto a Camina:</u></p> <ul style="list-style-type: none"> ○ No aborda el ámbito del transporte público, ni integra la actividad de caminar como parte de los trayectos urbanos. ○ No incide en concienciar sobre los beneficios que comporta la actividad física para la salud.

	<p>Nombre: Seguimiento de pasos – Podómetro gratis Desarrollador: Leap Fitness Group Categoría en Google Play: Salud y Bienestar Valoración: 4,6 ★ Enlace: https://play.google.com/store/apps/details?id=steptracker.healthandfitness.walkingtracker.pedometer</p>
<p>Puntos fuertes</p>	<p>Puntos débiles</p>
<ul style="list-style-type: none"> ○ Registra automáticamente los pasos, calorías quemadas, distancia andada, duración del trayecto, ritmo, etc. a lo largo del día. ○ Muestra los resultados en gráficos intuitivos y fáciles de consultar. ○ En el modo de seguimiento GPS, registra las rutas en el mapa con el GPS en tiempo real. ○ Posibilidad de desactivar el GPS, y contar los pasos con el sensor integrado para ahorrar batería. ○ Permite fijar objetivos (pasos diarios, distancia, calorías, duración etc.). ○ Diseño claro y muy sencillo. ○ La aplicación está completamente focalizada en la actividad de caminar. El usuario no puede distraerse con una multitud de funciones que pueden no interesarle. 	<p><u>Con respecto a sus competidores:</u></p> <ul style="list-style-type: none"> ○ No dispone de tutorial de uso al iniciar la app (<i>landing</i>). ○ Incluye publicidad (lo que puede molestar al usuario y que acabe escogiendo otra aplicación similar sin publicidad). ○ Carece de componente social. ○ No implementa técnicas de gamificación para motivar al usuario y mejorar el <i>engagement</i>. <p><u>Con respecto a Camina:</u></p> <ul style="list-style-type: none"> ○ No aborda el ámbito del transporte público, ni integra la actividad de caminar como parte de los trayectos urbanos. ○ No incide en concienciar sobre los beneficios que comporta la actividad física para la salud.

1.2 Objetivos del Trabajo

El objetivo de este TFM es el diseño y desarrollo de una aplicación para dispositivos móviles dirigida a los usuarios de la red de metro de Barcelona que, mediante técnicas de gamificación, fomente la incorporación de trayectos a pie lo más largos posibles en sus desplazamientos por la ciudad para incrementar su actividad física y, así, mejorar su salud.

1.2.1 Objetivos de la aplicación

La aplicación por diseñar y desarrollar deberá cumplir los siguientes **objetivos**:

O1.- Permitir la creación de una nueva cuenta de usuario, o el registro mediante cuentas de redes sociales. Asimismo, permitirá el acceso a la aplicación a usuarios ya registrados (*login*).

O2.- Posibilitar la creación de perfil del usuario y su posterior modificación.

O3.- Ser capaz de geolocalizar al usuario en un mapa.

O4.- Mostrar las estaciones de transporte de la red de Metro de Barcelona en un mapa.

O5.- Indicar al usuario cuál es la mejor ruta de metro para llegar al destino deseado, informándole de la estación más cercana a su ubicación.

O6.- Motivar e incentivar al usuario para que no inicie el trayecto desde la estación de metro más cercana, sino que lo haga desde estaciones más distantes y cubra el trayecto hasta la estación origen elegida a pie. El objetivo final es que incremente su actividad física.

O7.- Como estrategia de motivación, deberá facilitar la información sobre la caminata a pie que le supondría cada opción (kilómetros, tiempo estimado, etc.), y qué beneficios tendría para su salud (cuantitativos, como Kcal consumidas; y cualitativos, como beneficios para el corazón, etc.).

O8.- La aplicación deberá guiar al usuario hasta la estación origen de su trayecto (directamente o mediante la apertura de otra aplicación, como por ejemplo Google Maps).

O9.- Una vez completado el trayecto, el usuario deberá poder certificar en la aplicación que ha cumplido la misión, y acumular en su perfil los kilómetros adicionales andados gracias a su decisión de saltarse una o más estaciones de Metro.

O10.- La aplicación deberá utilizar técnicas de gamificación. Contará con un sistema de premios/trofeos y una clasificación para fidelizar al usuario.

O11.- El usuario podrá conocer su posición entre la del resto de usuarios, así como los trofeos y premios obtenidos por todos ellos.

1.2.2 Requisitos funcionales de la aplicación

En la tabla siguiente se enumeran los **requisitos funcionales** (RF) que debe tener la aplicación, y se relacionan con los objetivos (O) anteriores:

Requisito funcional	Objetivo relacionado
RF1. Gestión de usuarios y perfiles	
RF1.1 – Registro de nueva cuenta de usuario	O1
RF1.2 – Permitir utilizar cuentas de redes sociales para el registro	
RF1.3 – Login de usuario existente	
RF1.4 – Creación de perfil de usuario	O2
RF1.5 – Modificación de perfil de usuario	

RF2. Geolocalización	
RF2.1 – Gestión de permisos de ubicación en la aplicación	03
RF2.2 – Geolocalización del usuario en el mapa en tiempo real	
RF3. Mostrar red de Metro en el mapa	
RF3.1 – Acceder a la información necesaria de la red de Metro de Barcelona.	04
RF3.2 – Mostrar estaciones de Metro de Barcelona en el mapa.	
RF4. Gestionar la mejor ruta hacia el destino indicado	
RF4.1 – Habilitar sistema para que el usuario indique el destino	05
RF4.2 – Obtener la mejor ruta al destino (relacionado con RF3.1)	
RF4.3 – Determinar estación más cercana al usuario	
RF5. Motivar al usuario para elegir estaciones origen más distantes	
RF5.1 – Sugerir estación distinta a la más cercana	06 y 07
RF5.2 – Calcular y mostrar datos del trayecto a pie hasta la estación sugerida (kilómetros, tiempo, etc.)	
RF5.3 – Calcular y mostrar información motivacional (calorías, beneficios para la salud, etc.).	
RF5.4 – Ofrecer la opción de que el usuario consulte y seleccione estaciones alternativas a la sugerida.	
RF6. Guiar al usuario hasta la estación escogida	
RF6.1 – Guiar al usuario hasta la estación escogida mediante aplicación externa.	08
RF6.2 – Guiar al usuario desde la propia aplicación (OPCIONAL).	
RF7. Gestión de la realización del trayecto a pie	
RF7.1 – Mecanismo para que el usuario confirme el trayecto a pie realizado.	09
RF7.2 – Certificación automática desde la propia aplicación de que el trayecto se ha realizado mediante tracking de la posición del usuario (OPCIONAL).	
RF7.3 – Asignación de puntos obtenidos y actualización de logros (kms adicionales andados, calorías consumidas, etc.).	
RF8. Gamificación	
RF8.1 – Sistema de asignación de trofeos al usuario en función de sus logros.	010
RF8.2 – Sistema de clasificación	
RF9. Visualización de ranking y perfiles	
RF9.1 – Acceder al ranking y mostrarlo al usuario	011
RF9.2 – Permitir la visualización del perfil de otros usuarios (OPCIONAL)	
RF10. Gestión de notificaciones (OPCIONAL)	
RF10.1 – Notificaciones internas desde la propia aplicación (OPCIONAL)	06
RF10.2 – Notificaciones remotas (OPCIONAL)	

Debido al limitado tiempo disponible para la implementación se prevé poder satisfacer solo los requisitos funcionales no marcados como OPCIONALES. No obstante, no se descarta abordar alguno de ellos si las circunstancias lo permiten.

1.2.3 Requisitos no funcionales de la aplicación

Por último, se muestran los **requisitos no funcionales** (RNF) de la aplicación:

RNF1. – El desarrollo se llevará a cabo utilizando la IDE Android Studio, y el lenguaje de programación Java. El motivo de elegir Java frente a Kotlin es que se desconoce este último lenguaje.

RNF2. – Toda la gestión de usuarios (cuentas, perfiles, rankings, etc.) se realizará en un backend con el que la aplicación en el dispositivo móvil tendrá que comunicarse.

RNF3. – La información de la red de Metro de Barcelona se obtendrá mediante una API pública ofrecida por TMB o vía Google Maps.

RNF4. – La obtención de la mejor ruta hasta destino se obtendrá utilizando una API pública ofrecida por TMB o vía Google Maps.

RNF5. – La interfaz de usuario y el flujo de interacción deberán primar la facilidad de uso. La aplicación debe ser lo más sencilla posible.

RNF6. – Siempre que se produzca un error en la ejecución (imposibilidad de comunicar con el servidor, ausencia de conexión a internet, API's caídas) se deberá informar al usuario de manera clara.

1.3 Enfoque y método seguido

En los apartados siguientes se justificará el método de diseño escogido para desarrollar la aplicación, así como las primeras decisiones técnicas tomadas.

1.3.1 Estrategia de desarrollo

Dado que no se dispone de ninguna aplicación que proporcione las funcionalidades requeridas por la app **Camina**, ni en su totalidad ni parcialmente, es necesario desarrollar una aplicación completamente nueva.

Para el diseño de las funcionalidades, interfaces y flujos de la aplicación se seguirán metodologías de **diseño centrado en el usuario** (DCU). Estas técnicas ponen al usuario de la aplicación en el centro de todo el proceso. Se analizará en detalle a los usuarios potenciales, sus necesidades y sus contextos de uso. Con las conclusiones extraídas se realizará el diseño conceptual, funcional y visual que desembocará en el prototipado de la aplicación. El prototipo obtenido será finalmente evaluado para testear su usabilidad y la experiencia de usuario y, si es necesario, se realizarán los cambios necesarios.

Una vez obtenido el prototipo, para continuar con el desarrollo del software se seguirá un **modelo en cascada**. Superadas las fases de toma de requisitos y diseño (usando técnicas de DCU), se realizará la implementación y la verificación. Este modelo es el adecuado dado que la entrega del software está fijada (11 de diciembre de 2019, coincidiendo con la finalización de la PEC3) y se

conocen de antemano las funcionalidades básicas que la aplicación debe ofrecer (los requisitos están fijados).

No obstante, para la fase concreta de implementación (codificación) se usarán también técnicas de **desarrollo rápido de aplicaciones**. El objetivo será disponer de una aplicación funcional con los requisitos básicos lo más rápidamente posible para que, en sucesivas iteraciones, vaya perfeccionándose y se añadan más funciones. Esta elección se justifica porque, debido a la limitación temporal, es posible que no todas las funcionalidades se incluyan finalmente en la aplicación.

1.3.2 Tipo de dispositivo y plataforma de desarrollo de la aplicación cliente

Por la naturaleza de la aplicación, cuyo objetivo principal es fomentar la actividad física, se ha decidido desarrollar exclusivamente para **teléfonos móviles**, excluyendo las tabletas. Las personas no suelen caminar con *tablets* en la mano, mientras que sí lo hace con sus *smartphones*.

Una aplicación de este tipo hubiera sido buena candidata para ser desarrollada como aplicación híbrida, ya que no tiene grandes requisitos de rendimiento. Así podrían generarse de manera sencilla versiones para Android y iOS. Sin embargo, se decidió desarrollar una aplicación nativa para el sistema operativo **Android** por los siguientes motivos:

1. El tiempo disponible es limitado, y no se tenía experiencia con ningún entorno de desarrollo multiplataforma (como PhoneGap o Ionic).
2. De entre las plataformas de desarrollo en las que sí se tenía experiencia (iOS y Android), Android ocupa casi el 80% del mercado en España (datos de septiembre de 2019 proporcionados por StatCounter [7]). Uno de los objetivos es llegar al mayor número de usuarios posible, por lo que iOS queda descartado casi automáticamente.
3. El único dispositivo móvil del que se dispone para probar la aplicación es Android.

1.3.3 Elección de la plataforma de Backend

La aplicación móvil que se desarrollará deberá comunicarse con un servidor remoto con el que intercambiará datos. Este servidor remoto se encargará, por ejemplo, de almacenar los perfiles de los usuarios y proporcionar los datos necesarios para la elaboración de clasificaciones.

Para evitar la dificultad de montar y alojar un servidor propio se ha escogido la plataforma **Google Firebase** como servicio de Backend. Pese a que la base de datos que ofrece no es SQL, destaca por la sencillez de uso y la gestión de usuarios, que permite la identificación con cuentas de redes sociales de forma fácil. Al carecer de experiencia en el desarrollo de servidores, y dado que los servicios que se necesitarán por parte del servidor serán bastante simples, Firebase es sin duda la opción más adecuada.

1.4 Planificación del Trabajo

1.4.1 Recursos necesarios

Para el diseño y desarrollo de la aplicación que constituirá este TFM se utilizarán los siguientes recursos:

- MacBook Pro (Retina, 15 pulgadas, mediados de 2014). Intel Core i7 2,5 GHz, 16 GB RAM.
- Microsoft Office 365, básicamente:
 - Word, para la redacción de la memoria
 - Excel, para el diagrama de Gantt.
- Adobe Photoshop CC 2019 y Adobe Illustrator CC 2019, para la creación de recursos gráficos.
- Balsamiq Cloud y/o JustInMind para realizar el prototipado.
- Google Firebase como Backend as a Service (BaaS).
- Android Studio 3.5.1 (entorno de desarrollo integrado, IDE) para programar la aplicación Android.
- Se utilizará GitHub para el control de versiones.
- Dispositivo móvil Samsung Galaxy S8, con sistema operativo Android 9, como terminal de pruebas.

1.4.2 Planificación de las tareas a realizar

La fecha de inicio del proyecto es el miércoles 18 de septiembre de 2019, y el día límite de entrega es el 3 de enero de 2020. Dado que la asignatura del TFM tiene una carga de 12 créditos ECST (*European Credit Transfer System*), y cada crédito supone un tiempo de trabajo por parte del estudiante de 25 horas, **se ha considerado que se requerirán 300 horas para completar el proyecto.**

La dedicación al proyecto será de **2 horas al día de lunes a viernes, y 6 horas diarias el sábado y el domingo.** Para simplificar, en la planificación no se han descartado los días festivos: se ha mantenido la dedicación según su ubicación en la semana.

En la tabla siguiente se detallan todas las tareas, con su fecha de inicio y fin estimadas, y el número de horas de trabajo asignadas a cada una de ellas. Se ha previsto para cada bloque (que corresponde con cada entrega parcial de PEC) un margen de días sobrante que podría ser utilizado en el caso de presentarse desviaciones.

ACTIVIDAD	Inicio	Fin	Horas
PEC1 - Plan de trabajo	18/9/19	9/10/19	58
Lectura recursos aula	18/9/19	20/9/19	6
Búsqueda de información	21/9/19	22/9/19	8
Contexto y justificación	22/9/19	27/9/19	14
Objetivos del trabajo	28/9/19	29/9/19	10
Enfoque y metodología	29/9/19	2/10/19	8
Planificación del trabajo	3/10/19	5/10/19	8
Revisión de la memoria	5/10/19	6/10/19	3
Margen para desviaciones	6/10/19	8/10/19	-
Entrega PEC1	9/10/19	9/10/19	1
PEC2 - Diseño	10/10/19	30/10/19	60
Análisis	10/10/19	12/10/19	10
Investigación usuarios y requisitos	10/10/19	12/10/19	5
Investigación condiciones de uso	12/10/19	12/10/19	5
Diseño conceptual	13/10/19	16/10/19	12
Definición escenarios de uso	13/10/19	13/10/19	6
Definición flujos de interacción	14/10/19	16/10/19	6
Prototipado	17/10/19	22/10/19	20
Prototipado bajo nivel	17/10/19	19/10/19	6
Prototipado alto nivel	19/10/19	22/10/19	14
Diseño técnico	23/10/19	26/10/19	10
Diseño BD	23/10/19	24/10/19	4
Diseño de entidades y clases	25/10/19	26/10/19	4
API Cliente - Servidor	26/10/19	26/10/19	2
Evaluación	26/10/19	27/10/19	4
Evaluación del prototipo	26/10/19	26/10/19	2
Corrección errores y modificaciones	27/10/19	27/10/19	2
Revisión de la memoria	27/10/19	27/10/19	3
Margen para desviaciones	27/10/19	30/10/19	-
Entrega PEC2	30/10/19	30/10/19	1
PEC3 - Implementación	31/10/19	11/12/19	122
Preparación del Backend	31/10/19	1/11/19	4
Implementar aplicación y comentar código	2/11/19	4/12/19	106
Documentar memoria	5/12/19	7/12/19	6
Revisión de la memoria	7/12/19	7/12/19	3
Preparación de entregables	7/12/19	8/12/19	2
Margen para desviaciones	8/12/19	11/12/19	-
Entrega PEC3	11/12/19	11/12/19	1

ACTIVIDAD	Inicio	Fin	Horas
PEC4 - Entrega final	12/12/19	3/1/20	60
Finalización de la memoria	12/12/19	22/12/19	36
Elaboración de la presentación	22/12/19	26/12/19	10
Elaboración del vídeo	27/12/19	28/12/19	8
Preparación manual de la aplicación	29/12/19	29/12/19	3
Preparación de entregables	29/12/19	29/12/19	2
Margen para desviaciones	29/12/19	3/1/20	-
Entrega PEC4	3/1/19	3/1/19	1
TOTAL	18/9/19	3/1/20	300

En el [Anexo A: Diagrama de Gantt](#) se muestra gráficamente la planificación de tareas. La longitud de las líneas se corresponde con la duración de las actividades en **horas**. Para no perder la proporción al asignar las tareas a días, se ha utilizado diferentes tamaños para representar los días entre semana (rectángulos verdes, que comprenden solo dos horas de trabajo) y los días de fin de semana (rectángulos azules, a razón de 6 horas diarias).

En la siguiente figura se ha resaltado con un marco amarillo la cinta que simboliza los días:



Figura 5 - Detalle del diagrama de Gantt, resaltando la cinta que simboliza los días (diferenciando fines de semana)

De este modo se puede tener una representación de tareas en el calendario (pese a la que la distribución de horas en días no sea uniforme) y, al mismo tiempo, conservar la referencia del peso de cada tarea.

1.5 Análisis de riesgos y medidas de contingencia

Teniendo en cuenta los objetivos del proyecto, y tras enumerar los requisitos y establecer el calendario de trabajo, se identifican algunos riesgos que podrían hacer peligrar su éxito. En este apartado se analizan cada una de estas circunstancias y se proponen medidas de contingencia para mitigarlas en caso de presentarse:

RIESGO 1: Avería del portátil con el que se realizará el desarrollo del proyecto, que pueda provocar pérdida de los documentos de trabajo o del código desarrollado. En efecto, una avería que afectara al disco duro del portátil que se utilizará para la elaboración del TFM podría poner en peligro el código programado hasta el momento y todos los documentos que se van desarrollando.

MITIGACIÓN: Todos los documentos que se vayan elaborando (memoria del TFM, presentación, vídeo, manual, etc.) se almacenarán en una cuenta de OneDrive. Se abrirán directamente desde la nube y se trabajará siempre online. En lo que respecta al desarrollo del código de la aplicación, se utilizará un repositorio privado en GitHub que nos permitirá controlar las diferentes versiones y disponer de una copia de seguridad. También este documento se respaldará en el repositorio. De este modo el desarrollo podría continuar sin mayor problema utilizando otro PC/Mac.

RIESGO 2: Imposibilidad de encontrar una API gratuita para el cálculo rutas en Metro y distancias a pie. Dos requisitos funcionales de la aplicación son calcular la ruta óptima en metro entre dos direcciones dadas, y obtener la información necesaria de un trayecto a pie entre dos puntos. En la etapa de planificación del proyecto se desconoce si existen APIs públicas y gratuitas que ofrezcan estas funcionalidades. Sí sabemos que las APIs de Google Maps y Google Directions proporcionan esta información, pero es posible que sea necesario pagar por utilizarlas. Si no fuera posible disponer de APIs que ofrezcan estas funciones, como la limitación temporal de este proyecto no nos permite desarrollar el código necesario desde cero, quedaría fuera del ámbito del proyecto.

MITIGACIÓN: No es objeto de este TFM programar métodos que resuelvan rutas a pie ni trayectos en Metro para llegar de un punto a otro de la ciudad: no reinventaremos la rueda. En el caso de que no estuvieran disponibles las APIs públicas necesarias, se modelarían y programarían unos métodos de demostración que permitieran el desarrollo del resto de las funcionalidades que constituyen el elemento diferenciador de **Camina**.

RIESGO 3: Mayor complejidad de la esperada en la codificación. Consideramos que la etapa de codificación, además de ser la más larga, es también la más impredecible y difícil de estimar. Aprender a integrar y usar las APIs (si existen) puede ser especialmente complicado, y requerir más tiempo del proyectado.

MITIGACIÓN: En caso de necesidad se incrementaría el número de horas dedicadas a la implementación del código. A los días laborables se les ha asignado una carga de 2 horas para dedicar al TFM, que podría aumentarse hasta 4 horas si fuera necesario.

1.6 Breve resumen de productos obtenidos

Los productos que se obtendrán como resultado de este Trabajo Final de Máster serán:

- (a) Memoria final del trabajo. Se refiere a la versión definitiva de este documento.
- (b) Ejecutable de la aplicación para dispositivos Android desarrollada (archivo instalable .apk).
- (c) Código fuente de la aplicación comentado (archivo ZIP con las carpetas del proyecto listo para ser importado a Android Studio).
- (d) Presentación del proyecto (formato .ppt) en la que se explicará de forma sintética el trabajo llevado a cabo.
- (e) Vídeo de presentación del proyecto, que incluirá una breve demostración del funcionamiento de la app desarrollada.
- (f) Pequeño manual de usuario de la aplicación.

1.7 Breve descripción de los otros capítulos de la memoria

En los capítulos siguientes se irá desgranando el procedimiento de desarrollo de la aplicación.

El capítulo 2 está dedicado a las **técnicas de Diseño Centrado en el Usuario**. Se analizarán los usuarios potenciales de la aplicación, describiendo sus necesidades y convirtiéndolas en requisitos y funcionalidades. Por último, se definirán perfiles y escenarios de uso para ver cómo, cuándo y dónde utilizará el usuario la aplicación.

En el capítulo 3 se realizará el **diseño conceptual** y el **prototipado**. Se definirán los flujos de interacción, se implementará el árbol de navegación, se mostrarán los primeros esbozos del prototipo, y los resultados finales de baja y alta fidelidad. Finalmente se validará el prototipo.

El **diseño de la arquitectura** de la aplicación se expondrá en el capítulo 4. Se diseñará la base de datos, las entidades y las clases que estarán implicadas en la gestión de los diferentes procesos.

El capítulo 5 describirá los aspectos más relevantes de la **codificación**. Se justificarán las decisiones tomadas, y se detallarán las librerías, API's y *frameworks* utilizados en la implementación. Se detallarán los problemas encontrados y las soluciones adoptadas. Para concluir el capítulo se definirán las pruebas para validar el correcto funcionamiento de la aplicación.

En el capítulo 6 se expondrán las **conclusiones** del proyecto y las lecciones aprendidas durante su realización. Se reflexionará con espíritu crítico sobre la consecución de los objetivos iniciales, el seguimiento de las metodologías y la adecuación a la planificación realizada. Además, se apuntarán propuestas de mejora y posibles líneas de futuro.

Al final de esta memoria se incluirá un **glosario** que recogerá los términos y acrónimos más relevantes utilizados, y la lista de **referencias bibliográficas** usadas. El documento quedará cerrado con los anexos, donde se añadirán recursos adicionales que completarán el trabajo.

2 Diseño conceptual

2.1 Enfoque del producto y usuarios potenciales

La aplicación estaría dirigida a **personas de 30 a 75 años no deportistas**, familiarizados con el uso de smartphone, e **interesados por su salud** y por incrementar su actividad física para mejorarla. Un grupo especialmente interesante de usuarios son los pacientes diagnosticados con alguna de las enfermedades no transmisibles asociadas a la inactividad física (diabetes tipo II, hipertensión, hipercolesterolemia) y/o con exceso de peso.

Excluiríamos como usuarios potenciales a personas más jóvenes, puesto que a esas edades habitualmente aún no se ha manifestado el daño ocasionado por la inactividad física. Además, la población joven no está, por lo general, lo suficientemente concienciada con la salud como para interesarse por una aplicación de este tipo.

Es importante destacar que no se considera usuario potencial de esta app a las personas que practican deporte regularmente. Se asume que este colectivo no verá útil una aplicación de este tipo: no necesitan motivación para ejercitarse, y realizan la suficiente actividad física en sus entrenamientos como para que el incremento conseguido por las caminatas resulte relevante. Nos dirigimos a usuarios a los que no les gusta el deporte, pero que pese a ello quieren incrementar su actividad física de manera fácil.

Es imprescindible que la aplicación sea intuitiva y que tenga una interfaz de usuario atractiva y muy fácil de utilizar. Un porcentaje muy significativo de los usuarios serán personas de más de 60 años que, pese a estar familiarizados con los smartphones, no son especialmente hábiles en su manejo y huyen de las interfaces complejas.

2.2 Necesidades de los usuarios

El usuario potencial de esta aplicación necesita una herramienta sencilla y motivadora para incrementar su actividad física, que debe integrarse además en su vida cotidiana de manera no disruptiva. Requiere, por tanto:

- Una manera de **aumentar su actividad física** sin tener que acudir a gimnasios ni practicar deporte como tal. Nuestra propuesta es incorporar trayectos a pie lo más largos posible en los desplazamientos que realizar utilizando el transporte público.
- Para poder integrar los trayectos a pie con desplazamientos en Metro, necesita poder **consultar las ubicaciones de las estaciones de Metro**, recibir **información de los trayectos a pie que necesita para llegar a ellas** (tiempo, distancia, calorías, etc.), y también **del total del viaje** (a pie + Metro).
- **Motivación.** El perfil del usuario de esta aplicación, como ya hemos dicho, es una persona que no es deportista. No le gusta el ejercicio físico y, por lo tanto, requiere una herramienta que le recuerde los beneficios de mantener la suficiente actividad física, y que le motive a utilizarla regularmente.

2.3 Contextos de uso

Para ayudar a determinar cómo, cuándo y dónde utilizará el usuario la aplicación, lo más práctico es imaginar situaciones en las cuales usará cada una de las funcionalidades que le ofrece la aplicación.

Dentro del uso de la aplicación podemos diferenciar varias etapas:

- 1.) El usuario quiere realizar un desplazamiento desde su ubicación actual hasta una dirección determinada. Abrirá la aplicación para introducir la dirección destino y consultar el mejor trayecto en Metro. Se le presentarán varias alternativas, ordenadas por el tiempo de trayecto a pie (de mayor a menor). Una vez que seleccione una opción, idealmente la que la aplicación resaltará como “Ruta más saludable”, se le mostrará el trayecto exacto. A continuación, el usuario tendrá que elegir en qué estación subirá al metro: en la más cercana, la siguiente más cercana, etc. Una vez hecha la elección, podrá iniciar al trayecto. En esta fase el usuario se encontrará **en reposo en un determinado emplazamiento** (hogar, oficina, gimnasio, etc.) **o bien ya habrá iniciado la marcha y estará caminando por la calle.**
- 2.) Una vez clicado el botón de “Iniciar trayecto”, la aplicación Camina abrirá Google Maps para que guíe al usuario, a pie, hasta la estación de Metro que haya elegido para iniciar el trayecto. En este punto **el usuario se encontrará ya andando por la calle.**
- 3.) Una vez haya llegado a la estación de Metro escogida, el usuario realizará el trayecto en transporte público hasta la estación final de su recorrido. Saldrá del Metro y continuará a pie hasta su destino final. Una vez haya llegado a su destino, el usuario deberá interactuar con la aplicación Camina para certificar que realizó el trayecto seleccionado y, por tanto, que caminó todos los tramos a pie indicados. Puede hacerlo justo a la llegada a su destino o bien más tarde, cuando le venga bien. La casuística es variada. La interacción que se requerirá por su parte será mínima, simplemente se le pedirá que confirme el trayecto realizado pulsando un botón.
- 4.) En cualquier momento y lugar, el usuario podrá repasar sus logros y estadísticas o consultar la clasificación general para ver su desempeño en relación con el del resto de usuarios. Y probablemente realizará esta acción mientras lleva a cabo otras actividades. La atención que se requerirá del usuario será también muy pequeña: no tiene que introducir datos ni rellenar formularios. Será como consultar una red social. Será necesario que el usuario pueda llegar a esta parte de la aplicación de manera rápida y sencilla, y que la información le sea presentada visualmente de forma muy clara.

2.4 Fichas de usuario y escenarios

Hombre sedentario preocupado por su salud



“¿Correr? ¿Para qué? ¿Para ir a dónde? ¿Para escapar de qué?”

Nombre: Marcos

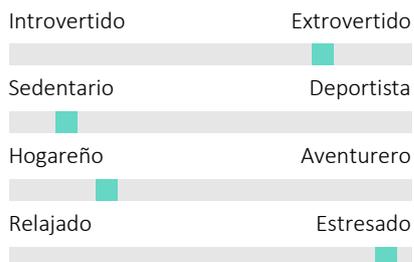
Edad: 43

Profesión: Service Manager

Familia: Soltero y sin hijos

Ciudad: Barcelona

Personalidad



algo hipocondríaco

techie

no le gusta el deporte

Bio

Marcos vive en Barcelona. Es soltero y no tiene hijos. De lunes a viernes acude a la oficina, también en Barcelona, o va a visitar a clientes. Siempre va de bólide, y entre informes, reuniones, videoconferencias, presentaciones y entregas, las jornadas laborales maratonianas se le pasan sin apenas darse cuenta. En su trabajo utiliza intensivamente el ordenador y el móvil.

Cuando llega a casa suele ser ya bastante tarde, con lo que apenas dispone de tiempo para bajar al súper, preparar algo sencillo, y cenar mientras ve alguna serie y consulta en su móvil las redes sociales. Dispone de un Mac y conexión a internet mediante FTTH, pero después de pasarse el día en la oficina, lo que menos le apetece es volver a encender un ordenador en casa.

Aunque disfruta haciendo excursiones por la montaña de vez en cuando los fines de semana, no le gusta el deporte y lleva una vida muy sedentaria. Probablemente debido a esto ha desarrollado hipercolesterolemia. Últimamente se preocupa por su salud, e intenta ser más activo: intenta caminar más, no utiliza los ascensores, etc. Se ha comprado una pulsera que monitoriza su actividad física, pero no ha dado el paso de apuntarse al gimnasio. Lo encuentra muy aburrido.

Escenario de uso

Son las 6.15 h de un miércoles y suena el despertador. Marcos llevaba ya un rato despierto, dando vueltas en la cama repasando mentalmente un par de temas complicados a los que tendrá que enfrentarse hoy en el trabajo.

Se levanta, se ducha rápidamente, se viste, y prepara el desayuno: copos de avena con leche y una taza de café. Engulle los cereales, apura de un sorbo el café y se cepilla los dientes. Coge el portátil y las llaves del coche. Pero antes de salir mira la hora y ve que va con tiempo. ¿Y si dejara el coche y fuera en transporte público a la oficina? Dicho y hecho. Coge el móvil, abre la aplicación **Camina** e introduce la ubicación de su trabajo. Con la mejor ruta en pantalla selecciona la tercera parada de Metro más cercana a su casa. En lugar de subirse en Urquinaona caminará hasta Universitat, y desde allí continuará en la línea roja hasta Plaça de Sants. En lugar de los 25 minutos que tardaría si hiciera todo el trayecto en metro, tardará 27. Pero en vez de andar solo 7 minutos (520 m), tendrá que andar 20 minutos (1500 m): 980 m más a pie que le supondrán quemar 60 calorías adicionales. Cierra la puerta con una sonrisa, y emprende el trayecto satisfecho.

Mujer con sobrepeso que quiere ser más activa



“Disfruto de la vida a mi propio ritmo. No sé qué es el estrés”

Nombre: Pilar

Edad: 50

Profesión: Profesora

Familia: Casada y con dos hijas

Ciudad: Barcelona

Personalidad



amante del campo

sedentaria

competitiva

Bio

Pilar está casada y es madre de dos niñas, de 10 y 17 años. Vive en l'Eixample Esquerra, en la ciudad de Barcelona. Desde pequeña quiso ser profesora, y ha logrado esa meta en su vida. Trabaja de lunes a viernes en un colegio público de Sant Martí impartiendo clase a alumnos de ESO.

En casa tiene ordenador y conexión a internet, y está muy familiarizada con el uso del smartphone, que utiliza a lo largo del día para gestionar el correo electrónico, chatear con la familia y con amigos, consultar redes sociales y navegar por la red.

Desde que tenía 14 años ha tenido tendencia al sobrepeso. Pese a que intenta comer sano y cocina casi sin sal, desde hace 5 años está diagnosticada de hipertensión arterial y toma Losartán a diario. Definitivamente Pilar no es una amante del deporte, pero le gusta bastante pasear, sobre todo por el campo.

Escenario de uso

Son las 16.00 h de un sábado, y Pilar ha quedado con un amigo para ir de compras en El Corte Inglés de Diagonal. Pilar coge el móvil y entra en la aplicación **Camina**. Lo primero que hace es consultar su clasificación. Desde que decidió saltarse 2 paradas de metro cada mañana cuando va al trabajo no para de ascender puestos. ¡Está bastante enganchada!

Introduce su destino y, se le muestra la ruta más rápida: 1 parada en la línea roja desde Urgell hasta Plaça Espanya; transbordo a la verde y 5 paradas más hasta Maria Cristina. Serían 23 minutos en total, 8 de los cuales serían a pie (unos 600 m). Elige caminar directamente hasta Plaça Espanya desde su casa (1,2 km) y coger la línea verde allí hasta María Cristina. Caminará en total 1,4 km y tardará 24 minutos. Esto supone 800 m más caminados, que con su peso implica un consumo estimado de 52 calorías quemadas más. Se pone en camino para no llegar tarde.

Cuando llega a su destino abre la aplicación de nuevo y marca la misión como completada. Los puntos sumados hacen que escale 2 posiciones. ¡Ya sólo está a 3 puestos de su compañero de trabajo!

2.5 Árbol de navegación

Con los requisitos funcionales establecidos en el apartado 1.2.2, y tras el análisis de los contextos de uso, estamos en disposición de diseñar el flujo de la interacción o árbol de navegación de la aplicación. El resultado se muestra en la siguiente figura. El detalle de cada pantalla se verá en el prototipado horizontal de alto nivel, en un apartado posterior.

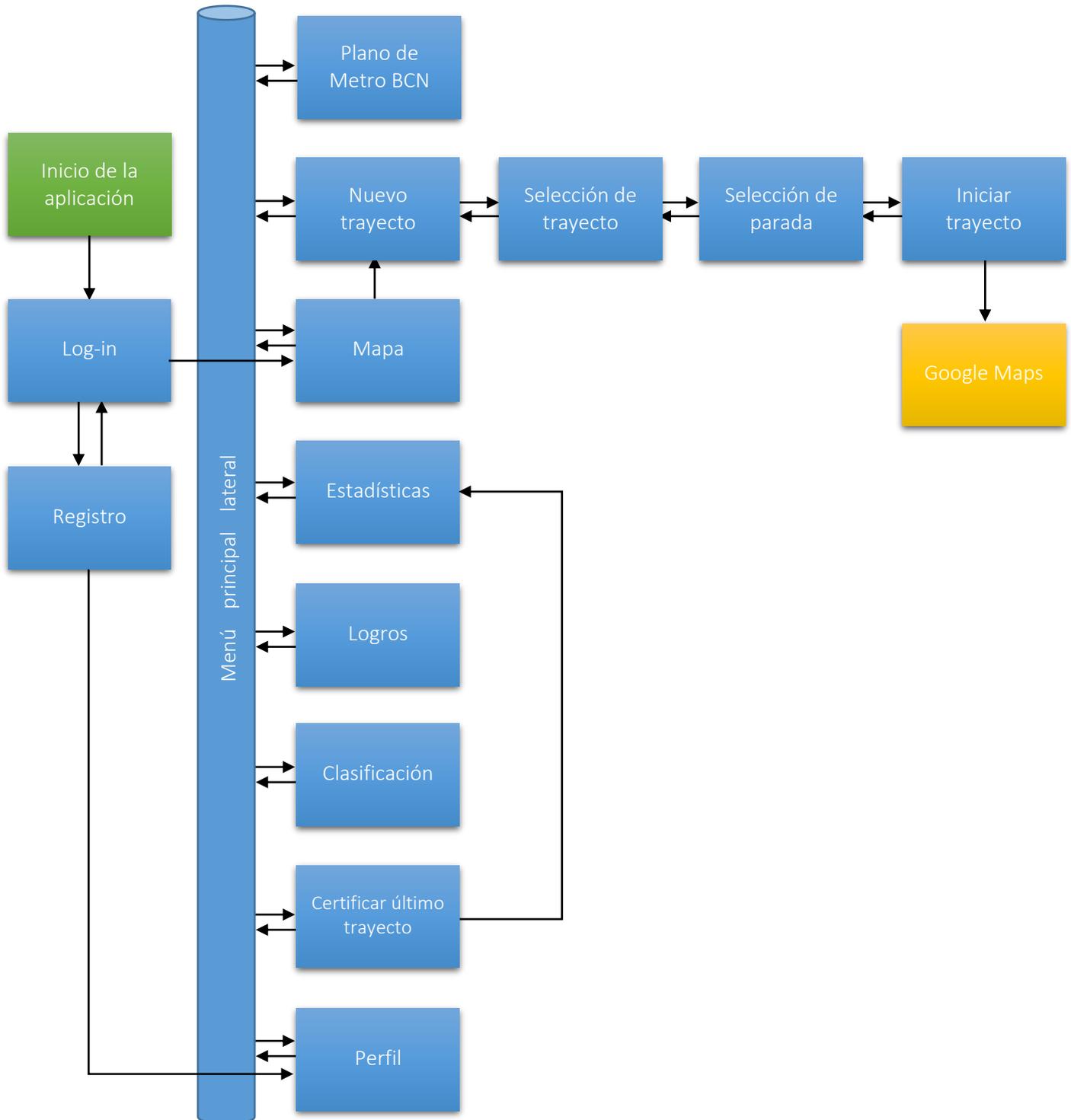


Figura 6 - Árbol de navegación de la aplicación

2.6 Prototipado

Tras realizar la investigación de los usuarios, determinar los requisitos y las condiciones de uso de la aplicación, y una vez finalizado su diseño conceptual con la definición de los escenarios de uso y los flujos de interacción, dedicaremos este apartado a su prototipado.

Se ha decidido realizar el prototipo de la aplicación en dos fases: un esbozo a mano o *sketching* rápido, para generar e iterar propuestas de manera muy dinámica; y a continuación un prototipado de alta fidelidad, para el que se ha utilizado la herramienta JustinMind.

2.6.1 Sketching

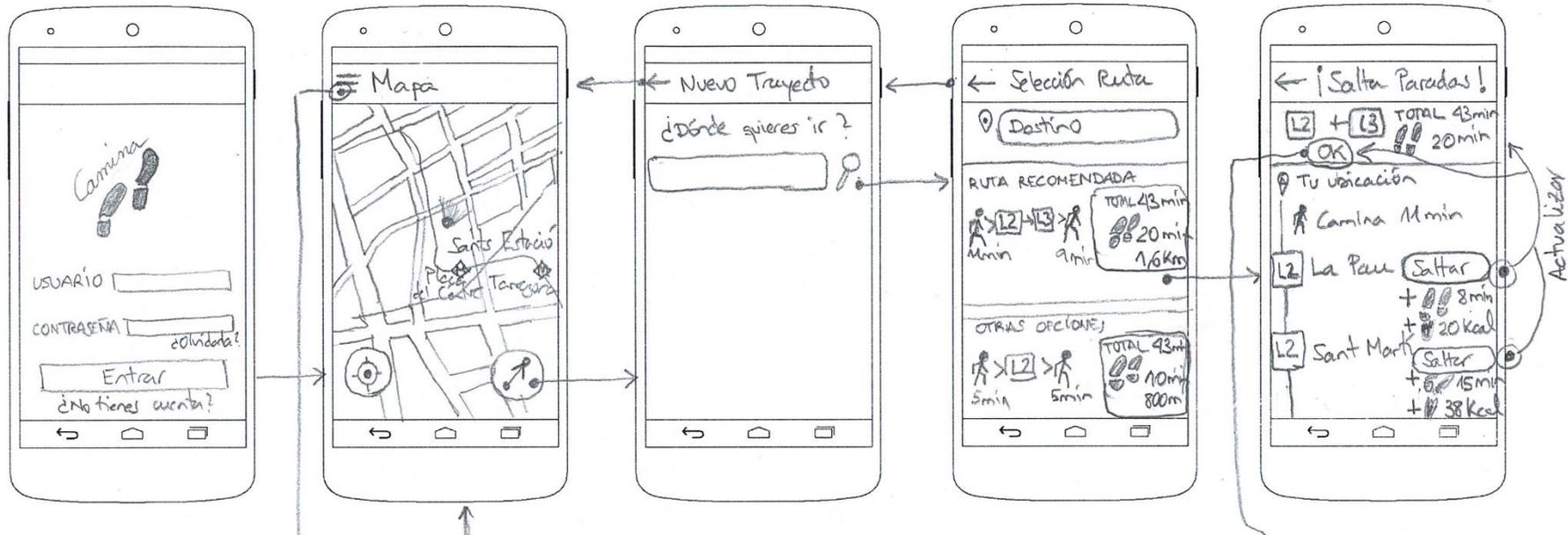


Figura 7 - Sketching (parte 1)

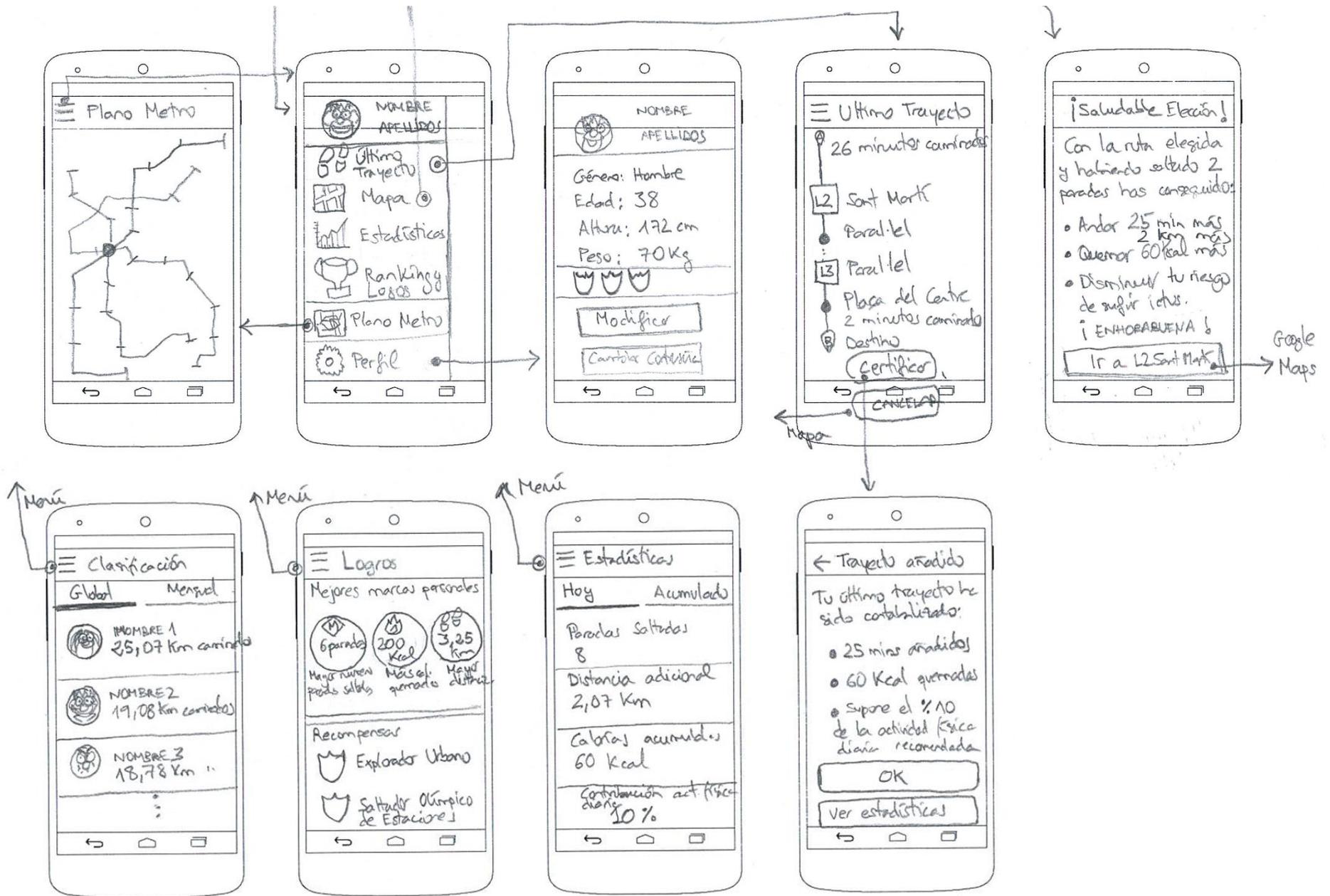


Figura 8 - Sketching (parte 2)

2.6.2 Prototipado horizontal de alta fidelidad

Basándonos en el sketching inicial, se procede a diseñar el prototipado horizontal de alta fidelidad con la herramienta JustinMind. Durante el proceso se concretan las ideas y se introducen algunos cambios en las pantallas.

Dado que el espectro de usuarios al que se destina la aplicación incluye a personas de edad avanzada, para el diseño de las pantallas se ha tenido siempre en mente que todos los elementos presentados (texto, iconos, botones) debían ser lo suficientemente grandes para asegurar su legibilidad.

También se ha intentado presentar la información de la manera más sencilla y clara posible, y simplificar al máximo las interacciones.

Se presenta a continuación el **resultado final**. Profundizaremos sobre las decisiones de diseño y trataremos los cambios realizados (tanto con respecto a la versión de sketching como con versiones previas del prototipo de alta fidelidad) en el apartado siguiente, dedicado a la revisión del diseño.

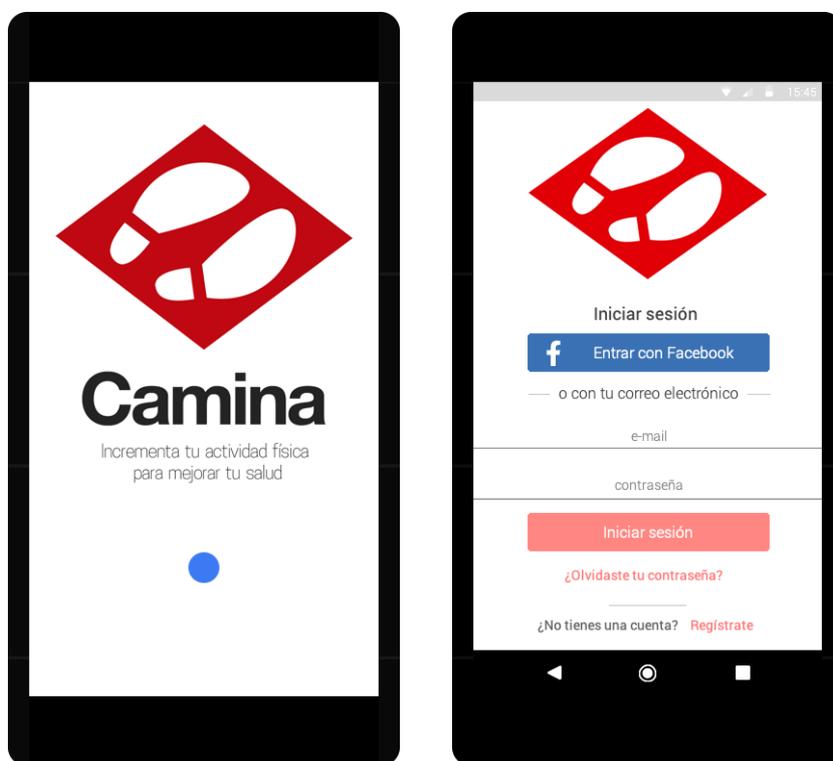


Figura 9 – Prototipo de alta fidelidad. Pantallas de splash e inicio (log-in)

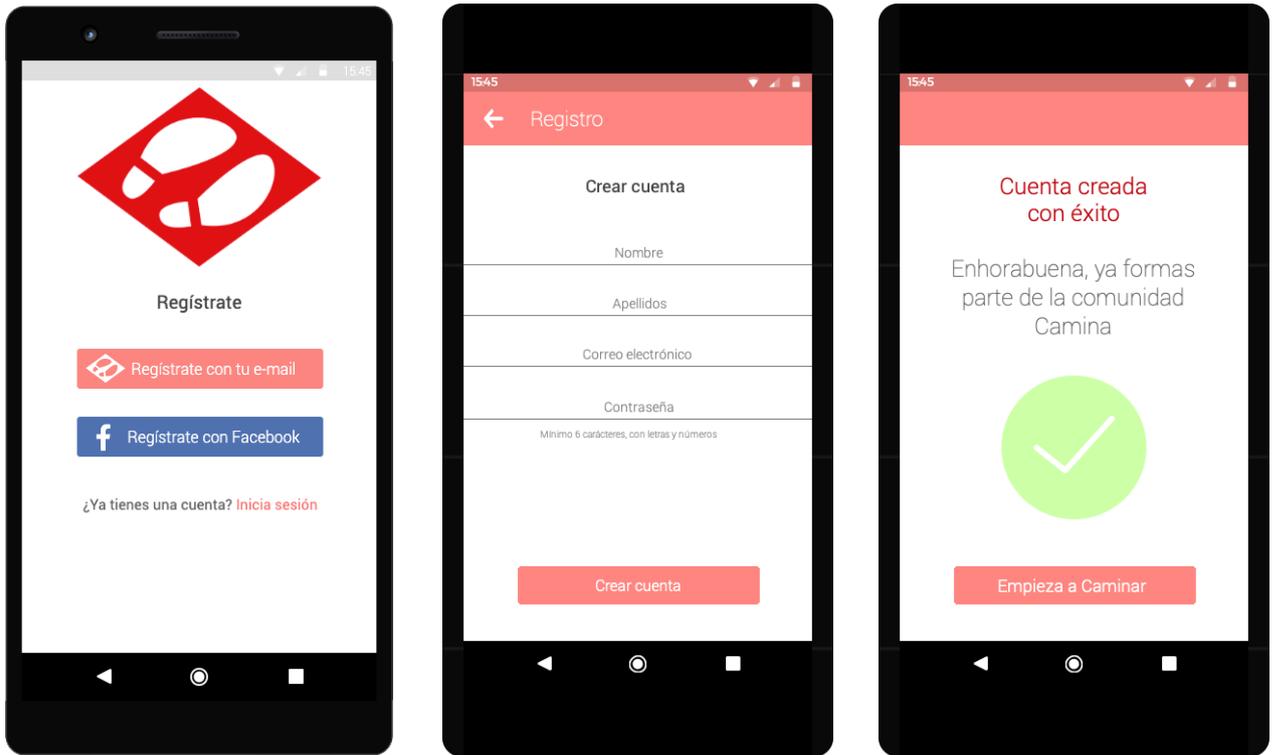


Figura 10 - Prototipo de alta fidelidad. Pantallas de registro (elección de método, introducción de datos, confirmación)

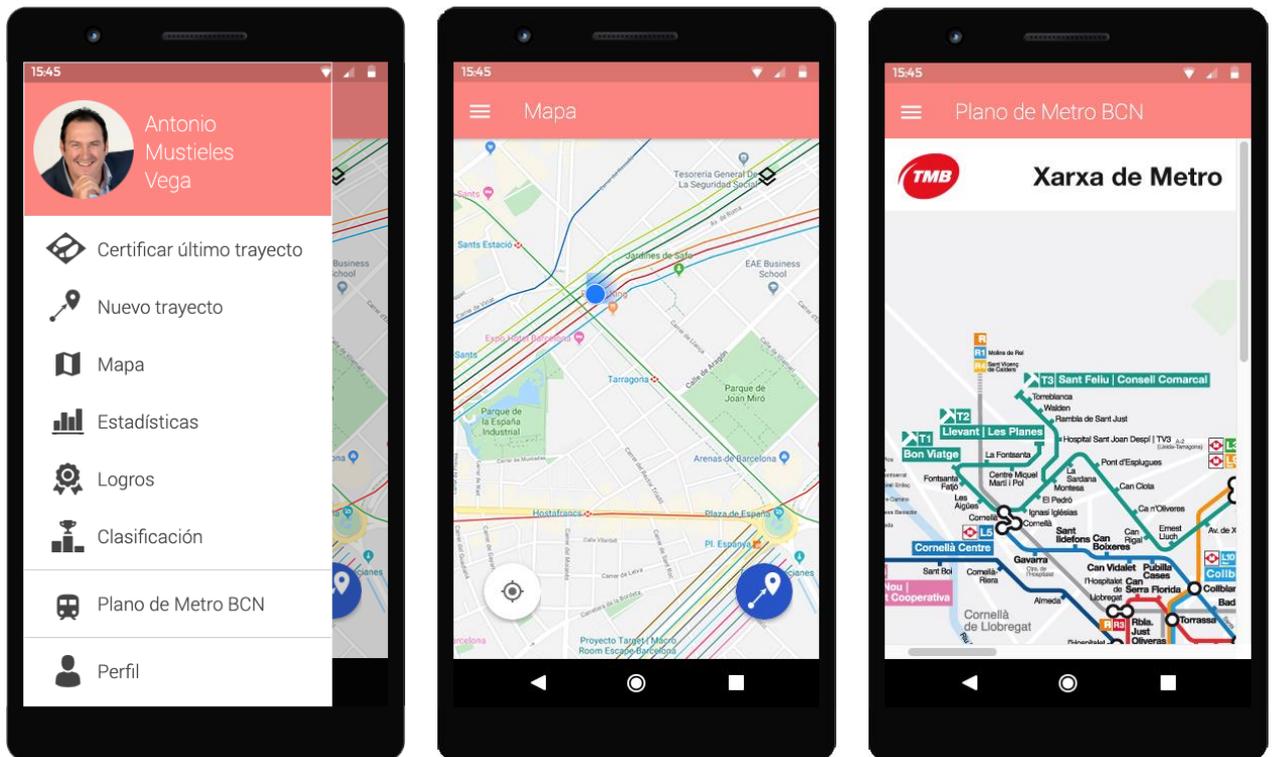


Figura 11- Prototipo de alta fidelidad. Menú principal lateral y Pantallas de Mapa y Plano de Metro BCN

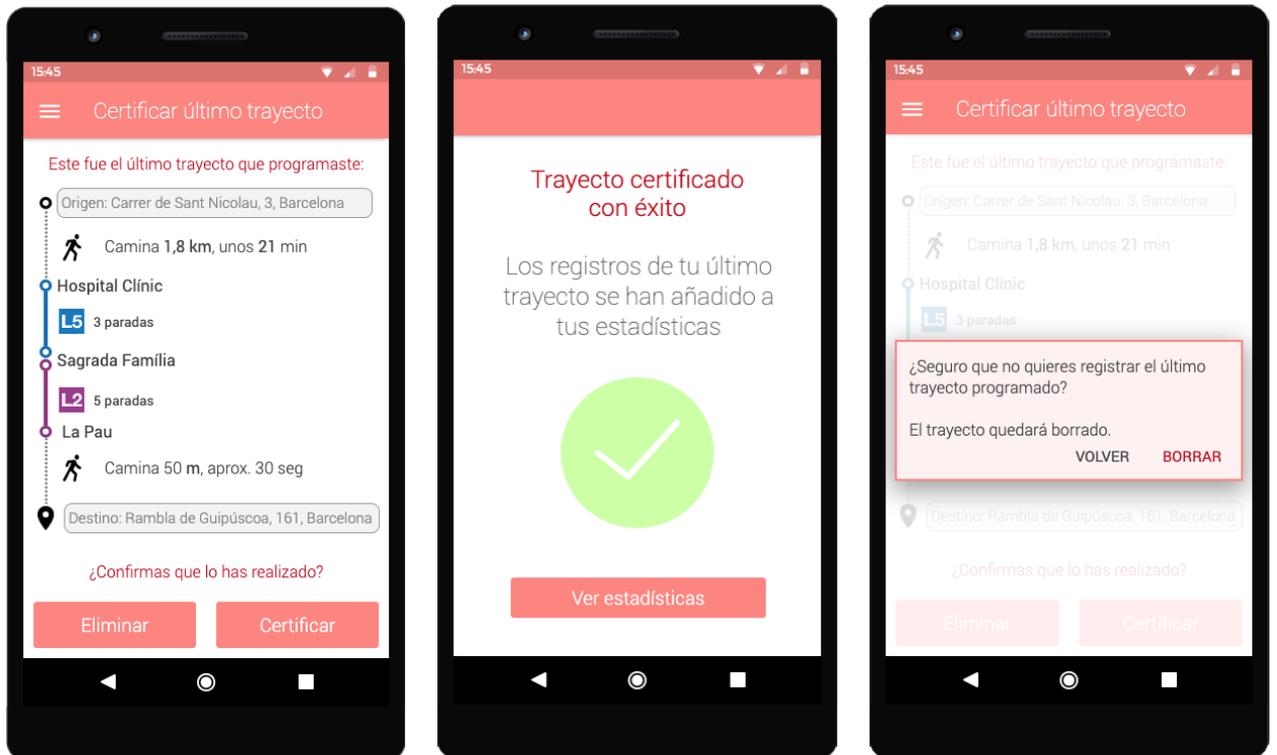


Figura 12 - Prototipo de alta fidelidad. Pantallas de Certificar último trayecto, mensaje de éxito, y notificación de confirmación para borrado.

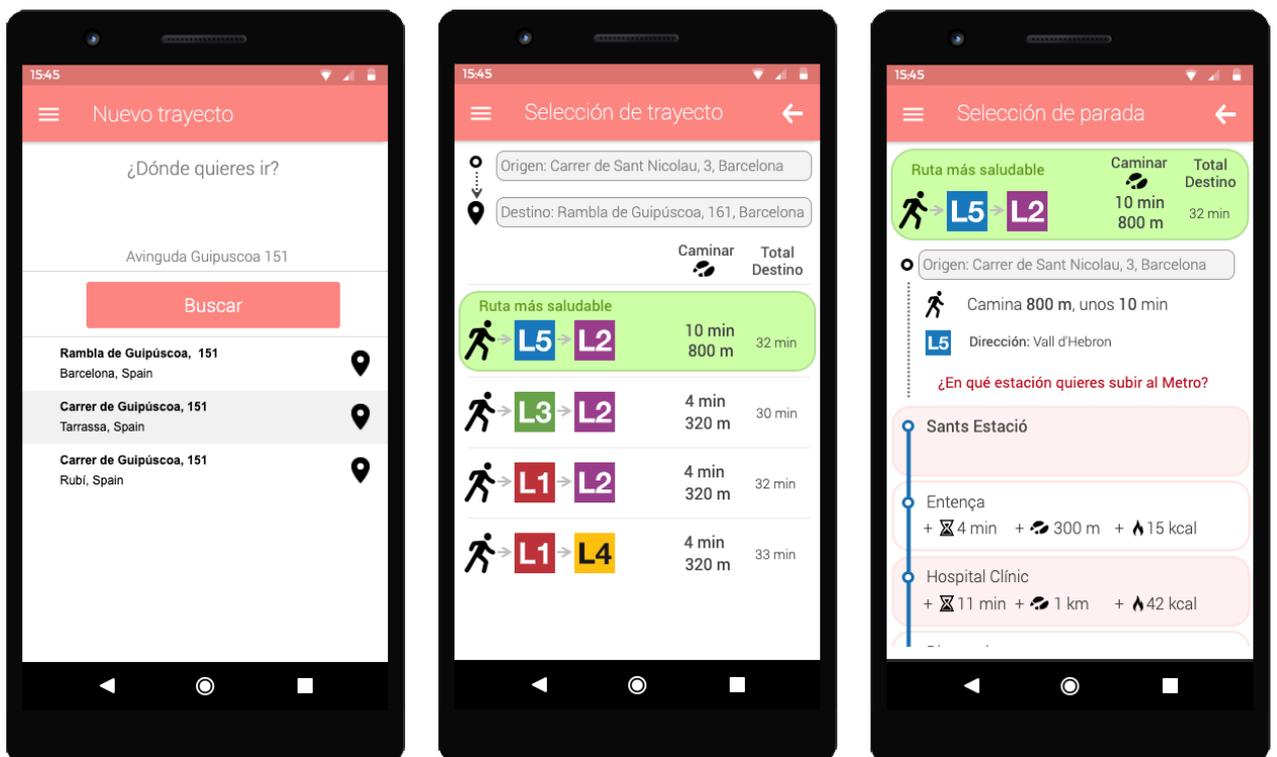


Figura 13- Prototipo de alta fidelidad. Pantallas de Nuevo trayecto, Selección de trayecto y Selección de parada

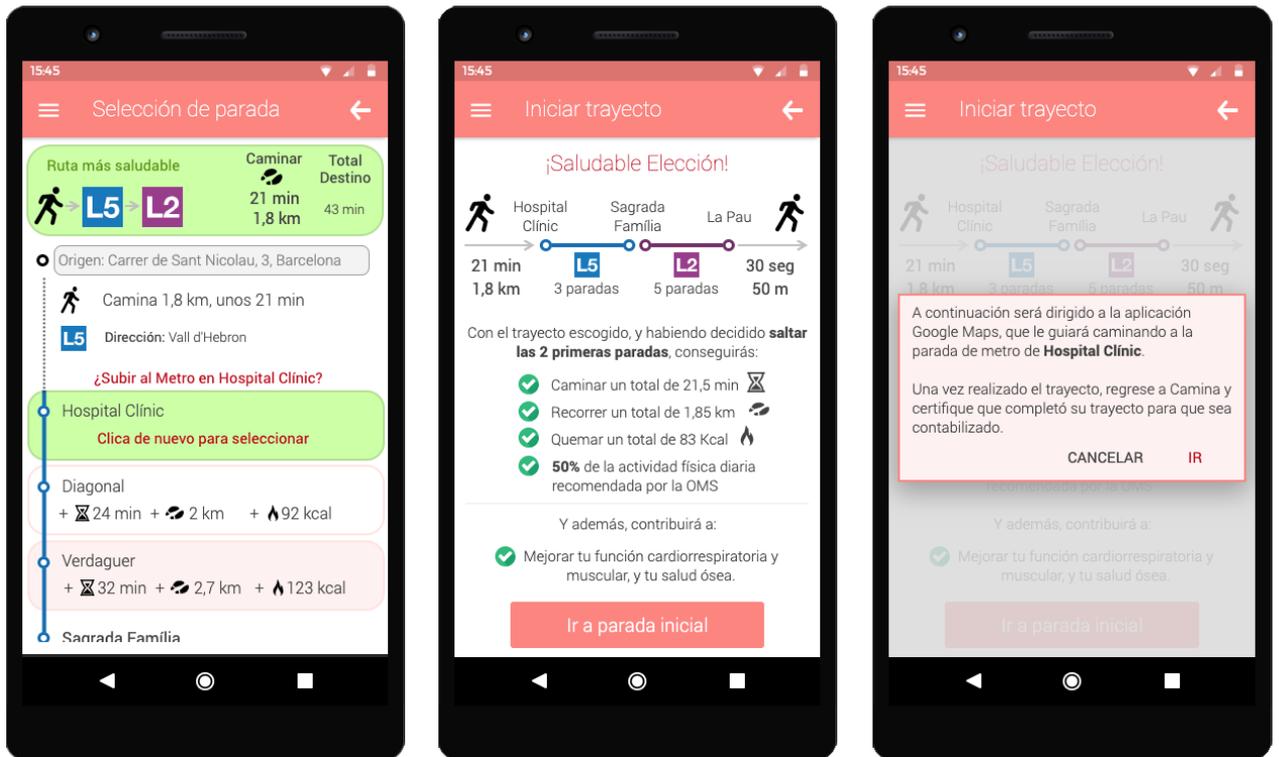


Figura 14- Prototipo de alta fidelidad. Pantallas de Confirmación de selección de parada, Iniciar trayecto, y notificación de confirmación de iniciar trayecto.

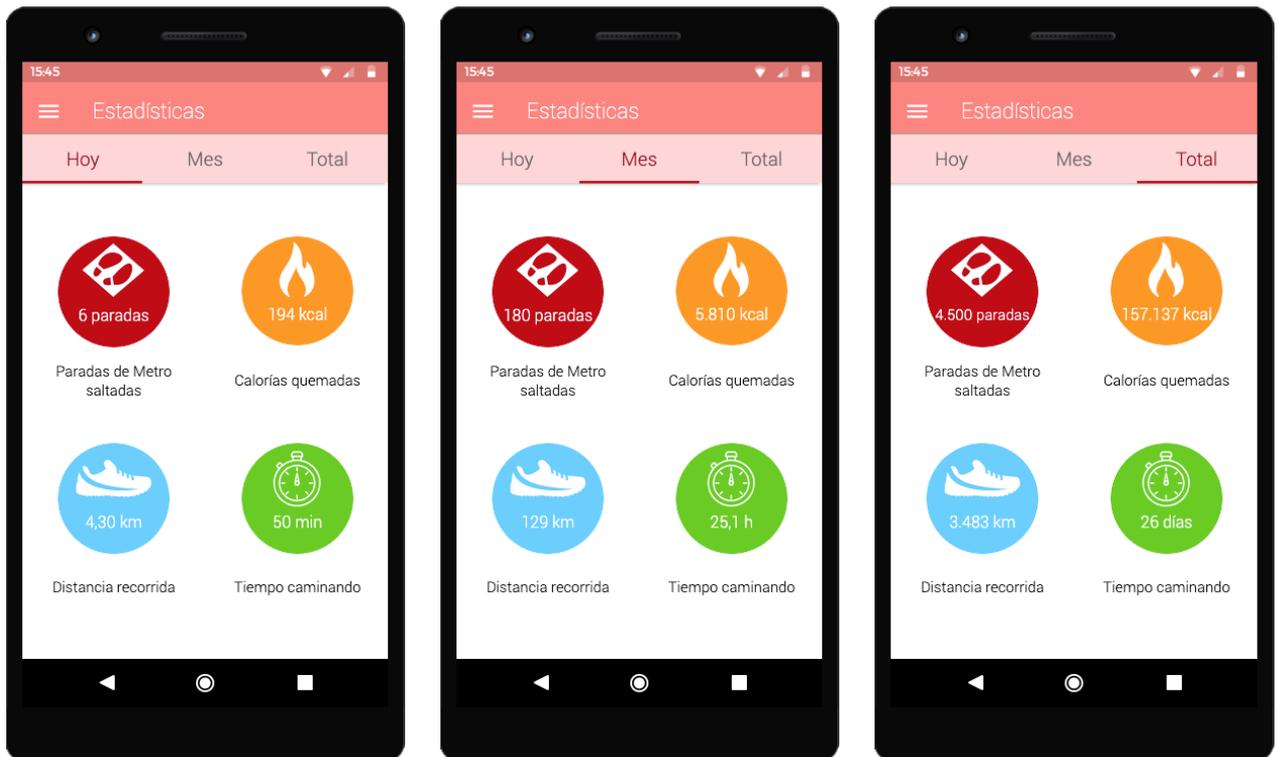


Figura 15 - Prototipo de alta fidelidad. Pantalla de Estadísticas diaria, mensual y total.

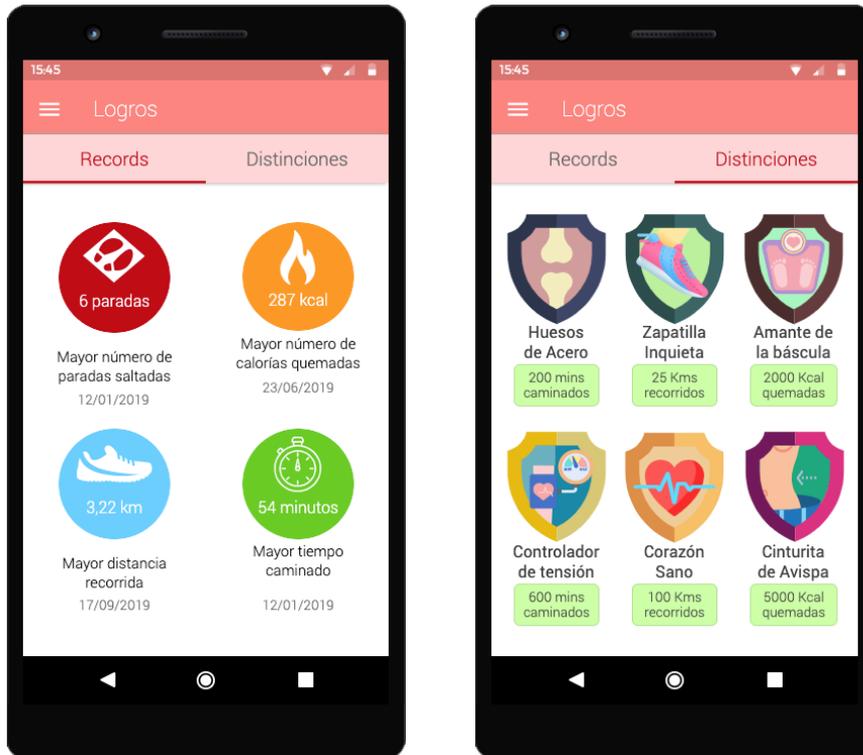


Figura 16 - Prototipo de alta fidelidad. Pantallas de Logros, pestañas Récor ds y Distinciones.

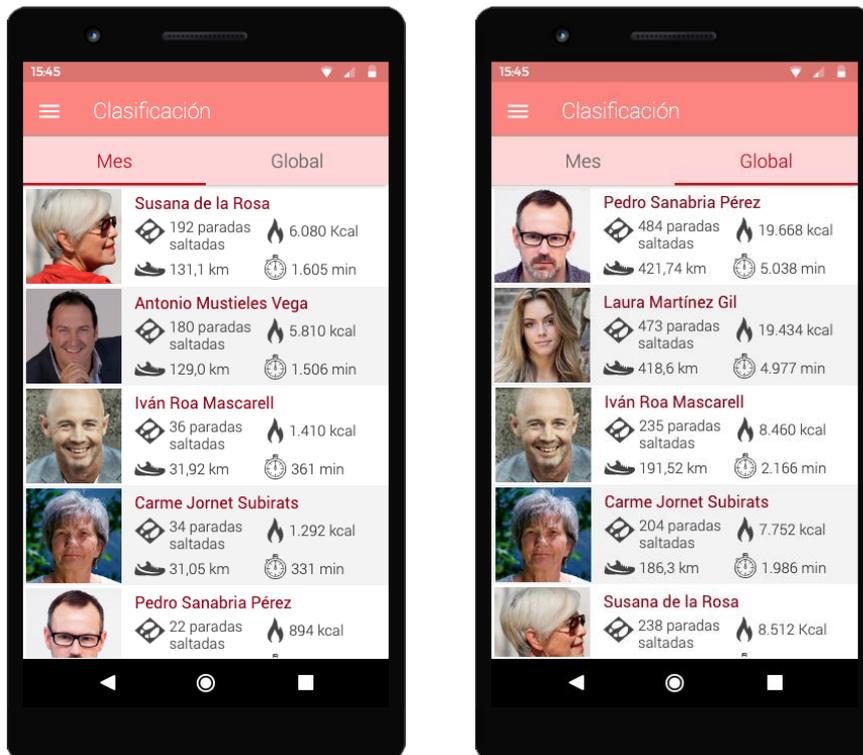


Figura 17- Prototipo de alta fidelidad. Pantalla de Clasificación mensual y global.

2.7 Evaluación y revisión del diseño

El árbol de navegación y el prototipo presentados en los apartados anteriores constituyen las versiones definitivas del diseño de la aplicación. Sin embargo, hubo versiones previas tanto del árbol de navegación como de algunas pantallas, que sufrieron modificaciones durante el proceso de desarrollo.

Ya en el paso de la versión del sketching inicial hasta el primer prototipo de alta fidelidad se introdujeron algunos cambios. Así, por ejemplo, se decidió introducir pestañas en la pantalla de **Logros** para separar los Récords y las Distinciones. El resultado final es una presentación mucho más limpia. También se aumentó el número de pestañas en la pantalla **Estadísticas**, pasando de dos (Hoy y Acumulado) a tres (Hoy, Mes y Total). Se consideró que un mes era una medida óptima para mostrar las estadísticas. Es un período mucho más largo que el día (demasiado volátil) y como se reinicia cada 30-31 días permite que los usuarios nuevos tengan también posibilidades de aparecer en las primeras posiciones (algo que no ocurriría con el acumulado total, que tiene mucha más inercia).

Pero los cambios más profundos surgieron cuando se presentó el prototipo de alta fidelidad interactivo a un grupo reducido de 3 usuarios potenciales. Coincidieron en que en algunas pantallas la interacción no quedaba clara y se prestaba a confusiones, por lo que se introdujeron cambios para subsanarlo.

Respecto al **árbol de navegación**, inicialmente se había pensado que la única vía para seleccionar un nuevo destino y el respectivo trayecto fuera desde la pantalla **Mapa**, clicando en el botón flotante de nuevo trayecto. Los usuarios indicaron que sería mejor disponer también de una vía directa desde el menú lateral, y se decidió incluir la opción “**Nuevo trayecto**”. Así, en la versión final del prototipo, se puede iniciar la secuencia de selección de destino/trayecto por ambas vías. Tiene sentido: el usuario puede tener claro su destino y no necesitar para nada consultar el mapa.

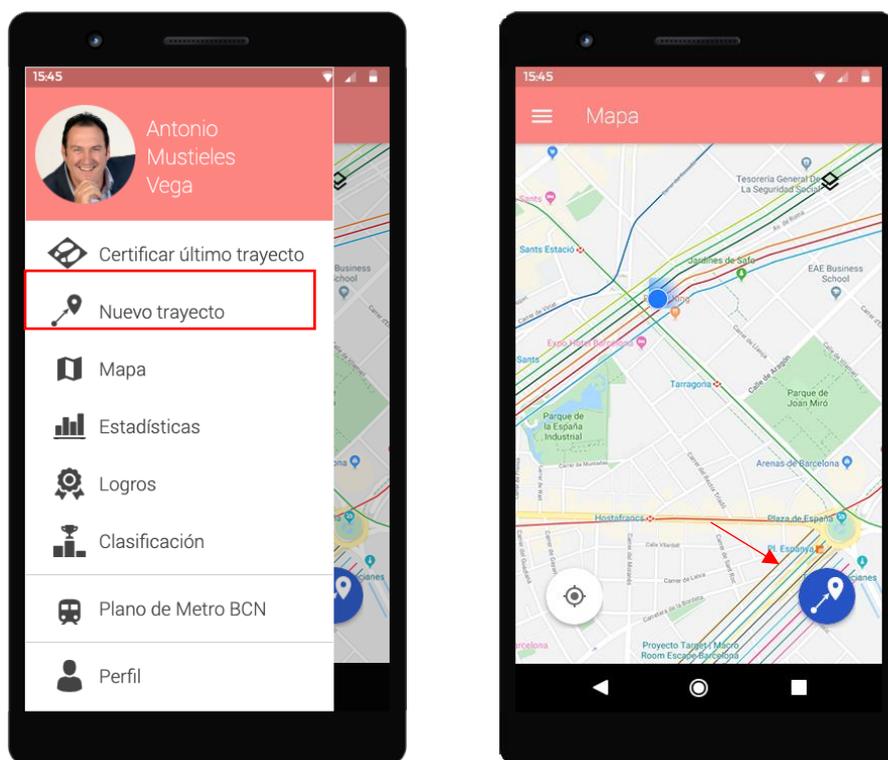


Figura 18- Detalle de la nueva opción introducida en el menú (Nuevo trayecto), y del botón flotante de Nuevo trayecto presente en la pantalla Mapa.

Respecto al modo de presentar la información, dos de los usuarios encontraron algo confusa la versión inicial de la pantalla de **Selección de trayecto** del prototipo de alta fidelidad. No les parecía correcto ni el título de las columnas (“Tiempo total a destino” y “Del que caminas”) ni el orden. Tras analizarlo, se encontraron unos títulos más claros y se cambió el orden de las columnas:

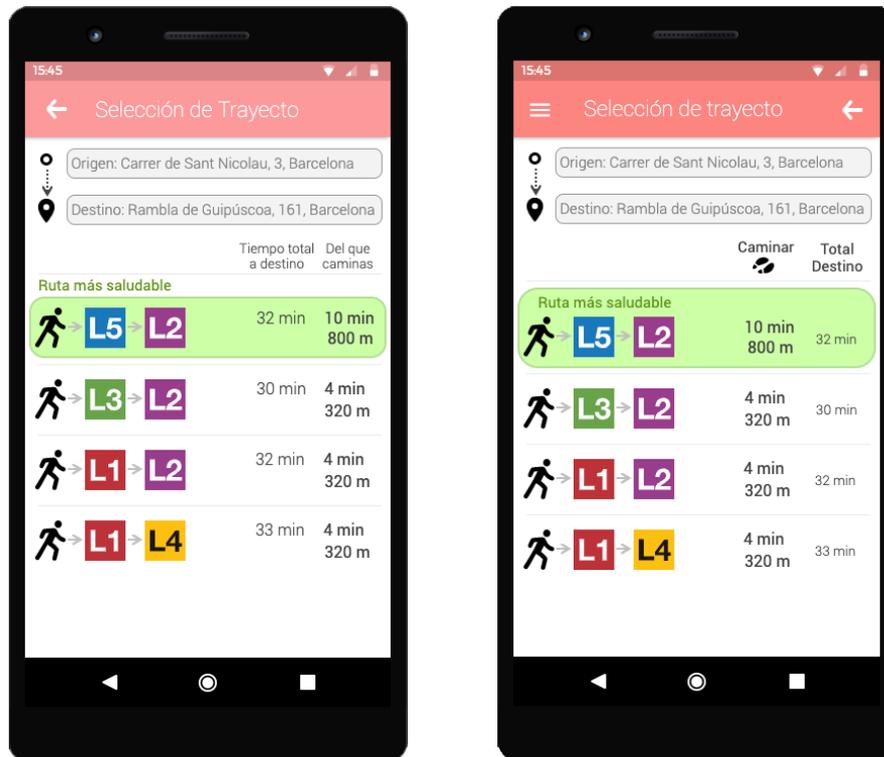


Figura 19 - Versión inicial (izquierda) y final (derecha) de la pantalla Selección de trayecto.

Otro cambio que puede apreciarse en la figura anterior es que, inicialmente, las pantallas de selección de destino/trayecto/parada se diseñaron para funcionar de manera exclusivamente secuencial: el usuario sólo podía ir a la siguiente pantalla o retroceder a la anterior (para modificar su elección). Los usuarios enseguida demandaron que pudiera accederse desde todas ellas al menú lateral. En efecto, si en medio del proceso de selección el usuario desistía, para volver al menú debía retroceder todas las pantallas, lo que resultaba tedioso. Se vio que era mucho más práctico introducir en todas ellas el acceso al menú, y poner la fecha de vuelta atrás en el lado opuesto.

Pero la pantalla que más controversia suscitó fue la de **Selección de parada**. Pese a que el concepto de *saltar parada* parecía una muy buena idea desde el planteamiento inicial del proyecto, a los usuarios no les quedaba tan claro. Se decidió rediseñarla. Para empezar, se modificó el título: se cambió de Saltar Paradas, a Selección de parada. Y en lugar de poner un botón de **Saltar** junto a cada estación (y los incrementos de tiempo, distancia y calorías que implicaría saltarse esa parada e ir andando a la siguiente), se pusieron directamente esos valores bajo el nombre de la parada. Con este nuevo planteamiento, la primera estación (la más cercana al usuario) no llevaría asociados valores incrementales. Por último, para facilitar la elección toda la fila de la tabla se convirtió en seleccionable. Cuando se presentó a los usuarios la nueva versión estuvieron de acuerdo en que el nuevo diseño era mucho más claro y fácil de entender.

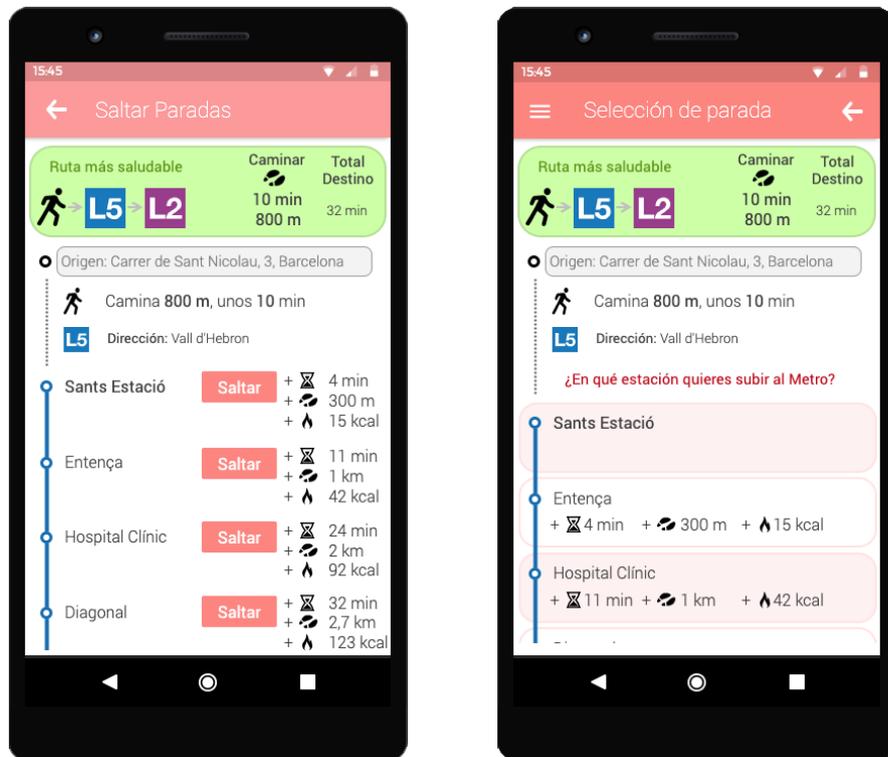


Figura 20 - Versión inicial (izquierda) y final (derecha) de la pantalla Selección de parada.

El último cambio que se realizó fue en la pantalla de **Certificar último trayecto**. Para clarificar su función se le cambió el nombre (de “Último trayecto” a “Certificar último trayecto”). Y para asegurar que la información cupiera en el caso de haber más trasbordos, se puso la información de tramos en orientación vertical. De este modo pudo incluirse también las direcciones origen y destino, información que los usuarios dijeron encontrar de utilidad (pues les facilitaba recordar el trayecto, en el caso de que la certificación se produjese días después de haberlo realizado). Se añadió también un botón para eliminar el trayecto (en caso de que no se hubiese completado).

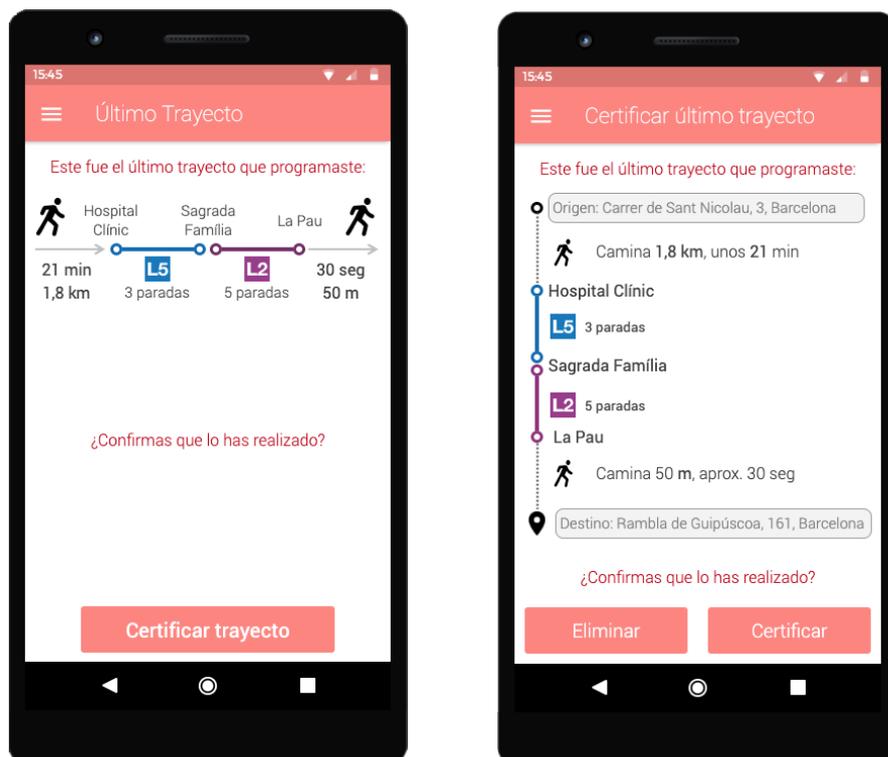


Figura 21 - Versión inicial (izquierda) y final (derecha) de la pantalla Certificar último trayecto.

3 Diseño técnico

Una vez finalizada la etapa de Diseño Centrado en el Usuario (DCU) entramos en el diseño técnico de la aplicación. Nos centraremos en la definición de los casos de uso y en el diseño de la arquitectura de la aplicación.

3.1 Definición de los casos de uso

La definición de los casos de uso es una técnica para recoger los requisitos de un nuevo sistema informático. Cada caso de uso proporciona un escenario que indica cómo debe actuar el sistema con el usuario o con otro sistema para alcanzar un objetivo concreto. Describe por tanto la secuencia de interacciones que tienen lugar entre el sistema (en este caso la aplicación objeto de este TFM), y sus actores en respuesta a un evento iniciado por un actor principal.

Se utilizará un diagrama en lenguaje UML (*Unified Modeling Language*) para representar los casos de uso y sus relaciones y, a continuación, se describirá cada uno de ellos.

3.1.1 Identificación de actores y casos de uso

3.1.1.1 Actores

En la aplicación Camina existe **un único actor**, el usuario que desea incrementar su actividad física maximizando los trayectos a pie en sus desplazamientos urbanos.

3.1.1.2 Casos de uso identificados

- CU-01: Registrarse en la aplicación
- CU-02: Logarse en la aplicación
- CU-03: Consultar perfil (se extiende con Crear perfil y Editar perfil).
- CU-04: Crear perfil.
- CU-05: Editar perfil.
- CU-06: Establecer nuevo destino.
- CU-07: Seleccionar trayecto preferido.
- CU-08: Elegir estación de metro inicial.
- CU-09: Iniciar trayecto.
- CU-10: Consultar estadísticas personales.
- CU-11: Consultar logros personales.
- CU-12: Consultar clasificación.
- CU-13: Ver ubicación en el mapa con las paradas de Metro cercanas.
- CU-14: Ver plano de Metro de Barcelona.
- CU-15: Certificar último trayecto realizado.

3.1.2 Diagrama UML de los casos de uso

En la siguiente figura se muestra el diagrama UML de los diferentes casos de uso identificados:

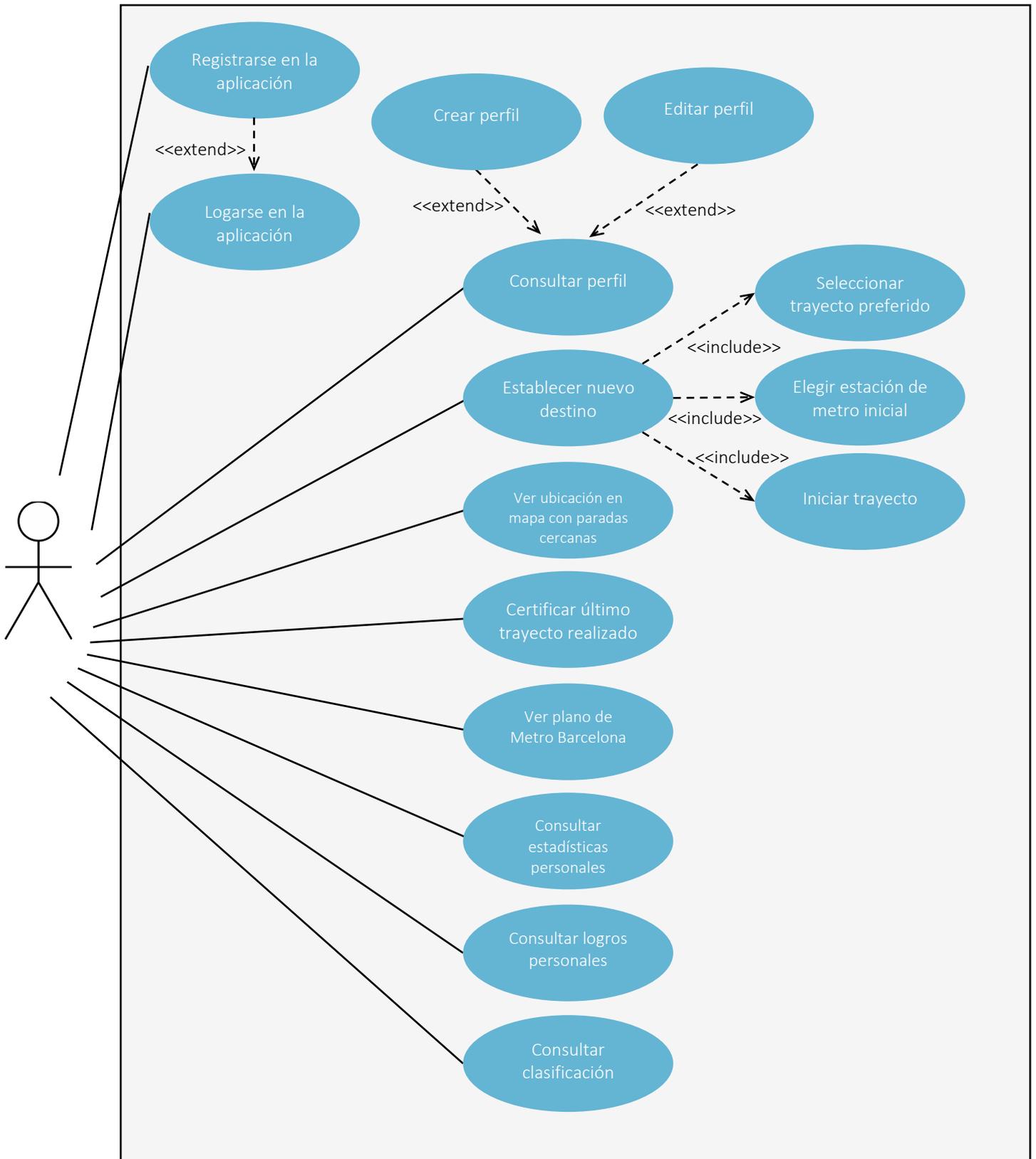


Figura 22 - Diagrama UML de los casos de uso

3.1.3 Detalle de los casos de uso

En este apartado se describen con más detalle los diferentes casos de uso, indicando los actores, las precondiciones, el flujo y las postcondiciones.

CU-01: Registrarse en la aplicación	
Descripción	Darse de alta en la aplicación
Actores	Usuario
Precondiciones	El usuario no puede estar registrado previamente (el e-mail no puede estar dado de alta ya en la BD).
Flujo	Desde la pantalla inicial, el usuario selecciona la opción correspondiente y accede a la página de Registro. Introduce los datos y clics Crear Cuenta.
Postcondiciones	<ul style="list-style-type: none"> • Si el correo no existía, se crea la cuenta y se notifica al usuario, que queda logado. Debe clics el botón Empezar a Caminar, que le lleva a la pantalla de Perfil, desde la que tiene acceso al menú principal de la aplicación. • Si el correo existía, se notifica al usuario que esa cuenta ya existe.

CU-02: Logarse en la aplicación	
Descripción	Hacer log-in en la aplicación
Actores	Usuario
Precondiciones	El usuario debe haberse registrado previamente en la aplicación o contar con una cuenta de Facebook o Google.
Flujo	Desde la pantalla de inicio, el usuario introduce su correo y contraseña para autenticarse en el sistema. También puede logarse con Facebook o Google.
Postcondiciones	<ul style="list-style-type: none"> • Si las credenciales son correctas, el usuario queda autenticado y accede a la pantalla de Mapa, desde la que tiene acceso al menú principal de la aplicación. • Si el e-mail o la contraseña no son correctos la aplicación muestra un mensaje de error.

CU-03: Consultar perfil	
Descripción	Ver el perfil de usuario
Actores	Usuario
Precondiciones	<ul style="list-style-type: none"> • CU-02 (haberse logado en la aplicación). • Estar en la página Perfil. • Que el usuario haya introducido previamente los datos de su perfil (CU-04).
Flujo	<ul style="list-style-type: none"> • Seleccionar la opción Perfil en el menú principal. También se llega automáticamente tras haberse registrado con éxito (CU-01).
Postcondiciones	<ul style="list-style-type: none"> • Si hay un perfil creado, se muestran los datos. • Si no había un perfil creado, se ofrece al usuario crearlo.

CU-04: Crear perfil

Descripción	Crear el perfil del usuario (nombre, edad, altura, peso, foto, etc.).
Actores	Usuario
Precondiciones	<ul style="list-style-type: none"> • CU-02 (haberse logado en la aplicación). • Estar en la página Perfil. • Que el usuario no haya creado ya previamente el perfil.
Flujo	<ul style="list-style-type: none"> • Seleccionar la opción Perfil en el menú principal. También se llega automáticamente tras haberse registrado con éxito (CU-01). • En la pantalla de Perfil, sin datos, clicar el botón para Crear Perfil. • Se muestran los campos y el usuario introduce los datos necesarios. • Clicar botón Crear.
Postcondiciones	<ul style="list-style-type: none"> • Finalizada la introducción de los datos, se almacena en la BD y se le muestran al usuario. • Se muestra el botón Editar.

CU-05: Editar perfil

Descripción	Modificar datos del perfil
Actores	Usuario
Precondiciones	<ul style="list-style-type: none"> • CU-02 (haberse logado en la aplicación). • Estar en la página Perfil. • Que el usuario haya introducido previamente los datos de su perfil (CU-04).
Flujo	<ul style="list-style-type: none"> • Seleccionar la opción Perfil en el menú principal. También se llega automáticamente tras haberse registrado con éxito (CU-01). • En la pantalla de Perfil, que muestra los datos actuales, clicar el botón Editar. • Se habilitan los campos para edición y el usuario puede modificar los valores. • Clicar botón Actualizar.
Postcondiciones	<ul style="list-style-type: none"> • Se deshabilitan los campos, que muestran los valores actualizados. • Se muestra el botón Editar.

CU-06: Establecer nuevo destino	
Descripción	Establecer un nuevo destino al que se quiere llegar
Actores	Usuario
Precondiciones	<ul style="list-style-type: none"> • CU-02 (haberse logado en la aplicación). • Estar en la página Nuevo Trayecto.
Flujo	<ul style="list-style-type: none"> • Seleccionar la opción Nuevo Trayecto en el menú principal. También se llega desde la página Mapa, clicando el botón flotante de Nuevo Trayecto. • Introducir una dirección en el campo de texto y clicar el botón Buscar. • Se muestra una lista de direcciones coincidentes. • El usuario elige la dirección deseada.
Postcondiciones	Se lleva al usuario a la página de Selección de trayecto.

CU-07: Seleccionar trayecto preferido	
Descripción	Elegir el trayecto deseado para llegar al destino, de entre todos los posibles que muestra la aplicación
Actores	Usuario
Precondiciones	<ul style="list-style-type: none"> • CU-02 (haberse logado en la aplicación), GPS activado y permisos de geolocalización concedidos. • Haber completado CU-06 y, por tanto, estar en la página de Selección de trayecto.
Flujo	<ul style="list-style-type: none"> • El usuario clica sobre una de las filas de la tabla que muestra los posibles trayectos para llegar al destino. • Si quiere, puede volver a la pantalla anterior para modificar el destino.
Postcondiciones	Se lleva al usuario a la página de Selección de parada.

CU-08: Elegir estación de metro inicial	
Descripción	Elegir la estación de desde la que se iniciará el trayecto en metro
Actores	Usuario
Precondiciones	<ul style="list-style-type: none"> • CU-02 (haberse logado en la aplicación), GPS activado y permisos de geolocalización concedidos. • Haber completado CU-07 y, por tanto, estar en la página de Selección de parada.
Flujo	<ul style="list-style-type: none"> • El usuario clica sobre una de las filas de la tabla que muestra las sucesivas paradas desde las que puede iniciar el trayecto en metro. • Al hacerlo, se actualizan los datos del trayecto total (tiempo total, tiempo a pie, distancia a pie), y se notifica al usuario que debe volver a clicar sobre la fila para confirmar la selección. • El usuario clica de nuevo para confirmar o lo hace en otra fila para elegir otra parada. • Si desea elegir otro trayecto, puede retroceder a la página anterior.
Postcondiciones	Se lleva al usuario a la página de Iniciar trayecto.

CU-09: Iniciar trayecto	
Descripción	Revisar datos del trayecto e iniciarlo
Actores	Usuario
Precondiciones	<ul style="list-style-type: none"> • CU-02 (haberse logado en la aplicación), GPS activado y permisos de geolocalización concedidos. • Haber completado CU-08 y, por tanto, estar en la página de Iniciar trayecto.
Flujo	<ul style="list-style-type: none"> • El usuario revisa todos los datos del trayecto elegido y, si está conforme, clicla el botón Ir a parada inicial. • Si no está conforme y quiere elegir otra parada de metro para iniciar su viaje, puede retroceder a la página anterior.
Postcondiciones	Se abre Google Maps para que guíe al usuario en ruta a pie hasta la estación de metro escogida.

CU-10: Consultar estadísticas personales	
Descripción	Permite al usuario conocer sus registros (diarios, mensuales y globales)
Actores	Usuario
Precondiciones	<ul style="list-style-type: none"> • CU-02 (haberse logado en la aplicación). • Estar en la página de Estadísticas.
Flujo	<ul style="list-style-type: none"> • Seleccionar la opción Estadísticas en el menú principal. • Por defecto se muestran las estadísticas diarias. Para ver las mensuales y las globales se debe cliclar sobre las pestañas correspondientes.
Postcondiciones	Se muestran los registros (paradas saltadas, calorías quemadas, distancia recorrida y tiempo que se ha caminado) correspondientes al intervalo temporal seleccionado (día, mes o global).

CU-11: Consultar logros personales	
Descripción	Permite al usuario revisar sus logros personales: récords obtenidos en un trayecto, y distinciones recibidas.
Actores	Usuario
Precondiciones	<ul style="list-style-type: none"> • CU-02 (haberse logado en la aplicación). • Estar en la página de Logros. • Si no se ha realizado ningún trayecto aún con la aplicación, los valores de récord estarán a cero. Si no se ha recibido todavía ninguna distinción, el panel de trofeos estará vacío.
Flujo	<ul style="list-style-type: none"> • Seleccionar la opción Logros en el menú principal. • Por defecto se muestran los récords. Para ver las distinciones el usuario debe cliclar sobre la pestaña correspondiente. El usuario puede alternar entre las dos pestañas.
Postcondiciones	Se muestran los valores máximos de cada registro (paradas saltadas, calorías quemadas, distancia recorrida a pie y tiempo caminado) obtenidos hasta el momento en un único trayecto, junto con la fecha que sucedió. En la pestaña Distinciones se muestra la lista de trofeos conseguidos.

CU-12: Consultar clasificación

Descripción	Permite al usuario ver el ranking de todos los usuarios que utilizan la aplicación, respecto a la distancia caminada.
Actores	Usuario
Precondiciones	<ul style="list-style-type: none"> • CU-02 (haberse logado en la aplicación). • Estar en la página de Clasificación.
Flujo	<ul style="list-style-type: none"> • Seleccionar la opción Clasificación en el menú principal. • Por defecto se muestran la clasificación del mes en curso. Mediante una pestaña puede accederse a la clasificación global.
Postcondiciones	Se muestran la clasificación de usuarios respecto a la distancia caminada.

CU-13: Ver ubicación en mapa con paradas de Metro cercanas

Descripción	Permite al usuario ubicar su posición en el mapa, que muestra también las paradas de Metro cercanas.
Actores	Usuario
Precondiciones	<ul style="list-style-type: none"> • CU-02 (haberse logado en la aplicación), GPS activado y permisos de geolocalización concedidos. • Estar en la página de Mapa.
Flujo	<ul style="list-style-type: none"> • Seleccionar la opción Mapa en el menú principal.
Postcondiciones	Se muestra el mapa y se ubica en él la posición del usuario. Se posicionan las estaciones de Metro cercanas.

CU-14: Ver el plano de líneas de Metro de Barcelona

Descripción	Permite al usuario consultar el plano del Metro de Barcelona.
Actores	Usuario
Precondiciones	<ul style="list-style-type: none"> • CU-02 (haberse logado en la aplicación). • Estar en la página Plano de Metro BCN.
Flujo	<ul style="list-style-type: none"> • Seleccionar la opción Plano de Metro BCN en el menú principal.
Postcondiciones	Se enlaza con una URL de tmb.cat que muestra el plano de las líneas del metro en formato PDF.

CU-15: Certificar el último trayecto realizado	
Descripción	Permite al usuario recuperar el último trayecto proyectado con la aplicación y certificar que lo realizó.
Actores	Usuario
Precondiciones	<ul style="list-style-type: none"> • CU-02 (haberse logado en la aplicación). • Estar en la página Certificar último trayecto. • Haber realizado previamente CU-09 (Iniciar trayecto). • Si el usuario previamente no ha realizado CU-09 se le notificará que no hay ningún trayecto previo registrado.
Flujo	<ul style="list-style-type: none"> • Seleccionar la opción Certificar último trayecto en el menú principal. • Se le muestra al usuario el último trayecto que programó y dos botones para certificarlo o borrarlo.
Postcondiciones	<ul style="list-style-type: none"> • Si el usuario certifica que ha realizado el trayecto, se actualizan sus registros y se le lleva a la página de Estadísticas (deberá pulsar el botón correspondiente). • Si el usuario elige eliminar el trayecto (porque no lo realizó) se borra la pantalla y se informa que no hay ningún trayecto previo registrado.

3.2 Diseño de la Arquitectura

La aplicación Camina correrá en dispositivos Android cliente, que se comunicarán con diferentes servidores a través de API's. Como se vio en el apartado 1.3.3 se utilizará Firebase de Google como servicio de backend, para almacenar todos los datos concernientes al usuario. Para guardar las fotos de perfil de los usuarios usaremos Cloud Storage, también de Google.

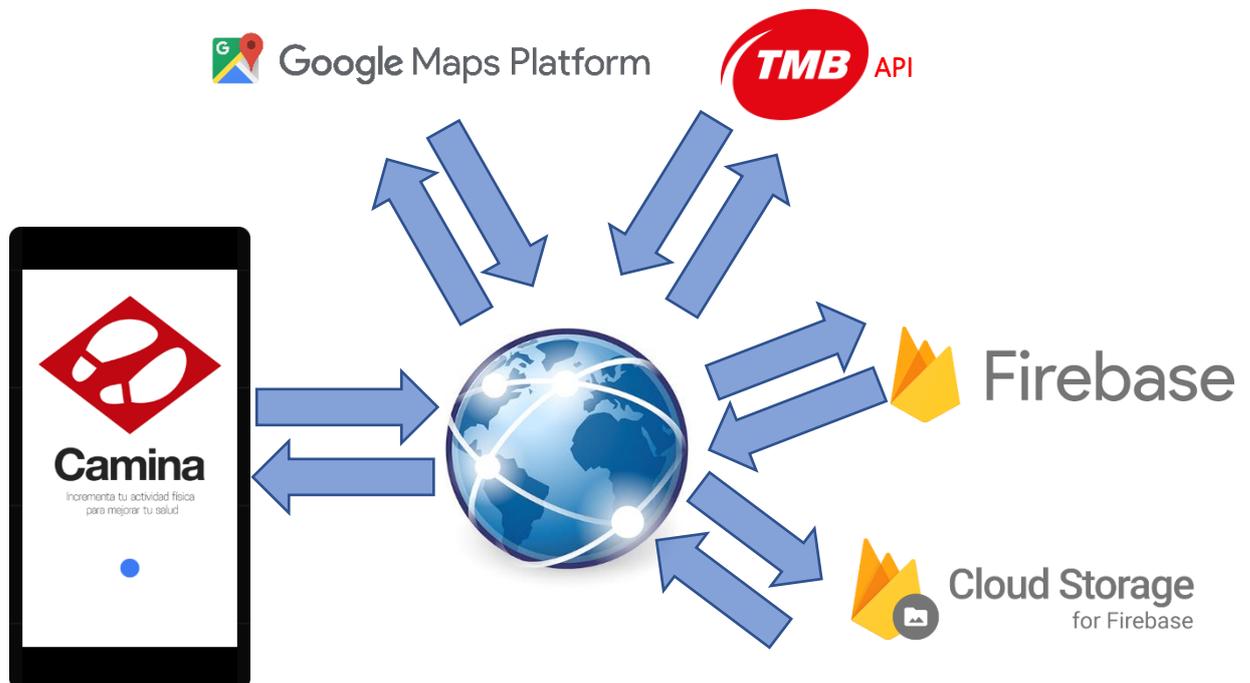


Figura 23 - Interrelación de la aplicación cliente con el servicio de backend y las API's de TMB y Google Maps

Para consultar los datos relativos a la red de Metro se utilizará la **API pública de TMB: Transit** para la ubicación de las paradas de Metro, y **Planner** para obtener los tiempos, distancias e indicaciones de rutas en metro y a pie se utilizarán.

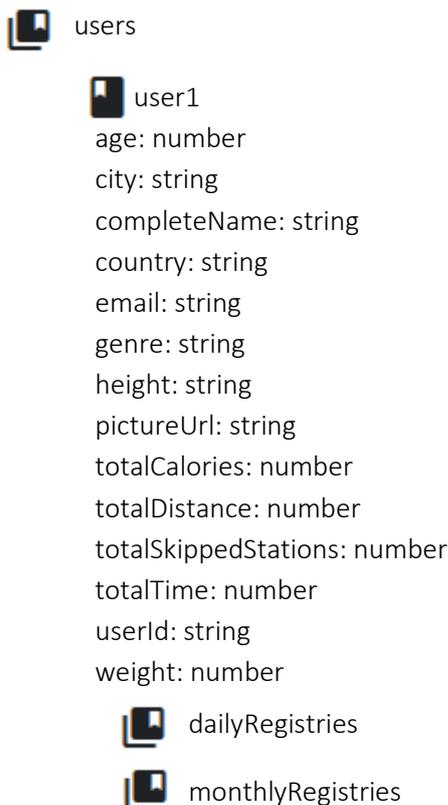
Para geolocalizar al usuario usaremos la API de **Google Maps** y, para autocompletar las direcciones introducidas por el usuario utilizaremos la API de **Google Places**.

3.2.1 Diseño de los datos

El servicio que se utilizará como *backend*, Firebase, no es un sistema de gestión de bases de datos relacionales, por lo que no es posible modelar los datos mediante un diagrama relacional.

Los datos se almacenan como objetos **JSON**, por lo que la base de datos podría representarse como un árbol JSON almacenado en la nube. No tendremos ni registros ni tablas, a diferencia de lo que encontramos en las bases de datos SQL. En su lugar contaremos con datos, documentos y colecciones. Cuando se añaden nuevos datos al árbol, estos se convierten en un nodo de la estructura JSON existente, y se le asigna una clave [8].

Tras un primer análisis de los datos que será necesario almacenar en la base de datos Firebase, determinamos que utilizaremos las siguientes colecciones (será revisado en el futuro, en la etapa de implementación).



Y estas son las subcolecciones que se incluyen en los documentos 'user':

- dailyRegistries
 - day_month_year
 - calories: number
 - day: number
 - distance: number
 - month: number
 - skippedStations: number
 - time: number
 - year: number

- montlyRegistries
 - month_year
 - calories: number
 - distance: number
 - month: number
 - skippedStations: number
 - time: number
 - year: number

3.2.2 Patrón de diseño del Software

Para el diseño del software se utilizará el paradigma **Modelo – Vista – Controlador (MVC)**. En este patrón se separan los **datos** de la aplicación, la **interfaz de usuario** y la **lógica de control** en tres componentes distintos (modelo, vista y controlador respectivamente).

Cada componente tiene una responsabilidad bien diferenciada, y el hecho de que estén separados facilita la escalabilidad de la solución y que los cambios realizados en uno de ellos no impacten necesariamente al resto.

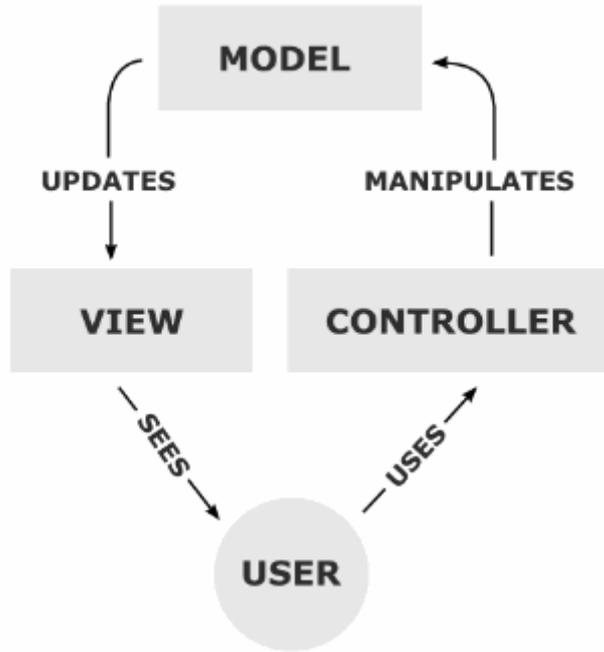


Figura 24 - Esquema del paradigma MVC, extraído de la Wikipedia [9]

En el caso de Android, la aplicación de este paradigma quedaría plasmado en el siguiente esquema:

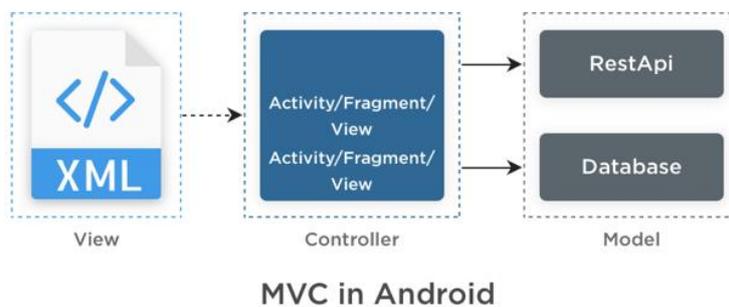


Figura 25 – Asimiliación del paradigma Modelo Vista Controlador a una aplicación Android [10]

La vista (interfaz de usuario) se define en los archivos xml; cada Activity/Fragments/View es el controlador y se encarga de responder a las interacciones de los usuarios y actualizar el modelo cuando es necesario; y el modelo gestiona el acceso a los datos, en nuestro caso vía APIs.

4 Implementación

4.1 Decisiones de diseño

4.1.1 El proyecto Android

La aplicación se ha desarrollado utilizando el IDE Android Studio 3.5.3. Se ha especificado como versión mínima de Android sobre la que correrá la aplicación, la Según Android Studio, eligiendo esta versión de plataforma llegaremos a un 85% de los usuarios. Consideramos que resulta un compromiso equilibrado entre llegar al mayor número de potenciales usuarios y ofrecer prestaciones suficientes del sistema operativo (fundamentalmente soporte para *Material Design* y la introducción de RecyclerView, que resulta imprescindible para esta aplicación).

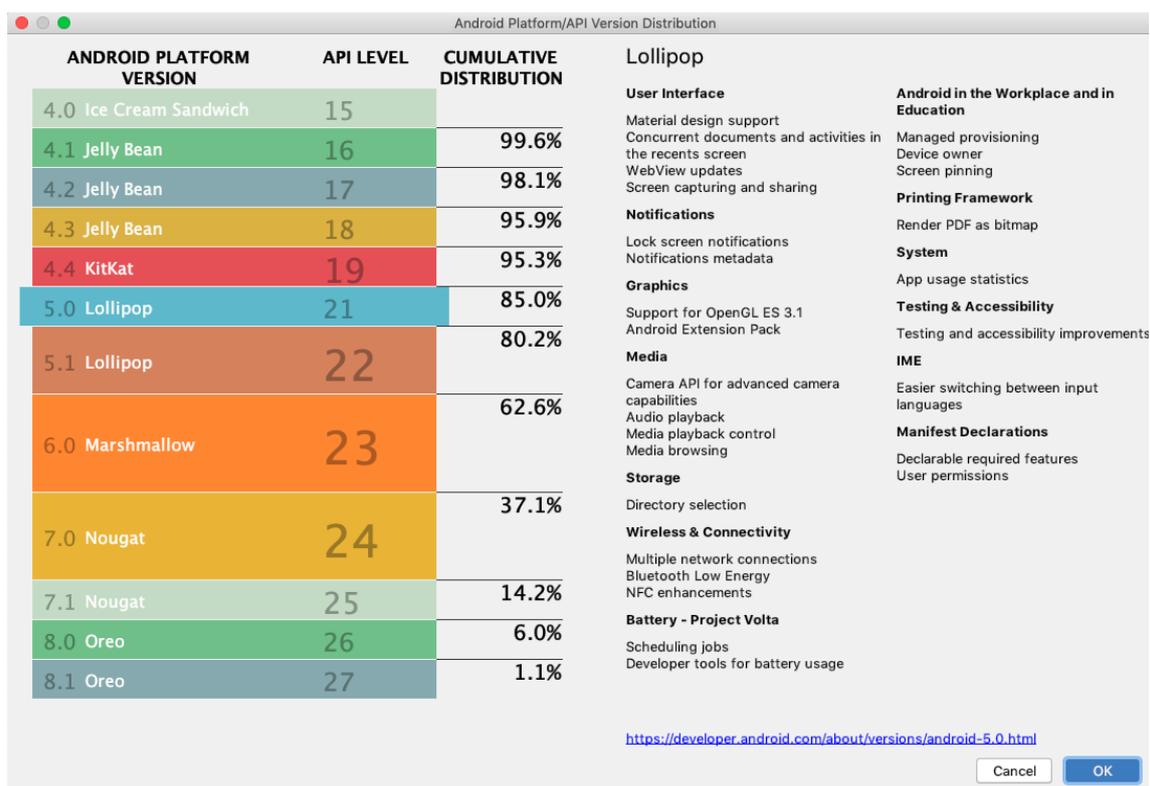


Figura 26 - Distribución de las diferentes versiones de la API de Android

El lenguaje utilizado para la programación ha sido Java.

4.2 Principales librerías utilizadas

Para agilizar algunas tareas se han usado las siguientes librerías de terceros:

- **MaterialDrawer** (by Mike Penz): Para la implementación del menú lateral Utilizada bajo licencia Apache v.2.0, <http://www.apache.org/licenses/LICENSE-2.0>

- **CircleImageView** (de Henning Dodenhof): Para la implementación de la imagen circular del perfil de usuario.
Utilizada bajo licencia Apache v.2.0, <http://www.apache.org/licenses/LICENSE-2.0>
- **Retrofit2** (de Square, Inc.): Para convertir las llamadas a la API HTTP de TMB en interfaces Java.
Utilizada bajo licencia Apache v.2.0, <http://www.apache.org/licenses/LICENSE-2.0>

IMPORTANTE: Se deben mantener las versiones incluidas en *Gradle*. No actualizarlas.

implementation 'com.squareup.retrofit2:retrofit:2.5.0'
implementation 'com.squareup.retrofit2:converter-gson:2.5.0'

- **okhttp3** (de Square, Inc.): Cliente HTTP. Para las llamadas a la API de TMB.
- **Picasso** (de Square, Inc): Para la descarga de las imágenes desde URLs de internet.
Utilizada bajo licencia Apache v.2.0, <http://www.apache.org/licenses/LICENSE-2.0>

4.3 APIs utilizadas

Se decidió utilizar siempre que fuera posible API's públicas y gratuitas, basadas en Open Data. El uso de la **API Google Maps** es casi inevitable, y su uso nunca estuvo en duda.

Se valoró desde el principio, ya en la fase de análisis de objetivos y requisitos, la utilización de Google Directions para obtener las indicaciones para llegar desde el punto A al punto B. Pero el servicio que ofrece Google es de pago, si bien proporciona un crédito inicial válido durante un año con el que podríamos haber realizado este TFM sin problemas. No obstante, tras analizar en detalle la **API pública de TMB** se vio que se podían obtener unos servicios equivalentes de manera gratuita. Ese fue el criterio clave para escoger la API para elaborar las rutas.

Por último, se decidió incorporar la **API Google Places**. Pese a ser también de pago, el saldo gratuito de evaluación es más que suficiente para este proyecto. En nuestra aplicación se utiliza para autocompletar las entradas del usuario cuando está introduciendo un destino.

En los subapartados siguientes se detalle un poco más la manera en que estas APIs se integran en **Camina** y los servicios que prestan. El orden en el que se presentan se corresponde con el orden de utilización en la app.

4.3.1 API Google Maps

La API de Google Maps se utiliza en este proyecto fundamentalmente para geolocalizar al usuario y mostrarle las paradas de Metro cercanas. Desde la opción del menú 'Mapa', que es también el punto de entrada a la aplicación tras el log-in (si no hay ningún trayecto pendiente de certificar), se ubica al usuario en un mapa en el que se muestran las estaciones de metro cercanas.

Se ha optado por ocultar la capa '**transit**' (transporte público) que ofrece Google, para poder situar marcadores personalizados para las paradas de Metro. Para conseguirlo es necesario configurar el estilo del mapa genérico utilizando un objeto JSON. También se ha desactivado la *toolbar* que viene por defecto, para impedir que el usuario pueda abrir el mapa directamente en

la app de Google Maps, ni conseguir indicaciones para llegar a un marcador en concreto directamente con esa aplicación.

Cuando el usuario clicla uno de los iconos de estación, se abre una ventana de información que indica el nombre de la parada y origen y destino de la línea a la que pertenece:



Figura 27 - Captura de pantalla Mapa - Geoposicionamiento de usuario con icono de parada e información

La API gestiona de manera automática el acceso a los servidores de Google Maps, la descarga de los datos, la visualización de mapas y la respuesta a eventos. El acceso a la misma se realiza utilizando el SDK específico para Android.

4.3.2 API pública TMB Transit

Para conseguir la ubicación de las distintas estaciones de metro se utiliza la API pública v1 de TMB, y concretamente el servicio **Transit**. Para poder utilizar esta API es necesario darse de alta en el Portal del Desarrollador (<https://developer.tmb.cat/>), registrar la aplicación, y conseguir un *id de aplicación* y una *key de aplicación*.

Existe una consulta específica para descargar todas las líneas de Metro:

```
https://api.tmb.cat/v1/transit/linies/metro?app_id={your_app_id}&app_key={your_app_key}
```

Sin embargo, tras múltiples problemas de parseado utilicé otro enfoque para conseguir esta información. Para cada una de las líneas de metro, realizo una consulta específica:

```
https://api.tmb.cat/v1/transit/linies/metro/{line_code}/estacions?app_id={your_app_id}&app_key={your_app_key}
```

Tras parsear la respuesta en forma de objeto JSON a las clases java necesarias, podemos acceder a todos los datos necesarios para colocar los marcadores personalizados en el mapa: latitud y

longitud de la ubicación de la parada, nombre de la línea, origen y destino de la línea, nombre de la estación, color de la línea, etc.

4.3.3 API Google Places

No se encontró una opción gratuita para la gestión de las direcciones o lugares que introduce el usuario cuando elige el destino, por lo que se decidió utilizar la **API Google Places**. Concretamente se utiliza el servicio de **Place Autocomplete**, que devuelve predicciones de lugares (negocios, direcciones, puntos de interés) conforme el usuario va introduciendo texto en el campo correspondiente del formulario de búsqueda.

La incorporación a la app la realizo lanzando un `widget Autocomplete` mediante un *intent* en modo pantalla completa.

```
public void onSearchCalled() {  
    // Set the fields to specify which types of place data to return.  
    List<Place.Field> fields = Arrays.asList(Place.Field.ID, Place.Field.NAME,  
        Place.Field.ADDRESS, Place.Field.LAT_LNG);  
    // Start the autocomplete intent.  
    Intent = new Autocomplete.IntentBuilder(  
        Autocomplete.ActivityMode.FULLSCREEN, fields).setCountry("ES") //Constrain  
            the results to Country: Spain  
        .setLocationRestriction(RectangularBounds.newInstance( //Restrict results  
            new LatLng(41.166217, 1.436145), //to province of BCN  
            new LatLng(42.013023, 2.839965)))  
        .build(this);  
    startActivityForResult(intent, AUTOCOMPLETE_REQUEST_CODE);  
}
```

Como se puede apreciar en el código de arriba, es posible acotar los resultados mediante unos límites rectangulares. En nuestro caso, dado que el uso de esta aplicación se restringe únicamente a la red de Metro de TMB, se ha restringido a la provincia de Barcelona. Se debe indicar también qué datos queremos que nos devuelvan en la respuesta. Los básicos para nuestra aplicación son las coordenadas del lugar destino.

4.3.4 API TMB Planner

La API de la cual se hace un uso más intensivo en la aplicación Camina es del componente **Planner** de la API de TMB v1. Se trata de un servicio de cálculo de rutas en transporte público, basado en el proyecto *open source OpenTripPlanner*. Aunque no lo explicita en la escueta documentación, también facilita trayectos a pie (con ciertas limitaciones).

En nuestra aplicación se utiliza para:

- 1.) Ofrecer los posibles trayectos combinados (caminando + red de Metro) desde su ubicación actual hasta el destino introducido. El formato de este tipo de consultas es el siguiente:

```
https://api.tmb.cat/v1/planner/plan?app_id={your_app_id}&app_key={your_app_key}&mode=SUBWAY%2CWALK&showIntermediateStops=TRUE&fromPlace=41.3756419945693335%2C2.149231681286226&toPlace=41.2874631%2C2.0730214
```

Dado que la consulta tiene múltiples parámetros, es necesario utilizar un Hash Map, que se define de la siguiente manera en el interfaz TMBService:

```
@GET(ApiTMBConstants.TMB_PLANNER_ENDPOINT)
Call<JsonObject> getPlannedRoute(@QueryParam Map<String, String> options);
```

La consulta se construye en el método AskItinerariesFromOriginToDestination() de la RouteSelectionActivity.:

```
// Construct the Map of query parameters to call TMBPlannerAPI to ask for routes to get from origin to destination
```

```
Map<String, String> TMBPlannerQuery = new HashMap<>();
TMBPlannerQuery.put("showIntermediateStops", "TRUE");
TMBPlannerQuery.put("mode", "SUBWAY,WALK");
TMBPlannerQuery.put("fromPlace", originDestination.getOriginCoordinates());
TMBPlannerQuery.put("toPlace", originDestination.getDestinationCoordinates());
```

```
Call<JsonObject> callTMBPlannerAPIResponse;
callTMBPlannerAPIResponse = new TMBAdapter().getPlannedRoute(TMBPlannerQuery);
```

Después se encola, con una *callback* para gestionar la respuesta:

```
callTMBPlannerAPIResponse.enqueue(new Callback<JsonObject>() {
    @Override
    public void onResponse(@NonNull Call<JsonObject> call, @NonNull
        Response<JsonObject> response) {

        //If a swipe to refresh was performed, stop the spin
        if (swipePerformed) {

            swipeRefreshLayout.setRefreshing(false);
            swipePerformed = FALSE;
        }

        JsonObject responseTMBPlannerAPI = response.body();
        if (responseTMBPlannerAPI != null) {

            // If a correct answer is obtained from server, parse the data
            ParseTMBPlannerAPIResponse (responseTMBPlannerAPI);
            // Finally, setup Recycle View
            setupRecyclerView();

        } else {

            //No itineraries to go from origin to destiny are available within
            TMB Metro network
            swipeRefreshLayout.setRefreshing(false);
            //And tell the user
            snack = Snackbar.make(coordinatorLayout, "No es posible ir al
            destino indicado utilizando la red de Metro de TMB",
            Snackbar.LENGTH_LONG);
            CoordinatorLayout.LayoutParams params =
            (CoordinatorLayout.LayoutParams)
            snack.getView().getLayoutParams();
            params.setMargins(0,0,0,0);
            snack.getView().setLayoutParams(params);
            snack.show();
        }
    }
}
```

Si la respuesta que recibimos del servidor no es nula, quiere decir que se ha encontrado una posible ruta que combine caminar (un máximo de 800 m, límite no modificable que impone la API) y Metro. En ese caso llamamos al método que se encarga del parseado de la respuesta. Pese a que inicialmente se pretendía realizar esta tarea utilizando la librería **retrofit2**, no fue posible conseguir que no diera errores, probablemente debido a la elevada complejidad del objeto JSON de respuesta.

Finalmente fue necesario realizar la conversión de manera más manual, utilizando directamente la librería `com.google.gson.Gson`. La clase java sobre la que se parsea la respuesta es `Plan`, dentro de `TMBPlannerAPI` y a su vez dentro de `model`.

```
// Convert and parse the date of API call
```

```
plan.setDate(Epoch2DateString(responseTMBPlannerAPI.getAsJsonObject("plan").getAsJsonPrimitive("date").getString()));
```

```
// We will use a Gson object to parse some objects into classes automatically
```

```
Gson g = new Gson();
```

```
// Parse JSON objects 'from' and 'to' into Point objects
```

```
Point = g.fromJson(responseTMBPlannerAPI.getAsJsonObject("plan").getAsJsonObject("from"), Point.class);
```

```
plan.setFrom(point);
```

```
point = g.fromJson(responseTMBPlannerAPI.getAsJsonObject("plan").getAsJsonObject("to"), Point.class);
```

```
plan.setTo(point);
```

```
// Gson class TypeToken is used to assist us in finding the correct Type for our class configuration when trying to get a List of objects of that class
```

```
Type itineraryListType = new TypeToken<ArrayList<Itinerary>>().getType();
```

```
List<Itinerary> itineraries =
```

```
g.fromJson(responseTMBPlannerAPI.getAsJsonObject("plan").getAsJsonArray("itineraries"), itineraryListType);
```

Aprovechamos este método de parseado para **ordenar los itinerarios** tal y como se los presentaremos al usuario. Recordemos que Camina es una aplicación para **fomentar la actividad física**. Preferiremos los trayectos en los que más se camine y que sean lo más eficientes posibles (que la duración total sea la menor posible).

- Ordenaremos primero por tiempo de caminata y distancia de caminata, de mayor a menor.
- Después por duración total, de menor a mayor.

```
Collections.sort(itineraries, new ItineraryComparator());
```

```
// Add the sorted list to the plan object
```

```
plan.setItineraries(itineraries);
```

- 2.) Calcular las distancias y el tiempo de trayecto **a pie** desde la ubicación actual del usuario hasta cada una de las estaciones de Metro de la primera línea del trayecto escogido. Con estas consultas podremos obtener los datos necesarios para facilitar al usuario la información de los beneficios que obtendrá si, en lugar de escoger la estación que tiene

más próxima, elige estaciones más lejanas. La consulta, en este caso, tiene el siguiente formato:

```
https://api.tmb.cat/v1/planner/plan?app_id={your_app_id}&app_key={your_app_key}&mode=WALK&showIntermediateStops=FALSE&fromPlace=41.375718843709244%2C2.149217056918744&toPlace=41.37551%2C%202.149382
```

Llegar a entender esta API ha sido una tarea bastante complicada. **La documentación existente es escueta e imprecisa, cuando no directamente errónea.** A fuerza de prueba y error se acaba aprendiendo su uso y sus limitaciones. Algunas de las limitaciones e incoherencias encontradas, que ha entorpecido enormemente el desarrollo, han sido:

- La línea 8 (Barcelona-Sant Boi Molí Nou), que en realidad es de Ferrocarrils de la Generalitat, aparece como integrante de la red de Metro en las consultas de trayectos (Planner). Sin embargo, no aparece en el servicio Transit (donde queda excluida por no ser estrictamente de TMB). Esto provoca que las estaciones de esta línea no aparezcan ubicadas en el mapa.
- De la documentación se deduce que es posible realizar consultas con los siguientes parámetros:

Planner

QUERY PARAMETERS

fromPlace required	string Example: "41.3755204,2.1498870" Origen de la ruta en coordenades lat/lon
toPlace required	string Example: "41.422520,2.187824" Destí de la ruta en coordenades lat/lon
date required	string/ <code>^(0[1-9] 1[012])-(0[1-9] 12 [0-9]3[01])-(19 20)d(2)\$/</code> Example: "MM-DD-YYYY" Dia d'arribada/sortida de la ruta (en funció del paràmetre arriveBy). Format mm-DD-YYYY. Per exemple 05-26-2014
time required	string/ <code>^(0[1-9] 1[012]):([0-5][0-9])(am pm)\$/</code> Example: "11:58am" Hora d'arribada/sortida de la ruta (en funció del paràmetre arriveBy). Per exemple 11:58am
arriveBy required	string Indica si el dia/hora especificats determinent la hora de sortida (false) o d'arribada (true)
mode required	string Example: "TRANSIT,WALK" Indica els modes de transport a utilitzar
maxWalkDistance	number Example: 300 Màxima distància caminant en la ruta resultant
showIntermediateStops	boolean Indica si es retornen les parades intermitges dins de cada part del trajecte resultant

Figura 28 - Documentación API TMB - Parámetros de consulta

Uno de los parámetros que nos hubiese interesado modificar es la 'maxWalkDistance'. Si se pudiese aumentar podríamos acceder a trayectos cuya primera parada de Metro estuviera más distante al usuario. Esto es especialmente útil para, por ejemplo, localidades del área metropolitana donde únicamente hay una estación de Metro.

Pues bien, se indique el valor que se indique para este parámetro en la consulta a la API, la respuesta que ofrece fija siempre el valor maxWalkDistance=800 (metros). Puede verse en el objeto JSON de respuesta, que incluye los **requestParameters**. Esto implica que, para poblaciones con una única parada, solo aquellos usuarios situados a menos de 800 m de distancia de la parada obtendrán itinerario para llegar a Barcelona. El resto recibirá como respuesta que no es posible llegar desde su ubicación actual al destino indicado utilizando la red de Metro.

- Probablemente lo más grave es la **inconsistencia de algunos resultados**. Un itinerario, denominado **Itinerary** en el objeto JSON que devuelve la consulta y en la clase sobre la que se parsea, está formado por varias **Leg**. Una **Leg** representa un trayecto, que bien puede ser andando (**mode: "WALK"**) o en Metro (**mode:"SUBWAY"**). Los objetos **Itinerary** tienen las propiedades **walkTime** y **walkDistance**. Por su parte, las **Leg** a pie (**mode: "WALK"**) tienen las propiedades **distance** y **duration**. La mayoría de las veces, conforme al sentido común, el valor de la propiedad **walkTime** del **Itinerary** se corresponde con la suma de todos los valores **distance** de las **Leg** a pie que lo integran. Lo mismo para la duración. Sin embargo, algunas veces esto no se cumple. No se ha encontrado ninguna explicación al problema. Y se trata de un error muy evidente: por ejemplo, vemos un campo **walkDistance** de 25m, en un **Itinerary** que tiene 3 **Leg** a pie, una con 300 m, otra con 128m y otra con 100m. Detectar estos errores y encontrar un método de sortearlos (no utilizar nunca el valor **walkDistance** del **Itinerary**, sino la suma de todos los valores **distance** de sus **Leg**) a pie ha sido una tarea complicada.

4.3.5 API FireBase

Camina utiliza los servicios de Firebase para llevar a cabo la autenticación de los usuarios y para almacenar cierta información (perfil, estadísticas, récords, distinciones, último trayecto realizado, etc.). En el momento de elaboración de esta documentación (para la PEC3), las funcionalidades relacionadas con FireBase no han sido implementadas todavía (detalle en el apartado 7. Revisión de la planificación (PEC3). Debido a este motivo, este apartado se ampliará posteriormente, una vez concluida la implementación.

4.4 Layouts

Una de las tareas que han resultado más laboriosas y complicadas ha sido la implementación de los *layouts* necesarios para presentar al usuario la información de los trayectos.

Concretamente tres pantallas han sido especialmente complicadas: Selección de trayecto, Selección de parada e Iniciar trayecto. Las podemos ver a continuación:

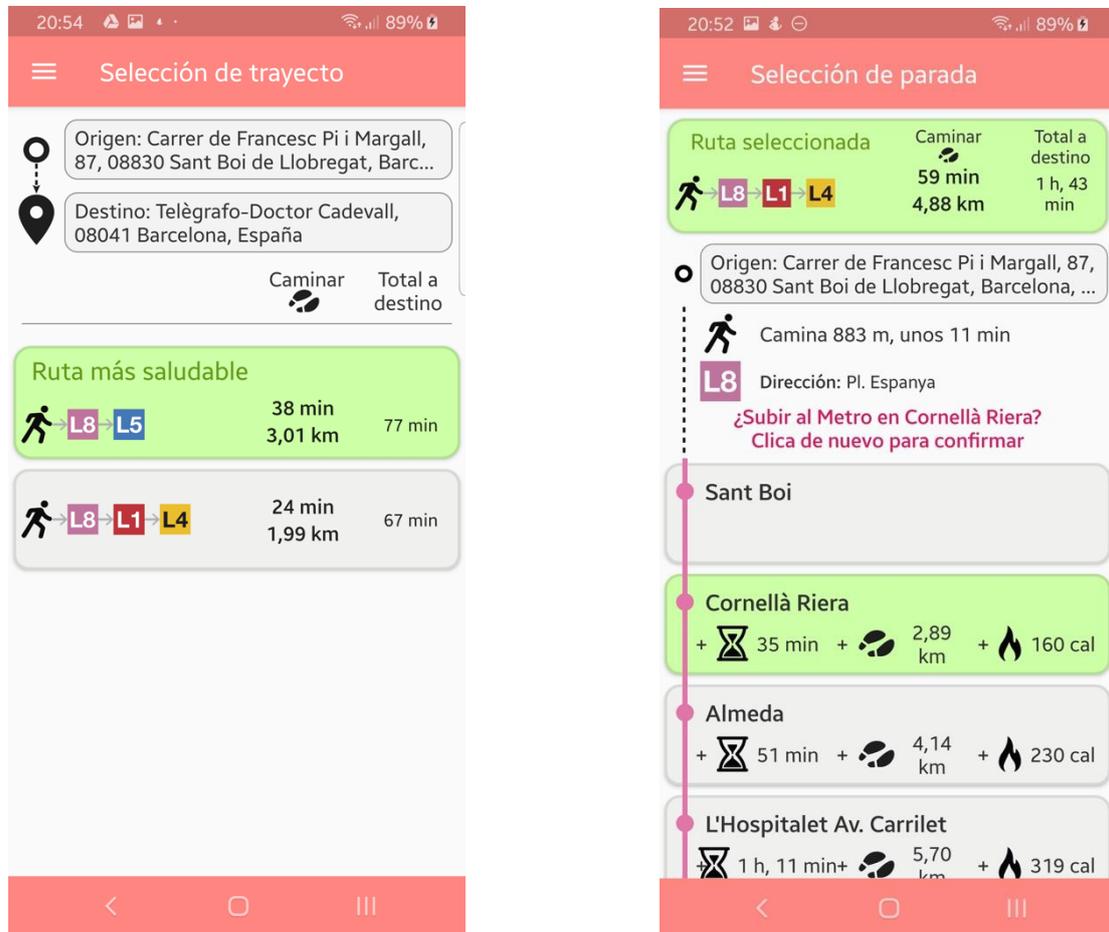


Figura 29 - Pantallas de Selección de trayecto y Selección de parada

En la primera de ellas, **Selección de trayecto**, la complejidad radica en la colocación de los elementos gráficos, las relaciones entre ellos, y con los elementos de texto. Son críticas las restricciones aplicadas para que, pese a que el texto crezca (las unidades van variando: a veces se indican minutos, otras minutos y segundos, y en ocasiones horas y minutos) nunca nada se superponga.

Además, los elementos gráficos de los ítems de la *Recyclerview* deben ser dinámicos: a veces tendremos un único icono de línea, mientras que otras veces podemos encontrarnos hasta 4. Y es que, aunque por lo que se puede deducir de la documentación de la API de TMB, el número de trasbordos está limitado a 2 (y, por tanto, nunca deberíamos ver más de 3 iconos), algunas consultas devuelven 3 conexiones. Por ello se decidió preparar los *layouts* para soportar un máximo de 4.

En la pantalla **Selección de Parada**, además de las cuestiones anteriores (que se repiten), se introducen 2 elementos gráficos adicionales (línea vertical discontinua que indica el trayecto andando, y línea continua del color de la primera línea de metro). Además, esta última línea debe colocarse superpuesta a los elementos del RecyclerView, y debe coincidir con exactitud en el punto medio de los iconos que simbolizan las estaciones (círculos del color de la línea). Estos iconos están colocados dentro de los ítems del RecyclerView, puesto que deben moverse junto con ellos cuando el usuario realiza scroll. La solución adoptada ha consistido en el siguiente *ConstraintLayout*:

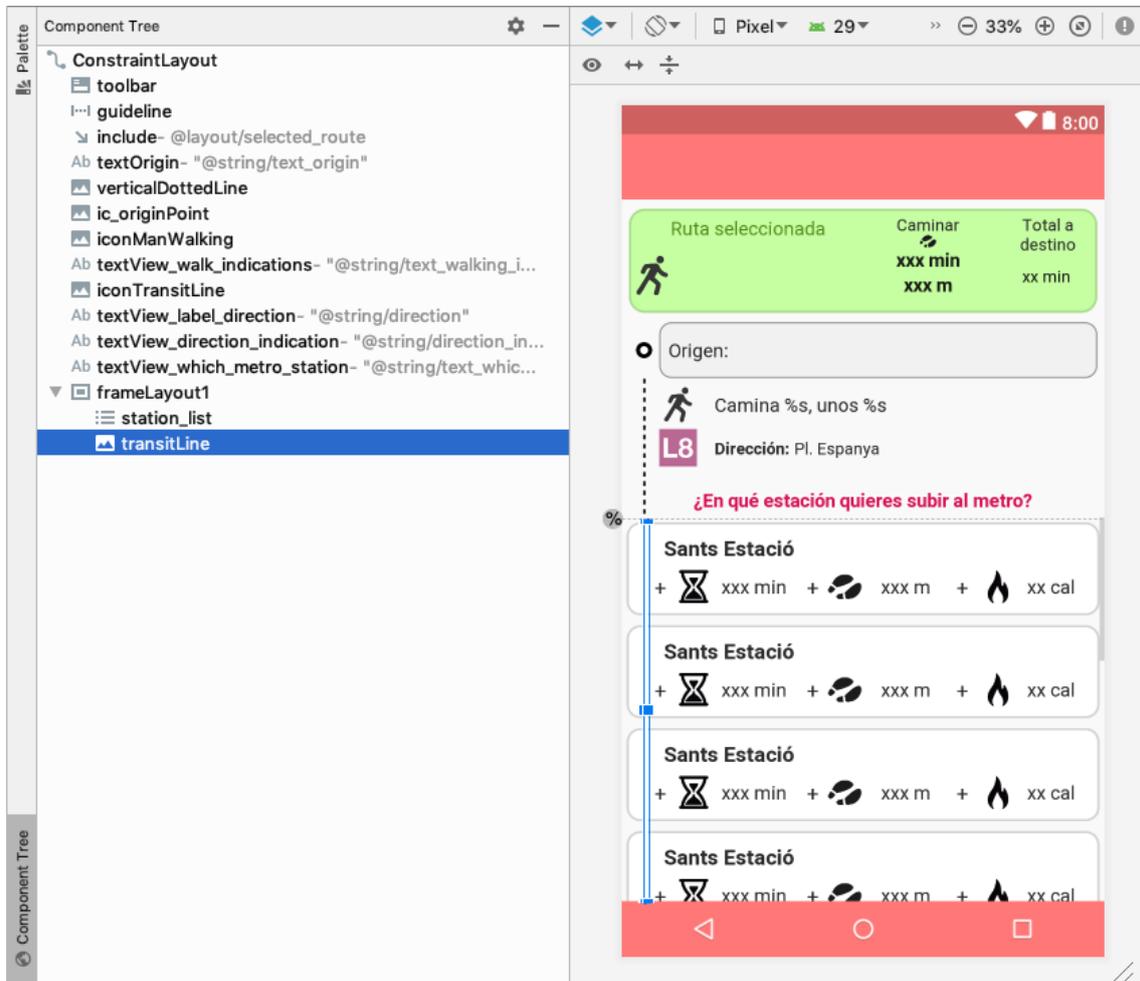


Figura 30 - Detalle del Diseño del layout de la pantalla Selección de parada

Y probablemente la pantalla más complicada de implementar haya sido la de Iniciar Trayecto:

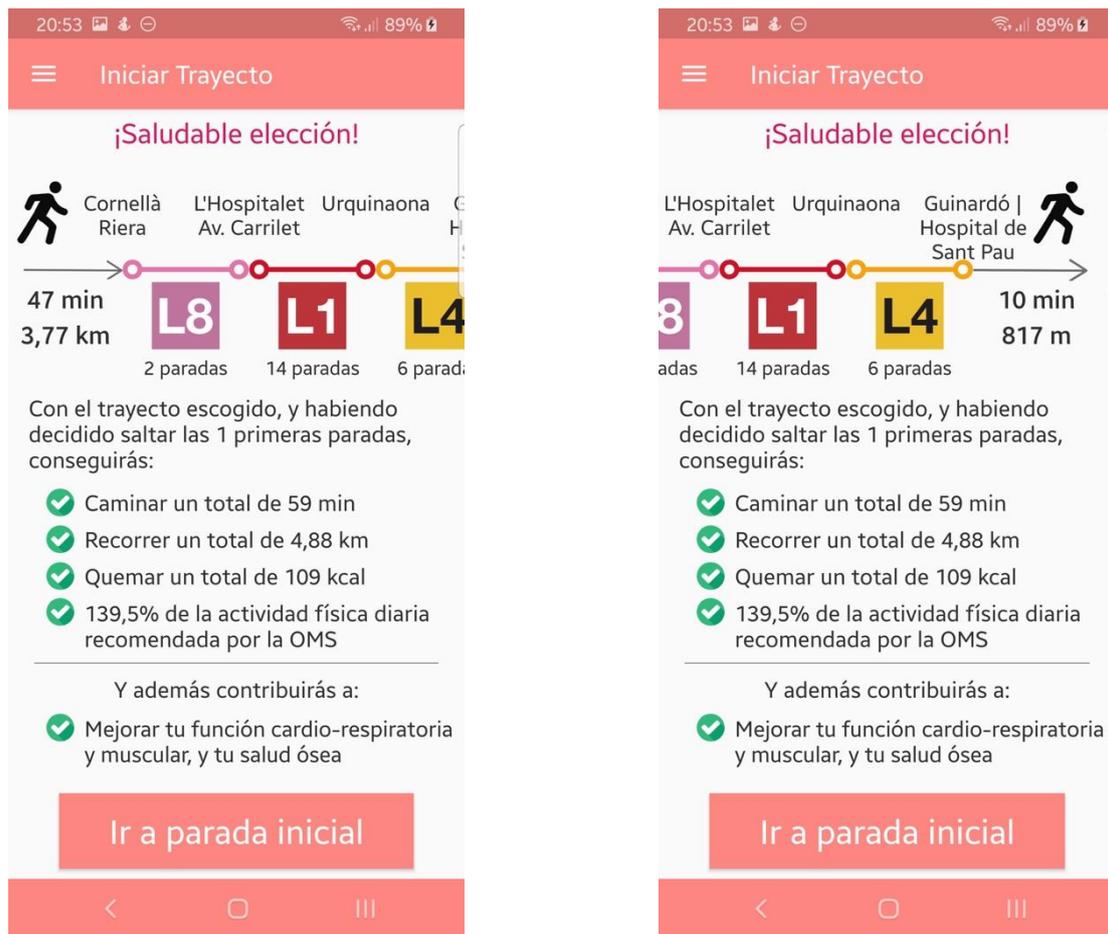


Figura 31 - Pantalla de Iniciar Trayecto - Detalle de Scrollview horizontal

A fin de poder incorporar todos los trayectos (Caminar – trayecto en línea metro 1 – trayecto en línea 2 ... - Caminar) ha sido necesario incluir un *ScrollView* horizontal. Dado que, de nuevo, el número de elementos a incluir es variable, la solución es complicada. Se consideró realizarlo con una lista o *RecyclerView*, pero dado que los elementos a mostrar son demasiado heterogéneos, no se encontró viable y se descartó. Finalmente se optó por realizar un diseño estático considerando el máximo de 4 líneas de metro, con la ayuda de 6 *guidelines* verticales. Inicialmente todos los elementos se ponen invisibles. Posteriormente, en tiempo de ejecución, una vez se conoce el número de líneas de metro que el usuario debe coger según el trayecto elegido, se pintan del color correspondiente a la línea de Metro, se colocan los iconos necesarios, y se hacen visibles únicamente aquellos elementos que realmente deben aparecer.

Y aquí aparecía un nuevo problema: el tramo final (icono de hombre caminando, flecha gris, y duración y distancia del trayecto a pie final) siempre debe aparecer, sea cual sea el número de líneas de metro. Para poder reubicarlo a la posición necesaria se modifica en el código la posición de las dos últimas guías verticales (que se utiliza para referenciar la posición horizontal de estos elementos), moviéndolas hacia la izquierda:

```
// Rearrange Layout: move the 2 last vertical guidelines to the appropriate position
```

```
// According to the number of routes made in Metro
```

```
ConstraintLayout.LayoutParams params = (ConstraintLayout.LayoutParams)
mVerticalGuidelines[4].getLayoutParams();
```

```

params.guideBegin = (int) TypedValue.applyDimension(TypedValue.COMPLEX_UNIT_DIP,
90+metroLines*100, getResources().getDisplayMetrics()); // Convert from dp's to pixels
mVerticalGuidelines[4].setLayoutParams(params);
params = (ConstraintLayout.LayoutParams) mVerticalGuidelines[5].getLayoutParams();
params.guideBegin = (int) TypedValue.applyDimension(TypedValue.COMPLEX_UNIT_DIP,
90+(metroLines+1)*100, getResources().getDisplayMetrics()); // Convert from dp's to pixels
mVerticalGuidelines[5].setLayoutParams(params);

```

En la siguiente imagen puede verse el árbol de componentes y el diseño del *layout*:

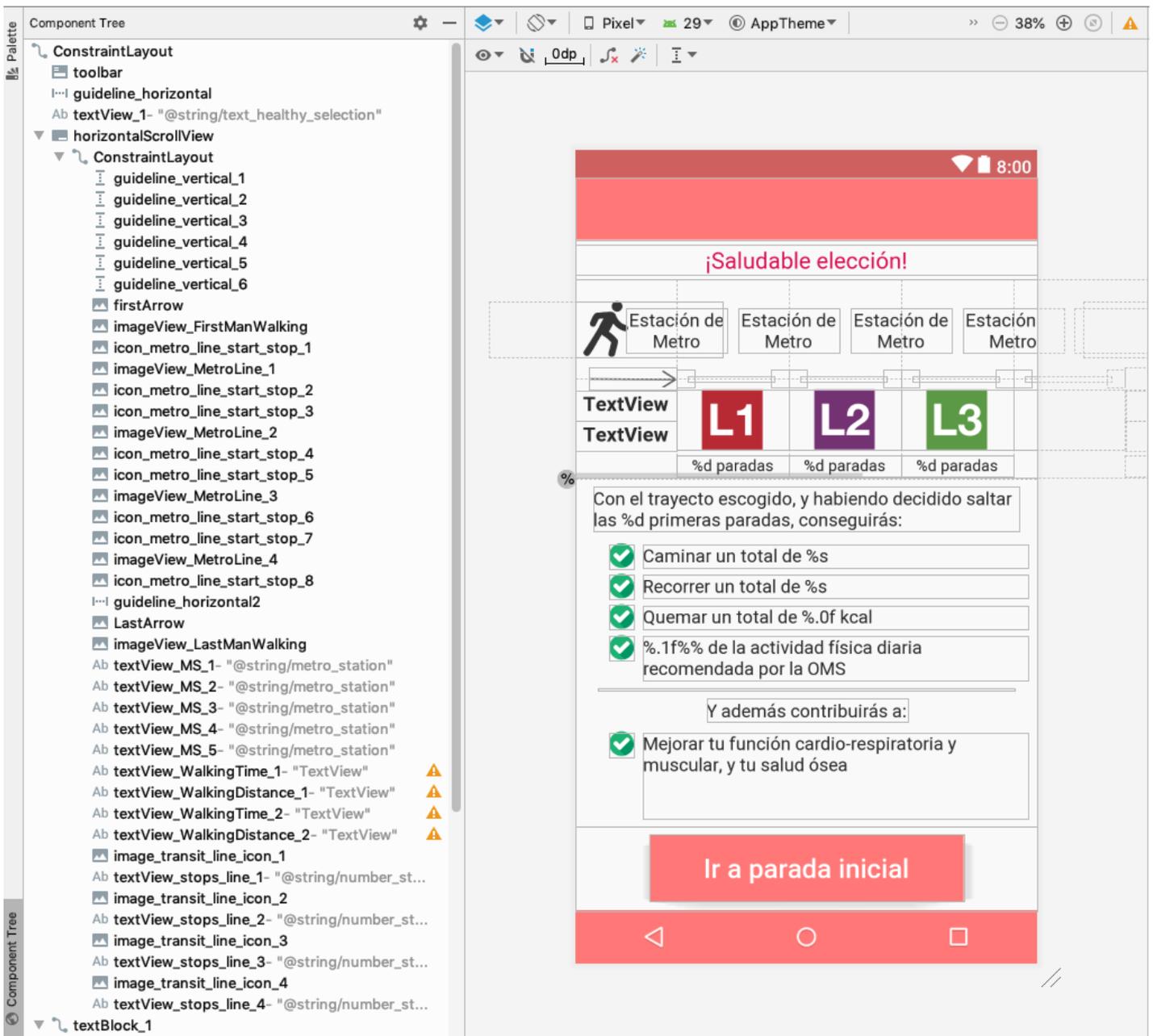


Figura 32- Árbol de componentes y diseño de *layout* de la pantalla *Iniciar trayecto*

5 Pruebas

Debido a la escasez de tiempo disponible (se analizará en el apartado siguiente) no ha podido establecerse un plan de pruebas sistemático a la finalización de la implementación de la aplicación. No se han diseñado tests unitarios ni funcionales utilizando *frameworks* específicos como *Espresso* o *Mockito*.

Únicamente se han podido realizar pruebas funcionales de manera manual, conforme se iban acabando las diferentes implementaciones. En ellas se revisaba si se cumplían los requisitos establecidos para la aplicación, y si se satisfacían los diferentes casos de uso.

Todas las pruebas realizadas hasta el momentos han sido correctas.

6 Revisión de la planificación

6.1 Punto de control PEC2

Después de haber completado la PEC2, que incluía la investigación de usuarios, el diseño conceptual, el prototipado y el diseño técnico de la aplicación, se realiza en este apartado una revisión de la planificación del proyecto. El objetivo es identificar el estado real del trabajo en relación con la proyección de tareas realizada, y establecer medidas correctoras si fuera necesario.

Por el momento no se han encontrado desviaciones significativas con respecto al calendario presentado en el Anexo A. A 30/10/2019, día de la entrega de la PEC2, se han concluido con éxito todas las actividades que se proyectaron. No obstante, es importante señalar que el tiempo requerido ha sido mayor al inicialmente previsto.

El prototipado de alto nivel ha sido la tarea que ha consumido más tiempo. Se han necesitado 24 horas para completarlo, 10 más de las previstas. Parte de este tiempo se ha recuperado de las 10 horas que se habían destinado para el diseño de la BD y de las entidades y clases. Al utilizar Firebase no ha sido necesario realizar un diagrama relacional. Por otra parte, tras consultarlo con el profesor, se decidió que no era necesario en este punto detallar el diagrama UML de las entidades y clases que se iban a utilizar en la aplicación. Al eliminar estas dos actividades se recuperó algo de tiempo. Pese a todo, se ha dedicado un total de 74 horas frente a las 60 proyectadas. Afortunadamente, en la planificación se había previsto 4 días para desviaciones, que han sido consumidos en su totalidad.

Respecto al cumplimiento del reparto de horas en días (2 horas en días laborables y 6 en sábados, domingos y festivos) debo decir que ha habido bastantes variaciones. Si bien los fines de semana se ha cumplido con lo estipulado, algunos días entre semana las obligaciones profesionales han impedido la dedicación prevista. No obstante, se ha corregido aumentando la carga otros días de la semana, de manera que el cómputo semanal se ha mantenido.

Para el control del tiempo he comenzado a utilizar la App *Productivity Challenge Timer*, que utiliza la técnica Pomodoro. No solo me ha permitido incrementar la productividad, sino que ofrece una manera sencilla de computar las horas dedicadas en cada jornada.

Con la experiencia de esta segunda entrega, se prevé que la fase de implementación demande aún un mayor número de horas adicionales. Por ello se ha tomado la siguiente decisión:

- El número de horas dedicado en días laborables de aumentará de 2 a 3.
- El número de horas dedicado los fines de semana se incrementará de 6 a 8.

6.2 Punto de control PEC3

Pese a que la dedicación a la PEC3, en número de horas, ha cumplido lo planificado (he dedicado un total de 131 h, frente a las 122 h proyectadas) no he conseguido implementar todas las funcionalidades que recogí en la fase de Diseño Centrado en el Usuario.

Del análisis de las causas se pueden establecer tres motivos principales:

1. La **escasa documentación disponible de la API de TMB y los errores que presenta**. No contar con información detallada y veraz del uso de una API hace que se vaya un poco a ciegas. Incrementa el número de horas necesarias para su comprensión y para aprender a utilizarla correctamente. Si además le añadimos que, en ocasiones, devuelve información incoherente, el escenario se complica aún más.
2. **La complejidad de diseño de algunos *layouts***. Como se indicó en el apartado 4.4, la dedicación a la implementación de las pantallas que presentan al usuario los trayectos ha sido muy elevada. Probablemente se subestimó la dificultad de este tipo de *layouts*, especialmente para alguien inexperto (los ejercicios vistos durante el Máster eran mucho más sencillos). **Probablemente haya sido lo más costoso en tiempo de toda la implementación.**
3. **Inexperiencia en el desarrollo de aplicaciones mínimamente complejas**, que lleva a plantear demasiadas funcionalidades creyendo erróneamente que podrán implementarse en el tiempo disponible. A la hora de llevarlo a la práctica, como implica la utilización de recursos que no se dominan (o que incluso se desconocen por completo) el tiempo necesario comienza a crecer exponencialmente.

6.2.1 Estado de la implementación

En este punto se han implementado todas las funcionalidades que tienen que ver con las API de transporte y localización, correspondientes a las opciones del menú **Mapa** y **Nuevo Trayecto** (que incluye a su vez las pantallas **Selección de trayecto**, **Selección de parada** e **Iniciar trayecto**). También está implementada la pantalla de **Plano de Metro de BCN** y la pantalla de **Perfil**. Con la finalización de esta última pantalla, he integrado la app con Firebase y FirebaseAuth: creación de cuenta del usuario, log-in de usuario, y la creación y modificación del perfil (que se guarda ya en *Firestore*).

Debo subsanar un error que provoca el cierre de la aplicación cuando se crea con éxito una cuenta nueva y, acto seguido, se envía al usuario a la pantalla de Mapa. No he podido averiguar por qué sucede todavía. Si se reinicia la aplicación puede continuarse con normalidad.

Queda pendiente lo relativo a guardar el último trayecto en Firebase *Firestore*, y permitir certificar al usuario que lo ha realizado: guardar el objeto *LastRoute* para el usuario, recuperarlo en la pantalla Certificar trayecto, y actualizar los registros. Debo decir que estos puntos los he llegado a realizar, pero he perdido la última versión debido a un error con Android Studio y he sido incapaz de recuperarlo, por lo que he tenido que volver a la versión anterior para esta entrega. También quedan por ser implementadas las pantallas que muestran las Estadísticas, los Logros y la Clasificación.

El grado de implementación lo situó en el 70% de las funcionalidades, que considero que representan el 85% de la complejidad.

6.2.2 Medidas a adoptar

A partir del 20 de diciembre tendré vacaciones en el trabajo, por lo que podré incrementar el número de horas dedicadas a este proyecto. Estimo que podré destinar 4 horas más por día (de lunes a viernes), lo que supondrá unas 50 horas adicionales a las 60 planificadas para la PEC4. Dedicaré estas horas extra a continuar con la implementación de la aplicación, **priorizando** las funcionalidades que faltan de la siguiente manera:

1.- Certificación del último trayecto. Como indicaba en el punto anterior, esto ya lo he conseguido implementar, pero he perdido el trabajo realizado. Debo rehacerlo y estimo que necesitaré 2 horas.

2.- Clasificación.

3.- Logros y Estadísticas.

Considero imprescindible la implementación de las dos primeras. Si no contara con tiempo suficiente para completar todas las funcionalidades, descartaría el punto 3. Con el prototipo JustinMind queda muy clara la funcionalidad, y creo que su implementación no va a aportarme nuevos conocimientos (los *layouts* son sencillos, y los datos a mostrar se obtendrán de manera directa de Firebase).

6.3 Punto de control PEC4

Con la adopción de las medidas indicadas al término de la PEC3, finalmente **he podido implementar en la app todas las funcionalidades proyectadas** inicialmente en este Trabajo Final de Máster.

El tiempo total dedicado a la implementación de la aplicación ha sido 171 horas (frente a las 122 planificadas inicialmente, un 40% más). Pese a haber tenido que incrementar notablemente mi dedicación, estoy muy satisfecho con el resultado final. Creo que la decisión ha sido acertada, puesto que si hubiese dejado funcionalidades sin desarrollar el TFM no hubiese quedado realmente completo.

7 CONCLUSIONES

7.1 Resultados obtenidos y lecciones aprendidas

A la finalización de este Trabajo Final de Máster se han logrado los objetivos planteados en las etapas iniciales:

- Se ha realizado la gestión integral de un proyecto real en todas sus etapas.
- Se ha conseguido desarrollar una aplicación Android completamente funcional.
- La aplicación cumple con todos los requisitos funcionales que se establecieron en las etapas iniciales de diseño.
- He mejorado notablemente mis competencias, especialmente las relacionadas con la integración de API's y el diseño de *layouts* complejos en aplicaciones Android.
- Me hubiera gustado haber realizado la codificación en Kotlin, que es el lenguaje de programación que marcará el futuro de Android, pero debido a lo ajustado del calendario no me ha sido posible.

Durante el diseño y, sobre todo, durante la implementación de la aplicación, me he topado con ciertas muchas dificultades que me han hecho aprender algunas lecciones importantes:

- La más valiosa, en mi caso, es que **la implementación suele requerir**, en general, **mucho más tiempo del que se proyecta inicialmente**. Esto es especialmente cierto si durante el proceso deben adquirirse nuevas habilidades o trabajar con API's que se desconocen.
- La calidad de la documentación de las APIs que se utilicen es básica y fundamental. Hasta tal punto que, en el futuro, será un criterio básico de decisión a la hora de elegir trabajar con una u otra.

Pese a que he logrado todos los objetivos que me marqué al inicio del proyecto, debo admitir que ha sido una tarea muy dura. La codificación ha sido compleja, y he tenido que aprender a utilizar muchas API's distintas: Maps, Places, Firestore, Cloud Storage (todas de Google) y, por supuesto, las API's públicas de TMB.

Quizás por ese motivo he tenido que dedicar unas **50 horas adicionales a la implementación**. Esto hizo que en la entrega de la PAC3 la aplicación no estuviera completa, y tuve que dedicar horas adicionales posteriormente para conseguir reconducir la situación y finalizar la aplicación.

Si bien considero que la metodología que he utilizado ha sido la correcta, debo reconocer que hice una **previsión demasiado optimista de la fase de implementación**. Sin duda lo tendré en cuenta en futuros proyectos.

7.2 Líneas futuras

La principal línea de trabajo pendiente de explorar y que queda para el futuro, es la utilización de una **API más global** (y deseablemente más robusta) para el cálculo de las rutas. Una candidata sería, por ejemplo, alguna de las proporcionadas por **OpenTripPlanner**, una familia de proyectos de software de código abierto que provee información sobre múltiples redes de transporte.

Esto permitiría, además, **hacer extensible la aplicación a otros medios de transporte públicos y a otras ciudades**.

Respecto a la posibilidad de saltar paradas, actualmente la aplicación solo lo permite al inicio de la primera línea de metro que forma parte del itinerario. En versiones posteriores podría flexibilizarse esta opción y permitir que el usuario cubriera caminando otras partes del itinerario (y por tanto se saltara estaciones en otros sitios: entre conexiones de líneas o al final de la última línea).

Se podrían **incluir además otros elementos de gamificación** para aumentar el *engagement* del usuario, como retos, torneos, etc. e **introducir recompensas reales** (mediante acuerdos con espórsors o administraciones).

Sería también muy conveniente **introducir notificaciones push** para espolear la motivación del usuario.

Por último, considero muy interesante (porque desconozco cómo hacerlo) investigar la posibilidad de que la propia aplicación hiciese la certificación automática de los trayectos configurados por el usuario, mediante seguimiento de su posición en tiempo real.

Lo cierto es que las posibilidades son muchas, pero no es menos cierto que el tiempo es limitado.

8 Glosario

API – *Application Programming Interface* (Interfaz de programación de aplicaciones).

APP – Vocablo inglés, adoptado en español para referirse a una aplicación software, especialmente cuando es descargada por un usuario a su dispositivo móvil.

DCU – Diseño Centrado en el Usuario.

ECST – *European Credit Transfer System* (Sistema Europeo de Transferencia de Créditos).

ENT – Enfermedades No Transmisibles.

IDE – *Integrated Development Environment* (Entorno de Desarrollo Integrado).

JSON – *JavaScript Object Notation* (Notación de Objeto de JavaScript). Es un formato de texto sencillo para el intercambio de datos.

OMS – Organización Mundial de la Salud.

SDK – *Software Development Kit* (Kit de Desarrollo de Software)

TFM – Trabajo Final de Máster.

TMB – *Transports Metropolitans de Barcelona* (Transportes Metropolitanos de Barcelona). Es la “denominación común de las empresas *Ferrocarril Metropolità de Barcelona, S.A.*, y *Transports de Barcelona, S.A.*, que gestionan la red de Metro y Autobús por cuenta del Área Metropolitana de Barcelona” [11].

UML – *Unified Modeling Language* (Lenguaje Unificado de Desarrollo). Es un lenguaje gráfico de modelado para visualizar, especificar, construir y documentar un sistema

VEMP – Vehículos Eléctricos de Movilidad Personal.

9 Bibliografía

- [1] **Organización Mundial de la Salud.** (2018, 23 de febrero). “Actividad física”. *Organización Mundial de la Salud* [artículo en línea]. [Fecha de consulta: 27 de septiembre de 2019]
<<https://www.who.int/es/news-room/fact-sheets/detail/physical-activity>>
- [2] **World Health Organization.** (2009). “Global health risks: mortality and burden of disease attributable to selected major risks”. *World Health Organization*. [Fecha de consulta: 26 de septiembre de 2019]
< https://www.who.int/healthinfo/global_burden_disease/GlobalHealthRisks_report_full.pdf>
Número ISBN 978 92 4 156387 1
- [3] **Guthold, R., Stevens, G.A., Riley, L. Bull, F.C.** (2018, 4 de septiembre). “Worldwide trends in insufficient physical activity from 2001 to 2016: a pooled analysis of 358 population-based surveys with 1.9 million participants”. *The Lancet Global Health*. [artículo en línea].
[Fecha de consulta: 26 de septiembre de 2019]
<[https://doi.org/10.1016/S2214-109X\(18\)30357-7](https://doi.org/10.1016/S2214-109X(18)30357-7) >
- [4] **Castro, C.** (2018, 5 de septiembre). “La cuarta parte de la población mundial no se mueve ni 22 minutos al día”. *El Independiente*. [artículo en línea].
[Fecha de consulta: 25 de septiembre de 2019]
<<https://www.elindependiente.com/vida-sana/2018/09/05/sedentarismo-global/>>
- [5] **Ministerio de Sanidad, Consumo y Bienestar Social, Instituto Nacional de Estadística** (2018, 26 de junio). “ENSE Encuesta Nacional de Salud España 2017”. *Ministerio de Sanidad, Consumo y Bienestar Social*. [artículo en línea]. [Fecha de consulta: 26 de septiembre de 2019]
<https://www.msbs.gob.es/estadEstudios/estadisticas/encuestaNacional/encuestaNac2017/ENSE17_pres_web.pdf>
- [6] **Organización Mundial de la Salud.** (2010). “Recomendaciones mundiales sobre actividad física para la salud”. *Organización Mundial de la Salud* [artículo en línea].
[Fecha de consulta: 27 de septiembre de 2019]
<https://apps.who.int/iris/bitstream/handle/10665/44441/9789243599977_spa.pdf?ua=1>
Número ISBN 978 92 4 359997 7
- [7] **StatCounter.** (2019). “Mobile Operating System Market Share Spain (Sept 2018 - Sept 2019)”. *StatCounter GlobalStats* [artículo en línea]. [Fecha de consulta: 28 de septiembre de 2019]
< <https://gs.statcounter.com/os-market-share/mobile/spain/#monthly-201809-201909>>
- [8] **Google Developers** (2019). “Estructura tu base de datos”. *Google* [artículo en línea].
[Fecha de consulta: 29 de octubre de 2019]
< <https://firebase.google.com/docs/database/web/structure-data?hl=es>>
- [9] **Wikipedia** (2019). “Modelo–vista–controlador”. *Wikipedia* [artículo en línea].
[Fecha de consulta: 29 de octubre de 2019]
<<https://es.wikipedia.org/wiki/Modelo–vista–controlador>>
- [10] **Soral, Rakshit** (2019). “MVC vs MVP vs MVVM for Android Application Development”. *Simform* [artículo en línea]. [Fecha de consulta: 29 de octubre de 2019]
<<https://www.simform.com/mvc-mvp-mvvm-android-app-development/>>

[11] **TMB** (2019). “Quiénes somos”. *Grupo TMB* [artículo en línea].
[Fecha de consulta: 28 de diciembre de 2019]
<<https://www.tmb.cat/es/sobre-tmb/conocenos/quien-somos>>

[12] **TMB** (2020). “Quiénes somos”. *Grupo TMB* [artículo en línea].
[Fecha de consulta: 01 de enero de 2020]
<<https://www.opentripplanner.org> >

10 Créditos

ICONO	CRÉDITOS
	Icon made by Freepik from Flaticon .
	Icon made by Freepik from Flaticon .
	Icon made by Freepik from Flaticon .
	Icon made by Freepik from Flaticon .
	Icon made by Freepik from Flaticon .
	Icon made by Freepik from Flaticon .
	Icon made by Freepik from Flaticon .
	Icon made by Freepik from Flaticon .
	Icon made by Smashicon from Flaticon .
	Icon made by Freepik from Flaticon .
	Icon made by Google from Flaticon .
	Icon made by Freepik from Flaticon .

	Icon made by Google from Flaticon .
	Icon made by Freepik from Flaticon .
	Icon made by Freepik from Flaticon .
	Icon made by Smashicon from Flaticon .
	Icon made by smalllikeart from Flaticon .
	Icon made by Freepik from Flaticon .
	Icon made by Freepik from Flaticon .
	Icon made by Freepik from Flaticon .
	Icon made by Freepik from Flaticon .
	Icon made by pongsakornRed from Flaticon .
	Icon made by monkik from Flaticon .
	Icon made by Flat Icon from Flaticon .
	Icon made by Flat Icon from Flaticon .

	Icon made by pongsakornRed from Flaticon .
	Icon made by Flat Icon from Flaticon .
	Icons made by monkik from Flaticon .
	Icono hecho por Freepik de Flaticon
	Icono hecho por Freepik de Flaticon
	Icono hecho por Skyclick de Flaticon
	Icon made by Roundicons from Flaticon .
	Icon made by Smashicons from Flaticon .

		L1	L2	L3	L4	L5	L6	L7
L8	L9 Nord	L9 Sud	L10 Nord	L10 Sud	L11	L12		R1
R2	R2 Nord	R2 Sud	R3	R4	R5	R6	R7	R50
R60	R	S1	S2	S3	S4	S5	S6	S7
S8	S9	T1	T2	T3	T4	T5	T6	

Todos los iconos de líneas de Metro, FFGG, Funicular y Tram son propiedad de sus respectivas compañías/entes públicos. Se utilizan en este Trabajo Final de Máster con fines puramente educativos.

Anexo A: Diagrama de Gantt

Planificación del Proyecto

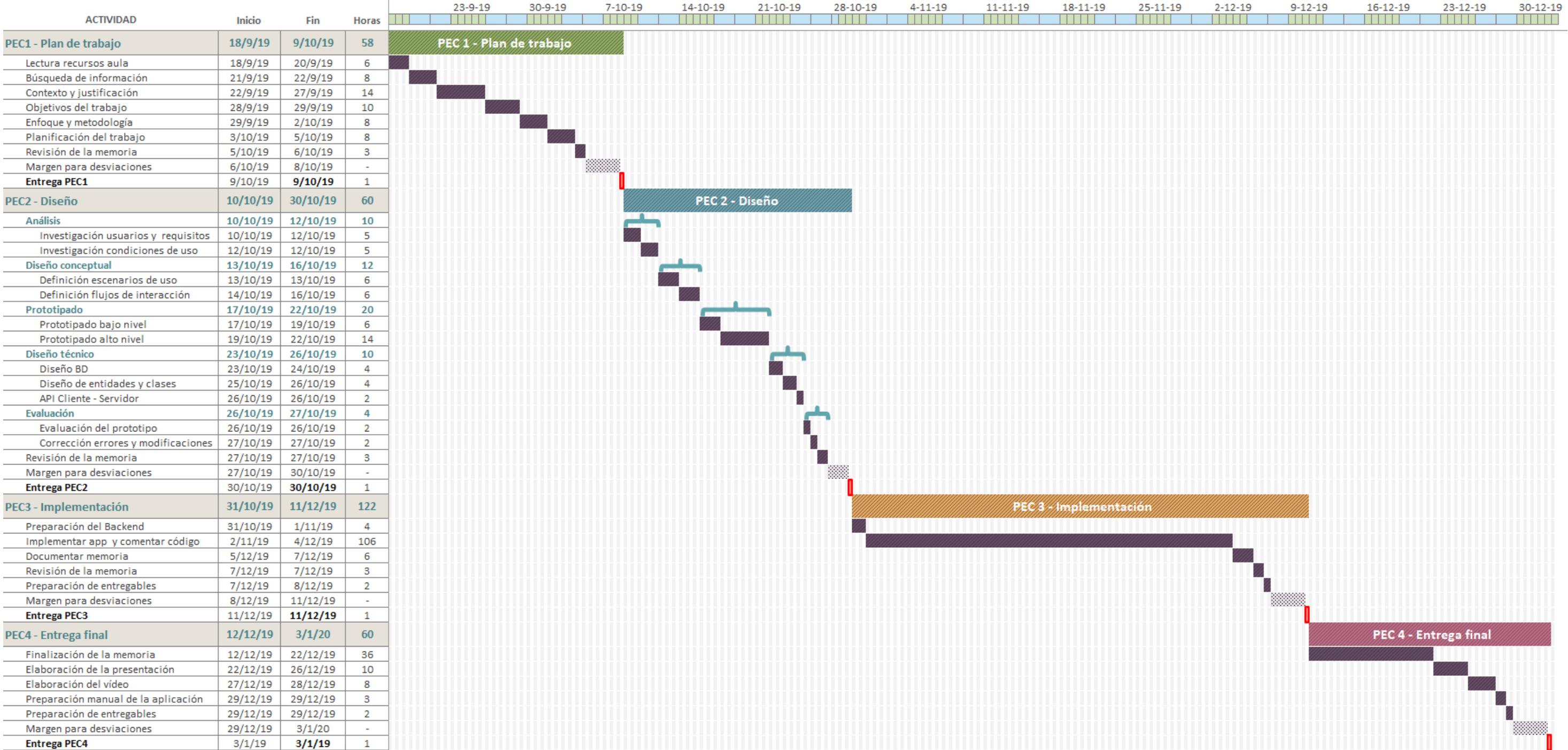


Figura 33 - Diagrama de Gantt de la planificación del TFM