



Universidad de Valladolid

Escuela de Ingeniería Informático

Trabajo de final de Grado

Grado en Ingeniería Informática

Mención en Ingeniería del Software

TURis-Route

Gestión de rutas de turismo

Autor:

Raúl Jiménez Bores

Tutora:

Margarita Gonzalo Tasis

Resumen

Existe una gran variedad de software destinado a ayudar y a hacer más amena la búsqueda de destinos para viajeros y turistas. Este software disponible en internet posee algunas limitaciones y funcionalidades de pago que produce el rechazo del usuario y que intente buscar las funcionalidades que desea en otro lugar.

Por estos motivos, la aplicación tratada en este proyecto se centrará en aprovechar una oportunidad de negocio encontrada en la mayoría de este tipo de aplicaciones, la capacidad de buscar y sugerir lugares de visita compaginables con la ruta al destino original del usuario. Además, será gratuita y poseerá disponibilidad en varios idiomas para llegar al máximo número de usuarios posibles.

Abstract

There is a wide variety of software designed to help and make the search for destinations more enjoyable for travellers and tourists. This software available on the internet has some limitations and paid functionalities that cause the user to reject it and try to find the functionalities he/she wants elsewhere.

For these reasons, the application dealt with in this project will focus on taking advantage of a business opportunity found in most of these types of applications, the ability to search and suggest places to visit that are compatible with the route to the user's original destination. In addition, it will be free and available in several languages to reach as many users as possible.

Agradecimientos

En primer lugar, deseo agradecer a mi tutora su implicación y dedicación en el desarrollo del proyecto, siempre dispuesta a solucionar mis dudas e inquietudes.

En segundo lugar, a mi familia, amigos y pareja, por haber estado acompañándome durante estos últimos meses de la carrera, animándome y ayudando en cualquier cosa que pudiese necesitar.

Por último, pero no menos importante, a mis compañeros de trabajo, por su interés y opiniones sobre el progreso del proyecto.

Muchas gracias.

Índice de contenidos

PARTE I: INTRODUCCIÓN.....	14
Capítulo 1 Introducción.....	15
1.1 Contexto	15
1.2 Motivación.....	15
1.3 Objetivos	15
1.4 Análisis aplicaciones similares.....	16
1.4.1 Google Trips	17
1.4.2 TripIt.....	18
1.4.3 Triposo.....	20
1.4.4 GPS Ruta Planificador.....	22
1.4.5 Conclusiones	23
PARTE II: TURIS-ROUTE	24
Capítulo 1 Visión global del proyecto.....	25
1.1 Tabla de definiciones	25
1.2 Metodología del proyecto.....	26
1.3 Restricciones del proyecto	28
Capítulo 2 Plan de trabajo.....	29
2.1 Organización del equipo	29
2.2 Plan de trabajo ideal.....	29
2.3 Gestión de costes	31
Capítulo 3 Riesgos.....	33
3.1 Introducción.....	33
3.2 Análisis y plan de gestión de riesgos	34
3.3 Desviaciones en la planificación original	40
Capítulo 4 Análisis	43
4.1 Requisitos	43
4.1.1 Requisitos Funcionales	43
4.1.2 Requisitos de información	47
4.1.3 Requisitos no funcionales	47

4.2	Cambios en los requisitos.....	49
4.3	Casos de uso	50
4.3.1	Especificación de casos de uso	51
4.4	Modelo de dominio.....	59
4.5	Diagramas de secuencia	61
4.5.1	CU Iniciar Sesión	61
4.5.2	CU Registro	62
4.5.3	CU Baja del sistema	62
4.5.4	CU Cerrar sesión	63
4.5.5	CU Cambio de contraseña	63
4.5.6	CU Crear ruta por sugerencia	64
4.5.7	CU Cambio de nombre.....	64
4.5.8	CU Ver Historial	65
4.5.9	CU Ruta por búsqueda.....	65
4.5.10	Diagrama de secuencia de algoritmo de sitios cercanos	66
Capítulo 5 Diseño		67
5.1	Arquitectura.....	67
5.2	Patrones Utilizados	70
5.2.1	M.V.P. (Modelo Vista Presentador).....	70
5.2.2	D.A.O. (Data Access Object).....	71
5.3	Esquema base de datos.....	71
5.4	Diagrama de despliegue	73
Capítulo 6 Implementación		75
6.1	Algoritmos propios	75
6.1.1	Algoritmo para búsqueda de sitios de interés cercanos.....	75
6.2	Dependencias.....	77
6.3	Herramientas utilizadas en el desarrollo del proyecto	77
6.3.1	Firebase.....	77
6.3.2	Ionic.....	78
6.3.3	Mapbox.....	79
6.3.4	Angular	80

Capítulo 7 Pruebas.....	81
7.1 Casos de Prueba.....	81
7.2 Criterios de evaluación.....	85
7.3 Resultados en Android.....	86
7.4 Resultados en navegador web	87
WEBGRAFÍA	91
BIBLIOGRAFÍA	93
ANEXOS 94	
Anexo 1	95
Manual de Instalación	95
Configuración de Android.....	95
Anexo 2	97
Manual de usuario.....	97

Índice de Figuras

Figura 1.4.1—1 Menú principal de la aplicación Google Trips.....	17
Figura 1.4.1—2 Mapa con puntos de interés de la aplicación Google trips	17
Figura 1.4.2—2 Sitios de interés cercanos en la aplicación TripIt.....	19
Figura 1.4.2—1 Creación de ruta en la aplicación TripIt	19
Figura 1.4.3—1 Destinos populares, mapa con puntos de interés y detalles de destino de la aplicación Triposo.....	21
Figura 1.4.4—1 Creación de ruta con sugerencias y almacenamiento de rutas Ruta planificador app	22
Figura 1.2—1 Fases de la metodología UPEdu.....	26
Figura 3.2—1 Criterios impacto/probabilidad	34
Figura 4.2—1 Diagrama de casos de uso de la aplicación.....	51
Figura 4.3—1 Modelo de dominio de la aplicación	60
Figura 4.5.1—1 Diagrama de secuencia del caso de uso iniciar sesión.....	61
Figura 4.5.2—1 Diagrama de secuencia del caso de uso registro	62
Figura 4.5.3—1 Diagrama de secuencia del caso de uso baja del sistema	62
Figura 4.5.4—1 Diagrama de secuencia del caso de uso cerrar sesión	63
Figura 4.5.5—1 Diagrama de secuencia del caso de uso cambio de contraseña.....	63
Figura 4.5.6—1 Diagrama de secuencia del caso de uso crear ruta por sugerencia	64
Figura 4.5.7—1 Diagrama de secuencia del caso de uso cambiar nombre.....	64
Figura 4.5.8—1 Diagrama de secuencia del caso de uso ver historial	65
Figura 4.5.9—1 Diagrama de secuencia del caso de uso ruta por búsqueda	65
Figura 4.5.10—1 Diagrama de la búsqueda de lugares cercanos	66
Figura 4.5.10—2 Diagrama de clases de profile	69
Figura 4.5.10—3 Diagrama de clases de home	69
Figura 5.2.1—1 Patrón mvc vs mvp.....	70
Figura 5.2.1—2 Ejemplo patrón mvp de la aplicación	70
Figura 5.2.1—3 Ejemplo patrón mvp de la aplicación	70
Figura 5.3—1 Esquema no relacional basado en elementos y colecciones en lo relativo a usuarios	72

Figura 5.3—2 Esquema no relacional basado en elementos y colecciones en lo relativo a puntos de interés recomendados por la aplicación	72
Figura 6.3.1—1 Logotipo de Firebase de Google	77
Figura 6.3.2—1 Logotipo del framework Ionic	78
Figura 6.3.3—1 Logotipo de Mapbox	79
Figura 6.3.4—1 Logotipo del framework Angular.....	80
Figura Anexo II—2 Captura de pantalla de la aplicación del inicio de sesión y registro	97
Figura Anexo II—3 Captura de pantalla de la aplicación sobre la creación de una ruta por búsqueda.....	98
Figura Anexo II—4 Captura de pantalla de la aplicación sobre crear una ruta por sugerencias de destinos de la aplicación	98
Figura Anexo II—5 Captura de pantalla de la aplicación sobre el perfil.....	98
Figura Anexo II—6 Captura de pantalla de la aplicación sobre el historial de rutas	98
Figura Anexo II—7 Captura de pantalla de la aplicación sobre cambio de nombre	98
Figura Anexo II—8 Captura de pantalla de la aplicación sobre cambio de contraseña .	98
Figura Anexo II—9 Captura de pantalla de la aplicación sobre ajustes	98

Índice de Tablas

Tabla 1.4.5—1 Definiciones útiles para el proyecto	26
Tabla 3.2—1 Riesgo: incumplimiento del plan.....	35
Tabla 3.2—2 Riesgo: Ausencia de familiaridad con el entorno.....	35
Tabla 3.2—3 Riesgo: escasez de detalle en los requisitos	36
Tabla 3.2—4 Riesgo: problemas de salud del equipo.....	36
Tabla 3.2—5 Riesgo: dificultades con el trabajo en equipo	37
Tabla 3.2—6 Riesgo: malfuncionamiento de los servidores.....	38
Tabla 3.2—7 Riesgo: problemas de usabilidad	38
Tabla 3.2—8 Riesgo: cambios en los requisitos.....	39
Tabla 3.2—9 Riesgo: rechazo del software.....	39
Tabla 3.2—10 Riesgo: pérdida de datos	40
Tabla 3.3—11 Tabla planificación real.....	42
Tabla 4.1.1—1 Requisito Funcional 1	43
Tabla 4.1.1—2 Requisito funcional 2	43
Tabla 4.1.1—3 Requisito Funcional 3	43
Tabla 4.1.1—4 Requisito Funcional 4	44
Tabla 4.1.1—5 Requisito Funcional 5	44
Tabla 4.1.1—6 Requisito Funcional 6	44
Tabla 4.1.1—7 Requisito Funcional 7	44
Tabla 4.1.1—8 Requisito Funcional 8	45
Tabla 4.1.1—9 Requisito Funcional 9	45
Tabla 4.1.1—10 Requisito Funcional 10	45
Tabla 4.1.1—11 Requisito Funcional 11	45
Tabla 4.1.1—12 Requisito Funcional 12	46
Tabla 4.1.1—13 Requisito Funcional 13	46
Tabla 4.1.1—14 Requisito Funcional 14	46
Tabla 4.1.1—15 Requisito Funcional 15	46
Tabla 4.1.2—1 Requisito de información 1	47
Tabla 4.1.2—2 Requisito de información 2	47

Tabla 4.1.3—1 Requisito No Funcional 1.....	47
Tabla 4.1.3—2 Requisito No Funcional 2.....	47
Tabla 4.1.3—3 Requisito No Funcional 3.....	48
Tabla 4.1.3—4 Requisito No Funcional 4.....	48
Tabla 4.1.3—5 Requisito No Funcional 5.....	48
Tabla 4.1.3—6 Requisito No Funcional 6.....	48
Tabla 4.1.3—7 Requisito No Funcional 7.....	49
Tabla 4.1.3—8 Requisito No Funcional 8.....	49
Tabla 4.3.1—1 Caso de uso iniciar sesión	52
Tabla 4.3.1—2 Caso de uso Registro.....	53
Tabla 4.3.1—3 Caso de uso Cerrar sesión	53
Tabla 4.3.1—4 Caso de uso baja del sistema	54
Tabla 4.3.1—5 Caso de uso ver historial de lugares visitados	55
Tabla 4.3.1—6 Caso de uso crear ruta por búsqueda	56
Tabla 4.3.1—7 Caso de uso buscar lugares en el mapa	56
Tabla 4.3.1—8 Caso de uso crear ruta por sugerencia de la app.....	57
Tabla 4.3.1—9 Caso de uso cambiar nombre de usuario.....	58
Tabla 4.3.1—10 Caso de uso cambiar contraseña	59
Tabla 7.1—1 Caso de prueba registro de usuario.....	81
Tabla 7.1—2 Caso de prueba iniciar sesión	82
Tabla 7.1—3 Caso de prueba cerrar sesión	82
Tabla 7.1—4 Caso de prueba baja del servicio	82
Tabla 7.1—5 Caso de prueba crear ruta mediante búsqueda	83
Tabla 7.1—6 Caso de prueba crear ruta mediante sugerencias.....	83
Tabla 7.1—7 Caso de prueba ver historial de lugares visitados.....	84
Tabla 7.1—8 Caso de prueba conocer la ubicación del usuario.....	84
Tabla 7.1—9 Caso de prueba controles de mapa	84
Tabla 7.1—10 Caso de prueba cambio de nombre.....	85
Tabla 7.1—11 Caso de prueba cambio de contraseña del usuario	85

PARTE I: INTRODUCCIÓN

Capítulo 1 Introducción

1.1 Contexto

El uso de dispositivos móviles también conocidos como smartphones es visto como algo prácticamente irremplazable en la sociedad de hoy en día. Se hace difícil pensar en contactar con alguien o realizar una gran variedad de cosas cotidianas sin tener en mente un teléfono móvil. Esto es algo lógico, ya que desde que llegó a nuestras vidas ha ido obteniendo importancia y mejoras que facilitan muchos ámbitos de nuestra vida, así como contactar con personas, jugar, encontrar ubicaciones, agenda, calculadora y navegador en internet entre innumerables otras.

Este proyecto plantea sumergirse en este mar de funcionalidades para incrementar la capacidad de un usuario con un smartphone. La aplicación más utilizada hoy en día para gestionar mapas y elaborar rutas es Google Maps, pero a pesar de lo eficiente y completa que es, existen todavía muchas mejoras y posibilidades ya que al ser una aplicación tan general no se centra en ningún foco concreto. Este proyecto considerado como Trabajo Fin de Grado (TFG) consistirá en el desarrollo de una aplicación móvil que guíe al usuario al lugar de interés que desea visitar sugiriendo visitas compatibles con su ruta original y sugiriendo destinos por la ciudad.

1.2 Motivación

El mundo de la tecnología móvil es inmenso y en parte gracias a esta inmensidad, todavía existen una gran cantidad de ideas que desarrollar o funcionalidades que perfeccionar. Teniendo esto en cuenta, este proyecto se centra en ofrecer al usuario una funcionalidad que como se comentaba, ayudará al viajero (turista) a perderse por los lugares que visita ofreciendo sugerencias compatibles con el destino original en la que se podrán observar lugares de interés que quizás no sean tan importantes como el destino final, pero enriquecerán la ruta y la harán más amena.

En resumen, se trata de una aplicación que facilitará y mejorará la experiencia de aquellos turistas cuyos motivos son disfrutar de unas vacaciones llenas de cultura y visualmente atractivas.

1.3 Objetivos

El propósito de este trabajo final de grado es realizar una aplicación multiplataforma que planee rutas turísticas yendo más allá del destino que quiere visitar el usuario ofreciéndole lugares de interés cercanos que le puedan interesar. Además, la aplicación ofrecerá al usuario algunos de los lugares turísticos más importantes de la ciudad para su visita. Dado que es una aplicación destinada a turistas, es vital que tenga disponibles varios idiomas.

El resto de los objetivos vendrá detallado en la sección de análisis en el presente documento.

Teniendo esto en cuenta podemos dividir los objetivos en dos, aquellos que nos aportarán conocimientos para realizar y detallar el proyecto, y por otro lado los relacionados con las funcionalidades de la app:

- Objetivos de proyecto:
 - Estudio completo de la documentación de Mapbox para el correcto uso e implementación de mapas y servicios que ofrece la plataforma en nuestra app.
 - Estudio completo de frameworks y entornos que puedan ayudar a conseguir el desarrollo de la app.
 - Generación de productos derivados de la aplicación de la ingeniería del software como diagramas y modelos, así como una memoria completa y detallada que los explique.
 - Creación de la aplicación funcional.
- Objetivos de la aplicación:
 - Ofrecer sugerencias sobre destinos turísticos (edificios, museos, iglesias, etc.) al usuario.
 - Realizar rutas a dichos destinos y a cualquier otro que desee el usuario.
 - Mostrar lugares cercanos a la ruta o destino del usuario que puedan serle de interés.
 - Disponibilidad de varios idiomas.

1.4 Análisis aplicaciones similares

A continuación, evaluaremos y analizaremos una serie de aplicaciones y servicios web similares al proyecto que vamos a desarrollar. Esta es una práctica interesante para incluir buenas ideas que ya hayan tenido éxito en otras apps y descartar otras opciones que no hayan sido bien recibidas por los usuarios.

1.4.1 Google Trips

Se trata de una aplicación que simula un guía turístico portátil. Esta aplicación tiene las ventajas de poder utilizarse offline y por eso ha tenido una buena acogida. La funcionalidad consiste en crear un itinerario a tu gusto, aunque también existe la modalidad de que la aplicación lo cree automáticamente por ti.

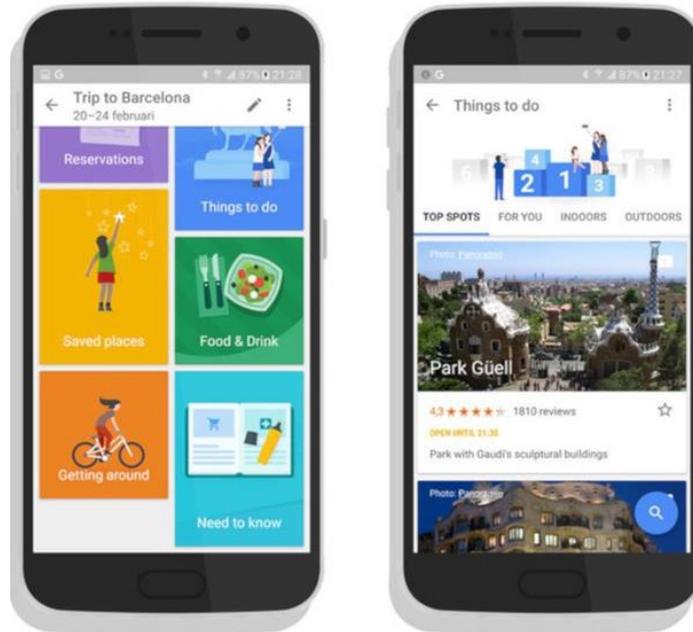


Figura 1.4.1—1 Menú principal de la aplicación Google Trips

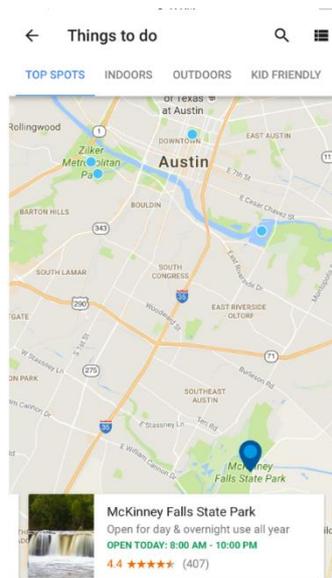


Figura 1.4.1—2 Mapa con puntos de interés de la aplicación Google Trips

Ventajas:

- Funcionalidad offline: al no depender de internet para funcionar, evita los costes del roaming en países extranjeros y la dependencia de cobertura en la zona.
- Sincronización con tu cuenta de Gmail para facilitar datos útiles para tus viajes automáticamente (horarios de vuelos, fechas de salida y entrada en hoteles...).
- Elaboración del itinerario personalizado
- Elaboración de itinerario automático por Google.

Desventajas:

- No dispone de seguimiento de rutas en tiempo real.
- Se centra en los lugares de interés más importantes de cada ciudad.
- Grandes descargas para poder facilitar el funcionamiento offline.
- Sincronización obligatoria con la cuenta de Google.
- Acceso a datos privados como el correo para escanear fechas de viaje vuelos e información.

Conclusión: con el análisis de esta aplicación podemos apreciar que el hecho de poder utilizarla offline es un gran punto, a pesar de las descargas que esto supone. El acceso excesivo a datos privados no es algo que a los usuarios les atraiga, por lo que en nuestro proyecto evitaremos dentro de lo posible tocar datos sensibles. Cabe destacar que esta aplicación fue retirada del mercado por Google, ya que las funcionalidades que poseía eran muy semejantes a las de Google Maps y la popularidad de esta hizo migrar a los usuarios de Google trips.

1.4.2 TripIt

Es una aplicación que se centra en el manejo y gestión de itinerarios, haciendo la vida un poco más fácil al usuario, reuniendo toda la información sobre fechas y horas de vuelos, coches de alquiler, horas de entrada al hotel etc.

Puntuación play store: 4.4/5

Comentarios destacados: Cuando se agregan múltiples destinos al itinerario, los tiempos de carga y espera son muy elevados. – La eficiencia de la aplicación ha de ser un factor tenido muy en cuenta.

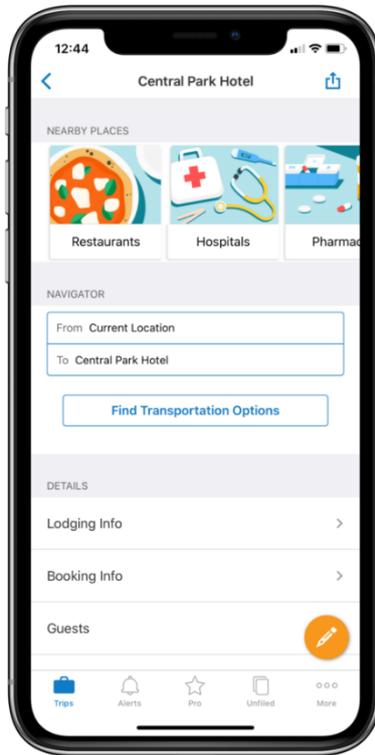


Figura 1.4.2—2 Creación de ruta en la aplicación TripIt

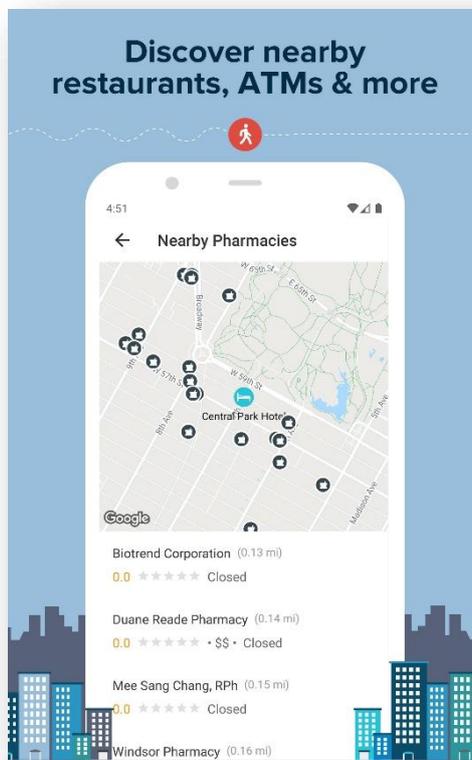


Figura 1.4.2—1 Sitios de interés cercanos en la aplicación TripIt

Ventajas:

- Dispone de toda la información referente a tu viaje.
- Uso intuitivo.
- Destaca lugares cercanos que puedan ser útiles en tus viajes.
- Permite la opción de escoger que correos deseas enviarle a la app para que extraiga la información sobre el viaje.

Desventajas:

- No crea un itinerario para rutas turísticas en la ciudad.
- Requiere información sobre el viaje que extrae de tus correos electrónicos.
- Optimizada en inglés

Conclusión: Puede parecer que no tiene mucha semejanza con este proyecto, pero no es del todo así, dispone de un mapa en el que se pueden observar los sitios de interés más cercanos, además destaca por su facilidad de uso e intuitividad. Este último factor se le considera muy importante ya que uno de los objetivos del proyecto es desarrollar una app user friendly, cuya complejidad sea fácilmente abordada por el usuario.

Una de las grandes desventajas de esta app es que esta optimizada para un solo idioma, para poder aumentar el abanico de usuarios que utilicen la aplicación, sobre todo teniendo en cuenta que se trata de viajeros, el sistema debería estar disponible y optimizado para varios idiomas.

1.4.3 Triposo

Esta es la aplicación que posiblemente más se asemeja a nuestros requisitos, se trata de un servicio que ofrece, según la localización sitios donde comer, alojarse y visitas sugeridas según la ubicación del usuario. Cuenta con las categorías de lugares de interés, comer fuera, vida nocturna y compras entre otras.

Puntuación play store: 4.3/5

Comentarios destacados: ¿Alguien sabe cómo cambiar el idioma de la aplicación?
– Resulta imprescindible disponer en una app de viajes de varios idiomas, además esta opción tiene que estar a la vista.

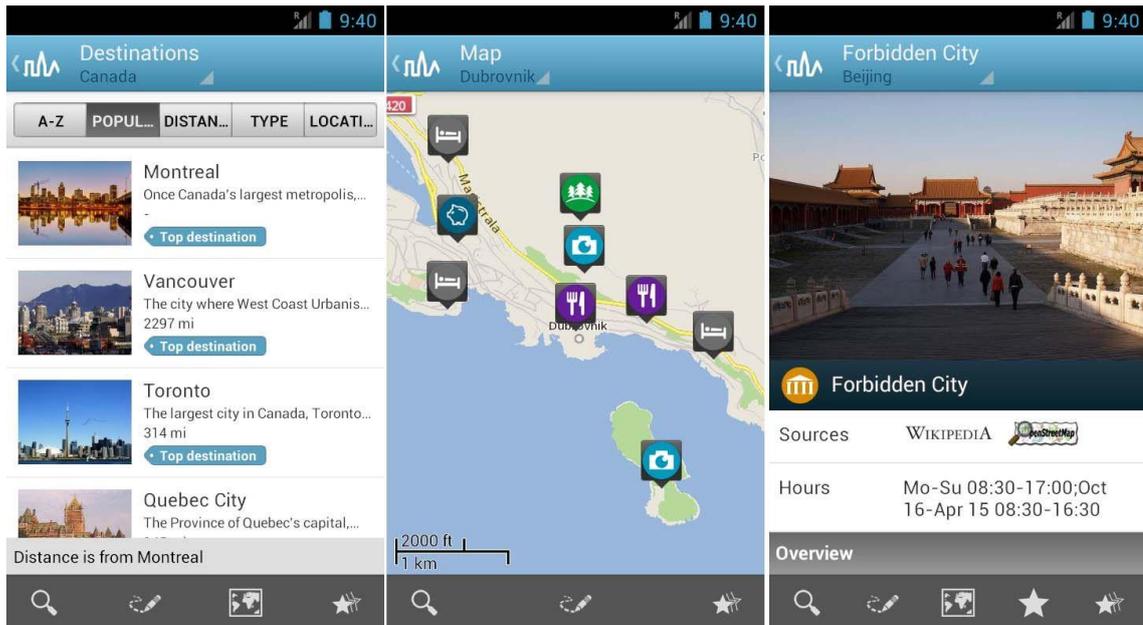


Figura 1.4.3—1 Destinos populares, mapa con puntos de interés y detalles de destino de la aplicación Triposo

Ventajas:

- Funcionalidad offline.
- No necesita información privada aparte de la ubicación.
- Intuitiva y fácil de usar.
- Calcula rutas al sitio escogido
- Amplia comunidad de usuarios

Desventajas:

- Solo está disponible en inglés.
- No es capaz de realizar rutas con varios lugares de interés en tu itinerario.

Conclusión: destaca por su facilidad de uso, la poca información personal que necesita y por su funcionamiento offline. Estos son importantes aspectos para valorar en nuestra app, teniendo en cuenta que en cuanto menos uso de información privada hagamos, la satisfacción del usuario será mayor siempre y cuando la aplicación funcione y cumpla unos requisitos de usabilidad.

El hecho de que el idioma exclusivo sea el inglés puede frenar a muchos usuarios de utilizarla. Los usuarios van a ser viajeros de un gran número de países, y a pesar de que el inglés sea el idioma más hablado del mundo, la satisfacción del usuario se incrementaría si incluyésemos español y francés adicionalmente.

1.4.4 GPS Ruta Planificador

Esta aplicación se centra en planificación de rutas pudiendo añadir múltiples paradas en la ruta. Además, cuenta con un sistema de localización de sitios cercanos a la ubicación del usuario.

El estudio de esta aplicación se centrará en analizar como realizan la localización de sitios cercanos y que opiniones tienen los clientes sobre ello.

Puntuación play store: 4.4/5

Comentario destacado: La aplicación funciona muy bien pero no guarda más de 10 rutas, para poder guardar más es necesario pagar una suscripción premium y las actualizaciones y mejoras que meten con la suscripción no merecen la pena.

Ventajas:

- Rutas con varias paradas
- Permite guardar rutas “prediseñadas”
- Permite localizar sitios de interés cerca de la ubicación del usuario.
- Versión en castellano.

Desventajas:

- Límite de rutas que se pueden guardar.
- Posee funcionalidades de pago.
- Mala optimización en castellano

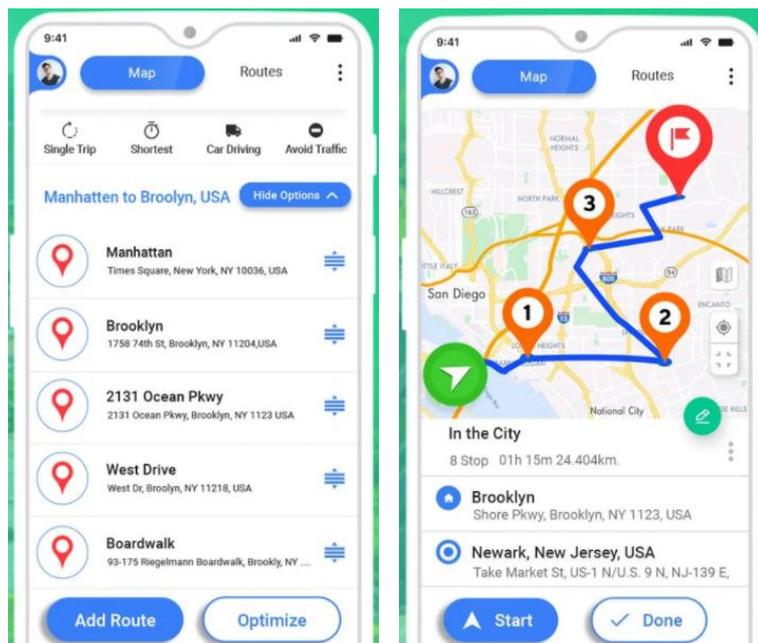


Figura 1.4.4—1 Creación de ruta con sugerencias y almacenamiento de rutas Ruta planificador app

Conclusión: es muy útil para el modelo de negocio del proyecto este sistema de localización de sitios cercanos que se puede personalizar para que se adapte al diseño de la aplicación. Muchos de los comentarios de la aplicación se quejan de las funcionalidades de pago, esto es un indicativo de que hay que intentar en la medida de lo posible que la aplicación sea completamente gratuita, sin hacer distinciones entre usuarios premium y gratuitos. Por último, la optimización en varios idiomas debe de ser un punto importante del proyecto.

1.4.5 Conclusiones

Hoy en día existen múltiples aplicaciones y sistemas web que permiten a un usuario planear un itinerario ideal para sus viajes turísticos, teniendo incluso facilidades para encontrar medios de transporte adecuados. Se puede observar que, aunque la funcionalidad offline acarrea grandes descargas, es algo muy apreciado por los usuarios y se debe tener en cuenta, de igual manera los idiomas han de ser valorados para disponer de la aplicación en al menos dos idiomas.

Finalmente observamos que ninguna de estas aplicaciones está destinada a sugerir lugares cercanos a la ruta original, para que este pueda visitarlos libremente durante su paseo. Nuestra idea de negocio debe centrarse en ese punto, sugiriendo lugares como destinos principales al usuario y por otro lado ofreciendo lugares de interés cultural cercanos a la ruta deseada.

PARTE II: TURIS-ROUTE

Capítulo 1 Visión global del proyecto

1.1 Tabla de definiciones

Término	Significado
UPEDU	Unified Process for EDUcation, es un proceso de desarrollo software característico especializado para la educación, se compone de una serie de buenas prácticas que permiten agilizar y esclarecer el desarrollo software. Es una variación del proceso unificado.
RUP	Rational Unified Process, es un proceso de desarrollo de software, desarrollado por Rational Software, que actualmente pertenece a IBM. Constituye una metodología estándar utilizada para análisis, diseño, implementación y documentación de software.
IBM	International Business Machines corporation, es una empresa multinacional estadounidense que fabrica hardware y software para computadoras y ofrece otros servicios en una amplia gama de áreas de la informática.
TFG	Trabajo Final de Grado
POI	Point of Interés, en el documento se utiliza para referirse a lugares con relevancia cultural como iglesias, museos, etc.
SCSS	Sass Cascading Style Sheets, se corresponde con un metalenguaje anidado utilizado para definir y personalizar estilos.
HTML	HyperText Markup Language, es un lenguaje de marcas que se utiliza para el desarrollo de páginas web.
TS	TypeScript, es un lenguaje de programación de tipado estático y superset de JavaScript, lo que significa que puede ejecutar programas de la tecnología JavaScript.

CRUD	Create Read Update Delete, estas siglas se utilizan para referirse a un conjunto de operaciones que realizan escritura borrado modificación y lectura de elementos relacionados con bases de datos.
MVC	Modelo Vista Controlador, patrón de diseño software en el que se separa los datos y las vistas de la lógica de negocio de las operaciones que se realizan sobre ellas y los datos
API	Application Programming Interfaces, conjunto de definiciones y protocolos que se utilizan para desarrollar e integrar software, permitiendo la comunicación entre aplicaciones software siguiendo un conjunto de reglas.
IOS	Sistema operativo de la empresa Apple

Tabla 1.4.5—1 Definiciones útiles para el proyecto

1.2 Metodología del proyecto

Para el desarrollo del proyecto se ha adoptado la metodología UPEDU. Es una variación del RUP (IBM Rational Unified process) y hereda de esta muchas propiedades características. El proceso de desarrollo según dicha metodología consiste en distintas fases: inicio, elaboración, construcción y transición. Se trata de una metodología en espiral e iterativa donde cada una de las fases mencionadas anteriormente se compone de distintas iteraciones. El proyecto se va refinando hasta que se consigue la versión final del producto. Cada iteración del proceso de desarrollo está compuesta por uno o más hitos.

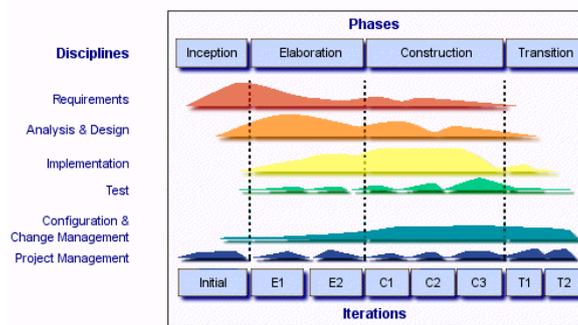


Figura 1.2—1 Fases de la metodología UPEDU

En caso de que surjan nuevos requerimientos o requisitos se inicia una nueva iteración en el proceso hasta conseguir el entregable. A continuación, se detalla el proceso de desarrollo con la metodología UPEDU y sus fases:

Inicio: la fase de inicio consiste en definir que funcionalidades se van a realizar en esta iteración, analizar los riesgos de dichas fases, presupuestos y requisitos. Esta fase de inicio tiene como objetivo asegurar la viabilidad de dicha iteración evaluando sus costes y riesgos, pudiendo llegar a dejar funcionalidades sin realizar debido a que no presentan una viabilidad convincente.

Dentro de las tareas indicadas para esta fase se encuentran:

- Determinar requisitos restricciones y alcance del proyecto.
- Capturar casos de uso críticos.
- Análisis y gestión de riesgos del proyecto.
- Determinar la arquitectura a seguir.
- Costes y viabilidad.
- Preparación del entorno de desarrollo.

Elaboración: Se trata de la fase previa a la construcción o desarrollo real del software, en esta se crearán y utilizarán distintos modelos para dar estabilidad al proyecto teniendo en cuenta la arquitectura elegida en la fase anterior.

Dentro de las tareas indicadas para esta fase se encuentran:

- Esclarecer los requisitos y casos de uso más críticos para aumentar su comprensión y evitar riesgos.
- Establecer los planes de iteración detalladamente para la fase de construcción.
- Creación de modelos para validar la arquitectura.
- Evaluación de costes y calendarización más ajustada de la fase de construcción teniendo en cuenta nueva información.

Construcción: En esta fase, teniendo en cuenta los modelos y las pautas a seguir elaboradas en las fases anteriores se procederá a completar el desarrollo del software, es decir, se obtendrán versiones parcialmente funcionales del producto. Cabe destacar que al ser un desarrollo iterativo y en espiral en cada fase de construcción se irá obteniendo una versión superior y más completa del producto final.

Dentro de las tareas indicadas para esta fase se encuentran:

- Obtención y elaboración de las versiones funcionales del producto.
- Refinación del desarrollo y diseño del producto parcial, este debe de estar preparado para la fase de transición con el usuario.
- Control de calidad del producto.

- Evaluación del cumplimiento de requisitos.

Transición: Esta fase consiste en las pruebas por parte del usuario del software elaborado en la fase de construcción, siendo posible la realización de pequeñas correcciones en este. El objetivo de esta fase es garantizar que el producto esté completamente disponible para entregar.

Dentro de las tareas indicadas para esta fase se encuentran:

- Talleres para mostrar el software al usuario.
- Pruebas con usuarios.
- Correcciones del software basadas en las experiencias de los usuarios.
- Lanzamiento del producto.

Cada una de estas fases pueden ser elaboradas en varias iteraciones según la metodología UPEdu. La fase de inicio de nuestro proyecto constará de 1 fase, elaboración y construcción de dos fases, y finalmente, transición de una fase. Estas fases son variables según se evalúe el seguimiento del proyecto, aunque en un principio son las que se van a elaborar.

1.3 Restricciones del proyecto

A partir de un análisis previo del alcance y objetivos del proyecto, se extraen las siguientes restricciones:

- **Tiempo:** Se dispone para la realización del proyecto un total de 12 créditos, lo que se traduce en 300 horas de trabajo.
- **Asignaturas pendientes:** El alumno posee 3 asignaturas pendientes para el primer cuatrimestre y hasta que no tenga la nota aprobada de todas no podrá presentar el proyecto.
- **Requisitos de plataforma:** La aplicación será instalable en dispositivos Android cuya versión sea igual o superior a la 8. La segunda versión más utilizada después de la 9, a la cual también se le da soporte ya que es superior.
- **Requisitos software:** Se acuerda el uso de Ionic con TypeScript, tecnología actualmente en auge para el desarrollo de aplicaciones híbridas. Además, como modelo de base de datos se acoge Firebase.

Capítulo 2 Plan de trabajo

2.1 Organización del equipo

Un equipo de desarrollo software se define como un conjunto de personas con cualidades técnicas orientadas al desarrollo de productos software que cooperan y trabajan en equipo para construir un producto totalmente operativo. El cliente es una parte vital para el éxito de un proyecto, mediante encuestas, entrevistas y pruebas, pero al no participar directamente en el desarrollo del proyecto no se le ha incluido en este apartado de organización de equipo.

Para que este proyecto tenga éxito, existen distintos perfiles necesarios para llevar a cabo todas las tareas del proyecto, son los siguientes:

- **Analista:** Responsable de captar las necesidades del cliente y asegurar que la solución de software propuesta se ajustara a ellas. A él le corresponde la elicitación de requisitos y reuniones con el cliente.
- **Programador:** A partir de la fase de elaboración, es el responsable del desarrollo final del software, esto incluye documentar el código.
- **Arquitecto de software:** Encargado del diseño de la solución adaptada, es decir, pensar en la aplicación antes de que sea desarrollada.
- **Tester:** Encargado de realizar las pruebas comprobando el funcionamiento correcto de cada parte del software. Hay gran interés en que este rol lo tome un usuario del servicio en vez de alguien propio del equipo para evitar que vea el software con “buenos ojos”.
- **Jefe de proyecto:** Encargado del seguimiento del proyecto, realización del plan de desarrollo y asignación de roles en el equipo de desarrollo.
- **Administrador de sistemas:** Encargado del diseño y configuración de la base de datos y del entorno de desarrollo.

En el caso del proyecto abordado en este documento, todos los roles excepto uno serán desempeñados por el alumno Raúl Jiménez Bores, siendo responsable del proyecto. La tutora del TFG Dña. Margarita Gonzalo Tasis desempeñará el rol externo de cliente, dando información al alumno sobre los detalles del software requerido. Por último, el rol de Tester será desempeñado por personas voluntarias externas al proyecto y por el alumno.

2.2 Plan de trabajo ideal

El plan ideal de un proyecto se realiza antes de su inicio para comprobar su viabilidad en torno a tiempo y costes del desarrollo. En caso de que sea viable, se toma el plan como

Turis-Route: Aplicación para la generación de rutas

una guía a seguir para el desarrollo del producto de tal forma que, si el plan se cumple, se asegure el éxito del proyecto. Debido a circunstancias personales, no se podrán dedicar un número de horas exactas diarias al proyecto, por lo que se planificará en semanas y en cada una de estas se llevarán a cabo un total de 20 horas fijas.

Nombre	Duración	Inicio	Fin	Responsable
Fase de inicio	3 semanas	jue. 08/10/20	mié. 28/10/20	-
Contexto del proyecto	1º Semana	jue. 08/10/20	mié. 14/10/20	Jefe de Proyecto
Búsqueda de la metodología	1º Semana	jue. 08/10/20	mié. 14/10/20	
Búsqueda y análisis de servicios similares	1º Semana	jue. 08/10/20	mié. 14/10/20	Jefe de proyecto
Definición del plan de trabajo	2º Semana	jue. 15/10/20	mié. 21/10/20	Jefe de proyecto
Análisis y gestión de riesgos	2º Semana	jue. 15/10/20	mié. 21/10/20	Jefe de Proyecto
Documento de presupuestos y costes	3º Semana	jue. 22/10/20	mié. 28/10/20	Jefe de proyecto
Identificación de requisitos inicial	3ºSemana	jue. 22/10/20	mié. 28/10/20	Analista
Instalación y preparación del entorno	3º Semana	mar. 22/10/20	mié. 28/10/20	Administrador de sistemas
Fase de elaboración	4 semanas	jue. 29/10/20	mié. 25/11/20	-
Especificación en detalle de requisitos	4º Semana	jue. 29/10/20	mié. 04/11/20	Analista
Diagrama casos de uso	4º Semana	jue. 29/10/20	mié. 04/11/20	Arquitecto de software
Especificación de casos de uso	5º Semana	jue. 05/11/20	mié. 11/11/20	Arquitecto de software
Modelado de dominio	5º y 6º Semana	jue. 05/11/20	mié. 18/11/20	Analista
Diseño y configuración de la base de datos	7º Semana	jue. 19/11/20	mié. 25/11/20	Administrador de sistemas
Fase de construcción	5 semanas	jue. 26/11/20	mié. 30/12/20	-

Implementación de las vistas del servicio web	8º Semana	jue. 26/11/20	mié. 02/12/20	Programador
Elaboración de la base de datos	9º Semana	jue. 03/12/20	mié. 09/12/20	Programador
Implementación de la funcionalidad del servicio	9º, 10º, 11º, 12º Semana	jue. 03/12/20	mié. 30/12/20	Programador
Fase de transición	4 semanas	jue. 31/12/20	vie. 22/01/21	-
Planificación de las pruebas con usuarios	13º Semana	jue. 31/12/20	mié. 06/01/21	Tester
Población de la base de datos para pruebas	14º Semana	jue. 07/01/21	mié. 13/01/21	Programador
Elaboración del manual para el uso de la aplicación	14º Semana	jue. 07/01/21	mié. 13/01/21	Tester
Pruebas de las distintas funcionalidades con el usuario	15º Semana	jue. 14/01/21	mié. 20/01/21	Tester
Corrección del software según la experiencia del usuario	16º Semana	jue. 21/01/21	mié. 27/01/21	Programador

Tabla 2.2—1 Plan de trabajo del proyecto

2.3 Gestión de costes

A continuación, se desglosarán los gastos estimados referentes al proyecto, desde recursos humanos hasta recursos físicos. Se comenzará por el coste del trabajador, este será calculado con el sueldo medio por hora de un ingeniero informático y teniendo en cuenta que en la planificación cada semana cuenta de 30 horas y que el proyecto idealmente se desarrollará en 16 semanas.

Después, por los costes del equipo utilizado en el proyecto, se necesitará un ordenador y un teléfono móvil smartphone. Se han analizado varias páginas de renting de ordenadores y se ha calculado el precio medio por mes del alquiler de estos, que es de unos 30 euros al mes, mientras que el alquiler de un teléfono móvil es de 10 euros al mes.

Se tendrán en cuenta también los gastos generales que generará el desarrollo del proyecto, como electricidad, gas, internet y alquiler. Tanto el gas como la electricidad se aproximarán teniendo en cuenta las 20 h de trabajo semanal. Respecto del internet serán 20 euros al mes, de los cuales se descontarían aquellas horas que no pertenecen a la jornada laboral, de la misma manera que el alquiler, que el mes completo serían 400 euros. Teniendo en cuenta la jornada de trabajo de 20 horas semanales y que una semana tiene 168 horas, el alquiler y el wifi se quedan en 48 euros/mes y 4 euros/mes respectivamente.

Turis-Route: Aplicación para la generación de rutas

Cabe destacar que estos costes son ideales y no se ha tenido en cuenta la posibilidad de que se manifieste algún riesgo, en caso de que esto sucediese, se sumará al coste y se detallará en el documento de costes reales del proyecto.

Concepto	Descripción	Coste(euros)
Trabajador	320 horas * 10,00€/h	3200€
Teléfono móvil	4 meses * 10€/mes	40€
Ordenador	Equipo de gama media 4 meses * 30€/mes	120€
Electricidad	4 meses * 40€/mes	160€
Gas	4 meses * 30€/mes	120€
Wifi	4 meses * 4€/mes	16€
Alquiler	4 meses * 48€/mes	192€
Coste total inicial	Sin IVA	3848€
IVA 24 %		923,52€
Total		4771,52€

Tabla 2.3—1 Costes del proyecto

Capítulo 3 Riesgos

3.1 Introducción

El desarrollo de un producto software es un proyecto que involucra una gran cantidad de recursos para poder ser llevado a cabo, es por eso por lo que una parte muy importante de este proceso es la identificación de los riesgos que supone desarrollar dicho producto software.

Para comenzar, explicaremos lo que significa un riesgo en este contexto:

Un riesgo es cualquier acontecimiento futuro que pueda afectar de forma negativa o positiva a un proyecto. No todos los riesgos tienen la misma probabilidad de ocurrir, ni el mismo impacto, por lo que a la hora de tratarlos nos preocuparemos de solucionar primero los más problemáticos. Los riesgos conllevan cambios, elecciones y dudas. Podremos conocer o no los riesgos, o simplemente son impredecibles.

Existen distintos tipos de riesgos y se clasifican de la siguiente manera:

- Riesgos de negocio: Ponen el peligro el plan, si se materializan supondrán un aumento de recursos.
 - Presupuesto
 - Tiempo
 - Personal
- Riesgos técnicos: Ponen en peligro la calidad del software resultante. Pueden relacionarse con:
 - Diseño: La arquitectura escogida no se ajusta al proyecto
 - Requisitos: Unos requisitos pobremente detallados o escasos.
 - Validación: escasez de pruebas.
 - Tecnologías desconocidas.
- Riesgos de producto: ponen en peligro la finalización del proyecto, si se dan, este deberá ser cancelado.
 - Viabilidad del producto en el mercado
 - Presupuesto: no se dispone de presupuesto suficiente para terminar el proyecto. Esto suele ser derivado de una mala planificación y gestión de costes ya que algunos proyectos fracasan porque nunca deberían de comenzar.

Ante los riesgos se pueden llevar a cabo distintas estrategias:

- Aceptar el riesgo: se decide seguir adelante con el proyecto sin tomar medidas contra el riesgo, en la mayoría de los casos esto se hace con

riesgos poco probables y que suponen un impacto mínimo, o con riesgos que supondrían mayores costes eliminar del que produciría en caso de materializarse.

- Transferir el riesgo: Se trata de modificar el plan para cambiar que parte debe soportar dicho riesgo.
- Evitación del riesgo: reduce la probabilidad de que el riesgo se materialice a cero, normalmente este tipo de resolución conlleva costes adicionales más elevados.

3.2 Análisis y plan de gestión de riesgos

Para el análisis de riesgos, se necesita establecer la exposición de este riesgo, es decir, una medida calculada en base al impacto y a la probabilidad de que este riesgo se materialice. A continuación, se muestra una tabla de donde se saca este valor:

Impacto/ Probabilidad	Muy Alto	Alto	Medio	Bajo	Muy Bajo
Catastrófico	Alto	Alto	Moderado	Moderado	Bajo
Crítico	Alto	Alto	Moderado	Bajo	Ninguno
Marginal	Moderado	Moderado	Bajo	Ninguno	Ninguno
Despreciable	Moderado	Bajo	Bajo	Ninguno	Ninguno

Figura 3.2—1 Criterios impacto/probabilidad

A continuación, se muestra las tablas con los riesgos identificados para este proyecto, todos están expuestos detalladamente y poseen un identificador.

Identificador	R-01
Título	Incumplimiento del plan de desarrollo
Descripción	Debido a diversos motivos se puede llegar a superar el tiempo estimado para la realización de las tareas y no se cumplen los plazos del plan.
Categoría	Riesgos de negocio
Impacto	Crítico

Turis-Route: Aplicación para la generación de rutas

Probabilidad	60%
Exposición	Alto
Plan de contingencia	Priorización de las tareas a desarrollar y revisión del plan de desarrollo para llevar a cabo el proyecto con los menores recursos posibles.

Tabla 3.2—1 Riesgo: incumplimiento del plan

Identificador	R-02
Título	Ausencia de familiaridad con el entorno o herramientas de trabajo
Descripción	El equipo de desarrollo no tiene suficiente experiencia con el entorno o las herramientas de trabajo y tecnologías de desarrollo y supone un coste adicional de tiempo llevar a cabo las tareas.
Categoría	Riesgos de negocio
Impacto	Crítico
Probabilidad	30%
Exposición	Bajo
Plan de contingencia	Impartición de cursos gratuitos para el equipo sobre las tecnologías del proyecto previamente al inicio de este o en horas fuera del proyecto.

Tabla 3.2—2 Riesgo: Ausencia de familiaridad con el entorno

Identificador	R-03
Título	Escasez de detalle en los requisitos

Descripción	Debido a un documento pobremente elaborado, los requisitos no quedan claros por lo que no se pueden realizar correctamente las tareas.
Categoría	Riesgos técnicos
Impacto	Moderado
Probabilidad	40%
Exposición	Alto
Plan de contingencia	Reelaborar el documento de requisitos teniendo en cuenta los errores previos y asegurándose de comprender la información dada por el cliente

Tabla 3.2—3 Riesgo: escasez de detalle en los requisitos

Identificador	R-04
Título	Problemas de salud en el equipo de desarrollo
Descripción	Debido a bajas y problemas sanitarios puede llegar a ser complicado cumplir los plazos del plan
Categoría	Riesgos de negocio
Impacto	Crítico
Probabilidad	60%
Exposición	Alto
Plan de contingencia	Revisión del plan de desarrollo para adaptarse a la realidad debido a estas bajas médicas

Tabla 3.2—4 Riesgo: problemas de salud del equipo

Identificador	R-05
Título	Dificultades con el trabajo en equipo
Descripción	Problemas para la cooperación entre los miembros del equipo que pueden llegar a retrasar las tareas del plan
Categoría	Riesgos de negocio
Impacto	Crítico
Probabilidad	30%
Exposición	Baja
Plan de contingencia	Llevar a cabo reuniones con periodicidad, exponiendo distintas dudas y opiniones para poder elegir entre todos la mejor forma de proceder.

Tabla 3.2—5 Riesgo: dificultades con el trabajo en equipo

Identificador	R-06
Título	Malfuncionamiento de los servidores de alguna de las dependencias de la app
Descripción	Caída del Servidor por avería o fallo eléctrico. Pueden ser servidores de proveedores externos cuya caída imposibilitaría el funcionamiento de nuestra app.
Categoría	Riesgos de negocio
Impacto	Crítico

Probabilidad	15%
Exposición	Bajo
Plan de contingencia	Restauración del sistema en otro servidor mediante una copia de seguridad.

Tabla 3.2—6 Riesgo: malfuncionamiento de los servidores

Identificador	R-07
Título	Problemas de usabilidad
Descripción	El manejo del sistema no se adapta a los usuarios objetivo, llevando más tiempo de lo pensado aprender a usar el servicio.
Categoría	Riesgos de negocio
Impacto	Crítico
Probabilidad	30%
Exposición	Alto
Plan de contingencia	Prestar especial atención a las experiencias de los usuarios en las pruebas con el sistema, evaluando su satisfacción y la dificultad que les supone utilizar el sistema.

Tabla 3.2—7 Riesgo: problemas de usabilidad

Identificador	R-08
Título	Cambios en los requisitos
Descripción	Aparición de nuevos requisitos en el proyecto

Categoría	Riesgos técnicos
Impacto	Crítico
Probabilidad	20%
Exposición	Bajo
Plan de contingencia	Análisis y evaluación del impacto del cambio requerido, replanificación del plan de trabajo.

Tabla 3.2—8 Riesgo: cambios en los requisitos

Identificador	R-09
Título	Rechazo del software por el cliente
Descripción	El software desarrollado es rechazado por el cliente en base a distintos problemas que hacen que no se ajuste a su petición.
Categoría	Riesgos de negocio
Impacto	Crítico
Probabilidad	30%
Exposición	Alto
Plan de contingencia	Realizar más pruebas y correcciones del software hasta que el cliente dé su aprobación

Tabla 3.2—9 Riesgo: rechazo del software

Identificador	R-10
----------------------	------

Título	Perdida de datos relacionados con el proyecto
Descripción	Malfuncionamiento del equipo local en el que se desarrolla el proyecto. Esto puede ser debido a múltiples causas y provoca que la información que no se encuentre guardada en un sistema Backup se pierda
Categoría	Riesgos de negocio
Impacto	Crítico
Probabilidad	30%
Exposición	Alto
Plan de contingencia	Sincronizar el proyecto local con el Backup frecuentemente para así lograr que la información perdida sea lo menos posible

Tabla 3.2—10 Riesgo: pérdida de datos

3.3 Desviaciones en la planificación original

A menudo, en un proyecto software se producen retrasos, que no son ni más ni menos que la manifestación de los riesgos comentados en la sección anterior. Dependiendo del plan de contingencia estos riesgos pueden afectar en mayor o menor grado a la planificación, pero siempre tienen algo de impacto sobre el proyecto. A continuación, se expondrá el seguimiento del plan real, con las desviaciones y los riesgos manifestados.

Nombre	Duración	Inicio	Fin	Responsable
Fase de inicio	3 semanas(80h)	jue. 08/10/20	mié. 30/10/20	-
Contexto del proyecto	1º Semana	jue. 08/10/20	mié. 14/10/20	Jefe de Proyecto
Búsqueda de la metodología	1º Semana	jue. 08/10/20	mié. 14/10/20	
Búsqueda y análisis de servicios similares	1º Semana	jue. 08/10/20	mié. 14/10/20	Jefe de proyecto
Definición del plan de trabajo	2º Semana	jue. 15/10/20	mié. 21/10/20	Jefe de proyecto

Turis-Route: Aplicación para la generación de rutas

Análisis y gestión de riesgos	2º Semana	jue. 15/10/20	mié. 21/10/20	Jefe de Proyecto
Documento de presupuestos y costes	3º Semana	jue. 22/10/20	mié. 28/10/20	Jefe de proyecto
Identificación de requisitos	3ºSemana	jue. 22/10/20	mié. 28/10/20	Analista
Instalación y preparación del entorno	3º Semana	mar. 22/10/20	mié. 28/10/20	Administrador de sistemas
Fase de elaboración	4 semanas y media(90h)	jue. 02/11/20	mié. 29/11/20	-
Especificación en detalle de requisitos	4º y 5ºSemana	jue. 02/11/20	mié. 12/11/20	Analista
Diagrama casos de uso	4º y 5º Semana	jue. 02/10/20	mié. 12/11/20	Arquitecto de software
Especificación de casos de uso	5º y 6º Semana	jue. 13/11/20	mié. 20/11/20	Arquitecto de software
Modelado de dominio	6º y 7º Semana	jue. 10/11/20	mié. 25/11/20	Analista
Diseño y configuración de la base de datos	7º y 8º Semana	jue. 19/11/20	mié. 29/11/20	Administrador de sistemas
Fase de construcción	5 semanas y media(110h)	jue. 30/11/20	mié. 07/01/21	-
Implementación de las vistas del servicio web	8º y 9º Semana	jue. 30/11/20	mié. 07/12/20	Programador
Elaboración de la base de datos	9º y 10º Semana	jue. 08/12/20	mié. 15/12/20	Programador
Implementación de la funcionalidad del servicio	9º, 10º, 11º, 12º y 13ºSemana	jue. 08/12/20	mié. 07/01/21	Programador
Fase de transición	4 semanas(80h)	jue. 08/01/21	vie. 06/02/21	-
Planificación de las pruebas con usuarios	14º Semana	jue. 08/01/21	mié. 15/01/21	Tester
Población de la base de datos para pruebas	15º Semana	jue. 16/01/21	mié. 23/01/21	Programador
Elaboración del manual para el uso de la aplicación	15º Semana	jue. 16/01/21	mié. 23/01/21	Tester
Pruebas de las distintas funcionalidades con el usuario	16º Semana	jue. 24/01/21	mié. 01/02/21	Tester

Turis-Route: Aplicación para la generación de rutas

Corrección del software según la experiencia del usuario	17° Semana	jue. 02/02/21	mié. 06/02/21	Programador
--	------------	------------------	------------------	-------------

Tabla 3.3—11 Tabla planificación real

Se puede apreciar que el plan real cuenta con algunos retrasos respecto al ideal, vamos a explicarlos detalladamente a continuación.

- En la fase de inicio, hubo un retraso de dos días debido a que esa semana hubo una carga mayor de trabajo en el entorno laboral del alumno, por lo que no se pudieron dedicar las horas semanales requeridas. Termina con un total de 2 días de retraso.
- Los retrasos de la fase de elaboración se deben, en parte a los retrasos conllevados por la fase anterior y a la manifestación de un riesgo de cambios en los requisitos en la segunda iteración de esta fase, por lo que se tuvo que volver a elaborar el documento de requisitos con las nuevas necesidades. Al final el impacto respecto de la planificación no es muy elevado ya que las últimas tareas de la fase se pudieron lograr sin verse muy afectadas por los retrasos anteriores. Tiene un total de 4 días de retraso.
- En suma a los retrasos arrastrados, se añade la manifestación del riesgo de falta de conocimientos con las tecnologías a emplear, que acaban retrasando el fin de la fase en 7 días en total.
- Por último, la fase de transición se lleva a cabo sin complicaciones en el tiempo esperado.

Como se puede apreciar, respecto de la fecha de fin original del proyecto, 22 de enero de 2021, a la fecha de fin real del proyecto 6 de febrero de 2021, hay 14 días (2 semanas) de diferencia.

Capítulo 4 Análisis

4.1 Requisitos

Los requisitos de un proyecto software son aquellas funcionalidades, información o comportamientos que el producto tiene que presentar para adecuarse a aquello pedido por el cliente. A continuación, realizamos una elicitación y análisis de los requisitos del sistema.

4.1.1 Requisitos Funcionales

Identificador	RF-01
Título	Mostrar lugares de interés
Descripción	El sistema deberá mostrar al usuario lugares de interés de una ciudad para poder crear rutas hacia ellos

Tabla 4.1.1—1 Requisito Funcional 1

Identificador	RF-02
Título	Ubicación del usuario
Descripción	El sistema debe permitir al usuario visualizar su ubicación en tiempo real

Tabla 4.1.1—2 Requisito funcional 2

Identificador	RF-03
Título	Personalización de ruta
Descripción	El sistema deberá permitir al usuario seleccionar los lugares de interés que desea incluir en la ruta.

Tabla 4.1.1—3 Requisito Funcional 3

Turis-Route: Aplicación para la generación de rutas

Identificador	RF-04
Título	Resumen de ruta
Descripción	El sistema deberá mostrar la ruta prediseñada.

Tabla 4.1.1—4 Requisito Funcional 4

Identificador	RF-05
Título	Confirmación y creación de rutas
Descripción	El sistema deberá permitir al usuario aceptar la ruta prediseñada.

Tabla 4.1.1—5 Requisito Funcional 5

Identificador	RF-06
Título	Seguimiento de ruta
Descripción	El sistema deberá mostrar al usuario la ruta aceptada en la pantalla del dispositivo con seguimiento de ruta incluida.

Tabla 4.1.1—6 Requisito Funcional 6

Identificador	RF-07
Título	Finalizar una ruta
Descripción	El sistema permitirá finalizar la ruta cuando el usuario lo considere oportuno o cuando se llegue al destino final.

Tabla 4.1.1—7 Requisito Funcional 7

Identificador	RF-08
Título	Registro de usuario
Descripción	El sistema deberá permitir al usuario registrarse en el sistema.

Tabla 4.1.1—8 Requisito Funcional 8

Identificador	RF-09
Título	Inicio de sesión
Descripción	El sistema deberá permitir al usuario iniciar sesión en el sistema.

Tabla 4.1.1—9 Requisito Funcional 9

Identificador	RF-10
Título	Cerrar sesión
Descripción	El sistema deberá permitir al usuario cerrar sesión en el sistema.

Tabla 4.1.1—10 Requisito Funcional 10

Identificador	RF-11
Título	Baja del sistema
Descripción	El sistema deberá permitir al usuario darse de baja del sistema. Borrando todo el contenido asociado al mismo.

Tabla 4.1.1—11 Requisito Funcional 11

Identificador	RF-12
Título	Modificación de ruta en tiempo real
Descripción	El sistema mostrara al usuario lugares de interés cercanos durante el recorrido de su ruta (no incluidos en la ruta original). Permitiendo añadirlos y recalculando la ruta.

Tabla 4.1.1—12 Requisito Funcional 12

Identificador	RF-13
Título	Recomendaciones sobre la ruta.
Descripción	El sistema deberá sugerir al usuario sitios de interés para visitar compatibles con su ruta original.

Tabla 4.1.1—13 Requisito Funcional 13

Identificador	RF-14
Título	Rutas innovadoras
Descripción	El sistema deberá ser capaz de ofrecer rutas diferentes dependiendo de los lugares previamente visitados por el usuario.

Tabla 4.1.1—14 Requisito Funcional 14

Identificador	RF-15
Título	Elección de rutas
Descripción	El sistema deberá ser capaz de permitir al usuario seleccionar para una ruta diferentes variantes (rápida o pintoresca) en función de sus intereses.

Tabla 4.1.1—15 Requisito Funcional 15

4.1.2 Requisitos de información

Identificador	RI-01
Título	Lugares visitados por el usuario
Descripción	El sistema deberá almacenar los lugares de interés que el usuario ha visitado.

Tabla 4.1.2—1 Requisito de información 1

Identificador	RI-02
Título	Datos del usuario
Descripción	El sistema deberá almacenar la contraseña, correo electrónico, nombre, apellidos y nombre del usuario.

Tabla 4.1.2—2 Requisito de información 2

4.1.3 Requisitos no funcionales

Otro tipo de requisitos son aquellos que relatan los atributos de calidad del software o características del funcionamiento de este. A continuación, se exponen los requisitos no funcionales que afectan al software a realizar:

Identificador	RNF-01
Título	No repeticiones en el email de las cuentas
Descripción	El email debe ser único para cada usuario.

Tabla 4.1.3—1 Requisito No Funcional 1

Identificador	RNF-02
Título	Características de la contraseña
Descripción	La contraseña ha de tener entre 6 y 12 caracteres

Tabla 4.1.3—2 Requisito No Funcional 2

Identificador	RNF-03
Título	Eficiencia
Descripción	El usuario dispondrá de botones de rápida accesibilidad para llegar a las funciones de la aplicación

Tabla 4.1.3—3 Requisito No Funcional 3

Identificador	RNF-04
Título	Compatibilidad
Descripción	El sistema deberá de ser compatible con las versiones Android 7 en adelante

Tabla 4.1.3—4 Requisito No Funcional 4

Identificador	RNF-05
Título	Formato
Descripción	El sistema deberá utilizar el formato UTF-8

Tabla 4.1.3—5 Requisito No Funcional 5

Identificador	RNF-06
Título	Disponibilidad
Descripción	El sistema debe de estar disponible las 24 horas del día.

Tabla 4.1.3—6 Requisito No Funcional 6

Identificador	RNF-07
Título	Privacidad de datos

Descripción	El sistema debe de proteger los datos sensibles proporcionados por el usuario
--------------------	---

Tabla 4.1.3—7 Requisito No Funcional 7

Identificador	RNF-08
Título	Facilidad de recuerdo
Descripción	La curva de aprendizaje debe ser significativamente menor cuando un usuario ya ha utilizado el sistema tiempo atrás.

Tabla 4.1.3—8 Requisito No Funcional 8

4.2 Cambios en los requisitos

Con el desarrollo del proyecto se manifestaron cambios en los requisitos para adaptarse mejor a las necesidades del producto final. En la segunda iteración de la fase de elaboración se realizaron ciertas modificaciones por lo que quedaron eliminados los siguientes requisitos

Identificador	Motivo
RF-03	El modelo final del producto permitirá escoger un solo lugar destino con sugerencias en la ruta, pero no permitirá personalizar los pasos intermedios de la ruta.
RF-04	Con el motivo de agilizar el proceso de creación de rutas, la ruta se mostrará directamente, no será necesario confirmar esa previsualización
RF-14	Se va a optar por un modelo de negocio en el que se deje libre al usuario por la ruta, es decir, no se especificará que ruta estricta seguir, sino que se le proporcionarán sugerencias cercanas a su ruta para que el usuario se desvíe de su ruta original con total libertad.
RF-15	

Tabla 4.2—1 Eliminaciones de requisitos

Otros de ellos sufrieron modificaciones en esta segunda iteración:

Identificador	RF-12
Título	Sugerencias de ruta en tiempo real
Descripción	El sistema mostrara al usuario lugares de interés cercanos durante el recorrido de su ruta (no incluidos en la ruta original).

Tabla 4.2—2 Modificación requisito 12

Por otro lado, se crearon dos requisitos nuevos:

Identificador	RF-16
Título	Cambio de nombre de usuario
Descripción	El sistema debe permitir al usuario cambiar su nombre de usuario.

Tabla 4.2—3 Requisito Funcional 16

Identificador	RF-17
Título	Cambio de contraseña de usuario
Descripción	El sistema deberá permitir al usuario cambiar su contraseña.

Tabla 4.2—4 Requisito Funcional 17

4.3 Casos de uso

En nuestro caso identificamos un único actor, el usuario que será el que realizará toda la funcionalidad de la app, desde crear rutas y ver su historial de lugares visitados hasta registrarse e iniciar sesión. En la siguiente imagen, se muestra el diagrama de casos de uso:

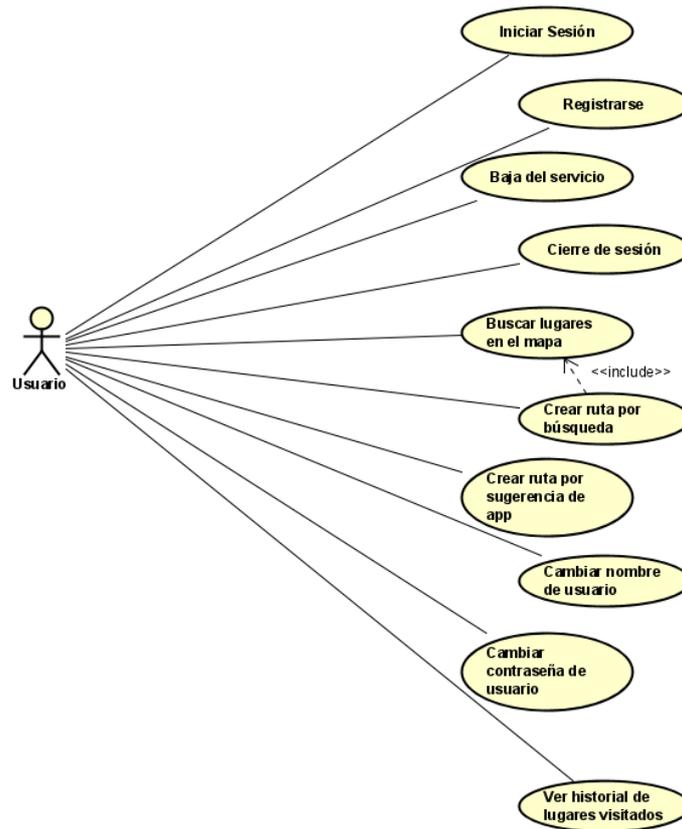


Figura 4.2—1 Diagrama de casos de uso de la aplicación

4.3.1 Especificación de casos de uso

A continuación, se muestran una serie de tablas detallando los casos de uso del diagrama anterior:

CU-01	Iniciar sesión
Versión	1.0
Actor	Usuario
Precondición	El usuario ha de estar registrado en el sistema.
Descripción	El usuario quiere identificarse en la aplicación con sus datos
Secuencia normal	1. El usuario introduce su email y contraseña

	<p>2. El usuario pulsa en el botón iniciar sesión</p> <p>3. Fin</p>
Postcondición	La sesión queda iniciada en el sistema
Secuencia alternativa	
Excepciones	El usuario no ha introducido alguno de los campos o alguno de estos es incorrecto. En este caso se mostrará el error y volvería al paso 1.

Tabla 4.3.1—1 Caso de uso iniciar sesión

CU-02	Registrarse
Versión	1.0
Actor	Usuario
Precondición	
Descripción	El usuario quiere registrarse en el sistema.
Secuencia normal	<p>1. El usuario pulsa registrarse</p> <p>2. El usuario introduce email, nombre, apellidos, nombre de usuario y contraseña</p> <p>3. El usuario pulsa el botón registrarse</p> <p>4. Fin</p>
Postcondición	El usuario queda registrado en el sistema
Secuencia alternativa	El usuario decide no registrarse y pulsa la flecha para ir atrás.

Excepciones	1. El usuario no ha introducido alguno de los campos o alguno de estos no sigue el formato adecuado. En este caso se volvería al paso 1.
--------------------	--

Tabla 4.3.1—2 Caso de uso Registro

CU-03	Cerrar sesión
Versión	1.0
Actor	Usuario
Precondición	El usuario ha de tener una sesión abierta en el sistema
Descripción	El usuario ha terminado de usar la aplicación y desea cerrar la sesión.
Secuencia normal	1. El usuario pulsa en el botón de cerrar sesión 2. Fin
Postcondición	La sesión se cierra y la aplicación vuelve a la ventana de iniciar sesión
Secuencia alternativa	
Excepciones	

Tabla 4.3.1—3 Caso de uso Cerrar sesión

CU-04	Baja del sistema
--------------	------------------

Versión	1.0
Actor	Usuario
Precondición	El usuario ha de estar registrado en el sistema y con la sesión abierta.
Descripción	El usuario quiere dejar de tener acceso a la aplicación con su cuenta.
Secuencia normal	<ol style="list-style-type: none"> 1. El usuario pulsa el botón de ajustes 2. Dentro de ajustes pulsa baja del sistema 3. Fin
Postcondición	El usuario queda borrado de la base de datos
Secuencia alternativa	
Excepciones	

Tabla 4.3.1—4 Caso de uso baja del sistema

CU-05	Ver historial de lugares visitados
Versión	1.0
Actor	Usuario
Precondición	El usuario ha de tener la sesión abierta en el sistema.
Descripción	El usuario quiere ver que lugares ha visitado desde que tiene la aplicación
Secuencia normal	<ol style="list-style-type: none"> 1. El usuario accede a su perfil

	<p>2. El usuario pulsa en historial de lugares visitados</p> <p>4. Fin</p>
Postcondición	El sistema muestra los lugares que el usuario ha visitado
Secuencia alternativa	
Excepciones	El usuario no ha realizado ninguna ruta por lo que el sistema le muestra que todavía no hay ningún sitio de interés en su historial

Tabla 4.3.1—5 Caso de uso ver historial de lugares visitados

CU-06	Crear ruta por búsqueda
Versión	1.0
Actor	Usuario
Precondición	El usuario ha de tener una sesión abierta en el sistema
Descripción	El usuario quiere crear una ruta a un lugar de interés
Secuencia normal	<p>1. El usuario accede al mapa</p> <p>2. El usuario busca el lugar de destino en el mapa</p> <p>5. El usuario selecciona el lugar correcto que el autocompletado le muestre</p> <p>6. La ruta se muestra en la pantalla destacando aquellos lugares pintorescos cerca de su camino.</p> <p>7. Fin</p>
Postcondición	

Secuencia alternativa	
Excepciones	Si el destino introducido es excesivamente lejano, la aplicación no aplica el algoritmo de extracción de lugares cercanos ya que tiene que realizar muchas consultas y demoraría demasiado la carga de la ruta.

Tabla 4.3.1—6 Caso de uso crear ruta por búsqueda

CU-07	Buscar lugares en el mapa
Versión	1.0
Actor	Usuario
Precondición	El usuario ha de tener una sesión abierta en el sistema
Descripción	El usuario desea buscar un lugar en el mapa
Secuencia normal	<ol style="list-style-type: none"> 1. El usuario accede al mapa 2. El usuario introduce en la barra buscadora el nombre del lugar 3. El mapa se centra en ese lugar añadiéndole un marcador 4. Fin
Postcondición	
Secuencia alternativa	
Excepciones	El lugar introducido no existe o es ambiguo, se recomienda pulsar en el autocompletado de la barra buscadora

Tabla 4.3.1—7 Caso de uso buscar lugares en el mapa

CU-08	Crear ruta por sugerencia de app
Versión	1.0
Actor	Usuario
Precondición	El usuario ha de tener una sesión abierta en el sistema
Descripción	El usuario quiere crear una ruta a un lugar de interés y desea que la app le recomiende o le muestre una lista de los más importantes.
Secuencia normal	<ol style="list-style-type: none"> 1. El usuario accede a la vista puntos de interés en la parte superior derecha de la ventana home 2. El usuario selecciona aquel lugar que desea visitar 3. El usuario hace click en el botón flotante “Go” en la parte inferior derecha de la pantalla. 4. La ruta se muestra en la pantalla. 5. Fin
Postcondición	
Secuencia alternativa	2b. El usuario desea volver al mapa y pulsa en la fecha en la esquina superior izquierda
Excepciones	

Tabla 4.3.1—8 Caso de uso crear ruta por sugerencia de la app

CU-09	Cambiar nombre de usuario
Versión	1.0

Actor	Usuario
Precondición	El usuario ha de tener una sesión abierta en el sistema
Descripción	El usuario quiere cambiar su nombre de usuario
Secuencia normal	<ol style="list-style-type: none"> 1. El usuario accede a su perfil 2. El usuario pulsa en el botón cambiar nombre 3. El usuario introduce su nuevo nombre 4. El usuario pulsa aceptar 5. Fin
Postcondición	El nombre de usuario ha sido cambiado en la base de datos
Secuencia alternativa	En cualquiera de los pasos el usuario puede pulsar la flecha en la esquina superior izquierda para volver atrás
Excepciones	

Tabla 4.3.1—9 Caso de uso cambiar nombre de usuario

CU-10	Cambiar contraseña de usuario
Versión	1.0
Actor	Usuario
Precondición	El usuario ha de tener una sesión abierta en el sistema
Descripción	El usuario quiere cambiar su contraseña
Secuencia normal	<ol style="list-style-type: none"> 1. El usuario accede a su perfil

	<ol style="list-style-type: none">2. El usuario pulsa en el botón cambiar contraseña3. El usuario introduce su nueva contraseña en los dos campos a rellenar4. El usuario pulsa el botón cambiar contraseña.5. El sistema comprueba que las contraseñas coincidan, en caso afirmativo cambia la contraseña en la base de datos.6. Fin
Postcondición	La contraseña ha sido cambiada en la base de datos
Secuencia alternativa	<p>En cualquiera de los pasos el usuario puede pulsar la flecha en la esquina superior izquierda para volver atrás.</p> <p>5.b En caso de que las contraseñas no sean coincidentes muestra un mensaje de error y vuelve al paso 3.</p>
Excepciones	En caso de llevar tiempo con la sesión iniciada en el sistema, se obliga a volver a iniciar sesión para poder cambiar la contraseña.

Tabla 4.3.1—10 Caso de uso cambiar contraseña

4.4 Modelo de dominio

A continuación, se presenta el modelo de dominio de la aplicación en el que se documentan las distintas entidades, atributos y operaciones que poseerá el sistema.

Turis-Route: Aplicación para la generación de rutas

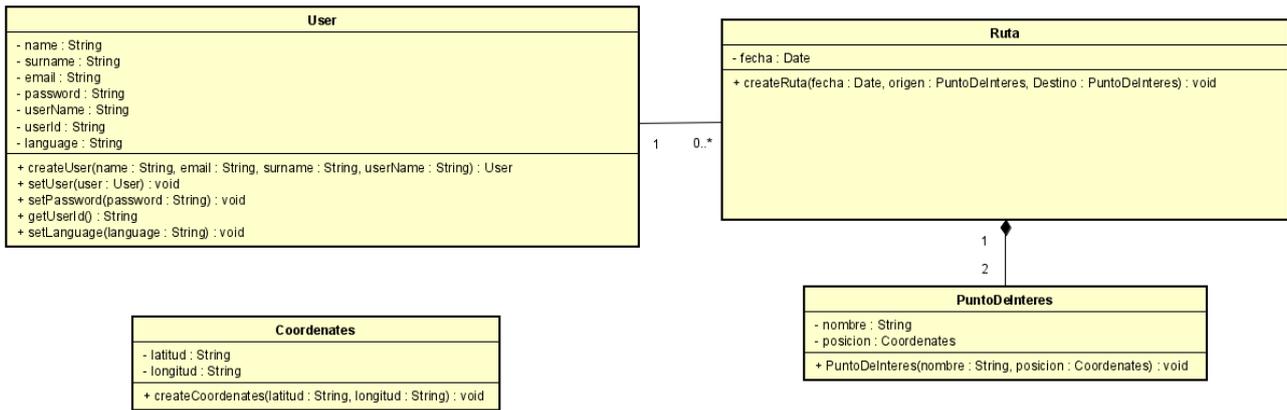


Figura 4.3—1 Modelo de dominio de la aplicación

En este apartado se detallará cada una de las clases y sus atributos del modelo de dominio:

User

- Descripción: clase que modela a un usuario de la aplicación
- Atributos:
 - name: nombre real del usuario de la aplicación.
 - surname: apellidos reales del usuario de la aplicación.
 - email: correo electrónico relacionado con el usuario de la aplicación.
 - password: contraseña ligada al usuario requerida para iniciar sesión en la aplicación.
 - userId: identificador de usuario que se corresponde con el identificador único de Firebase.
 - Language: cadena que se corresponde con el lenguaje preferido por el usuario, puede estar vacío en caso de que no haya cambiado su configuración.
 - userName: nombre del usuario en la aplicación

Ruta

- Descripción: clase que modela una ruta de un usuario.
- Atributos:
 - Fecha: la fecha se corresponde con el tiempo medido en horas minutos y segundos, a su vez día mes y año en el que se realizó la ruta

PuntoDeInteres

- Descripción: clase que modela un punto de interés.

- Atributos:
 - o nombre: nombre del punto de interés.
 - o Posición: coordenadas en las que se encuentra el punto de interés.

Coordenadas

- Descripción: clase que modela unas coordenadas.
- Atributos:
 - o latitud: latitud de las coordenadas.
 - o longitud: longitud de las coordenadas.

4.5 Diagramas de secuencia

4.5.1 CU Iniciar Sesión

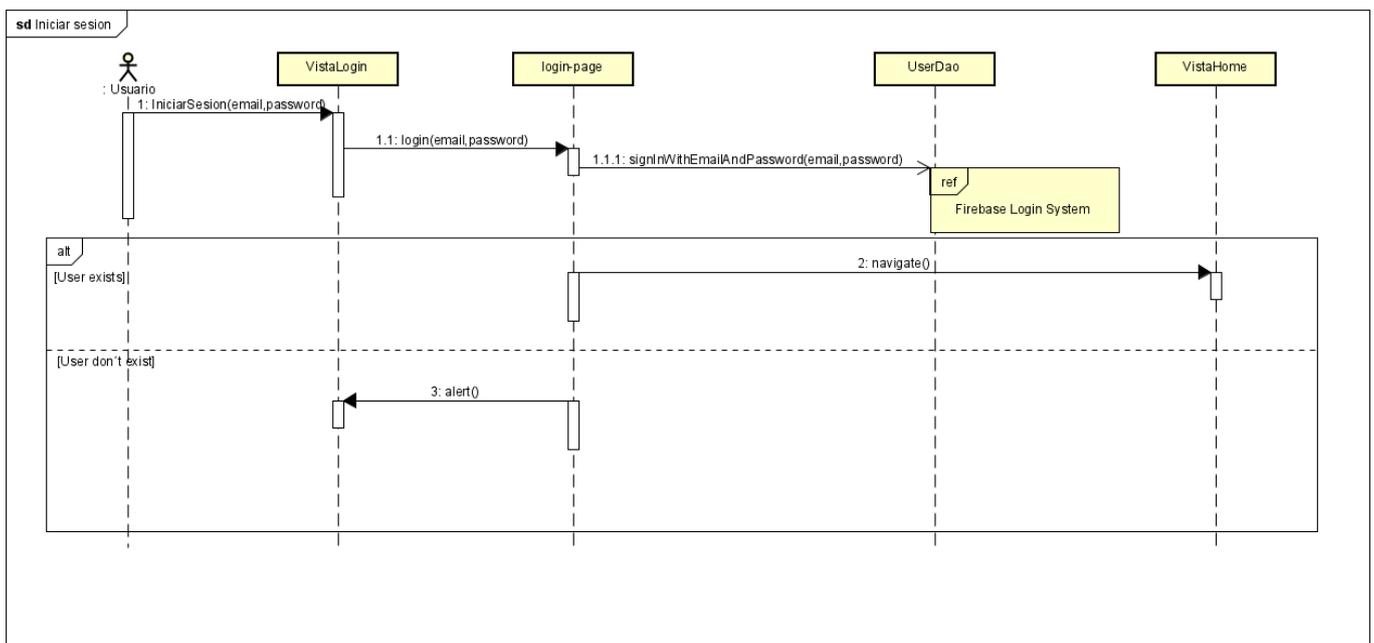


Figura 4.3.1—1 Diagrama de secuencia del caso de uso iniciar sesión

4.5.2 CU Registro

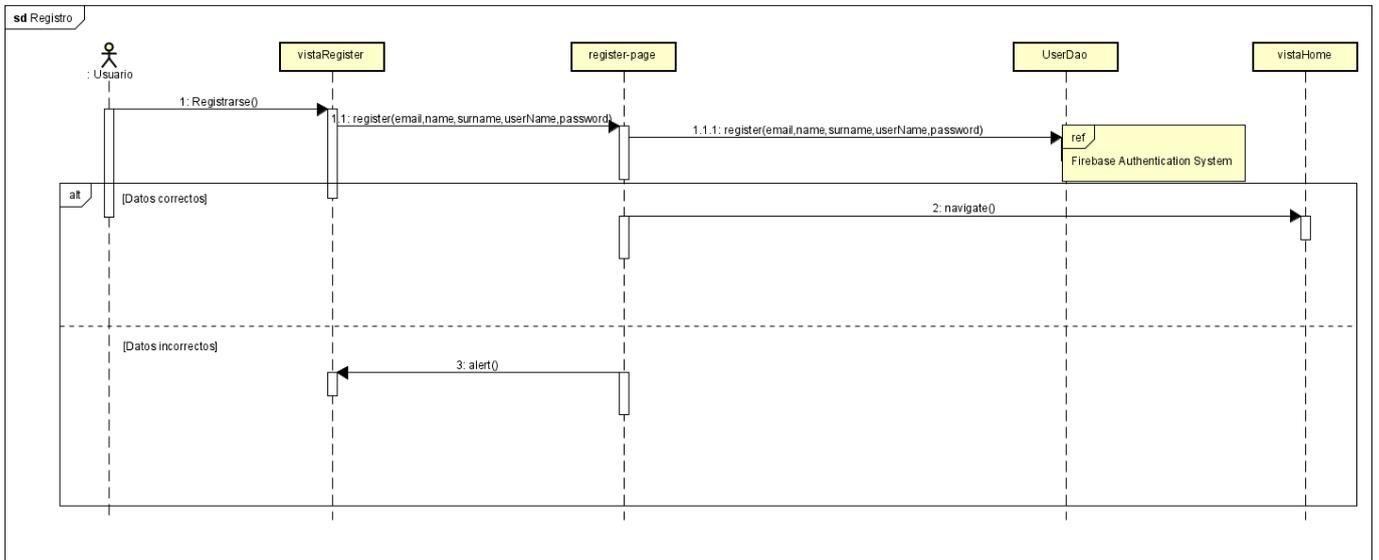


Figura 4.5.2—1 Diagrama de secuencia del caso de uso registro

4.5.3 CU Baja del sistema

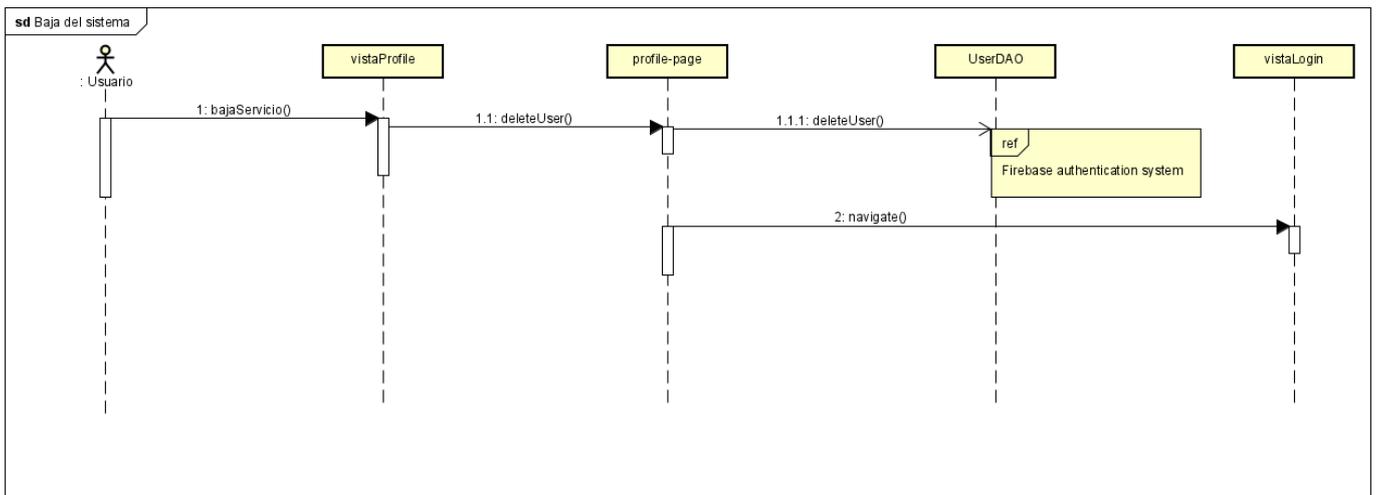


Figura 4.5.23—1 Diagrama de secuencia del caso de uso baja del sistema

4.5.4 CU Cerrar sesión

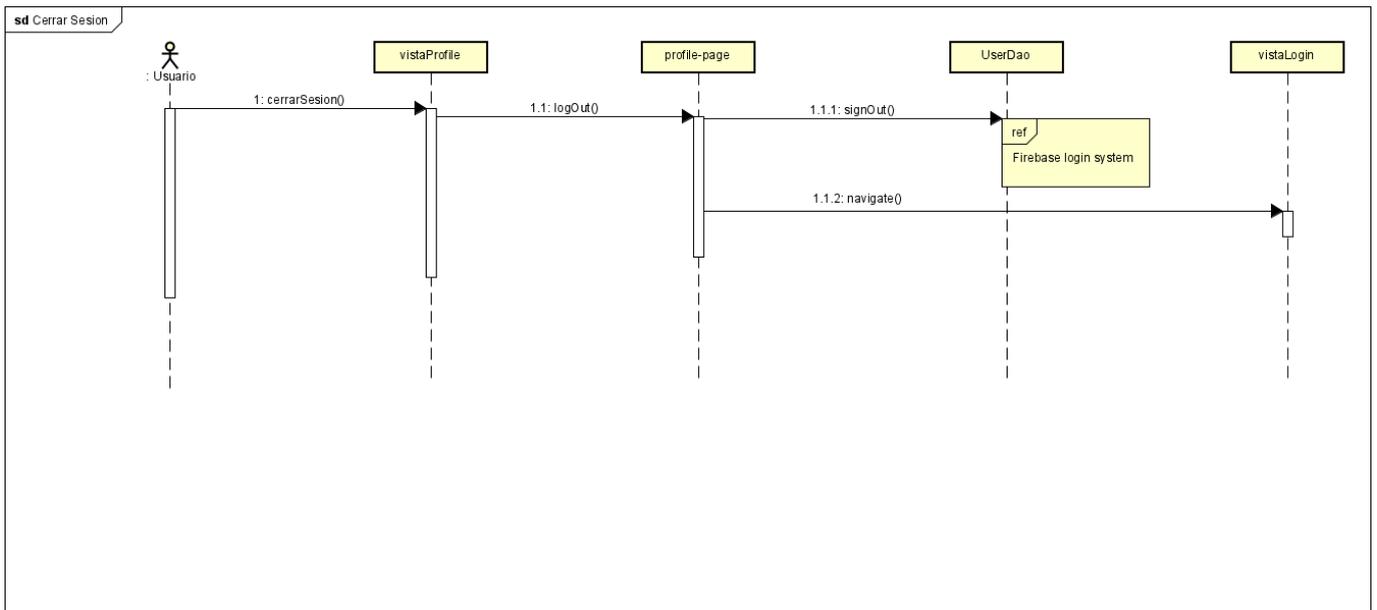


Figura 4.5.4—1 Diagrama de secuencia del caso de uso cerrar sesión

4.5.5 CU Cambio de contraseña

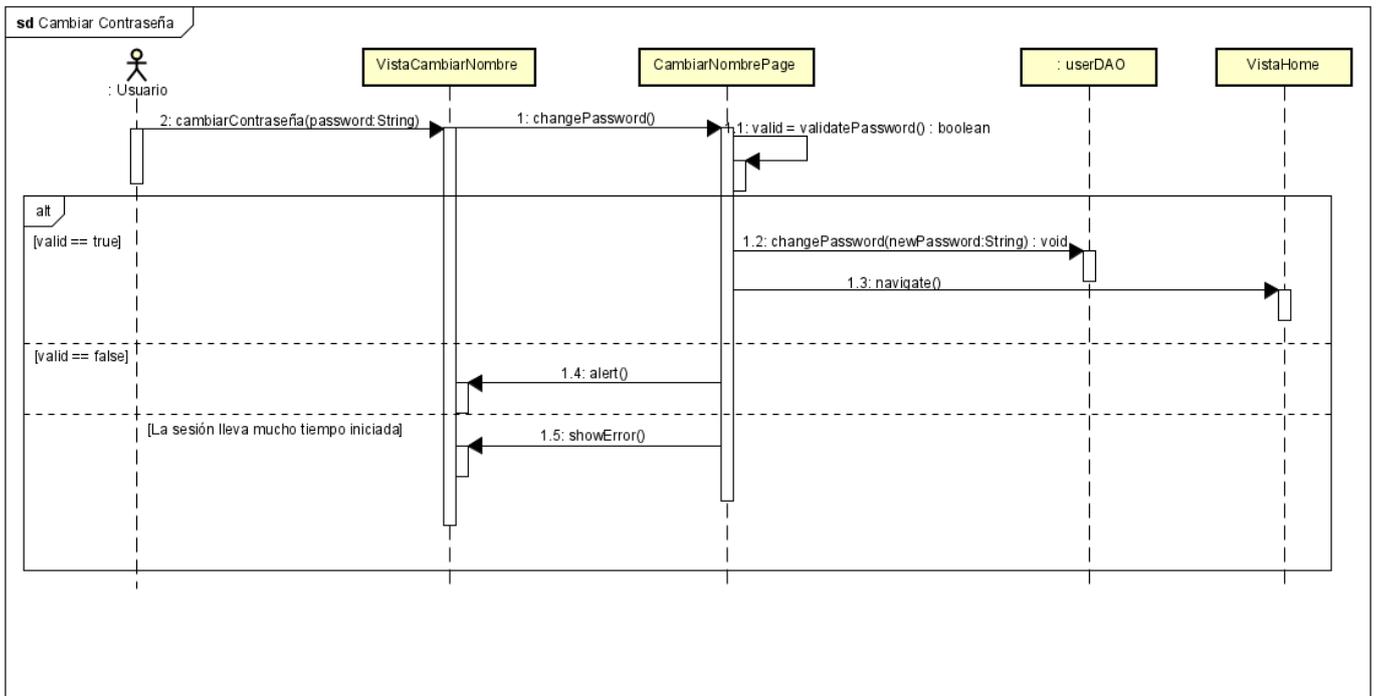


Figura 4.5.45—1 Diagrama de secuencia del caso de uso cambio de contraseña

4.5.6 CU Crear ruta por sugerencia

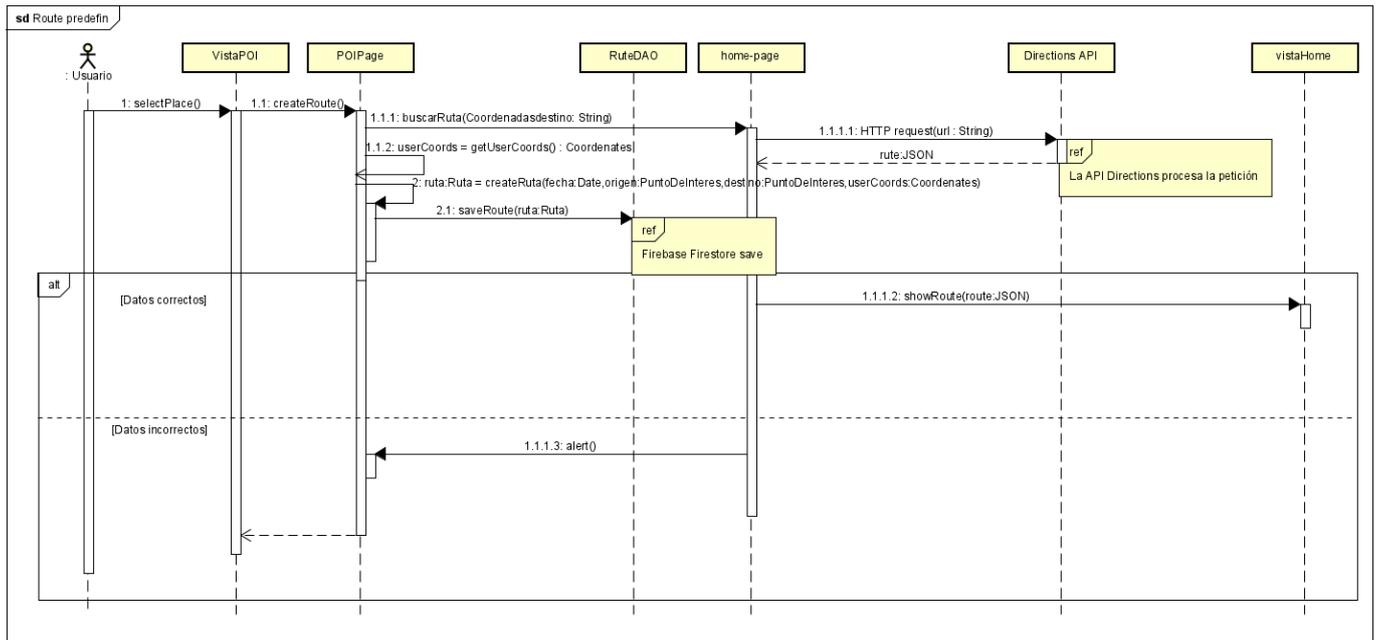


Figura 4.5.56—1 Diagrama de secuencia del caso de uso crear ruta por sugerencia

4.5.7 CU Cambio de nombre

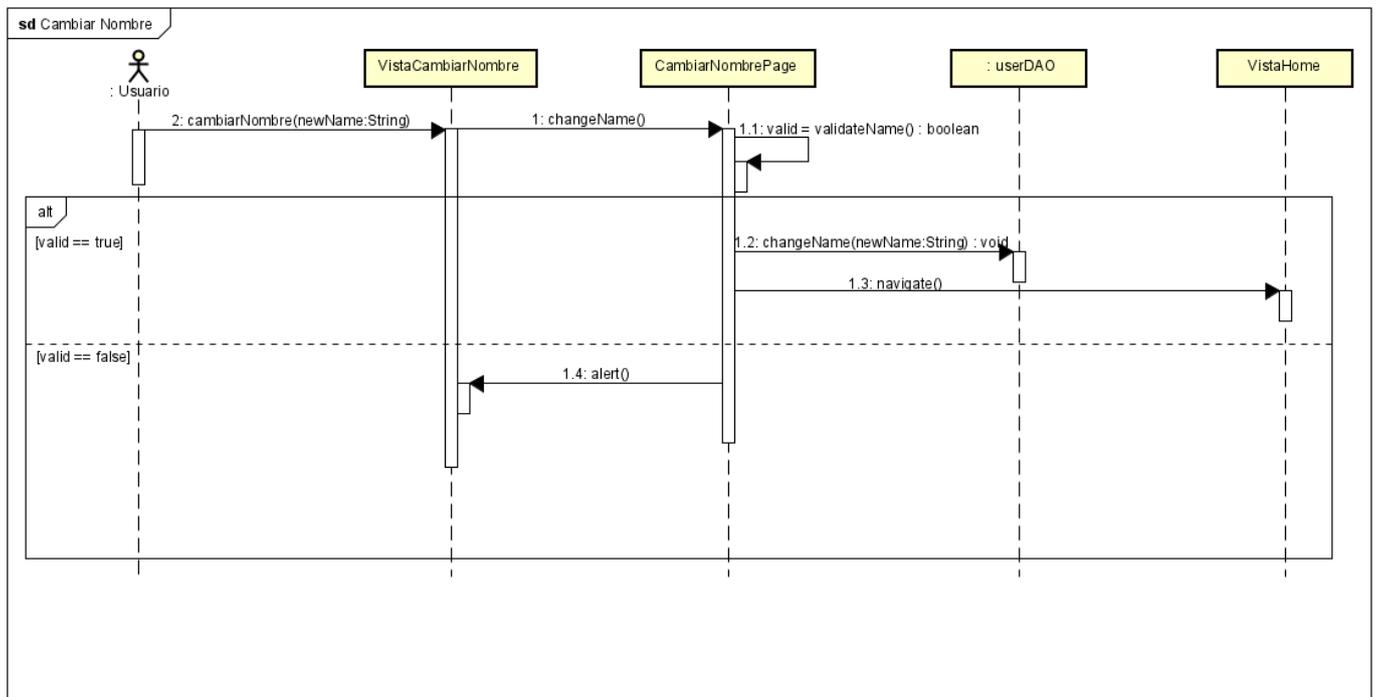


Figura 4.5.67—1 Diagrama de secuencia del caso de uso cambiar nombre

4.5.8 CU Ver Historial

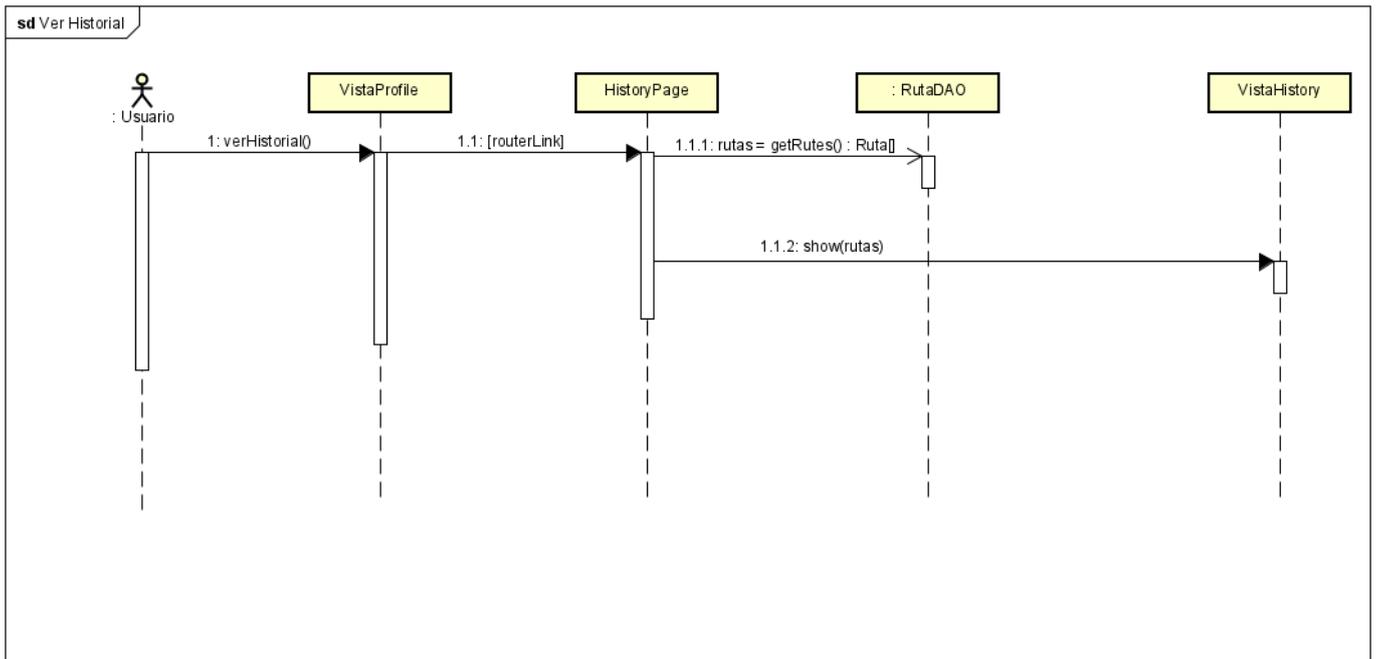


Figura 4.5.78—1 Diagrama de secuencia del caso de uso ver historial

4.5.9 CU Ruta por búsqueda

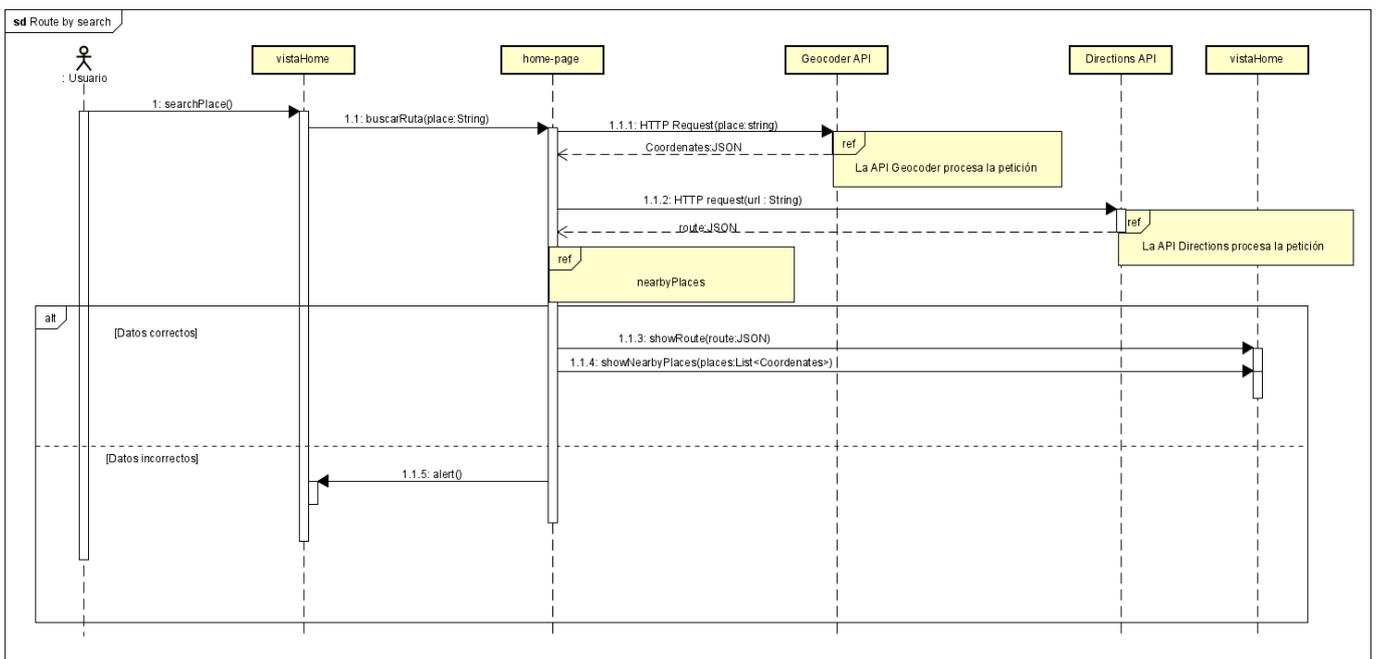


Figura 4.5.89—1 Diagrama de secuencia del caso de uso ruta por búsqueda

4.5.10 Diagrama de secuencia de algoritmo de sitios cercanos

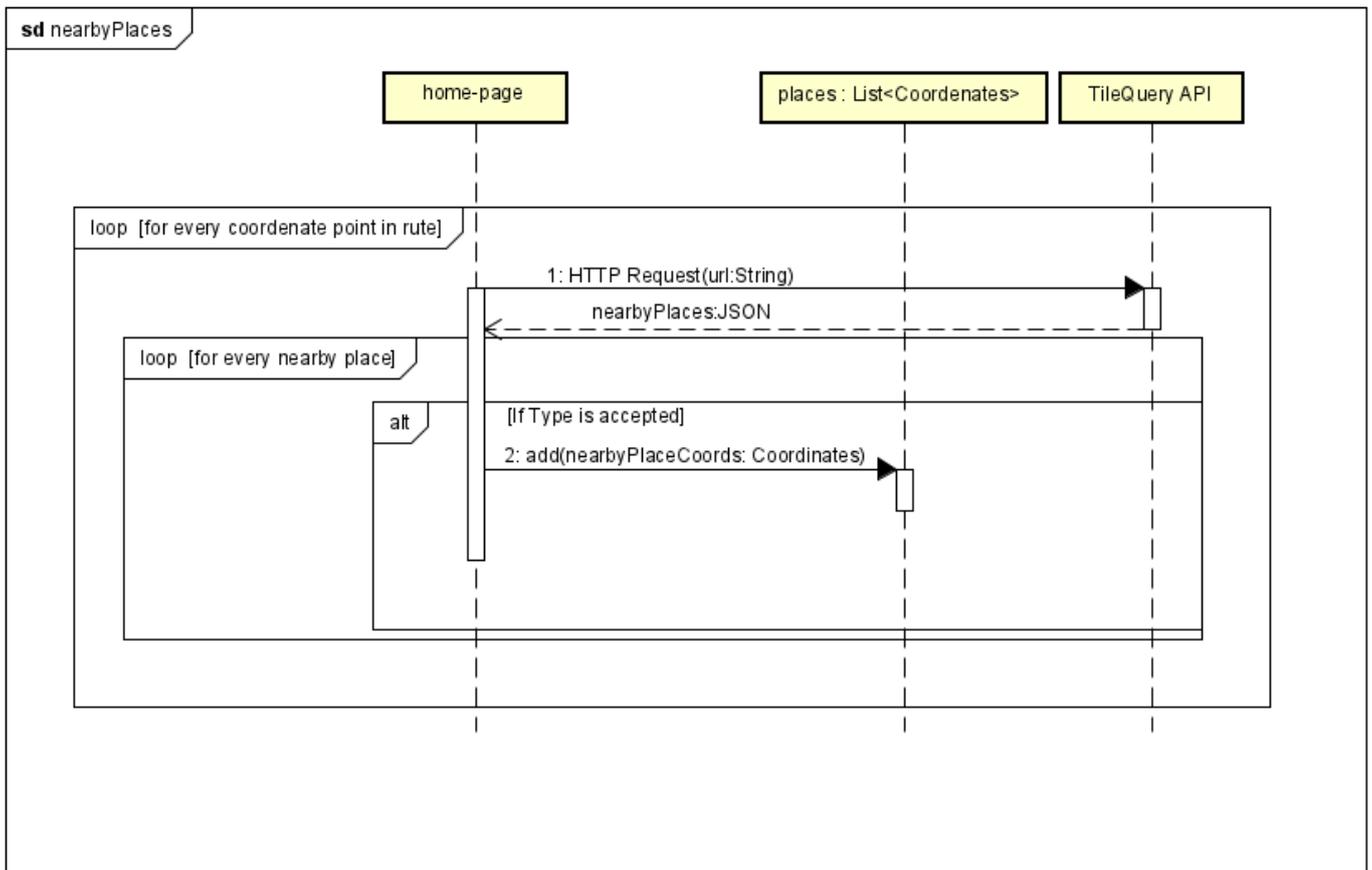


Figura 4.5.910—1 Diagrama de la búsqueda de lugares cercanos

Capítulo 5 Diseño

5.1 Arquitectura

El proyecto se estructura a muy alto nivel en los siguientes paquetes.

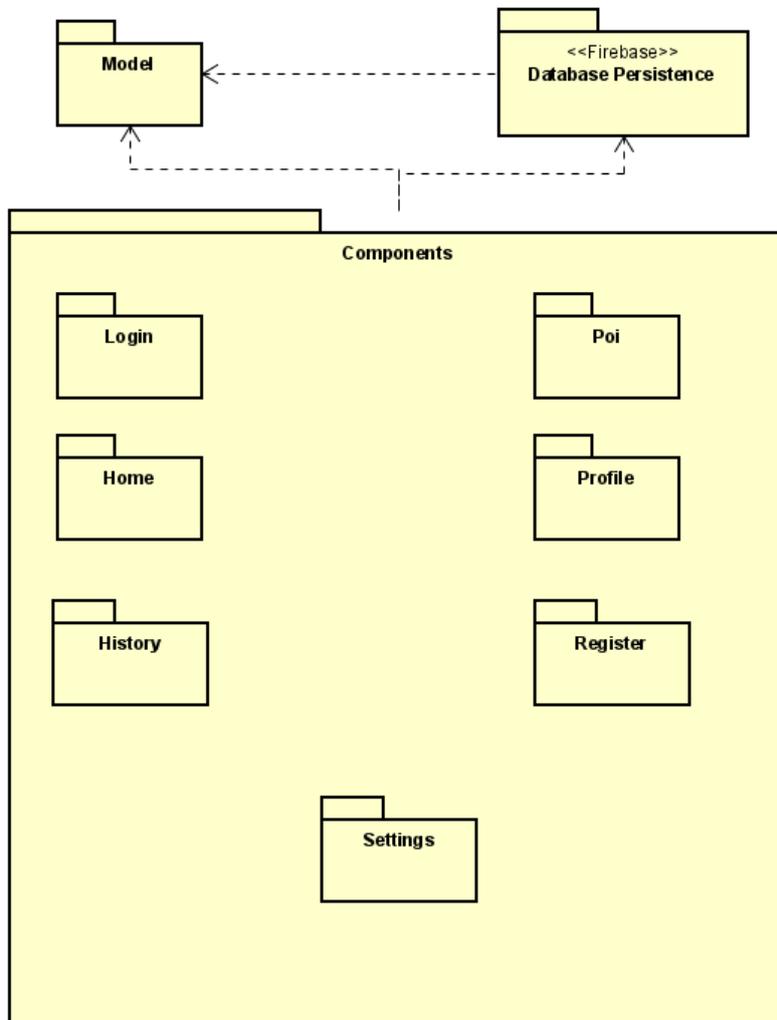


Figura 5.1—1 Diagrama de paquetes de alto nivel de la aplicación

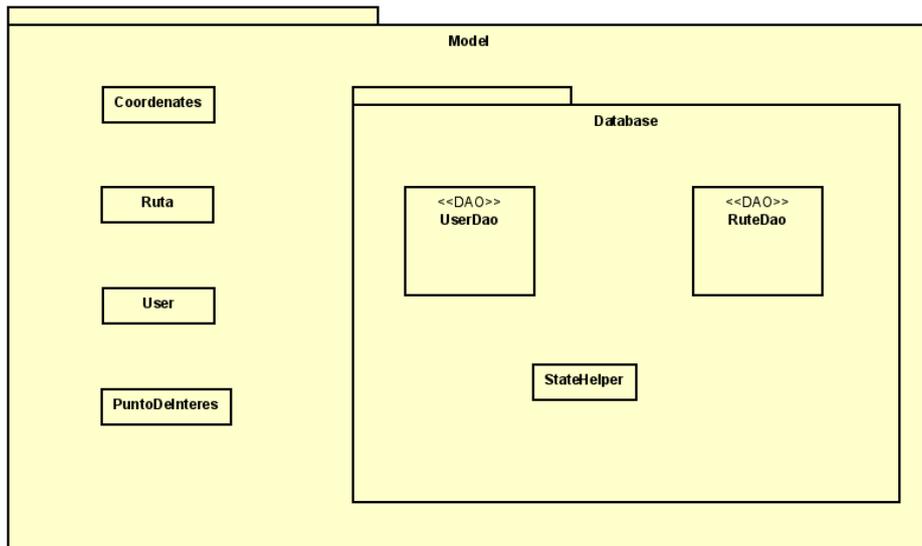


Figura 5.1—2 Diagrama de clases de model

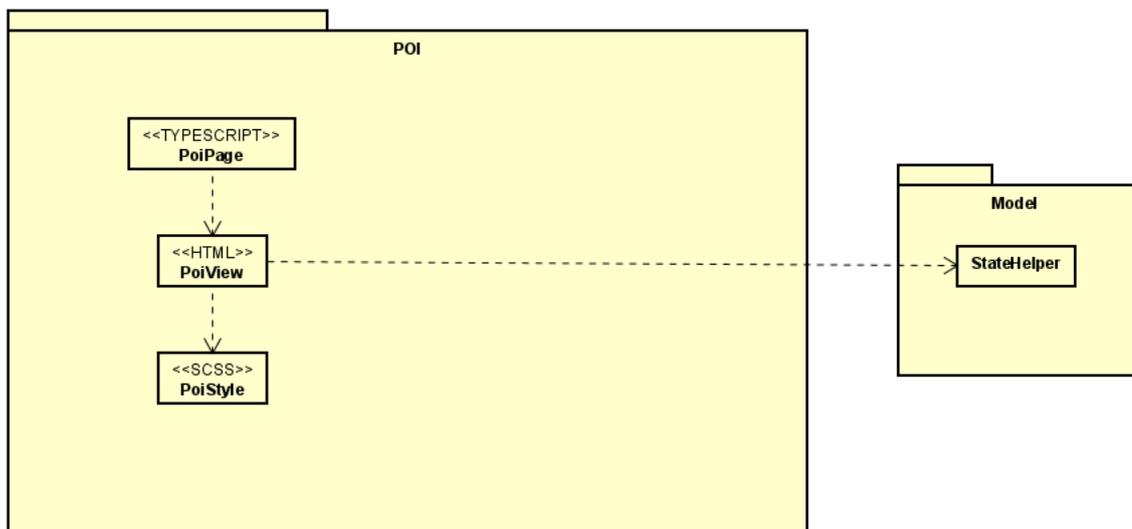


Figura 5.1—3 Diagrama de clases de POI

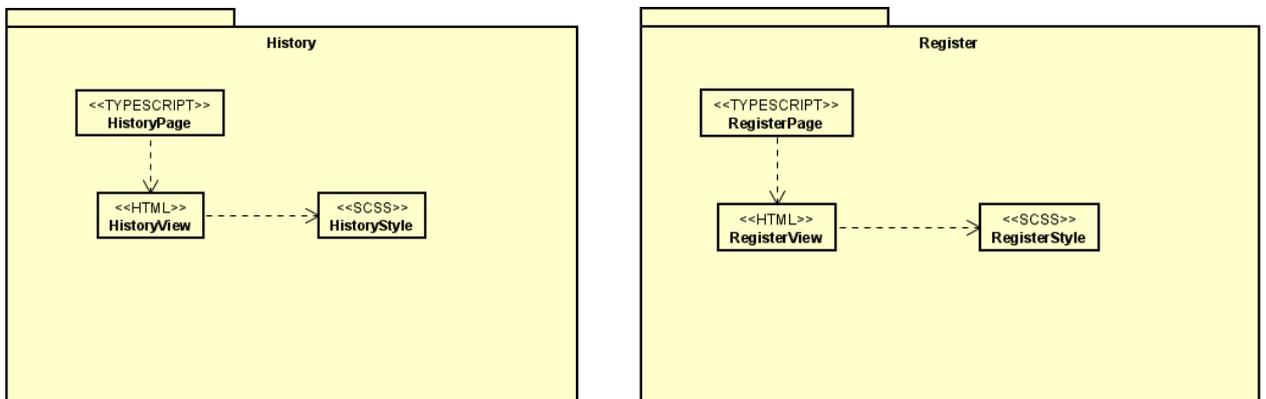


Figura 5.1—4 Diagrama de clases de history y register

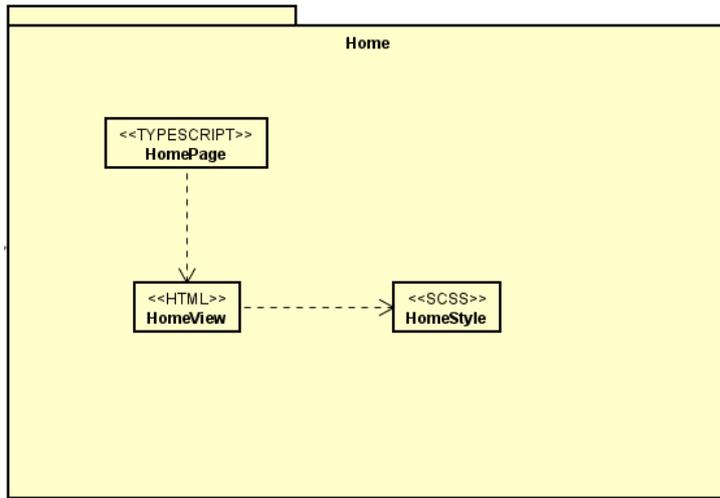


Figura 5.1—5 Diagrama de clases de home

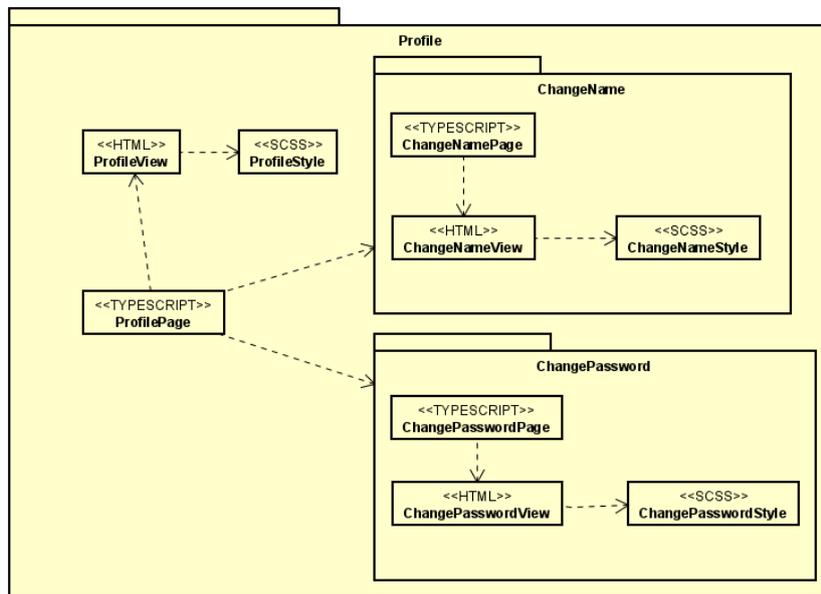


Figura 5.1—6 Diagrama de clases de profile

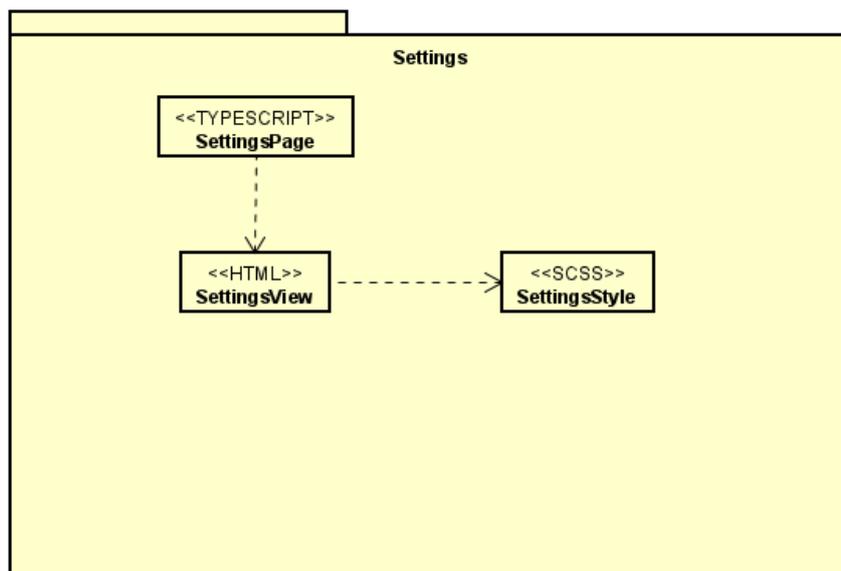


Figura 5.1—7 Diagrama de clases de settings

5.2 Patrones Utilizados

5.2.1 M.V.P. (Modelo Vista Presentador)

El patrón MVP, al igual que el MVC del que deriva, consiste en la separación de datos y principalmente las vistas de la lógica de negocio de la aplicación y del gestor de eventos y comunicaciones. Para ello se sirve de tres componentes:

- **Modelo:** es la representación de la información que contiene y manejará la aplicación. Es la capa de la lógica de negocio.
- **Vista:** capa encargada de mostrar los datos e información que contiene el modelo al usuario.
- **Presentador:** capa que asume toda la lógica de presentación, es el que comunica a la vista y al modelo, la vista no tiene conocimiento de que el modelo existe.

En el modelo MVC, la vista tiende a tener más lógica, en cambio, en el modelo MVP, esa lógica se encuentra en el presentador, siendo la única función de la vista representar la información que el presentador le proporciona.

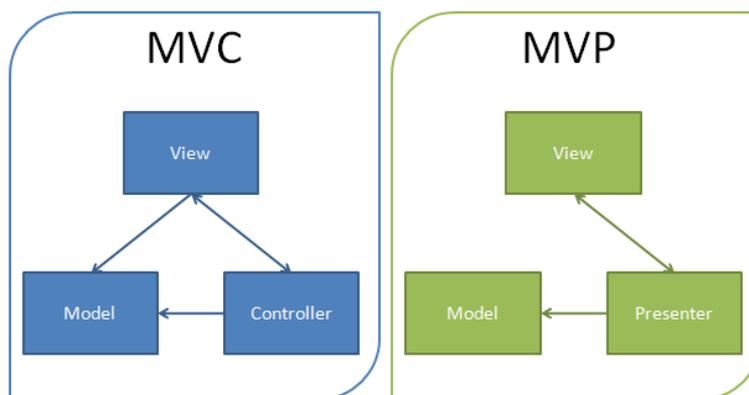


Figura 5.2.1—1 Patrón mvc vs mvp

A continuación, se muestra un sencillo ejemplo de este patrón en la app:

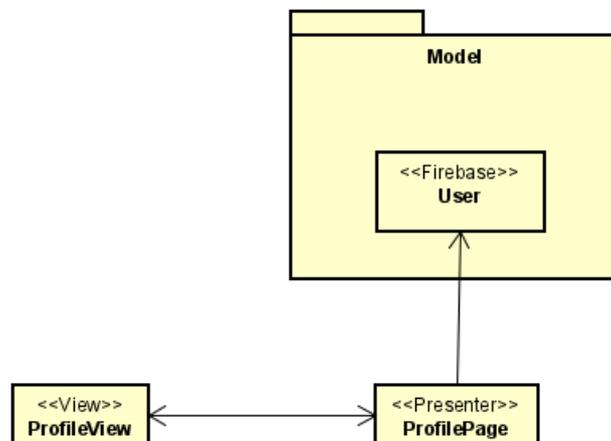


Figura 5.2.1—2 Ejemplo patrón mvp de la aplicación

5.2.2 D.A.O. (Data Access Object)

Este patrón consiste en abstraer y unificar la forma de acceso a la base de datos para ser lo más independiente posible del código y así no dependa de este, lo que supone un mayor grado de modularidad del software.

El patrón se ha empleado para realizar operaciones CRUD (Create, Read, Update y Delete) contra la base de datos Firebase donde se almacena la información correspondiente a usuarios, rutas y sitios de interés recomendados (POI).

En el siguiente diagrama se muestra el patrón D.A.O. tal y como es utilizado en nuestra aplicación.

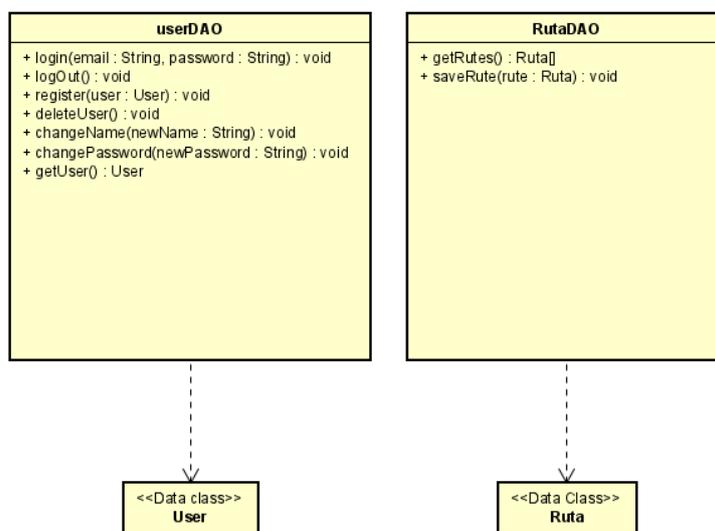


Figura 5.2.2—1 Patrón DAO en la aplicación

5.3 Esquema base de datos

Se ha decidido utilizar Firebase como base de datos para este proyecto, en concreto Firebase firestore ya que es más compleja que Firebase Realtime Database. Se trata de una base de datos no relacional, lo que ha supuesto algún que otro problema al realizar el diseño de los diagramas de esta. A su vez nos hemos apoyado en las funcionalidades de Firebase como Firebase authentication, que aporta una gestión de usuarios básica. Esto supone que no sea necesario almacenar la contraseña del usuario en base de datos ya que esta funcionalidad se encarga de su gestión. A continuación, se expondrán los siguientes esquemas no relacionales basados en documentos y colecciones correspondientes a la base de datos Firebase.

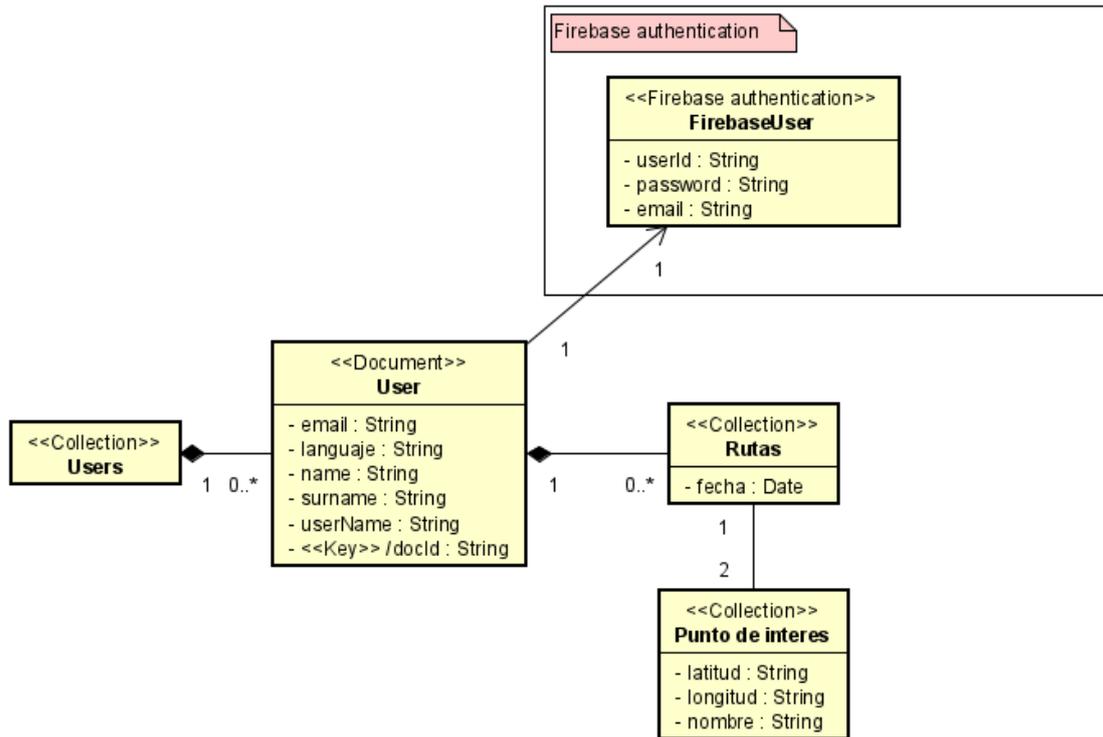


Figura 5.3—1 Esquema no relacional basado en elementos y colecciones en lo relativo a usuarios

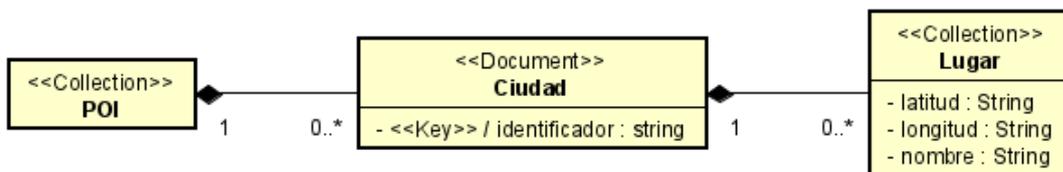


Figura 5.3—2 Esquema no relacional basado en elementos y colecciones en lo relativo a puntos de interés recomendados por la aplicación

5.4 Diagrama de despliegue

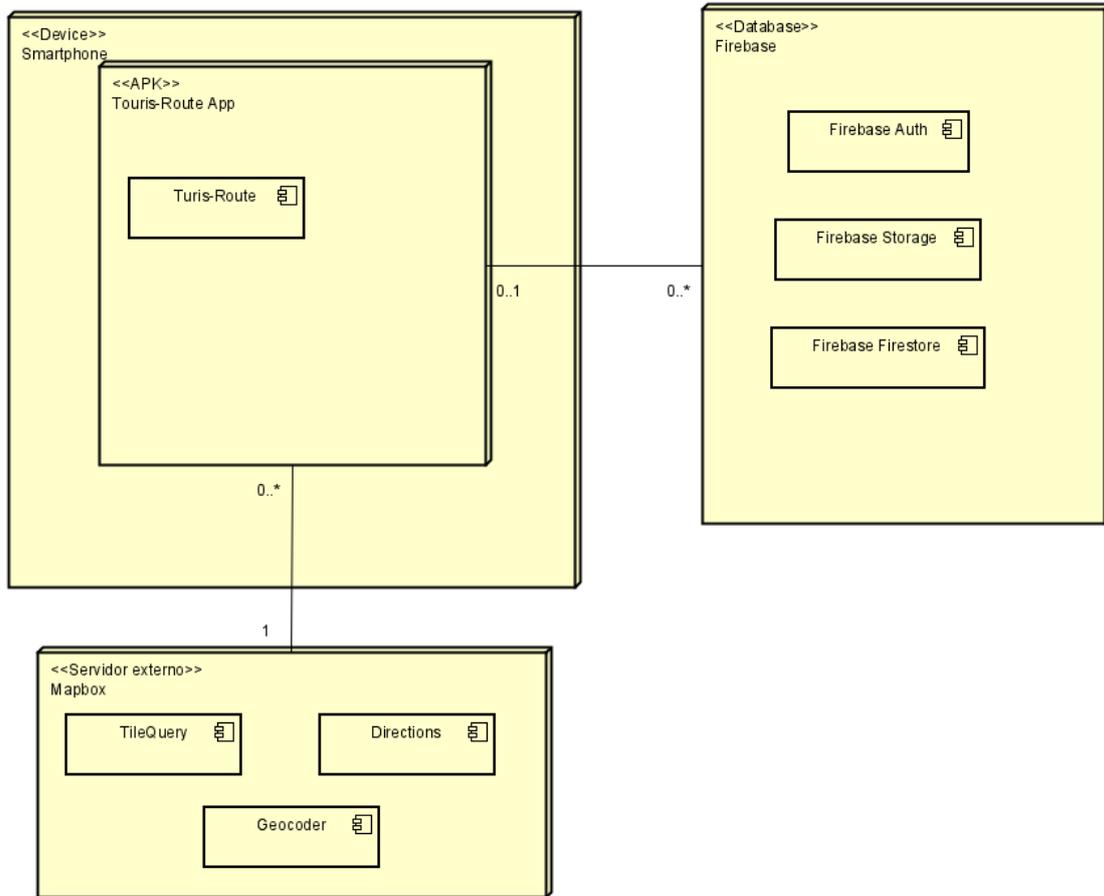


Figura 5.4—1 Diagrama de despliegue de la aplicación

Capítulo 6 Implementación

6.1 Algoritmos propios

6.1.1 Algoritmo para búsqueda de sitios de interés cercanos

Para comenzar, es necesario recopilar y aclarar un poco de información sobre mapbox y como se diseñan las rutas. La ruta se genera a partir de una llamada a una API de mapbox que devuelve la ruta entre los puntos que se desee, dentro de la llamada a la API se pueden añadir distintos parámetros:

- Profile: se trata del modo de transporte que se va a utilizar para realizar la ruta, en nuestro caso será andando ya que está pensada para rutas cortas, pero admite vehículos a motor, bicicleta y una opción para vehículos a motor teniendo en cuenta el tráfico
- Coordinates: son las coordenadas entre las que se va a generar la ruta, es decir, un origen y un destino. A su vez, se pueden añadir más de dos coordenadas si la ruta ha de pasar por algún lugar específico de camino al último destino.
- Opcionales: además cuenta con una serie de parámetros opcionales para la query, como velocidad máxima de la ruta, distintos tipos de waypoints o paradas intermedias etc.

Ahora, con un poco más de información sobre Mapbox y como genera las rutas, se conoce que una ruta se divide en múltiples puntos intermedios, en los cuales se producen giros, cambios de calle y demás reajustes en la dirección para poder llegar a tu destino correctamente y de una forma eficiente. El algoritmo desarrollado para encontrar puntos cercanos de interés a la ruta se genera a partir de estos puntos o waypoints.

Mapbox tiene disponible adicionalmente una API llamada Tilequery, que a partir de una coordenada busca los puntos de interés cercanos a esta. Además, admite una serie de parámetros extra para perfilar los resultados:

- Radius: La distancia aproximada en metros a la que como máximo se buscarán los sitios de interés.
- Limit: Número máximo de puntos de interés que se devuelven en el resultado.
- Geometry: Devuelve solo un único tipo de geometría, esto se utiliza para luego mostrar los resultados en el mapa, aunque en nuestro caso no especificaremos ninguna.
- Layers: Se puede especificar la capa o capas en las que se quiere realizar la búsqueda.

Es necesario decir, que estos resultados devuelven lugares de interés que no tienen relevancia cultural o histórica, como es el caso de bares, restaurantes, hoteles, etc. Por ello se filtrarán los resultados para obtener exactamente el tipo de lugar deseado.

A continuación, se muestra el pseudocódigo del algoritmo utilizado para mostrar los lugares de interés con relevancia cultural cercanos:

```
Para cada waypoint de la ruta {  
    Request(urlPeticiónAPITileQuery)  
    Cuando la Request se resuelve Then {  
        Para cada lugar de la respuesta {  
            Si (tipo = Theatre or church or Place of Worship or  
                Monument or University or Museum or Memorial or  
                ViewPoint or Attraction){  
                puntosDeInteres.add(lugar)  
            }  
        }  
    }  
}
```

Al final el conjunto de coordenadas relevantes se almacenará en el array puntos de interés para así mostrarse más adelante en el mapa. Por motivos de eficiencia y optimización, cuando se realiza una búsqueda a un lugar que esté a una distancia mayor de 3 km en línea recta, este algoritmo no se aplica, ya que llevaría tiempos de carga muy altos. La distancia entre ambos puntos se calcula con la fórmula de Haversine con el radio medio de la tierra: 6371 km, que es la fórmula general para determinar distancias en superficies esféricas. La llamada a la API TileQuery se formula con los parámetros adicionales “radius” y “limit”, para limitar el radio en el que se buscan los sitios de interés y el número de sitios que devuelve en el resultado para agilizar el algoritmo.

6.2 Dependencias

La aplicación cuenta con una serie de servicios o dependencias detalladas a continuación:

- **Firebase Authentication:** es un servicio que ofrece Firebase para la gestión de usuarios básica con un sistema de sesiones ya implementado.
- **Firebase Firestore:** es una base de datos no relacional basada en documentos. Al igual que todos los servicios de Firebase están siempre disponibles sin necesidad de preocuparnos por hostearlos.
- **Mapbox directions:** API cuyas consultas devuelven instrucciones para generar una ruta.
- **Mapbox tilequery:** API cuyas consultas devuelven una serie de lugares dadas unas coordenadas. Los lugares pueden ser bares, museos, parques, etc.
- **TranslateService:** es un paquete de terceros de código abierto para Angular, este simplifica el proceso de traducciones acudiendo a ficheros JSON con las traducciones a cada idioma.

6.3 Herramientas utilizadas en el desarrollo del proyecto

A lo largo de este proyecto se han utilizado una serie de herramientas que han ayudado al completo desarrollo de la aplicación Turis-Route. A continuación, vamos a explicar y detallar cada uno de ellos, dejando claro su objetivo y funcionalidad, así como las ventajas o desventajas que presentan.

6.3.1 Firebase

Firebase es una plataforma de Google en la nube para agilizar el desarrollo de aplicaciones web y móvil. Para esta agilización posee una serie de herramientas para desarrollo, crecimiento, monetización y análisis.



Figura 6.3.1—1 Logotipo de Firebase de Google

Para el desarrollo de este proyecto se han utilizado dos servicios de Firebase, por un lado, un servicio de autenticación de usuarios, que facilita la gestión general de estos, por otro lado, se ha utilizado la base de datos en la nube que posee Firebase, llamada FireStore o Cloud Storage. A continuación, se explica detalladamente cada una de estas:

- **Firestore Authentication:** Es una potente herramienta de Firebase que agiliza la gestión de usuarios evitando dedicar mucho tiempo a desarrollar el Backend de control de los mismos. Facilita los métodos y funciones básicas para el control de estos además de servicios de terceros como acceso rápido con redes sociales o Google, GitHub, etc. Posee a mayores un servicio de correo electrónico de confirmación de e-mail para nuevos usuarios.
- **Cloud Firestore:** es la base de datos más reciente de Firebase para el desarrollo de apps para dispositivos móviles. Cuenta con un modelo de datos nuevo e intuitivo pudiendo realizar consultas muy completas y rápidas. Se trata de una base de datos NoSQL para todo tipo de aplicaciones y entornos, que además funciona en tiempo real. Se ha utilizado en el proyecto para guardar datos adicionales de cada usuario y lugares de interés sugeridos por la app para realizar visitas a ciudades.

Entre las ventajas de este servicio se encuentran:

- La documentación es muy amplia y elaborada. La propia web de Firebase ofrece la documentación completa con ejemplos para ayudar a los desarrolladores.
- Ofrece un alto rendimiento debido a la escasa modificación del DOM
- Ofrece soporte para aplicaciones web, Android e iOS.

A pesar de todas estas ventajas, Firebase también posee sus contras como el complejo tratamiento de datos para acceder a la información, sobre todo cuando se trata de una estructura de datos grande y compleja. Por otro lado, existe un límite de peticiones para las cuentas que no sean de pago, en caso de que se popularice la aplicación será necesario pensar en un plan de pago que soporte las peticiones que se realicen.

6.3.2 Ionic

Ionic es un framework especializado en el desarrollo de aplicaciones híbridas basadas en HTML5, CSS3 y JavaScript o TypeScript. Las aplicaciones híbridas se llaman así porque combinan aspectos de las aplicaciones nativas y de las aplicaciones web según convenga.



Figura 6.3.2—1 Logotipo del framework Ionic

Las principales ventajas de este framework son:

- Alto rendimiento, ya que está diseñado para ser rápido gracias a la mínima manipulación del Dom y aceleraciones de transición por hardware.
- Uso de Angular para crear aplicaciones más ricas y robustas.
- Ionic posee un centro nativo, por lo que es fácil de entender para cualquier persona que haya desarrollado una aplicación nativa para Android o IOS.
- No es necesario modificar el código para producir aplicaciones Android, IOS o WEB.

Pero también posee desventajas:

- El rendimiento puede ser ligeramente menor que en aplicaciones nativas, aunque no es un problema siempre que no se trate de aplicaciones que requieran grandes cantidades de recursos.
- Al ser muy reciente, puede ser difícil encontrar feedback por parte de otros usuarios.
- Actualizaciones continuas debido a la juventud del framework, que hace que los usuarios tengan que estar actualizándose continuamente.

6.3.3 Mapbox

Mapbox es un proveedor de mapas en línea de código abierto. Los datos que utiliza Mapbox son extraídos tanto de bases de datos abiertas como OpenStreetMap, la Nasa y de propietarios privados.



Figura 6.3.3—1 Logotipo de Mapbox

A continuación, se muestran las principales ventajas de este proveedor de mapas:

- Todo es 100% personalizable desde cualquier color hasta poder crear tus propios estilos.
- Al ser de código abierto permite ver todas las funcionalidades que están en desarrollo, reportar cualquier problema o contribuir al desarrollo de Mapbox.

- Existen SDKs disponibles para prácticamente todas las plataformas.
- Fácil implementación de mapas básicos o servicios principales.

Como principal desventaja encontramos, al igual que Firebase, el límite de peticiones que se pueden realizar con la cuenta gratuita.

6.3.4 Angular

Angular es un framework opensource desarrollado por Google que facilita la creación y programación de aplicaciones web. Una de sus características principales es que separa totalmente frontend de backend, evitando escribir código repetitivo y favoreciendo el patrón MVC para aumentar la rapidez del desarrollo, haciendo la aplicación más modulable facilitando actualizaciones o modificaciones.



Figura 6.3.4—1 Logotipo del framework Angular

Mostraremos sus principales ventajas:

- Favorece la escalabilidad y modularidad.
- Las páginas se cargan al principio completas, haciendo que la navegación sea totalmente instantánea.
- Al usarse junto a Ionic facilita la creación de páginas web-responsive, es decir, adaptadas para uso en varios dispositivos.

Desventajas:

- Necesidad de una organización y estándares para evitar la falta de comprensión de la estructura del proyecto.
- Documentación en ocasiones no todo lo completa que podría ser.
- Problemas al migrar la aplicación ya que la mayoría de las funcionalidades va a estar muy integrada con clases que dependen de angular.

Capítulo 7 Pruebas

A continuación, se describirá en detalle el conjunto de pruebas desarrolladas para probar el funcionamiento de la aplicación. Son pruebas de caja negra guiadas por los casos de uso sobre todas las funcionalidades desarrolladas. Las pruebas serán en dispositivos Android y en navegadores web, a pesar de que el código es perfectamente compilable en iOS sin necesidad de ninguna modificación. Esto se debe, en un principio, a escasez de tiempo y, por otro lado, a la dificultad añadida de instalar software de terceros en un dispositivo iOS que supondría tener que hacer Jailbreak al dispositivo o pagar para subir la aplicación al store de Apple. Se entrará más en detalle en aquellas pruebas que han servido para detectar bugs o malfuncionamientos en la aplicación.

7.1 Casos de Prueba

CP-01	Registrar un usuario
Descripción	Crear un usuario en la base de datos con los datos personales suministrados
Pasos	Desde la ventana de login pulsar el botón registrarse, en la nueva ventana de registro introducir los datos especificados y pulsar el botón registrarse.
Resultado esperado	La base de datos se actualizará con el nuevo usuario creado, a su vez se iniciará sesión automáticamente y se lanzará la ventana home

Tabla 7.1—1 Caso de prueba registro de usuario

CP-02	Iniciar Sesión
Descripción	Iniciar sesión con el correo y contraseña pertenecientes a un usuario registrado
Pasos	Desde la ventana de login introducir los datos del usuario especificados y pulsar el botón iniciar sesión

Resultado esperado	Se iniciará sesión en la aplicación y se lanzará la ventana home
---------------------------	--

Tabla 7.1—2 Caso de prueba iniciar sesión

CP-03	Cerrar sesión
Descripción	Cerrar la sesión activa de un usuario en la aplicación
Pasos	Desde la ventana Home, pulsar el botón de perfil en la esquina superior derecha y a continuación pulsar el botón “cerrar sesión”.
Resultado esperado	La sesión del usuario se cerrará no permitiéndole acceder a las ventanas de la aplicación excepto a login y registro. Se lanzará la ventana login una vez pulsado el botón.

Tabla 7.1—3 Caso de prueba cerrar sesión

CP-04	Baja del servicio
Descripción	Borrar la cuenta de un usuario
Pasos	Desde la ventana Home, pulsar el botón de perfil en la esquina superior derecha y a continuación pulsar el botón “baja del servicio”.
Resultado esperado	El usuario es borrado de la base de datos y la aplicación sale de la sesión y se dirige a la ventana de login.

Tabla 7.1—4 Caso de prueba baja del servicio

CP-05	Crear ruta mediante búsqueda
Descripción	Creación de una ruta entre la posición del usuario y un destino buscado en el buscador de la aplicación en la ventana home

Turis-Route: Aplicación para la generación de rutas

Pasos	Desde la ventana home, pulsar en el buscador y escribir el lugar deseado como destino.
Resultado esperado	En la ventana home, se mostrará una ruta al sitio destino desde la ubicación del usuario mostrando lugares de interés cercanos a la ruta.

Tabla 7.1—5 Caso de prueba crear ruta mediante búsqueda

CP-06	Crear ruta mediante sugerencias
Descripción	Creación de una ruta a un destino con relevancia cultural sugerido por la aplicación.
Pasos	Desde la ventana Home, pulsar el botón de lugares en la esquina superior derecha y a continuación seleccionar el lugar sugerido que se desea visitar de la lista de lugares sugeridos. Cuando se sepa que lugar visitar, pulsar en el botón azul en la esquina inferior derecha para crear la ruta desde la ubicación del usuario hasta el destino seleccionado.
Resultado esperado	Creación de una ruta rápida entre la posición del usuario y el destino sugerido que haya seleccionado el usuario.

Tabla 7.1—6 Caso de prueba crear ruta mediante sugerencias

CP-07	Ver historial de lugares visitados
Descripción	Ver el historial de lugares visitados por el usuario.
Pasos	Desde la ventana Home, pulsar el botón de perfil en la esquina superior derecha, en la nueva ventana pulsar el botón “ver historial”.

Turis-Route: Aplicación para la generación de rutas

Resultado esperado	Lanzamiento de una ventana que muestre los lugares visitados por el usuario desde que se creó la cuenta.
---------------------------	--

Tabla 7.1—7 Caso de prueba ver historial de lugares visitados

CP-08	Conocer la ubicación del usuario.
Descripción	Representar las coordenadas del usuario en el mapa para que sean visibles.
Pasos	Desde la ventana Home, pulsar el botón de posición en el mapa en la esquina superior derecha.
Resultado esperado	Aparición de un punto azul de localización sobre el mapa en la posición del usuario.

Tabla 7.1—8 Caso de prueba conocer la ubicación del usuario

CP-09	Controles de mapa (zoom y desplazamiento)
Descripción	Movimiento del usuario por el mapa de la ventana home.
Pasos	Desde la ventana Home, uso de los controles en la esquina superior para acercar y alejar el mapa, uso del rato o de la pantalla táctil del móvil para desplazarse por el mapa.
Resultado esperado	El mapa responde al zoom y a los desplazamientos del usuario por el mapa.

Tabla 7.1—9 Caso de prueba controles de mapa

CP-10	Cambio de nombre de usuario
Descripción	Cambio de nombre del usuario con sesión iniciada en la app

Pasos	Desde la ventana Home, pulsar el botón de perfil en la parte superior derecha, a continuación, pulsar el botón cambiar nombre de usuario e introducir el nuevo nombre deseado
Resultado esperado	Cambio del nombre de usuario en la base de datos, apreciable en el perfil del usuario

Tabla 7.1—10 Caso de prueba cambio de nombre

CP-11	Cambio de contraseña del usuario
Descripción	Cambio de contraseña del usuario con sesión iniciada en la app
Pasos	Desde la ventana Home, pulsar el botón de perfil en la parte superior derecha, a continuación, pulsar el botón cambiar contraseña, introducir la contraseña actual y la nueva contraseña a la que se desea cambiar.
Resultado esperado	Cambio de contraseña del usuario en la base de datos. Necesitará introducir esta nueva contraseña para iniciar sesión la próxima vez.

Tabla 7.1—11 Caso de prueba cambio de contraseña del usuario

7.2 Criterios de evaluación

Para poder evaluar y valorar los resultados de las pruebas detalladas en el punto anterior, se ha seguido una serie de criterios para poder llegar a una nota entre 0 y 10 sobre el resultado de la prueba.

En la siguiente tabla se exponen los criterios utilizados:

Nota	Criterio	Estado
0-4	Los resultados de la prueba no coinciden con los esperados	Suspense

5-6	La prueba concluye con los resultados esperados, pero quebrantando requisitos de usabilidad de la aplicación (retrasos, seguridad, falta de eficiencia)	Aprobado
7-8	La prueba concluye con los resultados esperados y cumpliendo la mayoría de los requisitos de usabilidad.	Notable
9-10	La prueba concluye con los resultados esperados, cumpliendo todos los requisitos de usabilidad, siendo un caso de uso visual, intuitivo y eficiente	Sobresaliente

Tabla 7.2—1 Tabla criterios de evaluación

7.3 Resultados en Android

A continuación, se representan los resultados obtenidos de las pruebas anteriores de la aplicación ejecutándose sobre un dispositivo Android.

Caso	Resultado	Solución
CP-01	10/10	-
CP-02	10/10	-
CP-03	10/10	-
CP-04	10/10	-
CP-05	6/10: Se demora demasiado en buscar lugares alejados del usuario.	Fijar una distancia máxima a la que trazar rutas para evitar tiempos de demora demasiado altos para crear la ruta
CP-06	10/10	-
CP-07	10/10	-

CP-08	10/10	
CP-09	10/10	
CP-10	10/10	-
CP-11	10/10	-

Tabla 7.3—1 Resultados en Android

7.4 Resultados en navegador web

Se ha procedido a realizar las pruebas sobre varios navegadores web para comprobar el correcto funcionamiento de la aplicación independientemente del navegador utilizado. Los resultados a continuación se han extraído de pruebas en un navegador web, en concreto Google Chrome.

Caso	Resultado	Solución
CP-01	10/10	-
CP-02	10/10	-
CP-03	10/10	-
CP-04	10/10	-
CP-05	8/10: Se demora en buscar lugares alejados del usuario.	Fijar una distancia máxima a la que trazar rutas para evitar tiempos de demora demasiado altos para crear la ruta
CP-06	10/10	-
CP-07	10/10	-
CP-08	6/10: No es un problema al utilizarlo en un dispositivo móvil, sin embargo, si supone falta de precisión realizar el caso	La solución no parece residir en el ámbito de la aplicación, sino en la de

Turis-Route: Aplicación para la generación de rutas

	de prueba desde un ordenador, siendo esta variante y bastante imprecisa.	los sistemas de ubicación y posicionamientos de los ordenadores de sobremesa.
CP-09	10/10	-
CP-10	10/10	-
CP-11	10/10	-

Tabla 7.4—1 Resultados en Chrome

Los siguientes resultados se corresponden con el navegador web Microsoft Edge. Es importante tener en cuenta que su predecesor, Internet Explorer, no soporta funciones asíncronas por lo que el funcionamiento de la app se puede ver comprometido.

Caso	Resultado	Solución
CP-01	10/10	-
CP-02	10/10	-
CP-03	10/10	-
CP-04	10/10	-
CP-05	8/10: Se demora en buscar lugares alejados del usuario.	Fijar una distancia máxima a la que trazar rutas para evitar tiempos de demora demasiado altos para crear la ruta
CP-06	10/10	-
CP-07	10/10	-
CP-08	6/10: No es un problema al utilizarlo en un dispositivo móvil, sin embargo, si	La solución no parece residir en el ámbito de la

Turis-Route: Aplicación para la generación de rutas

	supone falta de precisión realizar el caso de prueba desde un ordenador, siendo esta variante y bastante imprecisa.	aplicación, sino en la de los sistemas de ubicación y posicionamientos de los ordenadores de sobremesa.
CP-09	10/10	
CP-10	10/10	-
CP-11	10/10	-

Tabla 7.4—2 Resultados en Microsoft Edge

Por último, se han realizado los casos de prueba en el navegador Mozilla Firefox.

Caso	Resultado	Solución
CP-01	10/10	-
CP-02	10/10	-
CP-03	10/10	-
CP-04	10/10	-
CP-05	8/10: Se demora en buscar lugares alejados del usuario.	Fijar una distancia máxima a la que trazar rutas para evitar tiempos de demora demasiado altos para crear la ruta
CP-06	10/10	-
CP-07	10/10	-
CP-08	6/10: No es un problema al utilizarlo en un dispositivo móvil, sin embargo, si supone falta de precisión realizar el caso	La solución no parece residir en el ámbito de la aplicación, sino en la de

Turis-Route: Aplicación para la generación de rutas

	de prueba desde un ordenador, siendo esta variante y bastante imprecisa.	los sistemas de ubicación y posicionamientos de los ordenadores de sobremesa.
CP-09	10/10	
CP-10	10/10	-
CP-11	10/10	-

Tabla 7.4—3 Resultados en Mozilla Firefox

WEBGRAFÍA

- [1] Blog Lowi (última actualización 21 de junio de 2009). ¿Qué es Google Trips? Organiza tus vacaciones en tu móvil (Última consulta 15 de octubre de 2020) [En Línea]. Disponible en: <https://www.lowi.es/blog/google-trips-organiza-vacaciones-en-tu-movil/>
- [2] Enrique Oriol (Última actualización 24 de junio de 2016). Ionic Vs React Native Vs NativeScript (Blog Enrique Oriol) (Última consulta 25 de octubre de 2020) [En Línea]. Disponible en: <http://blog.enriqueoriol.com/2016/06/ionic-vs-react-native-vs-nativescript.html>
- [3] Firebase. Firebase authentication oficial documentation (Última consulta 26 de octubre de 2020) [En Línea]. Disponible en: <https://firebase.google.com/docs/auth?hl=es-419>
- [4] Firebase (Última actualización 6 de enero de 2021). Firebase firestore documentación oficial (Última consulta 28 de octubre de 2020) [En Línea]. Disponible en: <https://firebase.google.com/docs/firestore?hl=es>
- [5] Google Play. TripIt (Última consulta 5 de noviembre de 2020) (Última consulta 5 de noviembre de 2020) [En Línea]. Disponible en: <https://play.google.com/store/apps/details?id=com.tripit&gl=ES>
- [6] Ionic Enterprise. Documentación oficial del framework Ionic (Última consulta 27 de noviembre de 2020) [En Línea]. Disponible en: <https://ionicframework.com/docs>
- [7] Ionic Enterprise. Documentación y cursos oficiales Ionic (Última consulta 28 de octubre de 2020) [En Línea]. Disponible en: <https://ionicframework.com/resources/learning>
- [8] Mapbox. Documentación oficial de la API Directions (Última consulta 9 de noviembre de 2020) [En Línea]. Disponible en: <https://docs.mapbox.com/api/navigation/directions/>
- [9] Mapbox. Geocoding in Mapbox (Última consulta 29 de noviembre de 2020) [En Línea]. Disponible en: <https://docs.mapbox.com/api/search/geocoding/>
- [10] Mapbox. Getting started with the Mapbox Directions API (Última consulta 9 de noviembre de 2020) [En Línea]. Disponible en: <https://docs.mapbox.com/help/tutorials/getting-started-directions-api/>
- [11] Mapbox. Mapbox GL JS Documentation (Última consulta 20 de noviembre de 2020) [En Línea]. Disponible en: <https://docs.mapbox.com/mapbox-gl-js/api/>
- [12] Mapbox. Mapbox Tilequery API documentación oficial (Última consulta 27 de diciembre de 2020) [En Línea]. Disponible en: <https://docs.mapbox.com/api/maps/tilequery/>
- [13] Mapbox. Markers and controls (Última consulta 14 de diciembre de 2020) [En Línea]. Disponible en: <https://docs.mapbox.com/mapbox-gl-js/api/markers/>

[14] MDN Web Docs (Última actualización 15 de octubre de 2020). Función Async (Última consulta 12 de enero de 2021) [En Línea]. Disponible en: https://developer.mozilla.org/es/docs/Web/JavaScript/Referencia/Sentencias/funcion_asincrona

[15] Miguh Ruiz (Última modificación 9 de agosto de 2017). Que es Firebase de Google (OpenWebinars) (Última consulta 25 de octubre de 2020) [En Línea]. Disponible en: <https://openwebinars.net/blog/que-es-firebase-de-google/>

[16] Quality Devs (Última actualización 31 de mayo de 2019). Que es Ionic y porque te interesa conocerlo si eres desarrollador web (Última consulta 25 de octubre de 2020) [En Línea]. Disponible en: <https://www.qualitydevs.com/2019/05/31/que-es-ionic-desarrollador-web/>

[17] Sites de Google. Gestión de proyectos software, tipos de riesgos (Última consulta 8 de noviembre de 2020) [En Línea]. Disponible en: <https://sites.google.com/site/gestiondeproyectossoftware/unidad-3-planificacion-de-proyecto/3-5-1-tipos-de-riesgos>

[17] Stack Overflow. Stack Overflow – Where developers learn, share and build careers (Última consulta 25 de enero de 2021) [En Línea]. Disponible en: <https://stackoverflow.com/>

[18] Trip It Enterprise. Documentación oficial de la aplicación Trip It (Última consulta 15 de octubre de 2020) [En Línea]. Disponible en: <https://www.tripit.com/web/uk/free>

[19] Wikipedia (Última vez editado, 2020, Dic 1). Metodologías de desarrollo software (Última consulta 17 de octubre de 2020) [En Línea]. Disponible en: https://es.wikipedia.org/wiki/Metodolog%C3%ADa_de_desarrollo_de_software

BIBLIOGRAFÍA

- [1] Banga, Cameron y Weinhold, Josh. Essential mobile interaction design. Perfecting Interface design in mobile apps. Editorial Addison-Wesley.
- [2] Cherny, Boris. Programming typescript. Make your typescript applications scale. Editorial O'Reilly.
- [3] Erich Gamma, Richard Helm, Ralph Johnson, John Vlissides. Design patterns: Elements of Reusable Object-Oriented Software. Editorial Addison-Wesley.
- [3] Griffith, Chris. Mobile app development with Ionic. Cross platform apps with Ionic, Angular and Cordova. Editorial O'Reilly.
- [4] Sridhar Rao Chivukula y Aki Iskandar. Web development with Angular and Bootstrap. Embrace responsive web design and build adaptative Angular web applications. Editorial Packt.
- [5] Stack Overflow contributors. Learning Firebase. Free unaffiliated eBook. Free eBook disponible en: <https://riptutorial.com/Download/firebase.pdf>

ANEXOS

Anexo 1

Manual de Instalación

En esta sección, se describirán los procesos para poner en marcha la aplicación, se hará una distinción dependiendo de qué sistema operativo se utilice. Por motivos de restricciones de tiempo solo se han hecho los manuales para Android, aunque el software es compilable para iOS y como aplicación web sin necesidad de modificar el código fuente.

Configuración de Android

Para poder instalar la aplicación, primero es necesario poder descargar la APK, en caso de que al descargarla diese un error por el que no puede descargar aplicaciones desconocidas, se deben seguir los pasos que se muestran a continuación para poder descargarla:

- Acceder a **ajustes**
- Acceder al apartado **Protección de privacidad**
- Pulsar sobre **Permisos especiales**
- Presionar la opción **Instalar aplicaciones desconocidas**
- Seleccionar la aplicación que va a descargar la APK
- Activar la opción **Autorizar descargas de esta fuente**

Una vez la APK se encuentra en el dispositivo, será necesario tener la opción para instalar aplicaciones de origen desconocido activada, por defecto esta opción viene desactivada en los smartphones Android. Para activarla es necesario seguir los siguientes pasos:

- Acceder a **ajustes**
- Acceder al apartado **Contraseña y seguridad**
- Pulsar sobre **Privacidad**
- Presionar la opción **Acceso especial a la aplicación**
- Seleccionar la aplicación que va a Instalar la APK
- Activar la opción que aparece para permitir la instalación de este tipo de aplicaciones.

En ocasiones este proceso variará un poco dependiendo del modelo de Smartphone que se utilice y de la versión del sistema operativo. También es posible que una vez descargada la APK, al intentar instalarla se abra automáticamente los ajustes para

permitir la instalación de aplicaciones de origen desconocido, por lo que esto simplificaría la tarea.

Una vez la aplicación esté instalada será necesario darle permisos de ubicación para el funcionamiento básico de la app. A continuación, se muestra como conceder este permiso:

- Acceder a **ajustes**
- Acceder al apartado **aplicaciones**
- Pulsar sobre **permisos**
- Presionar sobre el permiso de ubicación, u otros permisos en caso de que este no aparezca
- Seleccionar la aplicación Turis-Route
- Activar el permiso de ubicación para esta aplicación.

En todo momento en el caso de existir un buscador en el apartado de ajustes, se puede buscar la opción en concreto, lo cual ahorraría algunos pasos. Para que la aplicación funcione correctamente existen unos requisitos adicionales:

- Conexión a internet estable
- Tener el GPS activado

Anexo 2

Manual de usuario

Se trata de una aplicación intuitiva para encontrar el acceso a las funcionalidades de una forma rápida y familiar, para comenzar a usar la misma, es necesario disponer de una cuenta de usuario y estar iniciado sesión en la aplicación, para ello en la primera vista que se ofrece, se puede iniciar sesión o registrarse.

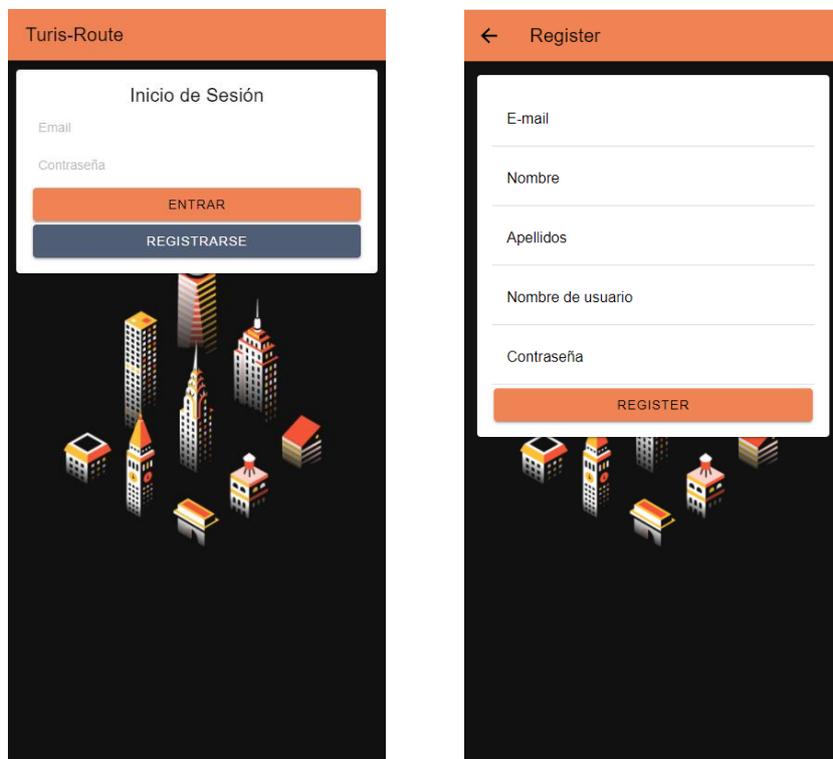


Figura Anexo II—2 Captura de pantalla de la aplicación del inicio de sesión y registro

El registro tiene ciertos requerimientos:

- El email debe de estar formado correctamente, es decir, tener un @, un dominio y .es o .com.
- Todos los campos son obligatorios y tienen que ser rellenados
- La contraseña ha de tener un mínimo de 6 caracteres

Una vez esté la sesión iniciada se visualizará la vista de la página principal, en la que se observa el mapa, con un buscador en la esquina superior izquierda, botones de control de

Turis-Route: Aplicación para la generación de rutas

mapa en la esquina superior derecha, y una barra de herramientas con las opciones puntos de interés, perfil, y ajustes, en este orden.

Una vez en esta vista, se podrá acceder a la funcionalidad principal de la aplicación, realizar rutas. Existen dos formas de realizar una ruta, la primera mediante el buscador, buscando nuestro destino la aplicación rellenará automáticamente una serie de opciones a partir de lo introducido en el buscador, una vez veamos la opción correcta pulsar sobre ella y la ruta se generará. Es necesario comentar que esta optimizada para recorridos cortos dentro de la ciudad, ya que además de generar la ruta buscará lugares cercanos que puedan interesar al usuario.

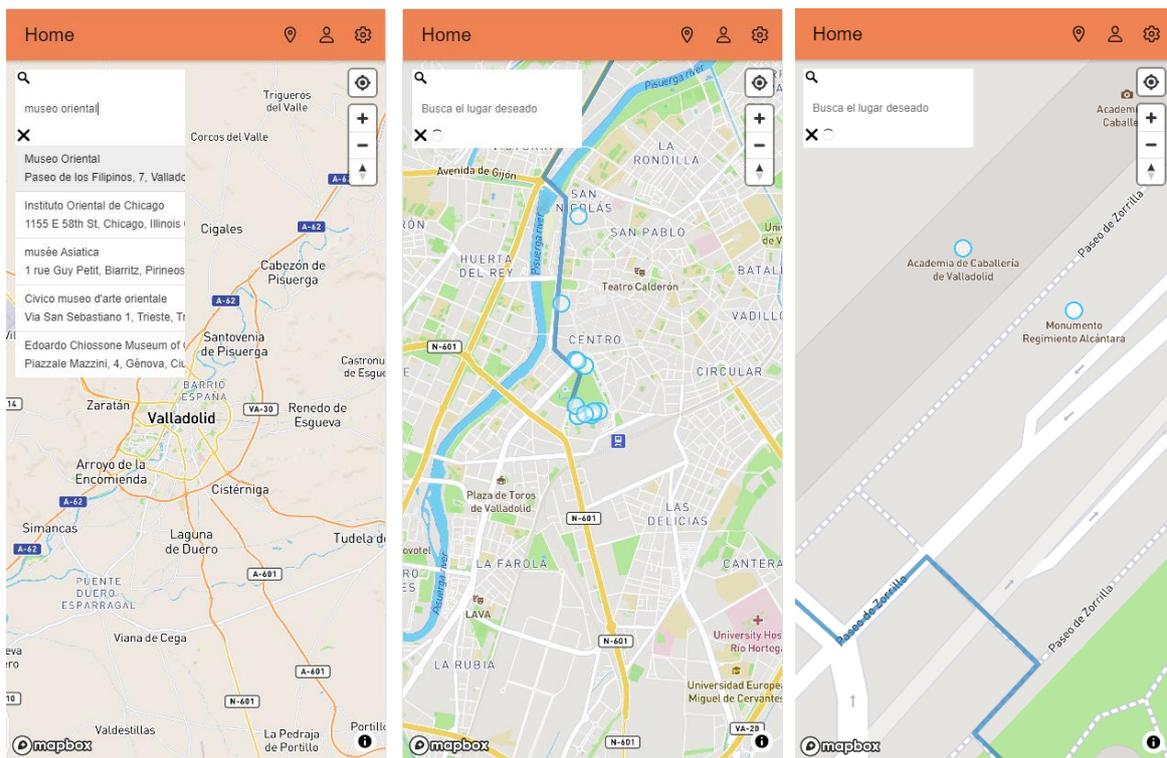


Figura Anexo II—3 Captura de pantalla de la aplicación sobre la creación de una ruta por búsqueda

Turis-Route: Aplicación para la generación de rutas

Por otro lado, la segunda forma de realizar una ruta es mediante las sugerencias de destinos posibles que te hace la aplicación, estas se encuentran en la vista de puntos de interés que se accede pulsando en el icono de ubicación en la barra de herramientas.

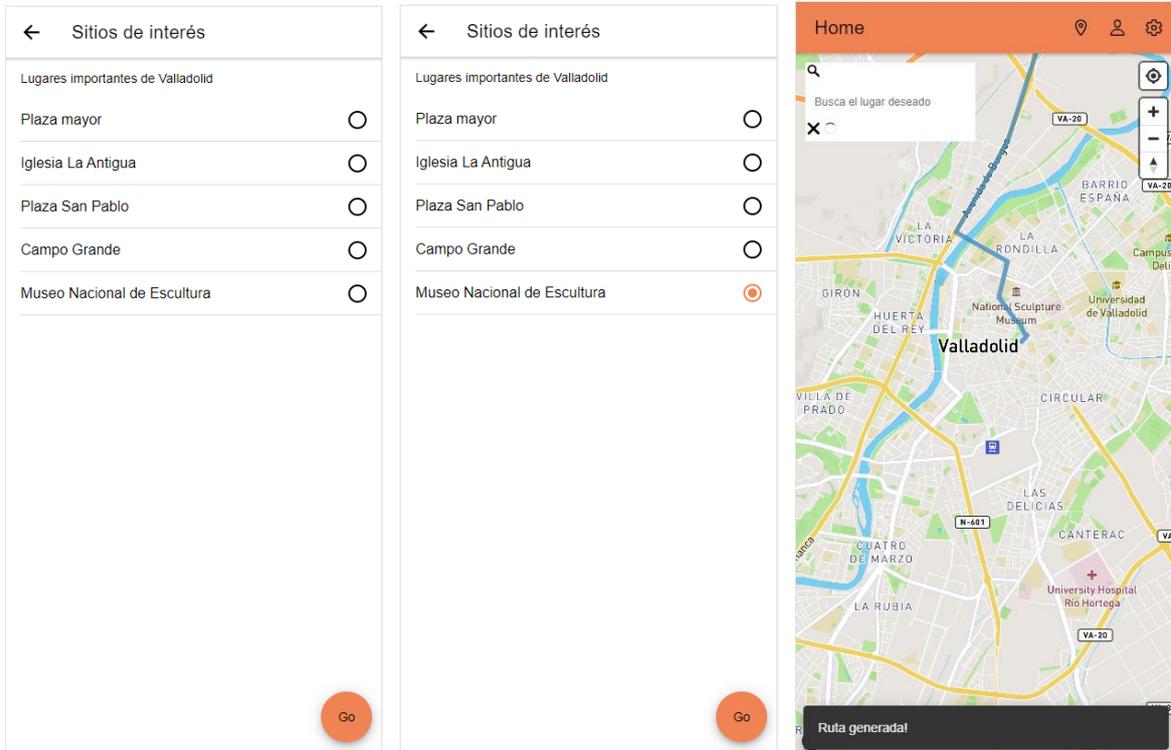


Figura Anexo II—4 Captura de pantalla de la aplicación sobre crear una ruta por sugerencias de destinos de la aplicación

Turis-Route: Aplicación para la generación de rutas

Una vez la ruta sea generada, esta se almacenará en un historial en la base de datos que podremos acceder desde la opción *historial* en la vista perfil que encontraremos pulsando el icono de perfil de la barra de herramientas de la página principal.



Figura Anexo II—5 Captura de pantalla de la aplicación sobre el perfil

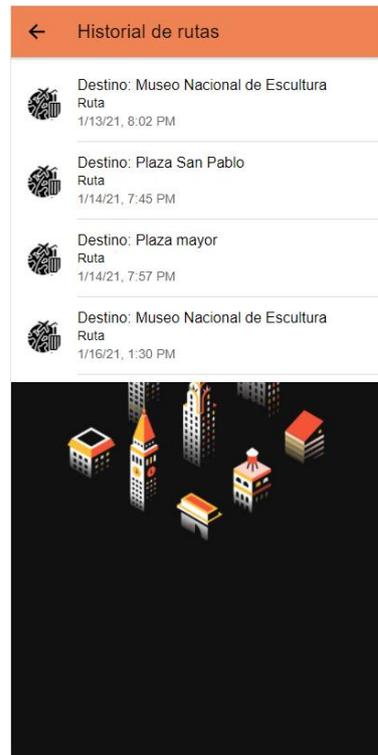


Figura Anexo II—6 Captura de pantalla de la aplicación sobre el historial de rutas

Desde la vista de perfil mostrada previamente, podemos acceder a tres funcionalidades de gestión de la cuenta del usuario, cambiar contraseña, cambiar nombre de usuario y cierre de sesión. Para ello debemos pulsar sobre el botón correspondiente y se abrirán las siguientes vistas.



Figura Anexo II—8 Captura de pantalla de la aplicación sobre cambio de contraseña



Figura Anexo II—7 Captura de pantalla de la aplicación sobre cambio de nombre

En particular, en el cambio de contraseña ha de escribirse la misma contraseña nueva en los dos campos, en caso de que presente alguna diferencia se le comunicara al usuario mediante un mensaje en la pantalla. También es posible que, si la cuenta lleva mucho tiempo con la sesión iniciada, el cambio de contraseña no funcione, por lo que será necesario renovar las credenciales volviendo a iniciar sesión en la app. El cierre de sesión es accesible mediante el botón de una puerta y una flecha pequeña en la esquina superior derecha en la barra de herramientas de la vista perfil.

Por último, desde la vista de la página principal si accedemos a ajustes podremos ver el botón de baja del sistema, el cual eliminará todos nuestros datos de usuario en la app y no podremos volver a acceder a la cuenta. Además, en esta vista se encontrará también el cambio de idioma, mediante el desplegable "idioma" que se muestra a continuación se podrá elegir entre castellano e inglés.



Figura Anexo II—9 Captura de pantalla de la aplicación sobre ajustes