



**UNIVERSIDAD DE TALCA  
FACULTAD DE INGENIERÍA  
ESCUELA DE INGENIERÍA CIVIL EN COMPUTACIÓN**

**Aplicación móvil para la planificación y búsqueda de  
rutas para viajes turísticos urbanos**

**CÉSAR ANTONIO BRAVO SAN MARTÍN**

Profesor Guía: LUIS GREGORIO SILVESTRE QUIROGA

Memoria para optar al título de  
Ingeniero Civil en Computación

Curicó – Chile  
Septiembre, 2020

## CONSTANCIA

La Dirección del Sistema de Bibliotecas a través de su encargado Biblioteca Campus Curicó certifica que el autor del siguiente trabajo de titulación ha firmado su autorización para la reproducción en forma total o parcial e ilimitada del mismo.



Two circular stamps and signatures are present. The left stamp is blue and reads "UNIVERSIDAD DE TALCA" at the top, "DIRECCIÓN" in the center, and "SISTEMA DE BIBLIOTECAS" at the bottom. A blue ink signature is written over the stamp. The right stamp is grey and reads "UNIVERSIDAD DE TALCA" at the top, "SISTEMA DE BIBLIOTECAS" in the center, and "CAMPUS CURICO" at the bottom. A blue ink signature is written over the stamp.

Curicó, 2022

*Dedicado a mis padres Luis y Jacqueline, y a mi hermana Paula quienes siempre me han apoyado durante mis estudios y fueron comprensivos en los momentos en los que lo necesité.*

## **AGRADECIMIENTOS**

Agradecimientos a los amigos que hice durante mi estadía en la universidad y a quienes conservo desde antes por hacer de mi carrera en la universidad mas entretenida. A mi profesor guía que me apoyó en los cambios al proyecto por confiar en mi. Al resto de profesores que me enseñaron a lo largo de la carrera por la formación profesional que me han dado. A la facultad de Ingeniería y al departamento de Ciencias de la Computación por permitirme participar en actividades extra-académicas que contribuyeron a mi desarrollo profesional. Y a Dios por darme la vida para recorrer este camino.

## TABLA DE CONTENIDOS

	página
<b>Dedicatoria</b>	<b>I</b>
<b>Agradecimientos</b>	<b>II</b>
<b>Tabla de Contenidos</b>	<b>III</b>
<b>Índice de Figuras</b>	<b>VII</b>
<b>Índice de Tablas</b>	<b>X</b>
<b>Resumen</b>	<b>XI</b>
<b>1. Introducción</b>	<b>12</b>
1.1. Descripción del Proyecto . . . . .	12
1.2. Motivación . . . . .	12
1.3. Contexto del proyecto . . . . .	13
1.4. Definición del problema . . . . .	14
1.5. Objetivos . . . . .	14
1.5.1. Objetivo general . . . . .	14
1.5.2. Objetivos específicos . . . . .	15
1.6. Alcances . . . . .	15
1.7. Trabajos relacionados . . . . .	16
1.8. Estructura del documento . . . . .	17
<b>2. Antecedentes</b>	<b>18</b>
2.1. Conceptos de Dominio . . . . .	18
2.1.1. Turismo urbano . . . . .	18
2.1.2. Puntos de interés . . . . .	19
2.2. Conceptos Metodológicos . . . . .	20
2.2.1. Metodología de desarrollo . . . . .	20
2.2.2. Metodología de evaluación . . . . .	22
2.3. Conceptos Tecnológicos . . . . .	23

2.3.1.	Tecnologías Back-end . . . . .	23
2.3.2.	Tecnologías Móviles . . . . .	25
2.3.3.	Gestión de Datos . . . . .	26
<b>3.</b>	<b>Marco Metodológico</b>	<b>27</b>
3.1.	Requisitos . . . . .	27
3.2.	Planificación . . . . .	29
3.3.	Inicio de iteración . . . . .	29
3.4.	Diseño . . . . .	31
3.4.1.	Arquitectura Física . . . . .	32
3.4.2.	Arquitectura Lógica . . . . .	32
3.4.3.	Diagrama clases . . . . .	32
3.4.4.	Diagrama de secuencia . . . . .	32
3.5.	Implementación . . . . .	35
3.6.	Pruebas . . . . .	36
3.7.	Retrospectiva . . . . .	36
<b>4.</b>	<b>Journey Planner</b>	<b>37</b>
4.1.	Concepción del Proyecto . . . . .	37
4.2.	Implementación de la Solución . . . . .	37
4.2.1.	Planificación del Proyecto . . . . .	37
4.2.2.	Desarrollo del Proyecto . . . . .	47
4.3.	Cierre del proyecto . . . . .	58
<b>5.</b>	<b>Evaluación de Journey Planner</b>	<b>59</b>
5.1.	Fases de Experimentación . . . . .	59
5.1.1.	Definición . . . . .	59
5.1.2.	Diseño de la Experimentación . . . . .	60
5.1.3.	Ejecución de la Experimentación . . . . .	62
5.1.4.	Análisis de la Experimentación . . . . .	64
5.2.	Revisión de resultados de experimentación . . . . .	69
<b>6.</b>	<b>Conclusiones</b>	<b>70</b>
6.1.	Discusión . . . . .	70
6.2.	Lecciones aprendidas . . . . .	72

6.3. Trabajo Futuro . . . . .	73
<b>Bibliografía</b>	<b>75</b>
<b>Anexos</b>	
<b>A: Descripción de las tareas realizadas por cada iteración</b>	<b>78</b>
A.1. Iteración 1 . . . . .	79
A.2. Iteración 2 . . . . .	80
A.3. Iteración 3 . . . . .	81
A.4. Iteración 4 . . . . .	82
<b>B: Resultados de la encuesta</b>	<b>83</b>
B.1. Formulario . . . . .	83
B.1.1. Datos generales . . . . .	83
B.1.2. Sección 2 . . . . .	84
B.1.3. Sección 3 . . . . .	85
B.1.4. Sección 4 . . . . .	85
B.1.5. Sección 5 . . . . .	86
B.2. Respuestas . . . . .	86
B.2.1. Sección 1 . . . . .	86
B.2.2. Sección 2 . . . . .	87
B.2.3. Sección 3 . . . . .	89
B.2.4. Sección 4 . . . . .	90
B.2.5. Sección 5 . . . . .	91
<b>C: Interfaces de la aplicación</b>	<b>92</b>
C.1. Vistas de inicio y búsqueda. . . . .	93
C.2. Vistas de gestión de usuario. . . . .	95
C.3. Mapa. . . . .	98
C.4. Vistas de creación y edición de viaje. . . . .	104
C.5. Vistas de creación de eventos. . . . .	106
<b>D: Historias de usuario expandidas</b>	<b>108</b>
D.1. Historias de usuario. . . . .	108

<b>E: Diccionario de datos</b>	<b>112</b>
E.1. users . . . . .	112
E.2. trips . . . . .	112
E.3. points . . . . .	113
E.4. events . . . . .	113



## ÍNDICE DE FIGURAS

	página
2.1. Fases del proceso PXP. . . . .	21
2.2. Actividades del proceso de experimentación en IS. . . . .	23
3.1. Tablero en Trello de la Iteración 1. . . . .	31
3.2. Arquitectura Física de la aplicación. . . . .	32
3.3. Arquitectura Lógica de la aplicación. . . . .	33
3.4. Diagrama de clases simplificado. . . . .	33
3.5. Diagrama de secuencia de registro y creación de un viaje. . . . .	34
4.1. Diagrama de actividades de la planificación de un viaje. . . . .	38
4.2. Modelo relacional de la aplicación. . . . .	42
4.3. Interfaces de registro e inicio de sesión. . . . .	43
4.4. Colección de peticiones en Postman. . . . .	44
4.5. Interfaces de agregar marcador y detalles de una ruta. . . . .	45
4.6. Interfaces de agregar un evento y búsqueda de viajes. . . . .	46
4.7. Interfaces de calificación de un evento y bloqueo de cámara. . . . .	46
4.8. Selección del destino. . . . .	47
4.9. Búsqueda de lugares. . . . .	47
4.10. Selección de la categoría. . . . .	48
4.11. Resultado de la búsqueda. . . . .	48
4.12. Información de un lugar. . . . .	49
4.13. Resultado de la búsqueda. . . . .	49
4.14. Búsqueda de ubicaciones por nombre. . . . .	50
4.15. Resultado de la búsqueda. . . . .	50
4.16. Agregar un marcador de forma manual. . . . .	50
4.17. Ingresar la información del marcador. . . . .	50
4.18. Pantalla de creación de un viaje. . . . .	51
4.19. Pantalla de inicio con el viaje creado. . . . .	52
4.20. Acceso a la búsqueda de viajes. . . . .	52
4.21. Ventana de búsqueda de viajes. . . . .	53
4.22. Resultado de la búsqueda. . . . .	53

4.23. Vista de un viaje. . . . .	54
4.24. Vista de información de un viaje. . . . .	54
4.25. Acceso a la búsqueda de viajes. . . . .	55
4.26. Selección del medio de transporte. . . . .	56
4.27. Selección del punto de partida. . . . .	56
4.28. Trazado de la ruta de viaje. . . . .	57
4.29. Información de la ruta. . . . .	58
C.1. Pantalla de inicio vacía. . . . .	93
C.2. Pantalla de inicio con viajes. . . . .	93
C.3. Pantalla de búsqueda de viajes vacía. . . . .	94
C.4. Pantalla de búsqueda de viajes con resultados. . . . .	94
C.5. Pantalla de inicio de sesión. . . . .	95
C.6. Pantalla de registro de usuario. . . . .	95
C.7. Pantalla de registro con código de verificación. . . . .	96
C.8. Pantalla de datos del usuario. . . . .	96
C.9. Pantalla de edición de usuario. . . . .	97
C.10. Pantalla de cambio de contraseña. . . . .	97
C.11. Mapa centrado en la posición del usuario. . . . .	98
C.12. Búsqueda por categoría. . . . .	98
C.13. Resultado de la búsqueda de ubicaciones. . . . .	99
C.14. Detalles de una ubicación seleccionada. . . . .	99
C.15. Pantalla para añadir un marcador al mapa. . . . .	100
C.16. Despliegue de tipos de ubicaciones para el marcador. . . . .	100
C.17. Selección de un marcador en el mapa. . . . .	101
C.18. Selección del medio de transporte. . . . .	101
C.19. Selección del punto de partida. . . . .	102
C.20. Ruta trazada en el mapa. . . . .	102
C.21. Detalles de la ruta trazada. . . . .	103
C.22. Trazado de ruta desde la ubicación del usuario a un punto en el mapa. . . . .	103
C.23. Pantalla de creación de un viaje. . . . .	104
C.24. Pantalla de información de un viaje. . . . .	104
C.25. Pantalla de edición de un viaje. . . . .	105
C.26. Pantalla de información de un viaje no perteneciente al usuario. . . . .	105

C.27. Pantalla de acceso a creación de un evento. . . . .	106
C.28. Pantalla de creación de un evento. . . . .	106
C.29. Despliegue de tipos de eventos. . . . .	107
C.30. Visualización de eventos en el mapa. . . . .	107

## ÍNDICE DE TABLAS

	página
2.1. Tipos y subtipos de ruta. . . . .	19
3.1. Resumen de historias de usuario del proyecto. . . . .	28
3.2. Planificación de Iteraciones. . . . .	29
3.3. Tareas realizadas en las iteraciones. . . . .	30
4.1. Tareas por sprint . . . . .	39
4.2. Detalle de las tareas de la Iteración 1. . . . .	41
5.1. Escala de Likert. . . . .	62
A.1. Descripción de las tareas de la Iteración 1. . . . .	79
A.2. Descripción de las tareas de la Iteración 2. . . . .	80
A.3. Descripción de las tareas de la Iteración 3. . . . .	81
A.4. Descripción de las tareas de la Iteración 4. . . . .	82
B.1. Datos generales . . . . .	86
B.2. Funcionalidad de gestión de usuario . . . . .	87
B.3. Funcionalidad de planificación de viaje . . . . .	87
B.4. Funcionalidad de trazado e información de la ruta . . . . .	88
B.5. Funcionalidad de búsqueda de viajes . . . . .	88
B.6. Corrección del sistema . . . . .	89
B.7. Utilidad del sistema . . . . .	89
B.8. Facilidad de uso del sistema . . . . .	90
B.9. Cómo le ha parecido la aplicación con respecto a . . . . .	90
B.10. Cómo le ha parecido la aplicación con respecto a . . . . .	91

## RESUMEN

Con el transcurso de los años, la cantidad de personas que salen de sus ciudades para realizar turismo ha ido en incremento. En este contexto, el turismo ha aprovechado las ventajas de las tecnologías de la información para mejorar la experiencia de usuario. En la actualidad existen varias aplicaciones web y móviles que ofrecen a sus potenciales usuarios servicios como hospedaje, y lugares de visita, entre otros. Sin embargo, los potenciales usuarios deben buscar alternativas para planificar su itinerario a través de folletos o guías turísticas.

La complejidad de planificar un itinerario turístico emerge cuando un turista tiene la intención de recorrer una ciudad pero considerando factores como el medio de transporte en el cual se encuentra el viajero, la distancia entre los puntos a realizar el trayecto, el tiempo disponible para esto, entre otros. Además el aspecto de la planificación se hace más crítico cuando el turista tiene un tiempo corto para realizar turismo.

Con el objetivo de abordar la problemática sobre planificación de turismo, se propone Journey Planner que es una aplicación móvil que permite realizar una planificación turística o de viaje considerando aspectos de tiempo, movilidad, rutas y lugares turísticos. Journey Planner fue desarrollado en cuatro iteraciones con la aplicación de Personal Extreme Programming (PXP). Por otro lado, Journey Planner es evaluado aplicando un proceso de experimentación en ingeniería de software, el cual permite validar el producto obtenido y evaluar las funcionalidades implementadas.

Los resultados de la evaluación sugieren un alto grado de satisfacción por parte de los sujetos participantes de la evaluación en términos de usabilidad, correctitud, funcionalidad y utilidad. Además Journey Planner es una aplicación móvil que se diferencia de las demás por su enfoque en la planificación que otras características que son bien abordadas por otras herramientas. En este sentido, Journey Planner ha demostrado ser útil para realizar turismo y para planificar viajes a otras ciudades. Sin embargo, queda un margen para mejoras en el ámbito de la visualización y la cantidad de información presentada.

# 1. Introducción

---

En este capítulo se presentan los aspectos fundamentales del proyecto, tales como la descripción de este y la motivación de la cual nace. Por otra parte, se presentan los objetivos que serán abordados durante su desarrollo junto con los alcances considerados para su ejecución.

## 1.1. Descripción del Proyecto

En este documento son descritos los procesos de diseño e implementación de una aplicación móvil para la planificación de jornadas de viaje, la cual permite al usuario seleccionar múltiples ubicaciones interactuando con un mapa y sugerir una ruta para recorrerlos, contando con diferentes medios de transporte o puntos de inicio.

Dado que al recorrer una ciudad, los usuarios no tienen conocimiento de ella, el sistema puede sugerir diversas locaciones en base a categorías, así como jornadas planificadas por otros usuarios a modo de orientación.

Adicionalmente, este contará con un módulo para informar de eventos que transcurren en una ciudad, con el fin de incentivar el recorrido de esta por parte de la misma comunidad.

Para evaluar las funcionalidades que se implementan, se cuenta con un tiempo de experimentación sobre la aplicación, con el fin de conocer la opinión de potenciales usuarios y validar el desarrollo.

## 1.2. Motivación

En la actualidad, las ciudades cuentan con atractivos turísticos de gran variedad con atracciones para diferentes tipos de personas. Por ello, estas ofrecen diversas actividades y

ubicaciones por conocer para los visitantes, quienes pueden verse desorientados al recorrer estas urbes.

Bajo esta premisa y, tomando a su vez experiencias cercanas, se concibe la idea del desarrollo de un proyecto que tenga como fin el desarrollo de una aplicación móvil que permita a sus usuarios la planificación para recorridos turísticos de diversa índole. Esto, con el fin de brindar una mayor facilidad al momento de realizar esta tarea teniendo la posibilidad de usar como guía rutas realizadas por otros usuarios en la misma ciudad sin requerir el uso de múltiples aplicaciones que no satisfacen de buena manera esta tarea.

### **1.3. Contexto del proyecto**

En relación a la premisa mencionada en la sección anterior, para el desarrollo del proyecto es necesario sentar las bases y definir los conceptos que forman parte del dominio asociado a la aplicación.

Iniciando por lo más fundamental, se debe entender en qué consiste el turismo. Este consiste, en su definición mas simple, en el acto de realizar un viaje por placer. A mayor profundidad, este puede definirse como el conjunto de múltiples interacciones derivadas por un desplazamiento temporal con diversos propósitos o motivaciones.

En este contexto, se considera como una jornada turística el conjunto de interacciones como resultado del recorrido por múltiples ubicaciones en una localidad tal como una ciudad o un pueblo. Estas ubicaciones pueden ser de diferente categoría dependiendo del atractivo turístico que ofrezcan, como por ejemplo museos o restaurantes, los cuales entregan servicios diferentes, pero cuentan con atractivos diferentes dependiendo de lo que busque quien realiza el viaje.

La aplicación a desarrollar será utilizada en ambientes como los descritos con anterioridad, con el fin de brindar una herramienta de apoyo al momento de realizar un recorrido entre varios puntos en una ciudad. Por ende, el marco del proyecto se encuentra en un ámbito del recorrido de localidades, las cuales cuentan con diversos lugares los cuales visitar, conectados por múltiples caminos que pueden ser recorridos por una variedad de medios de transporte.

Dado a que existe una gran variedad de localidades, son consideradas las que son de un carácter urbano, las cuales pueden poseer una variedad de puntos de interés en común que un viajero esperaría encontrar en ellas, como es el caso de los restaurantes, bares, plazas, lugares históricos, etc.

## **1.4. Definición del problema**

Al momento de planificar un recorrido y llevarlo a cabo, se desconocen los puntos de interés o lugares destacados, así como también los lugares en los cuales comer o retirar fondos en un cajero automático. A esto, se le suma el desconocimiento de las calles y el manejo de los tiempos que se disponen para recorrer la localidad, por lo que estas actividades son realizadas mediante múltiples aplicaciones móviles. Por otra parte en las urbes pueden llevarse a cabo eventos que sean atractivos para los visitantes en un contexto turístico, quienes no cuentan con esta información de manera accesible.

Estas tareas de planificación y realización de una jornada de recorrido son realizadas con varias aplicaciones las cuales no presentan una sinergia adecuada que facilite y sugiera al usuario de buena manera como realizar su recorrido. Debido a esto, el turista puede desorientarse en la ciudad, gastando más del tiempo estimado en un trayecto de lo que tenía estimado, no visitar algún punto de interés en la localidad o como también no asistir a algún evento esporádico en el destino turístico en cuestión.

Tomando en consideración lo anterior, los aspectos del problema que se abarcan con el desarrollo del proyecto están asociados a la dificultad para planificar una visita a una localidad, brindando un enfoque orientado al recorrido entre los diversos atractivos turísticos a recorrer y orientando a los usuarios mediante recomendaciones de ubicaciones o rutas realizadas por otros usuarios.

## **1.5. Objetivos**

En el desarrollo que se lleva a cabo tiene como objetivo abordar los aspectos descritos a continuación:

### **1.5.1. Objetivo general**

Guiar el proceso de organización de una jornada de turismo mediante una aplicación móvil que brinde recomendaciones e información sobre aspectos asociados a la planificación.



### 1.5.2. Objetivos específicos

- Identificar en que consiste la planificación de una jornada de viaje urbano y sus tipos.
- Asistir al usuario mediante recomendaciones de ubicaciones y lugares para realizar un recorrido turístico.
- Brindar una herramienta de geolocalización que permita al usuario conocer su posición al realizar su viaje.
- Aplicar una metodología de desarrollo de software para construir la aplicación.
- Aplicar una metodología de evaluación a la aplicación para determinar la efectividad del proyecto.

## 1.6. Alcances

En esta sección se abordarán los alcances de la realización del proyecto, indicando lo que se planea abordar y lo que se acotará tomando en consideración el tiempo de realización.

- En este trabajo se espera implementar un prototipo funcional de la propuesta, por lo tanto, se desarrollará una aplicación móvil simple con las facultades de recomendación y selección de ubicaciones, determinación de una ruta entre estos el cual cuenta también con un sistema de cuentas de usuario y manejo de esta para la gestión de viajes.
- Se contempla un periodo de pruebas para comprobar que las funcionalidades de la aplicación actúen de manera correcta y evaluar los resultados de la utilización de esta.
- Se aborda el desarrollo de la aplicación móvil y el periodo de pruebas para los dispositivos con sistema Android compatibles con la versión del kit de desarrollo de software a utilizar en la implementación. No se contempla un periodo de desarrollo y testeo en dispositivos con otros sistemas operativos.

- En el desarrollo del proyecto se abordan el ámbito relacionado al diseño y desarrollo de la aplicación. No se abordará la publicación y el modelo de negocios de la aplicación. Se aborda de esta manera ya que estas fases consecutivas al desarrollo, involucrarían un gasto monetario y análisis económico, al requerir de servicios externos los cuales son de pago.
- La información mostrada de las ubicaciones depende de la disponibilidad y completitud que entregue el servicio de consulta. Esto es, que ciertas ubicaciones pueden disponer de mayor información que otras y esta, en el caso de no existir, no será mostrada.
- Este trabajo se limita a la planificación de recorridos dentro de una ciudad que no quieran la utilización de medios de transporte los cuales no dependan del usuario, tales como el transporte público o servicios de renta de medios de movilización. No se contempla la planificación de viajes entre múltiples ciudades o con fines no turísticos.

## 1.7. Trabajos relacionados

Actualmente, existe un conjunto de herramientas individuales las cuales dan una solución a esta problemática. El uso de estas aplicaciones requiere que de manera frecuente se realicen cambios entre una aplicación y otra, lo cual resulta complicado para el usuario.

Con respecto a la geolocalización y la traza de rutas, podemos encontrar aplicaciones tales como Google Maps y Waze. La primera es una aplicación que cumple múltiples funcionalidades, tales como navegación entre puntos, conocer información de lugares, entre otros. Mientras tanto, la segunda se encuentra orientada para conducción vehicular.

Por otro lado, para el ámbito turístico la más conocida es TripAdvisor, que tiene su foco en la calificación y reseñas de lugares turísticos. Si bien, es posible generar viajes en esta aplicación, al momento de querer determinar como llegar a las ubicaciones, esta deriva dicha tarea a la aplicación Google Maps, la cual entrega el trayecto a recorrer para ese determinado punto.

## 1.8. Estructura del documento

El presente documento está compuesto por un inicio, cuerpo y fin del proyecto, distribuidos a lo largo de seis capítulos siendo estos, los señalados a continuación:

- **Capítulo 1: Introducción.** Presentación del proyecto y el contexto en el cual se envuelve, así como los objetivos que busca su desarrollo y los alcances que este presenta
- **Capítulo 2: Antecedentes.** Descripción del dominio en el cual se enmarca el proyecto, las tecnologías utilizadas para su desarrollo y la metodología de trabajo y evaluación mediante las cuales es implementado.
- **Capítulo 3: Marco Metodológico.** Detalles sobre como se lleva a cabo la metodología y el diseño formulado para la ejecución del proyecto.
- **Capítulo 4: Journey Planner.** En este capítulo se presenta el desarrollo de la implementación de la metodología aplicada, explicando las funcionalidades realizadas como parte del desarrollo.
- **Capítulo 5: Evaluación de Journey Planner.** Ejecución de la metodología de evaluación seleccionada, presentando un análisis y discusión de los resultados en contraste con los objetivos planteados.
- **Capítulo 6: Conclusiones.** Cierre del proyecto y discusión del trabajo realizado, señalando aspectos del trabajo futuro que puede ser llevado a cabo con el producto obtenido.

## **2. Antecedentes**

---

En este capítulo se presentan los antecedentes recopilados acerca del dominio en el cual se enmarca el proyecto. Iniciando por los conceptos relacionados al turismo, para posteriormente continuar con los aspectos referentes a las tecnologías y metodologías.

### **2.1. Conceptos de Dominio**

En esta sección se describen los conceptos asociados al área de conocimiento en el cual se enmarca el proyecto.

#### **2.1.1. Turismo urbano**

En la actualidad, la definición de turismo urbano toma en consideración la ciudad como espacio donde este se realiza, entendiéndola como un área compleja, con características dinámicas y estar condicionada por múltiples factores, actores e intereses, los cuales determinan su funcionamiento cotidiano. [19]

El turismo urbano surge con la demarcación de sitios y vistas que los visitantes debían conocer. Debido a que, si un turista eligiera viajar aleatoriamente por las calles de una ciudad, es posible que varios de sus recorridos no hubiesen resultado satisfactorios. [10]

#### **Recorrido turístico**

También denominados como rutas, circuitos, caminos o itinerarios, son los que invitan al turista a recorrer un trayecto en que predominan categorías patrimoniales tales como manifestaciones culturales, testimonios del pasado arqueológico, patrimonio artístico, industrial o espacios naturales.

Por otra parte, existen un tipos diferente de rutas, diferenciadas de las anteriores al presentar poco o nada de la realidad histórica y cultural del destino, pero incluyen otro tipo de valores agregados, tales como las rutas cinematográficas o literarias.

Finalmente, encontramos las rutas mixtas, que carecen de un eje temático definido y responden a una oferta genérica. Estas son diseñadas de manera artificial con una selección de hitos patrimoniales y recreativos los cuales son asociados sin un orden en específico. [20]

En vista de lo anterior, podemos clasificar los tipos de rutas como se muestra en el Cuadro 2.1.

Cuadro 2.1: Tipos y subtipos de ruta.

<b>Rutas</b>	<b>Subtipos</b>
Específicas Patrimoniales	Rutas del Patrimonio Etnológico
	Rutas del Patrimonio Histórico y/o arqueológico
	Rutas del Patrimonio Natural
	Caminos históricos e Itinerarios Culturales
Específicas imaginadas	Literarias-cinematográficas-televisivas
	Personalidades históricas y personajes ficticios célebres
	Mágicas esotéricas, míticas, legendarias
Genéricas o mixtas	Mosaico de recursos(rutas y circuitos)

### 2.1.2. Puntos de interés

Un punto de interés, en el contexto en el cual se sitúa el proyecto, corresponde a una atracción o un recurso turístico. Estas son definidas como elementos naturales, objetos culturales o hechos históricos, los cuales son adaptados o intervenidos y como resultado pueden ser utilizados como causa suficiente para motivar el desplazamiento de turistas [15].

Así mismo, otras definiciones apuntan a un concepto similar, en las cuales se define la atracción turística como que aquellos elementos que por sí mismos o en combinación con otros pueden despertar el interés para visitar una determinada zona o región [15].

Existen diversas definiciones como las previamente descritas, pero de ellas es posible extraer una idea general y conceptos en común. Es posible afirmar que un punto de interés o atracción turística son ubicaciones que generan el interés de realizar un desplazamiento a dicha ubicación.

## 2.2. Conceptos Metodológicos

En esta sección se describen las metodologías de desarrollo y evaluación que se utiliza para llevar a cabo la implementación y la validación del proyecto.

### 2.2.1. Metodología de desarrollo

Para realizar el desarrollo de la aplicación, es considerado el tiempo disponible para su ejecución y la flexibilidad de las metodologías conocidas. Bajo este contexto, dado a que el tiempo es acotado, se opta por la utilización de una metodología de desarrollo ágil. Por otra parte, dado a que es un desarrollo individual, se escogió la metodología Personal Extreme Programming, la cual involucra un bajo nivel de documentación generada, permitiendo un mayor tiempo dedicado a la implementación al no necesitar una gran cantidad de documentación como lo requieren las metodologías tradicionales.

Con esta elección, se busca flexibilidad y velocidad en el desarrollo, teniendo como fin obtener prototipos funcionales al final de cada iteración. Esto permite una mejor evolución del desarrollo y un producto final el cual ha sido adaptado a las necesidades en cada uno de los intervalos.

#### Personal Extreme Programming

Corresponde a una metodología adaptada del XP<sup>1</sup>, la cual conserva las prácticas de esta que puedan ser aplicables para un proyecto desarrollado por una persona y descarta los aspectos asociados al trabajo en equipo. A su vez, realiza modificaciones al Proceso de desarrollo de software personal<sup>2</sup>, manteniendo los principios pero disminuyendo la carga que este presenta para maximizar el rendimiento personal. [6]

Esta metodología se basa en los siguientes principios:

- El desarrolladores son responsables de aplicar las prácticas de PXP.
- Se debe medir, registrar y analizar el trabajo diario.
- Se debe aprender de las variaciones del desempeño y apuntar a mejorar el proceso en base a los datos recolectados en el proyecto.

---

<sup>1</sup>Extreme Programming: Metodología ágil de software orientada a grupos de trabajo pequeños.

<sup>2</sup>PSP: Personal Software Process

- PXP requiere de realización de pruebas continuas.
- El desarrollador debe tratar de automatizar la mayor cantidad de trabajo diario.

El proceso de desarrollo PXP se muestra en la Figura 2.1

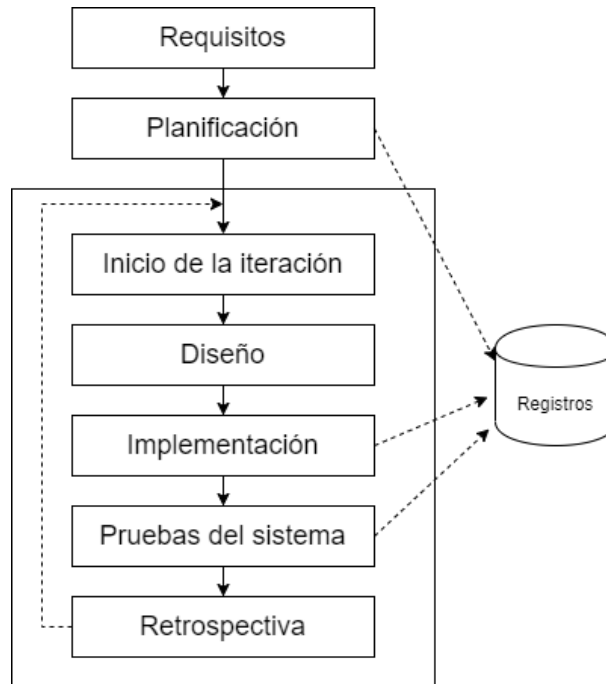


Figura 2.1: Fases del proceso PXP.

A continuación, se provee una breve descripción de estas fases:

- **Requisitos:** Durante esta fase, se crea un documento con los requerimientos funcionales y no funcionales del sistema.
- **Planificación:** Se toman decisiones tales como la elección del lenguaje de programación, el modelo de la aplicación, el framework a utilizar, entre otros. Se determinan una serie de tareas en base a los requerimientos listados en el documento. Estas son estimadas en base a la experiencia en proyectos previos.
- **Inicio de la iteración:** Indica el inicio de cada iteración, en el cual se seleccionan las tareas que serán implementadas en este ciclo. Estas suelen durar entre 2 o 3 semanas, dependiendo del alcance del proyecto.

- **Diseño:** En esta etapa son modelados los módulos y clases que serán implementadas durante la iteración en curso. Este diseño se hace en consideración a que los requisitos puedan variar en el futuro.
- **Implementación:** Se lleva a cabo la implementación y codificación de las clases definidas con anterioridad.
- **Pruebas del sistema:** En la fase de pruebas, es verificado el funcionamiento de las clases así como que estas cumplan con los requerimientos iniciales.
- **Retrospectiva:** Es el fin de la iteración, en el cual se busca la detección de problemas para ser solucionados en la siguiente iteración, tal como la correcta estimación de las tareas realizadas. Finalizada esta, se debe evaluar si quedan requerimientos para ser realizados en la siguiente iteración o marcar el fin del desarrolló, en caso de que se hayan cumplido los requerimientos y no haya defectos en el sistema.

### **2.2.2. Metodología de evaluación**

Con respecto a la metodología de evaluación, uno de los aspectos relevantes que a evaluar es la percepción de los posibles usuarios, por lo cual la elección de esta debe permitir la obtención de observaciones objetivas del producto en cuestión. Por ello, se considera el proceso de experimentación como el idóneo para la evaluación del software.

Tomando lo anterior en consideración, se estima que utilizando el Proceso de experimentación en Ingeniería de Software, es posible obtener una apreciación del producto generado junto con la validación de las funcionalidades implementadas.

### **Proceso de experimentación en Ingeniería de Software**

El proceso de experimentación consiste en identificar y comprender las distintas variables por las que se producen determinados resultados. Aplicada a la ingeniería de software, permite determinar las conexiones con el fin de validar o refutar con hechos las prácticas en las cuales se basa la construcción de un software. Su unidad básica es el experimento, el cual consiste en un modelado de las principales características de la construcción del software bajo condiciones controlables, permitiendo el estudio y comprensión de este [8].

Este proceso cuenta con determinadas actividades, las cuales quedan ilustradas en la Figura 2.2.

Estas fases, son brevemente descritas a continuación:



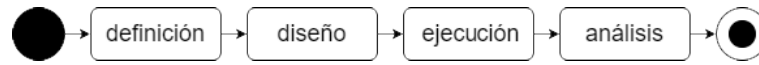


Figura 2.2: Actividades del proceso de experimentación en IS.

- **Definición:** Se especifican de manera general los aspectos fundamentales del experimento a realizar, entre estos, los objetivos de la experimentación.
- **Diseño:** Ya definido lo previamente mencionado, se diseña como será llevada a cabo. Esto consiste en definir las pruebas a realizar, los sujetos quienes las realizarán y los materiales necesarios para su ejecución.
- **Ejecución:** En esta fase se lleva a cabo el experimento. Con los materiales y los sujetos a disposición, se procede a dar las instrucciones a estos para su realización.
- **Análisis:** Los datos recolectados durante la ejecución son recolectados y son sometidos a análisis para posteriormente, ser contrastadas con los objetivos determinados durante la etapa de definición.

## 2.3. Conceptos Tecnológicos

En esta sección se describen de manera breve las tecnologías seleccionadas para la realización de la aplicación.

### 2.3.1. Tecnologías Back-end

Corresponden a las tecnologías que funcionan de manera interna, sin intervención directa del usuario. Para esta tarea, se opta por trabajar con Node.js, debido a la escalabilidad y popularidad de esta herramienta, la cual es reflejada en la comunidad y la cantidad de paquetes disponibles que facilitan el desarrollo con esta tecnología.

La alternativa considerada es Laravel, un framework para el lenguaje PHP, siendo el más popular para este. Las aplicaciones desarrolladas en este sistema utilizan el paradigma Modelo-Vista-Controlador(MVC) para su diseño. Cuenta como un ORM<sup>3</sup> llamado Eloquent, el que permite la gestión de las entidades en el modelo trabajado. [2]

<sup>3</sup>Object-Relational mapping: Modelo que permite transformar las tablas de una base de datos a tipos de un lenguaje orientado a objetos.

**Node.js**

Entorno de ejecución de JavaScript para el servidor de consultas, por sobre el cual funcionan los controladores de la lógica del sistema. Este funciona de manera asíncrona, disponibilizando un API o endpoints consultados mediante protocolo HTTP. Esta tecnología permite construir aplicaciones web escalables al presentar una estructura modular, la cual permite agregar dependencias de manera fácil mediante su gestor de paquetes. [23]

**NPM**

Es el gestor de paquetes para Node, el cual instala las dependencias de la aplicación y maneja la ejecución de scripts que permiten el despliegue de la aplicación. Es un proyecto de código abierto, el cual cuenta con una gran comunidad que lo soporta, así como usuarios que lo utilizan para publicar diversos componentes que pueden ser integrados a Node. [24]

**Express**

Es un framework para el desarrollo de aplicaciones web, el cual será utilizado para la comunicación con la base de datos y la alimentación la interfaz de usuario. Es de código abierto y permite crear aplicaciones web robustas y APIs. Funciona montando un servidor utilizando JavaScript como base, al cual se le realizan las consultas por parte de la interfaz mediante las solicitudes diseñadas. Al recibir las consultas, se comunica con el servidor de base de datos del cual extrae la información y envía la respuesta. [12]

**PM2**

PM2 es un gestor de aplicaciones para Node que permite su manipulación y monitoreo, permitiendo llevar un control sobre estas y manejando los errores que puedan presentarse durante la ejecución, evitando que esta deje de funcionar. Además, permite realizar balanceo de carga y el ingreso scripts de inicio para sistemas Linux. Ante la presencia de un error en el back-end de la aplicación, PM2 se encargará de mantener la aplicación activa para continuar su funcionamiento. [3]

**Docker**

Software para la automatización del despliegue de aplicaciones, el cual trabaja al nivel de kernel, brindando mayores ventajas en comparación a las máquinas virtuales. Permite

crear contenedores personalizados a partir de imágenes base, los cuales contienen las dependencias requeridas para correr la aplicación, permitiendo el despliegue del servidor de manera más fluida y automatizada. [13]

### 2.3.2. Tecnologías Móviles

Corresponden a las tecnologías con las cuales el usuario interactúa directamente. La tecnología con la cual se desarrolla el proyecto es Flutter, la cual permite el desarrollo simultaneo de aplicaciones para Android e IOS.

La elección de esta se debe principalmente a que permite el desarrollo para múltiples plataformas compilando de manera nativa para esta, lo cual facilita el mantenimiento de la aplicación al no tener que mantener códigos para ambas plataformas por separado. Por otra parte, estas tecnologías se encuentran entre las mas populares del último año [1], por lo que cuentan con comunidad activa y soporte constante.

Una alternativa considerada para el desarrollo es React Native, el cual es un framework para el desarrollo de aplicaciones multiplataforma utilizando JavaScript y desarrollado por Facebook. A diferencia de la tecnología escogida, React Native no compila el código a lenguaje nativo, por lo cual su eficiencia en el dispositivo es menor. [11]

#### **Flutter**

Es un kit de desarrollo de software en el cual permite el desarrollo de aplicaciones para dispositivos móviles, web y de escritorio. Utiliza el lenguaje de programación Dart, el cual es desarrollado por Google, al igual que Flutter. Las aplicaciones creadas mediante este SDK, están compuestas por Widgets, los cuales pueden contener estados mutables, que son los encargados de refrescar las vistas generadas. [18]

#### **Material Design**

Lenguaje de diseño que aplica para dispositivos tanto móviles como de escritorio, el cual contiene directivas independientes del tipo de plataforma y lenguaje para en el que se programa. [17]

### Google Maps Platform

Servicio que provee de mapas para diversos dispositivos y un API para interactuar con ellos. Esta plataforma permite también, obtener información de las ubicaciones y los detalles disponibles en su base de datos, entre otras facultades relacionadas a la geolocalización. Es el servicio de mapas con mayor cobertura y popularidad en la actualidad, contando con actualizaciones diarias de las ubicaciones por parte de sus usuarios. [4]

### 2.3.3. Gestión de Datos

Para la base de datos del proyecto, se opta por trabajar con el lenguaje de consultas SQL. La elección de esta tecnología se debe a la robustez que entrega el lenguaje SQL y la flexibilidad para el modelado de diversos esquemas que este brinda. Por su parte, la elección del gestor de base de datos es PostgreSQL debido a su robustez y velocidad a la hora de realizar consultas complejas, además de ser una herramienta de código libre.

Si bien existen alternativas viables que realizan buena sinergia con la aplicación, no son implementadas debido al bajo conocimiento sobre su funcionamiento. Una alternativa considerada es Firebase, el cual es un gestor de base de datos NoSQL utilizado ampliamente para desarrollo de aplicaciones móviles para Android. En esta, los datos son almacenados con formato JSON<sup>4</sup> en tiempo real, alimentando al cliente continuamente con la información mas reciente. [14]

### SQL

Lenguaje de consultas estructuradas para el modelo de datos, el cual trabaja utilizando un modelo relacional sobre el cual ejecuta sentencias. Mediante este, es posible administrar, mantener o recuperar información utilizando consultas a la base de datos. [16]

### PostgreSQL

Como motor y gestor de base de datos, se utiliza PostgreSQL, el cual es un sistema de bases de datos relacionales de código abierto utilizando el lenguaje SQL para operar sobre el. Entre sus características podemos encontrar el manejo de transacciones funcionales cumpliendo con las propiedades *ACID*<sup>5</sup>. [5]

---

<sup>4</sup>JavaScript Object Notation

<sup>5</sup>ACID es un acrónimo de Atomicity, Consistency, Isolation y Durability. En español: Atomicidad, Consistencia, Aislamiento y Durabilidad.

## 3. Marco Metodológico

---

En este capítulo se describe en detalle las etapas de la metodología elegida, especificando los ajustes que son realizados a estos al momento de su planificación y el transcurso del proyecto.

Como se mencionó en el capítulo previo, el desarrollo del proyecto será llevado a cabo utilizando la metodología ágil Personal Extreme Programming (XP). A pesar de ello, se han realizado ciertos ajustes en la realización de las etapas de esta metodología para adecuarla al tiempo disponible para la realización del proyecto.

### 3.1. Requisitos

La metodología XP, en su fase de requisitos, describe la generación de un documento de requisitos el cuál es abordado durante el desarrollo y permanece estable durante la implementación de este. Por la naturaleza del proyecto en desarrollo, se requiere de flexibilidad para adaptar las funcionalidades en cada entrega y la realización de una aplicación que se adecue a las necesidades surgidas en cada revisión. Por ello, se se optó por reemplazar el documento de requisitos dictado por la metodología y en su lugar realizar la generación de historias de usuario<sup>1</sup> en su lugar.

La creación de estas historias de usuario permite la definición de requisitos sin requerir de mayor documentación, lo cual se refleja en una disminución en el tiempo requerido para la ejecución de esta fase, el cual pasa a estar disponible para las siguientes etapas.

A continuación se presenta la estructura que sigue la especificación de historias de usuario seguida de un ejemplo:

---

<sup>1</sup>Las historias de usuario son representaciones de requisitos descritas en frases de mayor simplicidad, utilizadas en metodologías ágiles

*Como rol quiero funcionalidad para objetivo.*

Por ejemplo:

*Como usuario quiero añadir marcadores personalizados a un mapa para definir los lugares que visitaré.*

Siguiendo este esquema, son definidas las siguientes historias de usuario como se presentan en el Cuadro 3.1. Adicionalmente, se adjunta una extensión mas detallada de estos en los anexos.

Cuadro 3.1: Resumen de historias de usuario del proyecto.

Código	Historias de usuario.
HU1	Como usuario quiero añadir marcadores personalizados a un mapa para definir los lugares que visitaré.
HU2	Como usuario quiero conocer mi ubicación en el mapa para orientarme en este.
HU3	Como usuario quiero guardar la información de mis viajes para llevar un registro de ellos.
HU4	Como usuario quiero consultar una ruta y el tiempo para recorrer los lugares seleccionados.
HU5	Como usuario quiero seleccionar un medio de transporte para mi recorrido.
HU6	Como usuario quiero buscar lugares turísticos por categoría para agregarlos a mis marcadores.
HU7	Como usuario quiero conocer la información de un lugar para planificar mi visita.
HU8	Como usuario quiero conocer los eventos realizados en un lugar para poder complementar mi viaje.
HU9	Como usuario quiero buscar otros viajes y clonarlos para facilitar la planificación de mi viaje.
HU10	Como usuario quiero calificar los viajes y eventos para entregar retroalimentación a otros usuarios.
HU11	Como usuario quiero poder determinar como llegar a ubicación para poder realizar un trayecto a esta.
HU12	Como usuario quiero administrar mi información y viajes para llevar un control de mis datos.
HU13	Como desarrollador quiero evaluar la aplicación para determinar errores o mejoras en el diseño.

### 3.2. Planificación

En esta fase se determina el framework a utilizar y el modelo de la aplicación, lo cual es descrito en el capítulo de antecedentes.

Posteriormente, con las historias de usuario creadas en la fase previa, son definidas las iteraciones a realizar durante el proyecto y las tareas que serían abordadas en cada uno de ellos. Se determinó que la realización de estas tendrían una duración de 2 semanas.

La planificación de estas iteraciones es presentada en el Cuadro 3.2.

Cuadro 3.2: Planificación de Iteraciones.

<b>Sprint</b>	<b>Fecha de realización</b>
1	24 de Abril - 04 de Mayo
2	05 de Mayo - 18 de Mayo
3	19 de Mayo - 01 Junio
4	02 de Junio - 15 Junio

### 3.3. Inicio de iteración

Durante esta etapa son priorizadas las historias de usuario abordadas en cada iteración y las funcionalidades con mayor relevancia asociadas a ellas para ser implementadas a lo largo del periodo. Como resultado de esta tarea, se genera la información contenida en el Cuadro 3.3.

Para llevar un control de estas iteraciones se utilizó un tablero para desarrollo ágil en el cual era monitorizado el avance del proyecto. Para ello es utilizada la plataforma Trello<sup>2</sup> el cual se encuentra ejemplificado en la Figura 3.1. Estas tareas son ubicadas, dependiendo de su estado, en diferentes columnas, correspondientes a *en desarrollo*, *en pruebas*, *tarea realizada*.

<sup>2</sup>Es un software utilizado para la planificación de proyectos disponible para Web, Android e IOS.

Cuadro 3.3: Tareas realizadas en las iteraciones.

<b>Sprint 1</b>	
Historia de usuario	Tareas asociadas.
HU1	Realización de mock-ups y diagramas.
	Configuración del entorno de desarrollo.
	Añadir marcadores al mapa.
HU2	Mostrar la ubicación del usuario en el mapa
HU3	Registro e inicio de sesión de usuario.
	Carga y guardado de viajes.
	Persistencia en la aplicación.
<b>Sprint 2</b>	
HU1	Personalización de marcadores
HU4	Selección del punto de partida
	Determinar una ruta entre los marcadores
	Consulta de la duración del trayecto
HU5	Selección del medio de transporte
	Generar ruta considerando el transporte
HU6	Búsqueda de lugares con intereses turísticos.
	Obtención de atracciones cercanas.
HU7	Búsqueda de lugares por nombre.
	Mostrar detalles de una ubicación
<b>Sprint 3</b>	
HU8	Creación de eventos
	Mostrar eventos en el mapa
HU9	Búsqueda de viajes
	Vista de viajes de otros usuarios
	Clonación de un viaje
HU10	Calificar un viaje
	Orden de viajes por calificación
<b>Sprint 4</b>	
HU12	Edición y eliminación de viajes
	Edición y eliminación de cuenta
	Edición y eliminación de marcadores.
HU13	Pruebas y corrección de errores



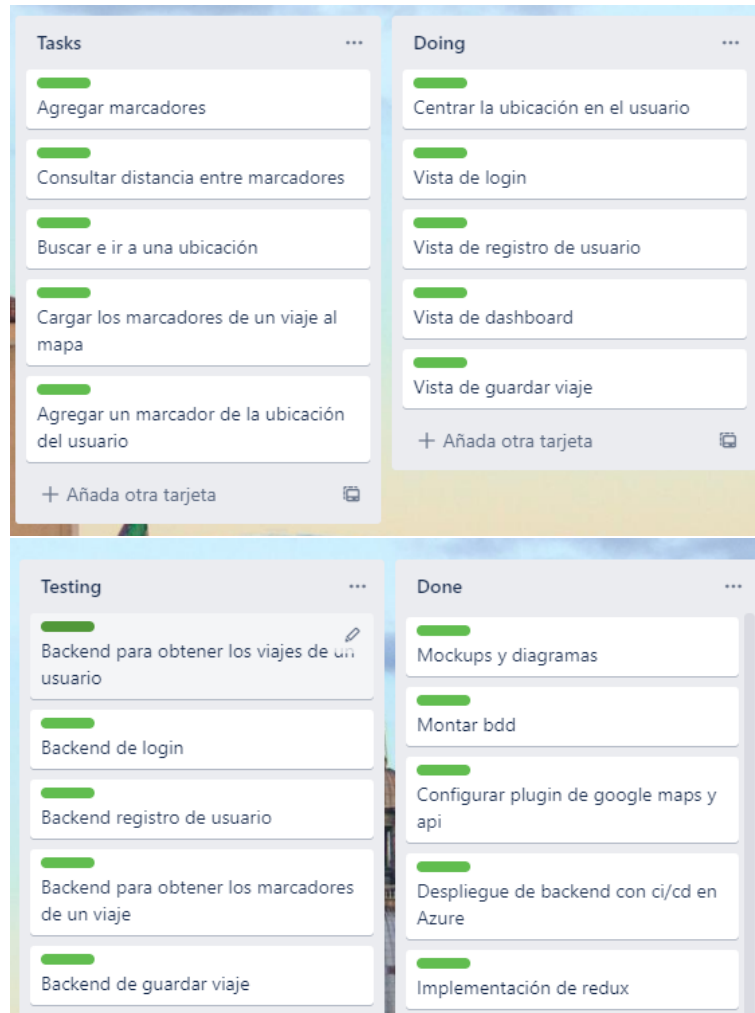


Figura 3.1: Tablero en Trello de la Iteración 1.

Estas tareas son definidas en base a la experiencia en desarrollo, siguiendo una estructura de desglose de trabajo con el fin de generar tareas unitarias las cuales puedan generar un entregable o producto del trabajo.

### 3.4. Diseño

En esta fase se define el diseño de la aplicación, el cual es el resultado de múltiples variaciones a lo largo del desarrollo dando como resultado lo expuesto en los siguientes apartados.

### 3.4.1. Arquitectura Física

Muestra la comunicación entre la arquitectura que da soporte a la aplicación. Esta corresponde al modelo Cliente-Servidor y se encuentra ilustrada en la Figura 3.2.

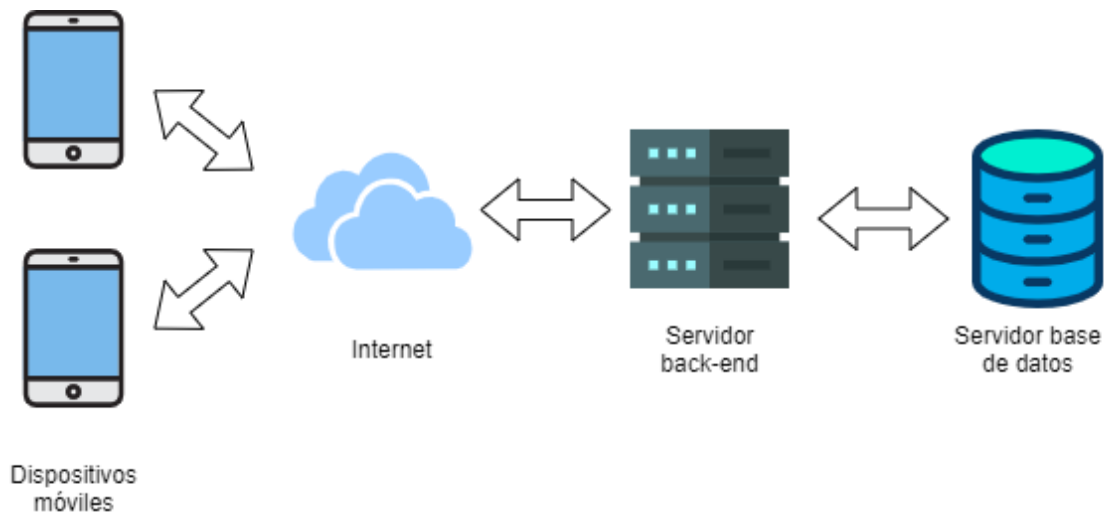


Figura 3.2: Arquitectura Física de la aplicación.

### 3.4.2. Arquitectura Lógica

Para la arquitectura lógica de la aplicación se utiliza el patrón Modelo-Vista-Controlador. En la Figura 3.3 se encuentra dicho modelo, el cual presenta la interacción entre los módulos determinados en un alto nivel de abstracción.

### 3.4.3. Diagrama clases

Se define el diseño de diagrama de clases para definir la estructura estática del sistema. En dicho diagrama, se han omitido las clases de menor importancia, tales como widgets visuales o utilidades, los atributos y los métodos, con el facilitar su visualización. Este se muestra en la Figura 3.4.

### 3.4.4. Diagrama de secuencia

En este diagrama, se presenta la interacción entre los actores para el registro de una cuenta de usuario y la creación de un viaje con múltiples marcadores, junto con su trazado de ruta. Figura 3.5

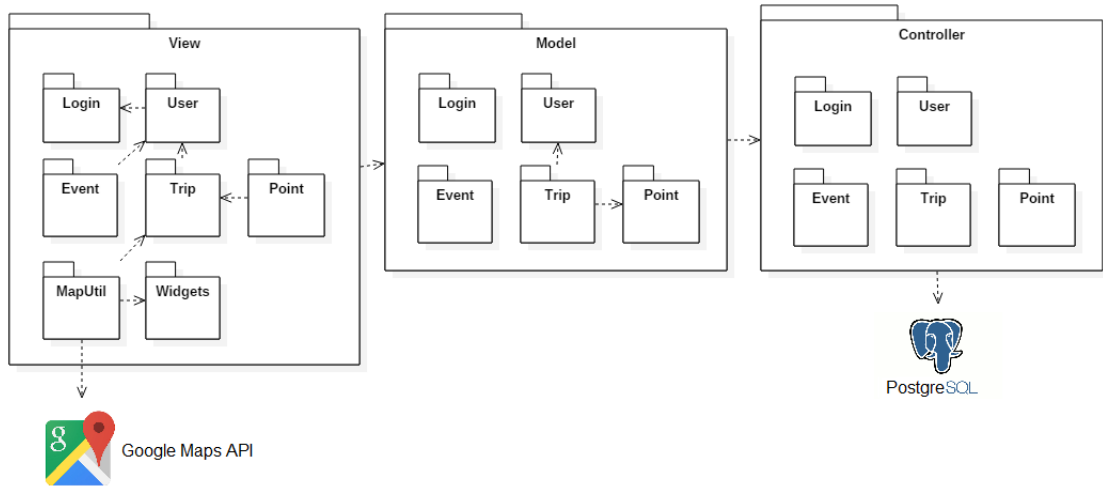


Figura 3.3: Arquitectura Lógica de la aplicación.

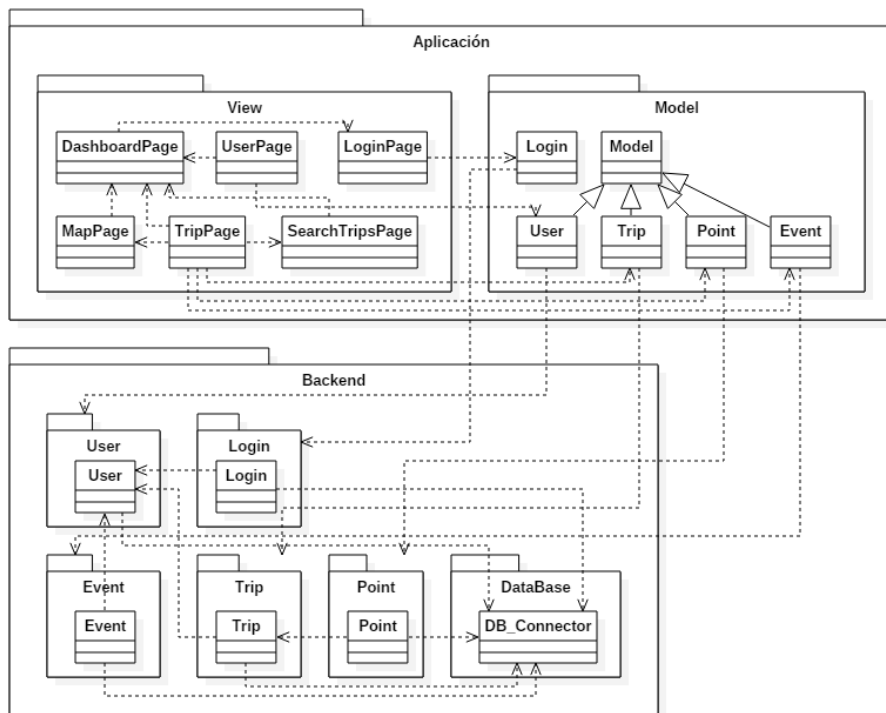


Figura 3.4: Diagrama de clases simplificado.

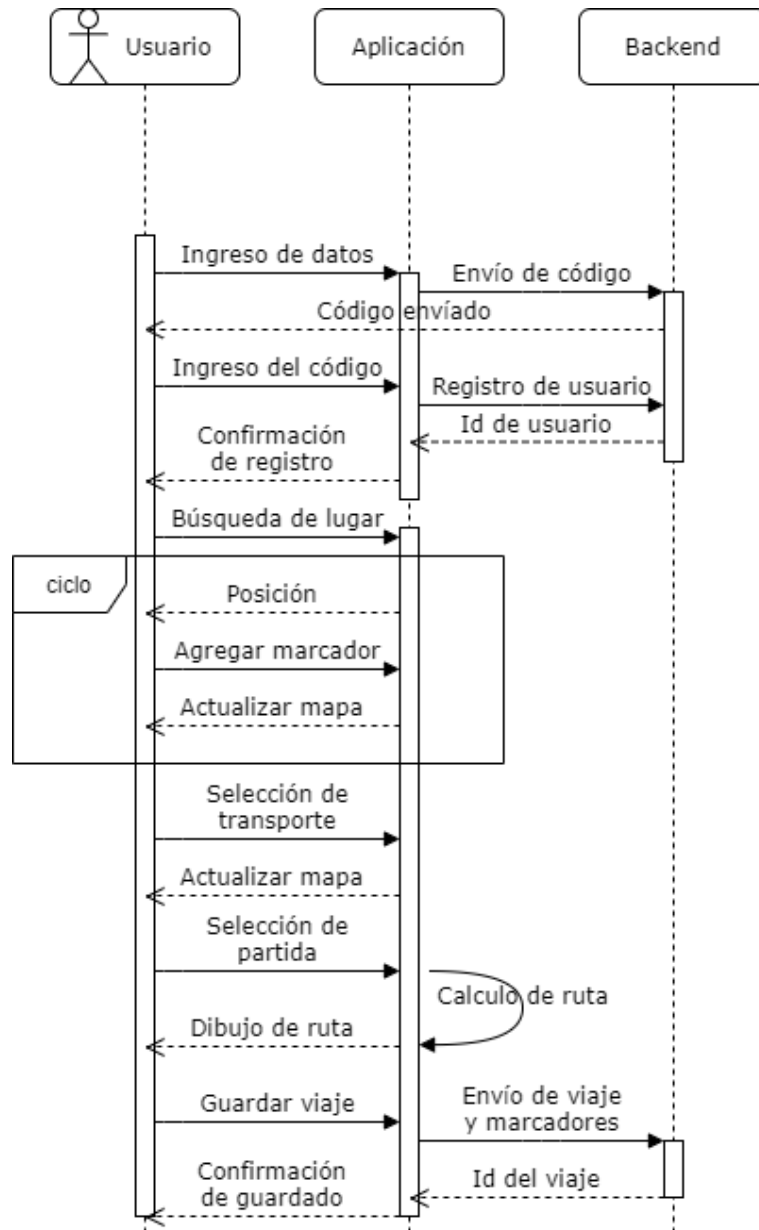


Figura 3.5: Diagrama de secuencia de registro y creación de un viaje.

### 3.5. Implementación

El desarrollo de cada iteración es llevado a cabo de la siguiente manera: en primera instancia, se aborda la codificación del back-end y la base de datos del proyecto, los cuales eran desplegados en la nube. Luego, se continúa con la implementación de la aplicación en Flutter, que era probada directamente en un dispositivo Android a medida que era implementada.

La comunicación entre el back-end y la aplicación se realiza siguiendo el patrón Modelo-Vista-Controlador como es señalado en el apartado de diseño. En él, la aplicación consulta y envía datos para ser procesados por parte del servidor, el cual los guarda en la base de datos y se encarga de rescatar la información solicitada y retornarla para ser mostrada en las interfaces.

Durante el desarrollo es utilizado el software de control de versiones Git, contando con un repositorio para la aplicación, alojado en Github<sup>3</sup> y otro par el back-end, alojado en Azure Repos<sup>4</sup>. Esto se debe a que la aplicación de back-end cuenta con una configuración para realizar CI-CD<sup>5</sup> en la nube de Azure<sup>6</sup>, lo cual permite que el periodo de pruebas sea realizado de manera mas ágil, al no requerir de participación para montar el servidor.

A mayor detalle, al realizar un commit en el repositorio donde se encuentra el back-end se activa la tarea programada que genera una imagen en Docker y la disponibiliza en el repositorio de imágenes Docker de Azure. Luego de esto, esta imagen era montada en la nube de Azure en el caso de que el proceso haya resultado exitoso. Si este presenta errores, se emite una notificación por correo la cual permite detectar fallas en el desarrollo. A grandes rasgos, el proceso realizado por el el script que genera la imagen del contenedor de Docker es descrito en el Algoritmo 1:

Por otra parte, la aplicación es desarrollada y sometida a pruebas directamente en un dispositivo Android mediante la herramienta de *hot-reload* de Flutter, la cual transmite los cambios al código directamente al dispositivo, sin la necesidad de compilar la aplicación nuevamente. Esta, mediante las variables configuradas se comunica directamente con el

---

<sup>3</sup>Es una plataforma web para el alojamiento de repositorios colaborativos que utilicen el control de versiones Git

<sup>4</sup>Es una plataforma web para el alojamiento de repositorios colaborativos que utilicen el control de versiones Git el cual se puede conectar fácilmente con la nube de Azure

<sup>5</sup>Continuous integration and Continuous Delivery es una forma de trajo en la cual los cambios efectuados en el repositorio son desplegados de manera automatizada.

<sup>6</sup>Servicio de computación en la nube perteneciente a Microsoft para el despliegue de aplicaciones web, contenedores, bases de datos, etc.

---

**Algorithm 1** Proceso del Dockerfile

---

- 1: **procedure** CREACIÓN DE LA IMAGEN DE DOCKER
  - 2: *tomando la imagen de node como base*
  - 3: *actualizar y descargar pm2*
  - 4: *copiar el directorio al contenedor*
  - 5: *instalar las dependencias*
  - 6: *exponer el puerto 3000*
  - 7: *levantar el servidor utilizando pm2*
- 

servidor back-end desplegado en Azure.

### 3.6. Pruebas

Para esta fase, siendo que la metodología recomienda la realización de Tests Unitarios para sortear las limitaciones de tiempo, se opta por realizar pruebas de código mediante debug y la validación del sistema a través de la experimentación, evaluando la implementación realizada en búsqueda de posibles defectos.

Para la realización de pruebas del sistema, se realizan debugs de código mediante impresión en consola del estado de la aplicación, junto al monitoreo de componentes que ofrece la herramienta DevTools<sup>7</sup> de Flutter. Por parte del servidor, es puesto a prueba mediante una colección de peticiones utilizando el software Postman<sup>8</sup> para verificar que su comportamiento es el esperado, junto con registros en consola de igual manera que la aplicación móvil tanto a nivel local como desplegado en la plataforma de Azure.

### 3.7. Retrospectiva

Transcurrida una iteración, se realiza una revisión junto al stakeholder de la aplicación, en este caso el profesor guía, quien entrega retroalimentación del trabajo realizado y eran evaluadas las funcionalidades presentes en cada versión del prototipo. De manera paralela, son presentados los avances prototipo con potenciales usuarios de la aplicación.

---

<sup>7</sup>Conjunto de herramientas para realizar debug de código en Dart y Flutter

<sup>8</sup>Es una aplicación web y de escritorio para realizar peticiones http.

## 4. Journey Planner

---

En este capítulo se describe la implementación de la solución iniciando por su concepción hasta su cierre, además el producto obtenido. Para ello, se definen hitos y funcionalidades significativas que permiten dar una idea del desarrollo general del proyecto.

### 4.1. Concepción del Proyecto

El proyecto surge como resultado de experiencias personales al recorrer diferentes ciudades visitando puntos emblemáticos y atractivos turísticos. Al realizar esta actividad las herramientas con las que disponía para la planificación de este no se complementan de manera satisfactoria, haciendo que esta tarea resulte tediosa y algo compleja.

Frente a esto, existen diferentes actividades a realizar al momento de planificar una visita turística, además de imprevistos comunes que se pueden llegar a presentar durante la realización del viaje, los que convergen en el proyecto llevado a cabo. Un diagrama para la planificación de un viaje puede encontrarse en la Figura 4.1

### 4.2. Implementación de la Solución

Conociendo los antecedentes descritos con anterioridad, se procede a realizar el desarrollo del prototipo para el proyecto. En este apartado, se describe la implementación del prototipo funcional y como es realizado este proceso.

#### 4.2.1. Planificación del Proyecto

En este apartado se presenta una descripción de las funcionalidades realizadas en cada una de las iteraciones, describiendo con completo detalle la Iteración 1, la cual se utiliza

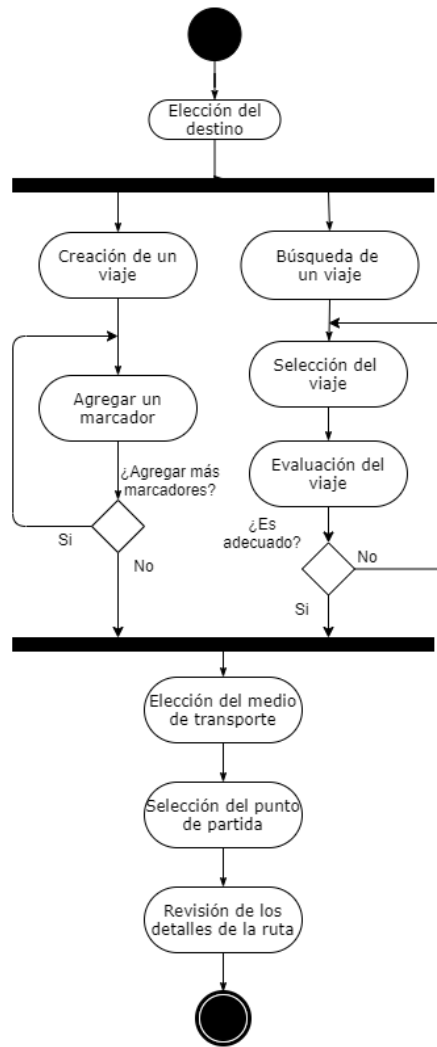


Figura 4.1: Diagrama de actividades de la planificación de un viaje.

como ejemplo de lo que es realizado con las sucesivas. La planificación de las funcionalidades se encuentra en el Cuadro 4.1



Cuadro 4.1: Tareas por sprint

Sprint	Periodo	Tareas
1	24/04 - 04/05	<ul style="list-style-type: none"> <li>- Diseño del diagrama de clases</li> <li>- Diseño del diagrama Entidad-Relacion</li> <li>- Despliegue de la base de datos</li> <li>- Diseño de mock-ups</li> <li>- Configuración del plugin de Google Maps</li> <li>- Back-end registro de usuario</li> <li>- Back-end ingreso de usuario</li> <li>- Back-end de guardar viaje</li> <li>- Agregar un marcador al mapa</li> <li>- Eliminar un marcador</li> <li>- Cálculo de distancia entre marcadores</li> <li>- Centrar la cámara en el usuario</li> <li>- Vista de registro de usuario</li> <li>- Vista de inicio de sesión</li> <li>- Vista de guardar viaje</li> <li>- Búsqueda de una ubicación</li> <li>- Persistencia de datos en la aplicación</li> </ul>
2	05/05 - 18/05	<ul style="list-style-type: none"> <li>- Listado de ubicaciones cercanas</li> <li>- Obtener los detalles de una ubicación</li> <li>- Agregar el marcador de una ubicación</li> <li>- Vista de personalización de un marcador</li> <li>- Selección del un punto de partida</li> <li>- Obtención de distancias y tiempos entre marcadores</li> <li>- Selección del medio de transporte</li> <li>- Implementación del algoritmo para generar la ruta</li> <li>- Cálculo del tiempo total del viaje</li> </ul>

3	19/05 - 01/06	<ul style="list-style-type: none"> <li>- Back-end para agregar eventos</li> <li>- Back-end búsqueda de viajes</li> <li>- Agregar eventos al mapa</li> <li>- Vista de agregar evento</li> <li>- Vista de resultados de búsqueda de viajes</li> <li>- Clonación de un viaje</li> </ul>
4	01/06 - 15/06	<ul style="list-style-type: none"> <li>- Back-end edición de un viaje y marcadores</li> <li>- Back-end calificación de eventos y viajes</li> <li>- Fijar la cámara al usuario</li> <li>- Back-end eliminación de un viaje y un usuario</li> <li>- Vista de edición de un viaje y un usuario</li> <li>- Widget de clasificación para viajes y eventos</li> </ul>

### Iteración 1

En la Iteración 1, en primera instancia, se realizan los diagramas de arquitectura de la aplicación junto con el diseño de mock-ups para las vistas del sistema. Luego, es configurado el entorno de desarrollo del proyecto, el cual utiliza el SDK<sup>1</sup> de Android para la aplicación y la instalación de los paquetes necesarios de Node.js para ejecutar el servidor de Express. Una vez realizado esto, se procede con el modelado y la implementación de este, para finalizar con un periodo de pruebas sobre el código implementado.

#### Inicio de la iteración

Durante esta fase, son seleccionadas las tareas a realizar durante la iteración. El conjunto de tareas elegidas se encuentran descritas en el Cuadro 4.2.

#### Diseño

Una vez determinadas las tareas que se realizarán en el sprint se prosigue con la fase de diseño. Durante este periodo es realizado el modelado de diagramas entre los cuales se encuentran:

- Diagrama de arquitectura física (Capítulo 3: Figura 3.2)
- Diagrama de arquitectura lógica (Capítulo 3: Figura 3.3)
- Diagrama de clases (Capítulo 3: Figura 3.4)

<sup>1</sup>Kit de desarrollo

Cuadro 4.2: Detalle de las tareas de la Iteración 1.

<b>Iteración 1</b>	
Tarea	Descripción
Diseño del diagrama de clases	Creación de la primera versión del diagrama UML que dicta la estructura de clases en la aplicación y el servidor.
Diseño del Modelo relacional	Creación de la primera versión del modelo relacional que llevará la base de datos.
Despliegue del servidor con CI/CD en Azure	Configuración del entorno de desarrollo de pruebas en el cual es desplegado el servidor de back-end en la plataforma Azure.
Despliegue de la base de datos en ElephantSql	Carga del script de base de datos en la plataforma para alojar base de datos ElephantSql.
Diseño de mock-ups	Diseño de las interfaces y la navegación básica del sistema.
Configuración plugin de Google Maps	Instalación del plugin de Google Maps para Flutter y realización de pruebas para probar sus funcionalidades.
Back-end registro de usuario	Implementación del módulo que registra usuarios en la base de datos.
Back-end ingreso de usuario	Implementación del módulo verifica los datos de ingreso de un usuario
Agregar un marcador	Implementación de la funcionalidad para agregar un marcador simple al mapa.
Eliminar un marcador	Implementación de la funcionalidad que permite eliminar un marcador seleccionado.
Cálculo de distancia entre marcadores	Implementación de la función que consulta por la distancia y la trayectoria entre dos marcadores.
Centrar la cámara en el usuario	Implementación de la funcionalidad que consulta por la ubicación del usuario y centra la vista del mapa en su posición.
Vista de registro de usuario	Implementación de la interfaz de registro de usuario.
Vista de inicio de sesión	Implementación de la interfaz de inicio de sesión.
Vista de guardar viaje	Implementación de la interfaz que permite ingresar la información de un viaje y guardarlo.
Back-end guardar viaje	Implementación del módulo que permite guardar un viaje asociado a un usuario y sus respectivos marcadores.
Buscar e ir a una ubicación	Implementación de la funcionalidad que permite buscar una ubicación por nombre y centrar la cámara en ella.
Persistencia de datos	Implementación de Redux para mantener el inicio de sesión y mantener los datos del usuario después del cierre de esta.

- Diagrama de secuencia (Capítulo 3: Figura 3.5)
- Modelo relacional (Figura 4.2)

Cabe destacar que se presentan las versiones finales de dichos diagramas, el cual es el resultado de la realización de todas las iteraciones realizadas. Esto es debido a que son de un carácter ilustrativo, con el fin de dar a conocer el trabajo en desarrollo. Un diccionario de datos sobre las entidades mas relevantes puede ser encontrada en los anexos de este documento.

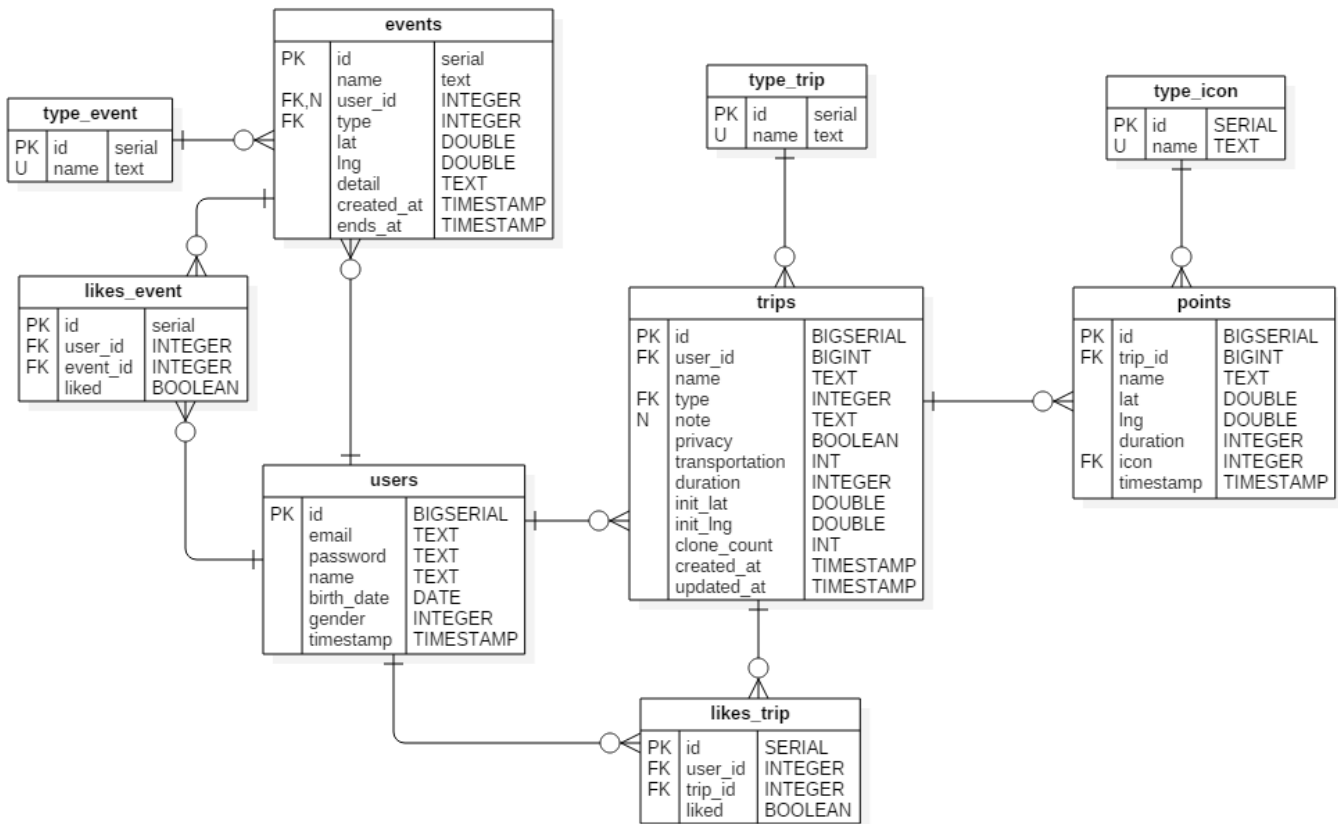


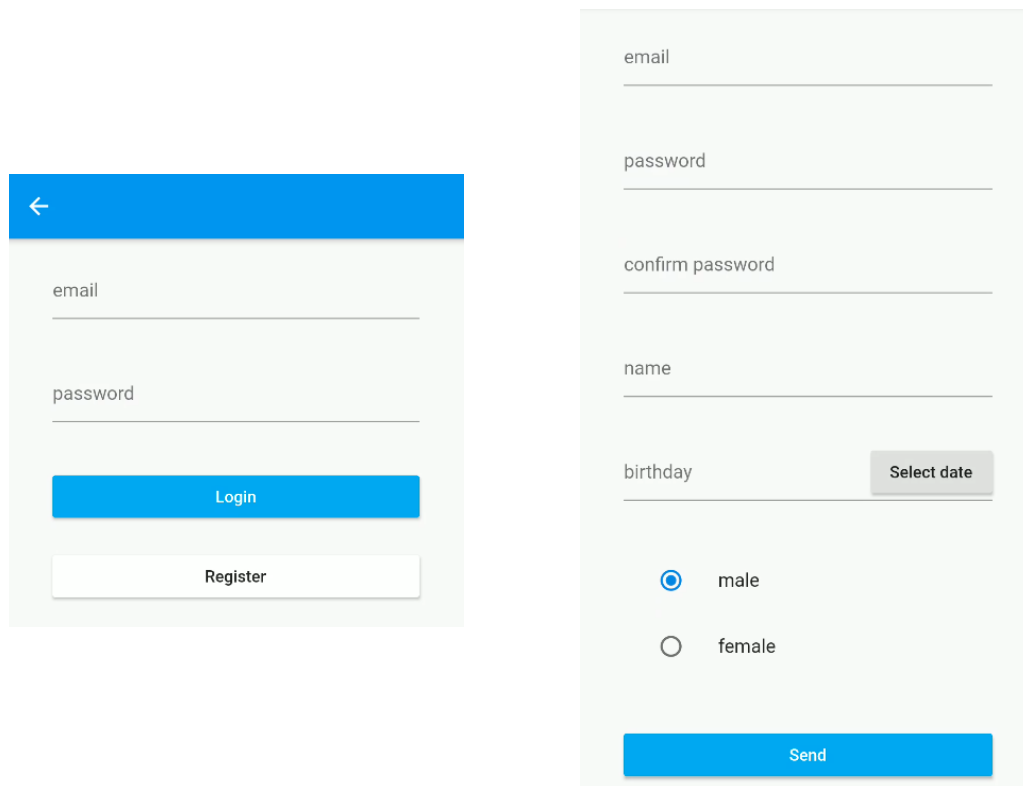
Figura 4.2: Modelo relacional de la aplicación.

### Implementación

Una vez efectuada la instalación de las dependencias mencionadas y realizado el diseño, se procede a inicializar los repositorios para el control de versiones de la aplicación y el back-end. Para el caso de este último, es creada una instancia de Docker en la nube de Azure para la integración continua desde el repositorio.

Ya inicializados los repositorios y configurada la integración continua, se comenzó con la codificación de las funcionalidades planificadas para esta iteración. Una vez finalizados los *endpoints*<sup>2</sup> de estas funcionalidades, se comenzó con el desarrollo de la aplicación.

En primera instancia, es realizada la navegación entre las vistas iniciales de la aplicación, las cuales son: la pantalla de inicio, la vista de registro de usuario, la vista de inicio de sesión, la vista de creación de un viaje y la pantalla del mapa. Una vez hechas estas interfaces, se añadieron los paquetes necesarios para mantener la persistencia de datos de la aplicación. Una muestra de estos resultados obtenidos como parte de la implementación puede apreciarse en la Figura 4.3.



The figure displays two mobile application screens. The left screen is a login form with a blue header bar containing a back arrow. It features two input fields labeled 'email' and 'password', a blue 'Login' button, and a white 'Register' button. The right screen is a registration form with a light green background. It includes input fields for 'email', 'password', 'confirm password', 'name', and 'birthday' (with a 'Select date' button), radio buttons for 'male' and 'female', and a blue 'Send' button.

Figura 4.3: Interfaces de registro e inicio de sesión.

### Pruebas del sistema

Comenzando con el back-end, para realizar las pruebas en este se utiliza el software Postman, probando cada uno de los endpoints implementados. Una vez concluidas las pruebas de este, se verificó que las funcionalidades implementadas en la aplicación funcionen de manera correcta con el contenedor desplegado en Azure y, de existir errores, se

<sup>2</sup>Rutas http mediante las cuales se comunica la aplicación con el servidor de back-end.

realizan las correcciones adecuadas. En la Figura 4.4 se encuentra la colección de peticiones http con las cuales es probado el servidor de Node.

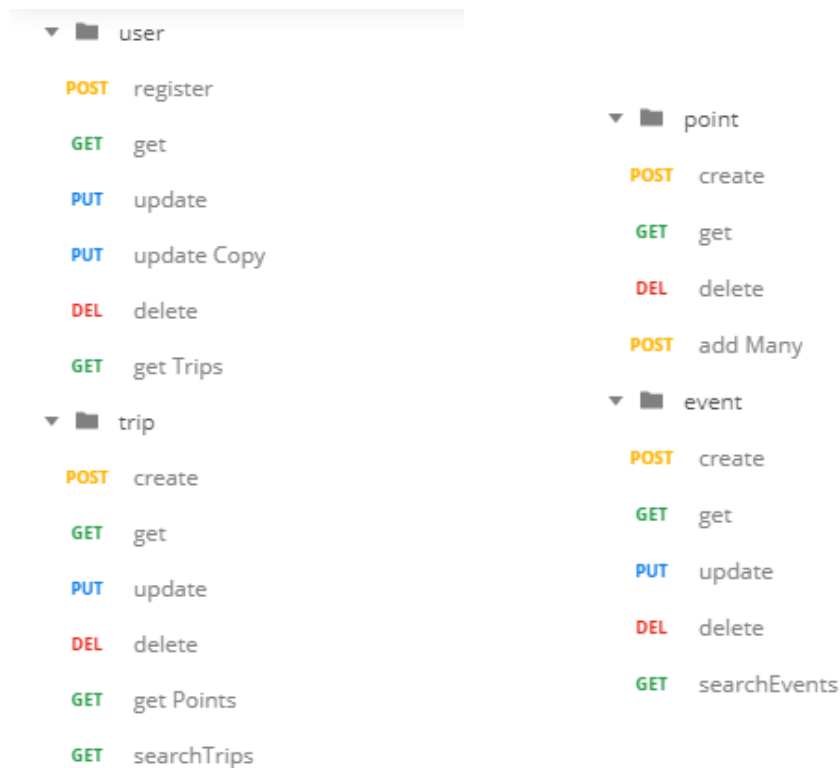


Figura 4.4: Colección de peticiones en Postman.

### Retrospectiva

Una vez concluida la iteración, se realiza una revisión de las funcionalidades implementadas en conjunto al profesor guía. Durante esta revisión se evalúa si el tiempo determinado por las tareas es el correcto y si quedaron tareas las cuales no pudieron ser realizadas. En este caso, son cumplidas todas las tareas asignadas, por lo cual se considera que la carga de trabajo es bien estimada y no hay tareas que requieran ser movidas para la siguiente iteración.

### Iteración 2

El desarrollo de la Iteración 2 se centro en la implementación de las funcionalidades mas relevantes de la aplicación. Estas consisten en gran parte a las tareas asociadas a la recomendación de lugares y el trazado de la ruta para recorrerlos.

Se comenzó con la funcionalidad que permite al usuario, dada una categoría, seleccionar un lugar de un listado para agregar un marcador al mapa. Una vez hecho eso, para realizar un trazado de ruta es necesario comunicarse con el API de Google Maps, al cual se le envían las coordenadas de los marcadores ingresados y responde con la distancia y tiempo entre ellos. En base a esta respuesta, se genera una lista de marcadores ordenados por la menor distancia entre estos y se realiza un nuevo llamado al API, considerando el punto de partida seleccionado y el medio de transporte, en el cual se solicita por un camino entre cada par de ubicaciones y se obtiene como respuesta la ruta para el recorrido. Con la ruta obtenida, se realiza el trazado en el mapa. Algunas de las interfaces resultantes pueden apreciarse en la Figura 4.5.

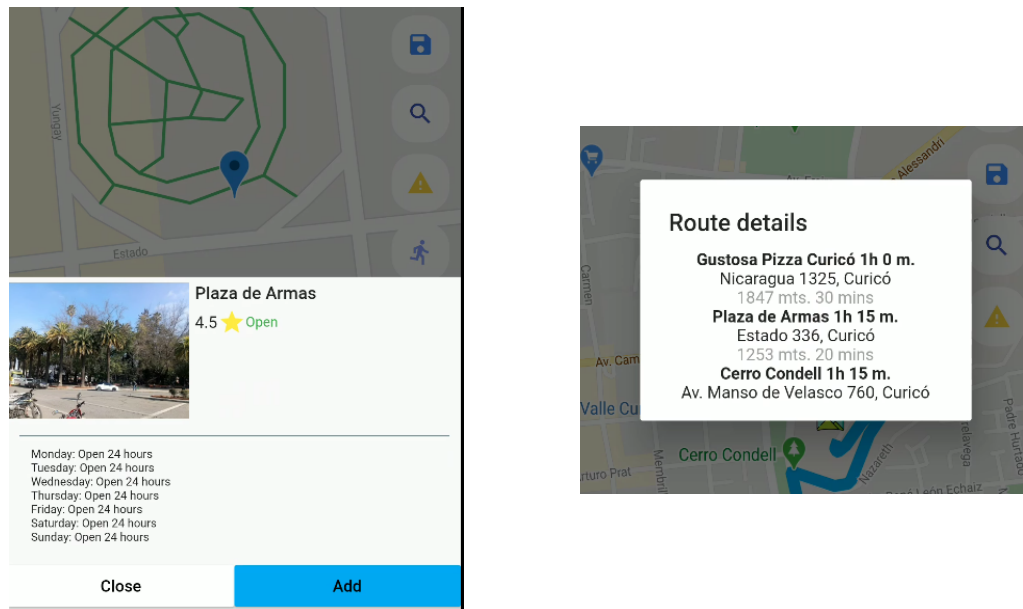


Figura 4.5: Interfaces de agregar marcador y detalles de una ruta.

### Iteración 3

A lo largo de esta iteración, se abordan las funcionalidades relacionadas a los eventos y la búsqueda de viajes realizados por otros usuarios. De la misma forma en la que es realizada la primera iteración, se comienza con el desarrollo del back-end de dichas funcionalidades, para posteriormente trabajar en las interfaces y su comunicación con el servidor. Las interfaces representativas de esta iteración se encuentran en la Figura 4.6.

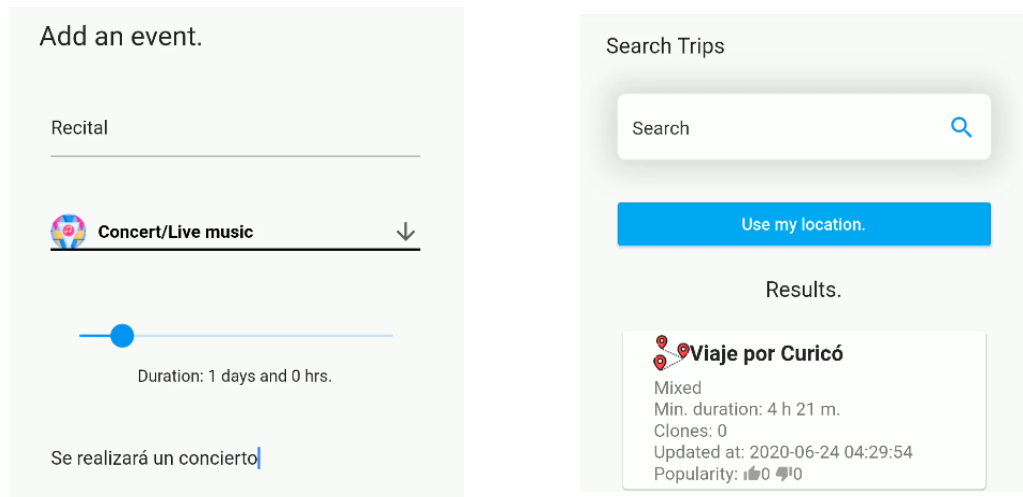


Figura 4.6: Interfaces de agregar un evento y búsqueda de viajes.

#### Iteración 4

Finalmente, con las funcionalidades fundamentales del sistema implementadas, en esta iteración se realiza la implementación de las funcionales de edición para las entidades correspondientes. Además, son añadidos los complementos que permiten la calificación de viajes y eventos, junto con la funcionalidad para fijar la cámara al movimiento del usuario. Las vistas relevantes que forman parte del resultado de este sprint son visibles en la Figura 4.7.

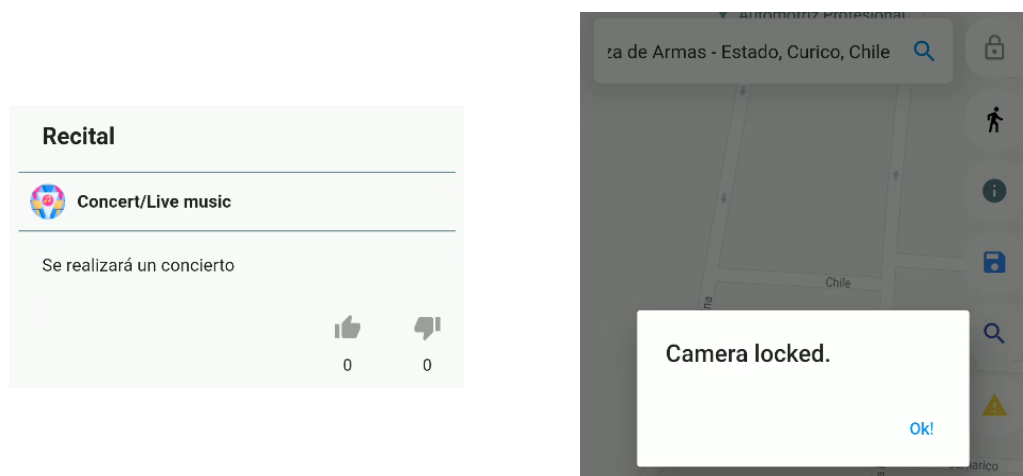


Figura 4.7: Interfaces de calificación de un evento y bloqueo de cámara.



### 4.2.2. Desarrollo del Proyecto

Para ilustrar el desarrollo del proyecto, en esta sección, se presentarán diversos ejemplos de las funcionalidades principales de la aplicación entregando una descripción de ellas y las interfaces generadas para realizar las tareas en cuestión. Dado a que la aplicación cuenta con un registro de usuario y gestión de este de manera estándar, se pondrá mayor atención a las características relacionadas a la creación de viajes y trazado de rutas.

#### Creación de un viaje

En ese apartado se es descrito como se realiza la planificación de un viaje con múltiples marcadores y el almacenamiento de este en la cuenta del usuario, por lo cual, para este proceso, ya se encuentra registrado en la base de datos. Para iniciar con el proceso, se requiere seleccionar un destino para el viaje desde la pantalla de inicio de la aplicación. Existen dos opciones para realizar esta tarea (Figura 4.8):

- utilizar el buscador de ubicaciones (Figura 4.9),
- utilizar la posición del usuario.

Independiente de la elección escogida la aplicación consultará por permisos para conocer la ubicación del usuario, si es que no los tiene, y procederá a abrir el mapa centrando la cámara en la ubicación elegida.

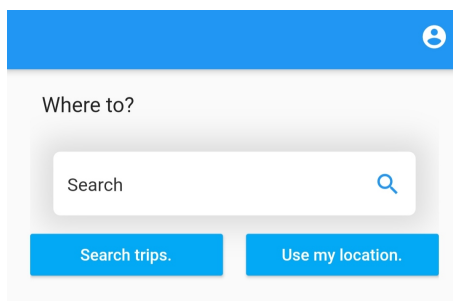


Figura 4.8: Selección del destino.

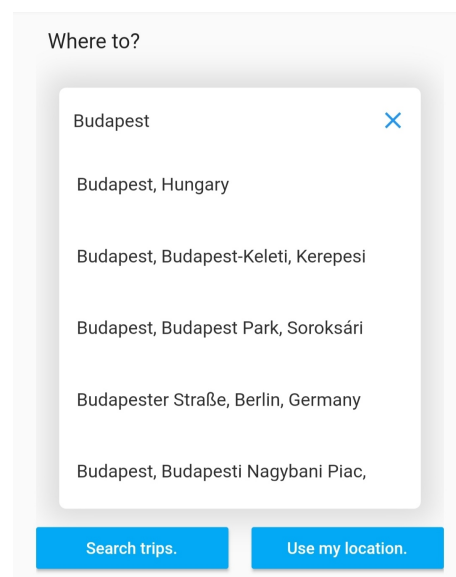


Figura 4.9: Búsqueda de lugares.

Continuando, se procede a agregar los puntos a visitar en el mapa. Para ello, existen tres formas de conseguirlo. La primera de ellas es utilizar el buscador de ubicaciones proporcionado en el menú lateral que se encuentra al lado derecho de la pantalla. Este provee con diferentes categorías para la búsqueda de ubicaciones (Figura 4.10), entregando un listado de estas en las cercanías (Figura 4.11), tomando como punto de referencia el centro de la pantalla proyectado en el mapa. Al seleccionar una, la cámara centrará la vista en la ubicación, entregando información básica sobre esta (Figura 4.12). La disponibilidad de esta información está ligada a la disponible en el API en la cual se consulta, por lo que ciertos detalles pueden ser visualizados en algunas ubicaciones, mientras que en otras no.

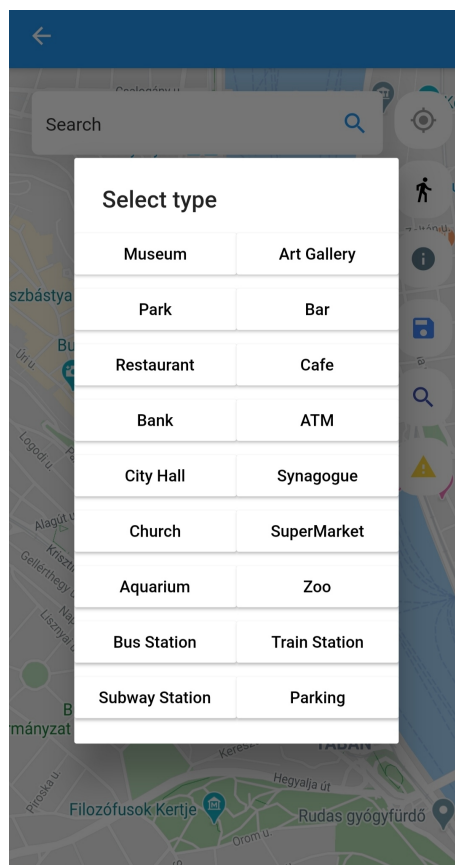


Figura 4.10: Selección de la categoría.

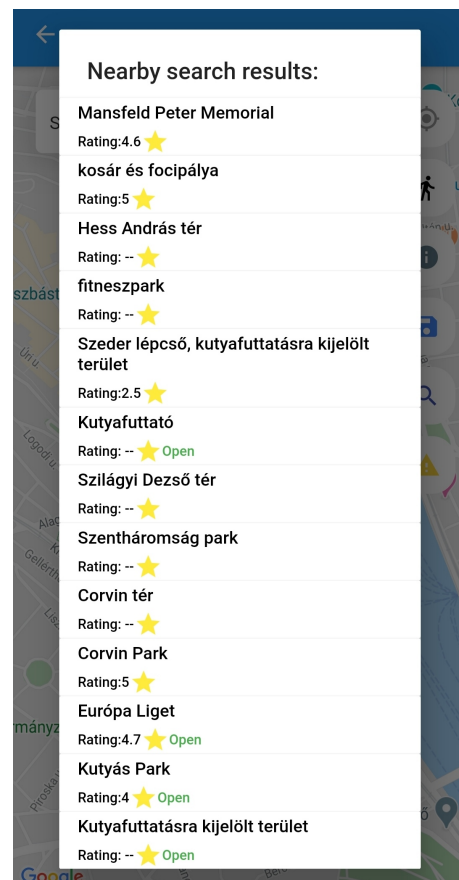


Figura 4.11: Resultado de la búsqueda.

Desde la ventana emergente cargada, es posible agregar un marcador o descartar la búsqueda. Al agregar un marcador, surgirá una nueva vista, en la cual se ingresan las características de este marcador tales como el nombre, que en este caso viene cargado pero puede editarse, el icono que el usuario quiera asignar a este marcador y la duración

estimada para esta visita (Figura 4.13). Una vez completado este proceso, se retorna al mapa con el marcador agregado.

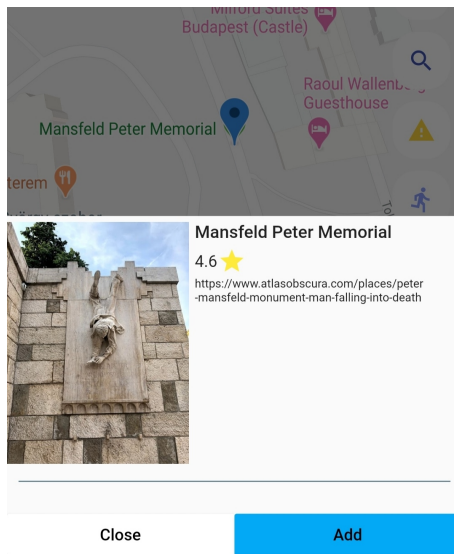


Figura 4.12: Información de un lugar.

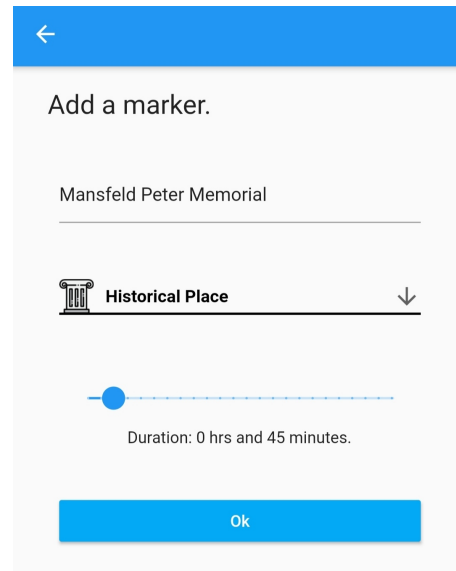


Figura 4.13: Resultado de la búsqueda.

La segunda forma para agregar puntos de interés es utilizar la barra de búsqueda superior (Figura 4.14), la cual dado el nombre de una ubicación, realiza un proceso similar al descrito previamente. Con el nombre obtenido, se obtienen los detalles disponibles de esta ubicación los cuales se muestran en una ventana emergente con las mismas opciones previas (Figura 4.15). De igual manera, como resultado de este proceso, obtenemos un nuevo marcador en el mapa.

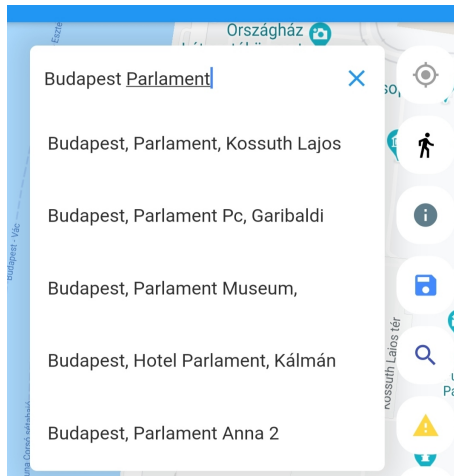


Figura 4.14: Búsqueda de ubicaciones por nombre.

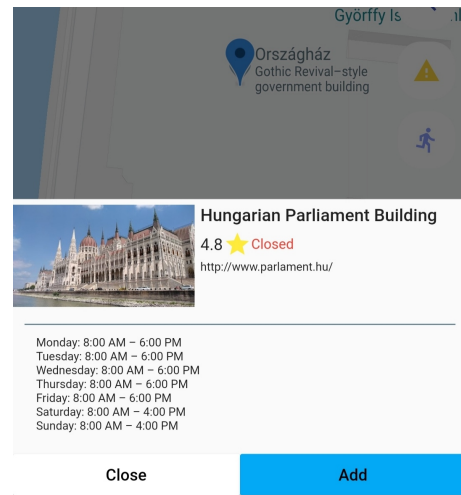


Figura 4.15: Resultado de la búsqueda.

Por último, es posible agregar un marcador tocando directamente la zona del mapa en la cual se desea agregar este (Figura 4.16). A diferencia de las opciones anteriores, para agregar este punto se mostrará en pantalla un nuevo botón el cual dirigirá a la pantalla donde se ingresan la información de este (Figura 4.17).



Figura 4.16: Agregar un marcador de forma manual.

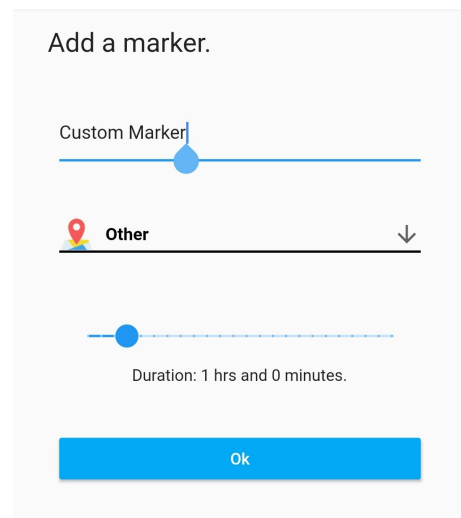
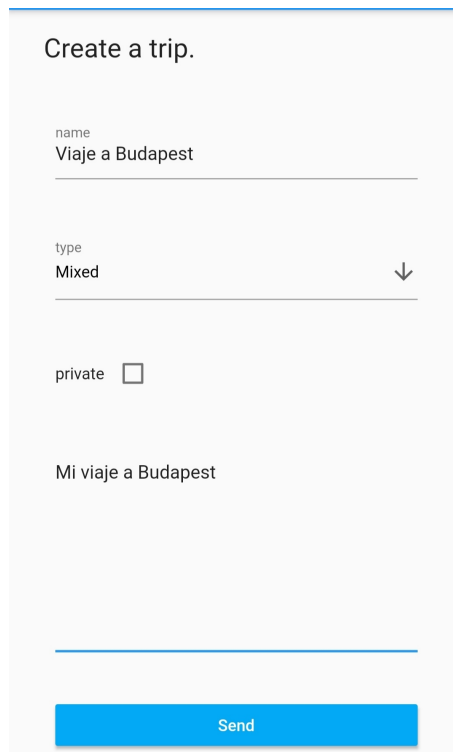


Figura 4.17: Ingresar la información del marcador.

Una vez ingresados los marcadores deseados, se procede a guardar el viaje. Para ello se debe pulsar la opción de guardar en el menú ubicado a la derecha, el cual lleva a la vista

de creación de un viaje (Figura 4.18). En ella, el usuario debe ingresar la información solicitada y finalizar el proceso, la aplicación envía la información ingresada al servidor de back-end que la almacena en la base de datos.



The screenshot shows a mobile application interface for creating a trip. The title is "Create a trip." Below the title, there are three input fields: a text field for "name" containing "Viaje a Budapest", a dropdown menu for "type" currently set to "Mixed" with a downward arrow, and a checkbox for "private" which is unchecked. Below these fields, the text "Mi viaje a Budapest" is displayed. At the bottom of the form is a blue button labeled "Send".

Figura 4.18: Pantalla de creación de un viaje.

Con el viaje ya creado, es posible acceder a él desde la pantalla de inicio de la aplicación, en la cual se muestran los viajes creados por el usuario (Figura 4.19).

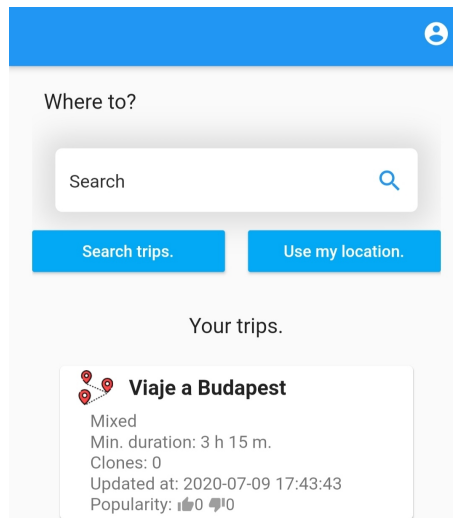


Figura 4.19: Pantalla de inicio con el viaje creado.

### Clonación de un viaje

En caso de que un usuario no conozca una ciudad o quiera guiarse por los viajes realizados por otros usuarios, dispone de la búsqueda de viajes. Para acceder a ella se debe pulsar el botón izquierdo ubicado bajo la barra de búsqueda desde la pantalla de inicio (Figura 4.20).

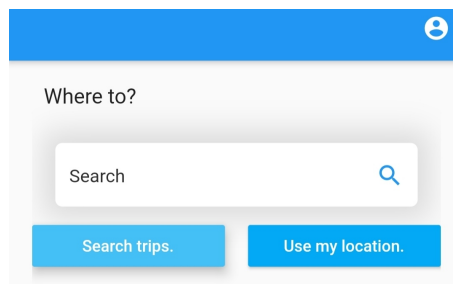


Figura 4.20: Acceso a la búsqueda de viajes.

Esta vista es similar a la pantalla inicial, en la cual el usuario puede buscar viajes ingresando a la barra de búsqueda o utilizando su ubicación (Figura 4.21). Como resultado de esto, se cargan los viajes cercanos a la ubicación seleccionada tomando en consideración un radio de 10Km (Figura 4.22).

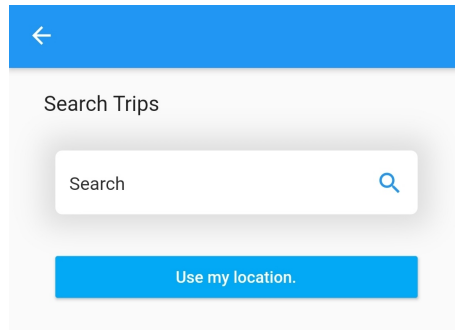


Figura 4.21: Ventana de búsqueda de viajes.

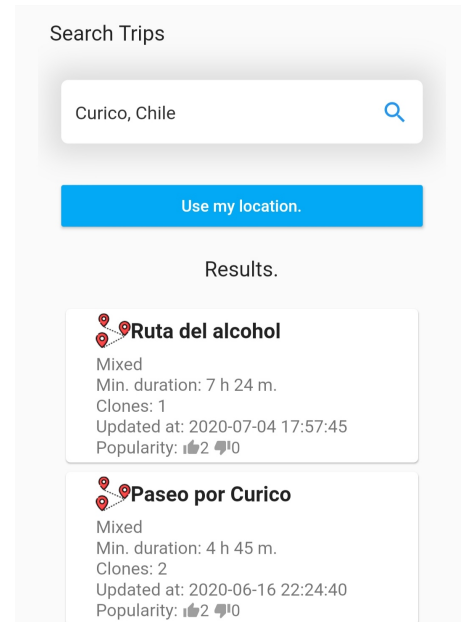


Figura 4.22: Resultado de la búsqueda.

Al ingresar a uno de ellos, se abre el mapa con los marcadores pertenecientes a este viaje (Figura 4.23). En esta instancia, no es posible realizar modificaciones a los marcadores dado a que el usuario no es el propietario de este. Para clonar este viaje, se debe seleccionar la opción de guardar presente en el menú derecho, la cual mostrará información ingresada por el propietario del viaje y se encuentran las opciones para efectuar la calificación de este (Figura 4.24).

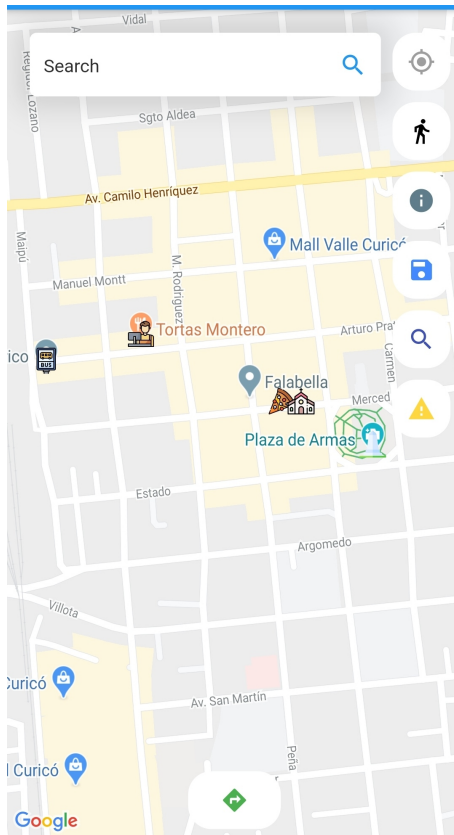


Figura 4.23: Vista de un viaje.

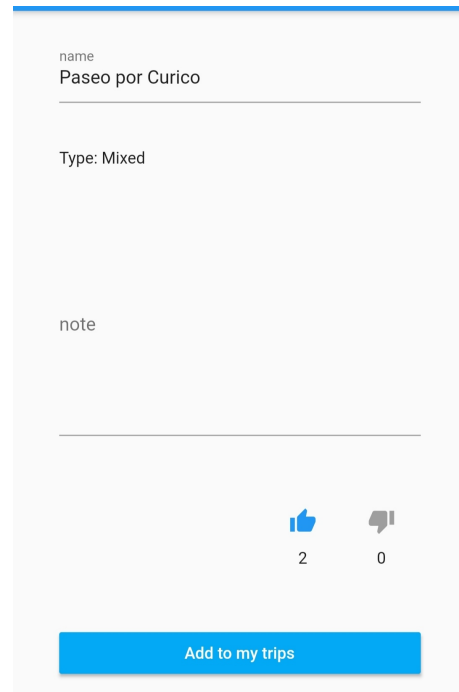


Figura 4.24: Vista de información de un viaje.

Si el usuario está registrado, en esta vista encontrará la opción de clonar el viaje, la cual creará una copia viaje asociado al usuario con los marcadores y la información del viaje en cuestión. Esta copia ahora será listada en la pantalla de inicio de la aplicación (Figura 4.25).



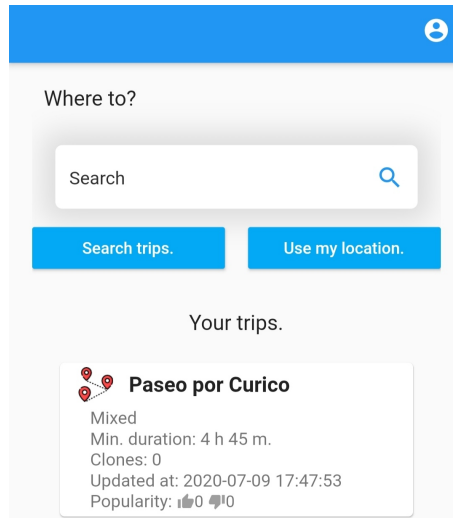


Figura 4.25: Acceso a la búsqueda de viajes.

Accediendo al viaje clonado, es posible realizar los ajustes deseados al viaje, tales como eliminar marcadores o añadir propios.

### Trazado de ruta

La segunda funcionalidad que es detallada es el trazado de la ruta de un viaje y la vista de sus detalles. Para ello, no es necesario poseer una cuenta en la aplicación, solo se necesita ingresar a un viaje creado por otro usuario como se realizó en la funcionalidad previamente explicada. Independiente de ello, una vez ingresado a un viaje, ya sea propio o de otro usuario, es posible elegir el medio de transporte, opcionalmente, por el cual se realizará el recorrido (Figura 4.26). Esta elección repercutirá en la generación de la ruta dado que los caminos entre estos varían. De no realizar este proceso, la aplicación cuenta con la opción, por omisión, de realizar el viaje a pie.

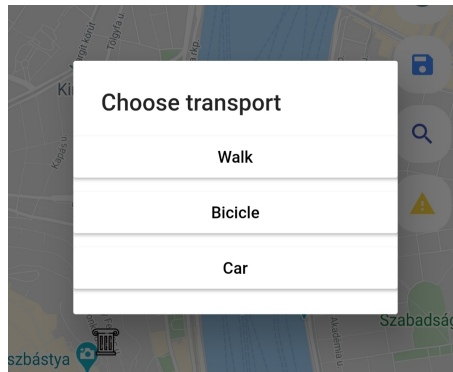


Figura 4.26: Selección del medio de transporte.

Una vez seleccionado el método de transporte, en caso de ser deseado, se debe pulsar el icono ubicado centro-inferior de la pantalla para trazar el camino. Al realizar esta acción surgirá una ventana emergente para que se seleccione el punto de partida para trazar el viaje (Figura 4.27).

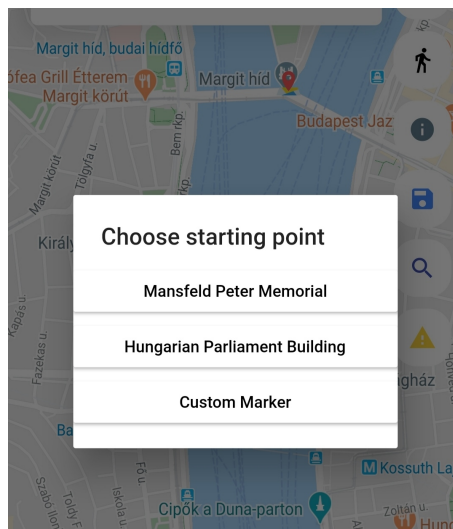


Figura 4.27: Selección del punto de partida.

Una vez hecha la selección la aplicación consultará al API por las distancias y tiempos entre cada uno de los puntos para determinar la ruta a dibujar. Se utiliza un *algoritmo greedy* que parte desde el punto seleccionado y salta al siguiente en base a la distancia entre los puntos. Cabe destacar que se descartó la utilización de algoritmos de cobertura mínima tales como Dijkstra y Kruskal, dado a que en ellos el resultado puede requerir

que el usuario vuelva a un punto previo para llegar al siguiente. Una vez finalizado este proceso, será dibujado en el mapa la ruta que pasa por los marcadores del viaje (Figura 4.28).

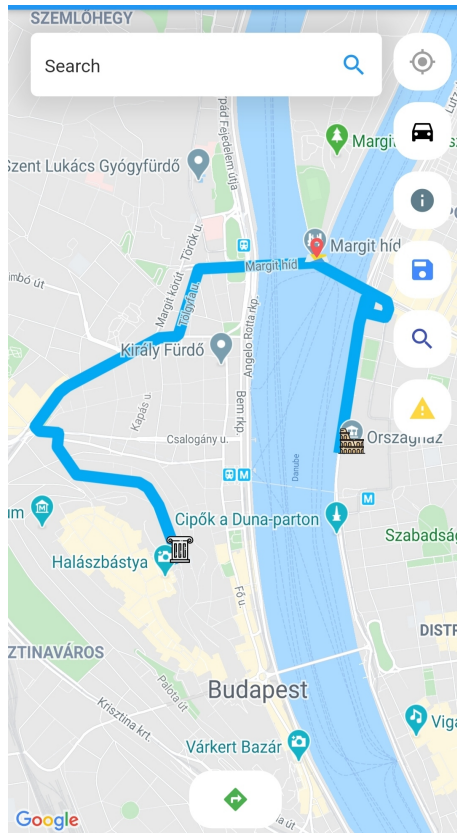


Figura 4.28: Trazado de la ruta de viaje.

Finalmente, con el viaje trazado, se procede a observar los detalles de la ruta generada, en el cual se indican las calles en donde se ubican los marcadores ingresados, el tiempo que se planea estar en ellos, la distancia entre la cual se ubican y cuanto tarda el recorrido entre uno lugar y el siguiente (Figura 4.29).

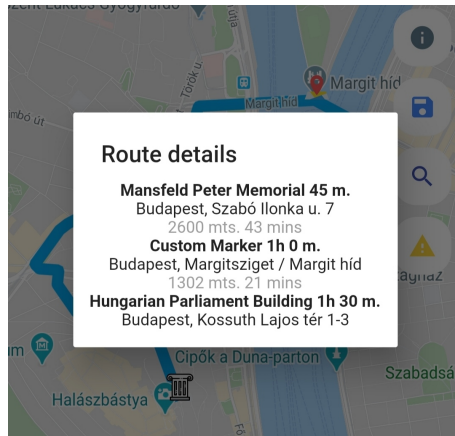


Figura 4.29: Información de la ruta.

### 4.3. Cierre del proyecto

Como resultado del desarrollo, se obtiene un archivo de instalación .apk para dispositivos Android. Con este archivo, es posible de instalar la aplicación al almacenarlo en el dispositivo y permitiendo la opción de seguridad correspondiente. Para utilizar la totalidad de las funcionalidades de la aplicación, se requiere que el dispositivo cuente con conexión a internet.

Para el despliegue del back-end se necesita una máquina con Docker instalado, que cuente con las siguientes especificaciones recomendadas:

- 2 Gb de RAM
- Un procesador de doble núcleo a 2GHz
- Sistema operativo Linux x64

Por otra parte, dado a que es posible generar una imagen en Docker conteniendo el back-end, este puede ser desplegado directamente en una plataforma en la nube que soporte el despliegue directo de imágenes de esta tecnología.

## 5. Evaluación de Journey Planner

---

En este capítulo se describe la aplicación de la metodología que evalúa el prototipo obtenido como resultado del desarrollo. Posteriormente, se realiza un análisis de los resultados obtenidos en contraste con los objetivos planteados para la evaluación.

### 5.1. Fases de Experimentación

Como es descrito en los antecedentes del proyecto, la ejecución de la experimentación es llevada mediante diferentes fases. En esta sección se describe como estas son llevadas a cabo indicando el propósito que estas cumplen y como es desarrollada cada fase.

#### 5.1.1. Definición

En primer lugar, como parte de la metodología se inicia con la fase de definición, en la cual se especifican los aspectos principales de la experimentación a realizar. Para ello se utilizó el método GQM<sup>1</sup> con el cual se facilita definir los siguientes aspectos como parte de la evaluación:

- **El objetivo del experimento**

Este corresponde a la evaluación de los objetivos propuestos inicialmente en la fase de definición del problema descrita con anterioridad.

- **El propósito del experimento**

Como propósito, el experimento busca conocer el nivel de satisfacción logrado como resultado del desarrollo de la aplicación desarrollada con respecto a las historias de usuario determinadas para su desarrollo.

---

<sup>1</sup>Goal-Question-Metric: Objetivo, pregunta y métrica.

- **El enfoque de calidad**

Este aspecto es evaluado con respecto a las funcionalidades que son ofrecidas con respecto a su usabilidad, funcionalidad, utilidad y completitud. Aspectos tales como la eficiencia, y confiabilidad, no serán evaluados debido a que no entran en el alcance definido para el proyecto.

- **La perspectiva y el contexto donde se realiza el experimento.**

El prototipo obtenido como resultado es evaluado por un grupo de diez sujetos los cuales son potenciales usuarios de la aplicación, en condiciones naturales para la aplicación. Esto es, disponibilidad en diferentes horarios y ubicaciones.

### **5.1.2. Diseño de la Experimentación**

Durante la fase de diseño, se especifican objetivos a un nivel mas bajo, los cuales serán utilizados para el modelado de la evaluación y las características de esta.

#### **Objetivos de evaluación**

Para el diseño de objetivos, son tomados como base el objetivo general y específicos del proyecto en conjunto con las historias de usuario definidas al inicio del desarrollo. Estas historias tienen como fin satisfacer este objetivo general, por lo cual permiten obtener información sobre el cumplimiento de este.

En base a ello, los objetivos determinados para la realización del experimento se encuentran indicados a continuación.

- Realizar la instalación de la aplicación en múltiples dispositivos móviles con sistema Android.
- Evaluar las funcionalidades asociadas con la planificación de un viaje y el trazado de su ruta.
- Recoger información sobre el funcionamiento y la apreciación de las características implementadas en el prototipo.

### Características de evaluación

Los aspectos de la aplicación a evaluar son indicados a continuación. Esta descripción corresponde a la entregada por la ISO/IEC 9126-1[7] para los apartados de Usabilidad y Funcionalidad, mientras para la Completitud es tomada como referencia la ISO/IEC 25000[21] y para la Utilidad la definición de la ISO/IEC 2382[22]. Ambos estándares tomados en consideración están asociados a la calidad en un producto de desarrollo de software.

- **Usabilidad:** Compete a la capacidad del software de ser comprendido, aprendido y utilizado de manera fácil e intuitiva
- **Funcionalidad:** Corresponde a la capacidad del software de cumplir con las necesidades requeridas y sus propiedades, siendo estas funciones las que satisfacen al conjunto de usuarios establecido o implícito.
- **Utilidad:** Es la cualidad en la que un software es capaz de proveer las solicitudes de los usuarios y continuar proveyendo el servicio bajo condiciones específicas.
- **Completitud:** Se refiere al grado de cumplimiento del producto en el cual se satisfacen los requerimientos y objetivos del usuario especificados.

### Instrumentos de Medición

Para llevar a cabo la experimentación se utilizará la plataforma para encuestas online Google Forms<sup>2</sup>. En ella, se ha diseñado un conjunto de sentencias que evalúan los aspectos mencionados en la sección previa. La distribución del formulario se encuentra dividida en cinco secciones, estructuradas de la siguiente manera:

- **Sección 1:** Datos generales del sujeto, tales como género, rango etario y ocupación.
- **Sección 2:** Datos sobre la utilización de la aplicación. Entre ellos:
  - Funcionalidad de gestión de usuario
  - Funcionalidad de planificación de viaje
  - Funcionalidad de trazado e información de la ruta

---

<sup>2</sup>Aplicación web para la realización y análisis de encuestas

- Funcionalidad de búsqueda de viajes
- Funcionalidad de información de eventos
- **Sección 3:** Evaluación sobre las funcionalidades implementadas.
  - Corrección del sistema
  - Utilidad del sistema
  - Facilidad de uso del sistema
- **Sección 4:** Apreciación general sobre la aplicación.
  - Cómo le ha parecido la aplicación con respecto a:
- **Sección 5:** Comentarios sobre la aplicación.

Las secciones correspondientes a la evaluación de la aplicación, vale decir, las secciones dos, tres y cuatro, son evaluadas mediante la escala Likert, la cual se utiliza para cuestionar el nivel de acuerdo o desacuerdo con respecto a una declaración [9]. Esta escala está compuesta por cinco niveles de satisfacción, de manera cualitativa como se muestra en Cuadro 5.1:

Cuadro 5.1: Escala de Likert.

Totalmente en desacuerdo	En desacuerdo	Ni en desacuerdo ni de acuerdo	De acuerdo	Totalmente de acuerdo
--------------------------	---------------	--------------------------------	------------	-----------------------

Con estas opciones, el sujeto que efectúa la evaluación debe señalar si está en desacuerdo o acuerdo con la sentencia correspondiente.

### 5.1.3. Ejecución de la Experimentación

Al momento de realizar la experimentación, es desplegado un contenedor de Docker con el back-end de la aplicación y una base de datos PostgreSQL en Azure para acceder a ella desde cualquier ubicación. Adicionalmente, se agregaron algunos viajes y eventos en la ciudad de Curicó para poder realizar en su totalidad las tareas señaladas. En la fase experimentación participan un total de diez personas, las cuales tienen edades que fluctúan entre los 22 y 25 años y cuentan con un manejo fluido en dispositivos móviles.



### **Protocolo de Experimentación**

La experimentación es planificada para ser realizada durante un periodo de dos semanas, en los cuales se les entrega la aplicación a los sujetos que hayan aceptado participar del experimento junto con una lista de actividades que deben ser desarrolladas con la aplicación. Además, a modo de ayuda, se preparó un video enseñando las funcionalidades de la aplicación, el cual puede ser utilizado como guía al realizar las actividades.

El conjunto de tareas para ser realizadas por los participantes del experimento es el siguiente:

- Registrar una cuenta.
- Añadir tres o más marcadores al mapa.
- Seleccionar un medio de transporte.
- Trazar una ruta entre los marcadores.
- Guardar un viaje y volver a abrirlo.
- Añadir un evento.
- Buscar un viaje realizado por otro usuario.
- Clonar un viaje buscado.
- Calificar un viaje o un evento.
- Añadir o eliminar marcadores en un viaje cargado.

### **Sesiones de Experimentación**

Como se mencionó con anterioridad, se dispone de un tiempo de dos semanas en los cuales el servidor de back-end y la base de datos se encontrarán activos para la realización de las pruebas. Durante este periodo de tiempo los usuarios pueden acceder a la aplicación y utilizar sus funcionalidades en cualquier momento para posteriormente llenar el formulario. Esto se debe a que en condiciones normales la aplicación debe ser funcional independiente de la ubicación del usuario y el tiempo en que es usada, siendo único requisito para que esta funcione la conexión a internet.

Una vez transcurrido el tiempo destinado para la realización de la experimentación, se procede a detener los servidores asociados a la aplicación.

#### 5.1.4. Análisis de la Experimentación

Ya finalizado el periodo de pruebas, se procede a analizar los resultados obtenidos durante la fase de experimentación. Para ello, se hacen uso de los gráficos proveídos por Google Forms. El análisis será centrado en las funcionalidades presentadas en el sistema, vale decir, las secciones dos, tres y cuatro.

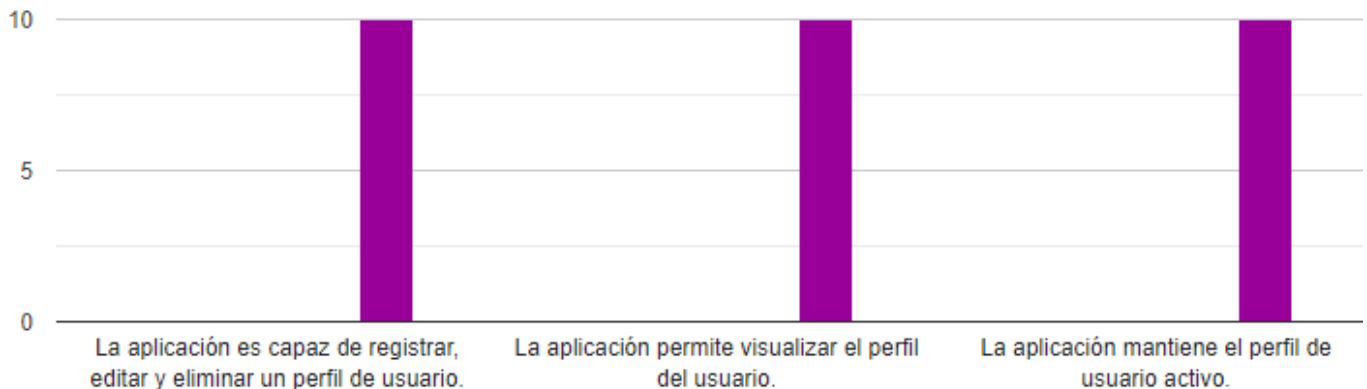
Los gráficos obtenidos como resultado de la experimentación poseen la siguiente escala de satisfacción:

- Totalmente de acuerdo
- De acuerdo
- Ni en desacuerdo ni de acuerdo
- En desacuerdo
- Totalmente en desacuerdo

#### Resultados Usuarios Registrados

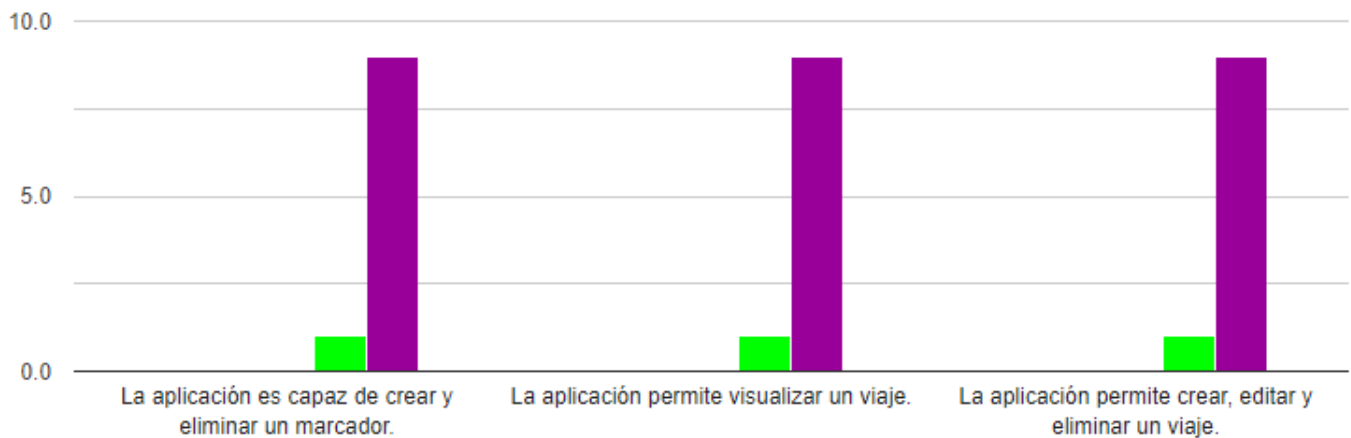
##### ■ Funcionalidad de gestión de usuario

En este apartado se obtiene un nivel de cumplimiento total en todos los aspectos presentados. Lo anterior indica que se han logrado el nivel de satisfacción deseado con respecto al registro, persistencia y gestión de un usuario, por lo cual esa funcionalidad no requiere de cambios.



■ **Funcionalidad de planificación de viajes**

En este apartado se obtiene un nivel de cumplimiento satisfactorio, habiendo aspectos a mejorar en cada una de las sentencias. De igual manera, en este aspecto se ha conseguido un nivel de satisfacción alto, pero existen aspectos en los cuales se puede mejorar. Se asocia esta respuesta particular a que al usuario le hubiese gustado recibir mayor retroalimentación a la hora de crear un viaje.



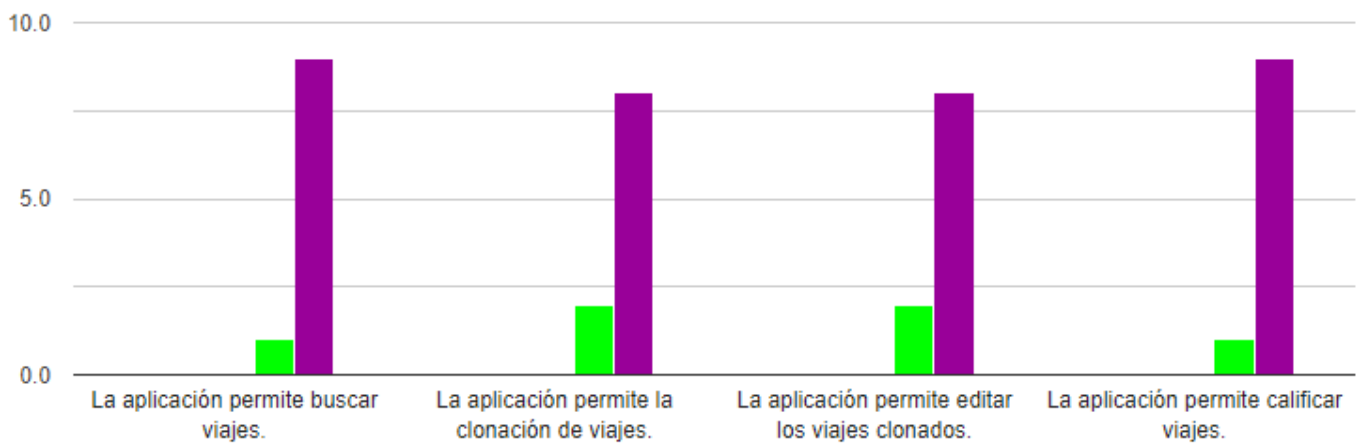
■ **Funcionalidad de trazado e información de la ruta**

En este apartado se obtiene un buen nivel de cumplimiento, contando con un cumplimiento total en la afirmación relacionada al medio de transporte. Por otra parte, cabe notar que en la afirmación relacionada a la totalidad de marcadores por los que pasa la ruta posee un nivel de satisfacción menor, lo cual puede estar relacionado a la posición de marcadores en puntos a los que la API a la cual se consulta no posee una ruta para el medio de transporte seleccionado.



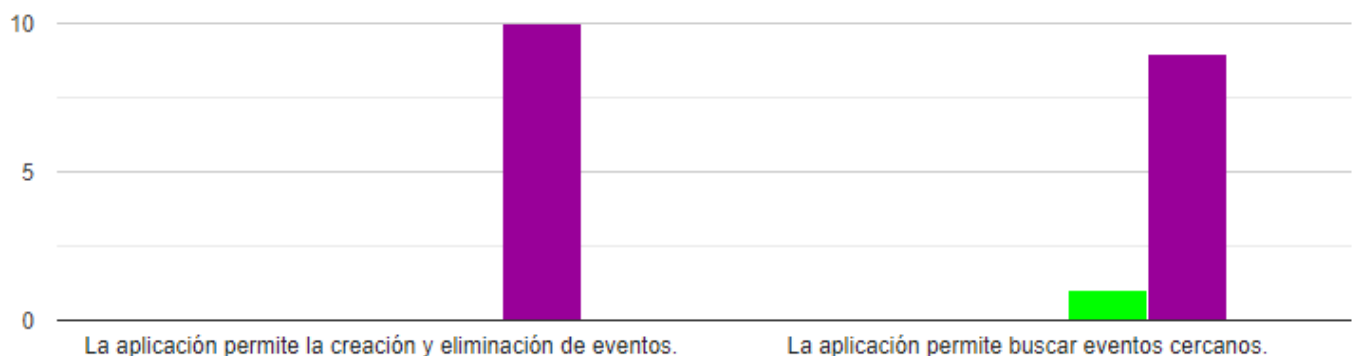
■ **Funcionalidad de búsqueda de viajes**

En este apartado se obtiene un nivel de cumplimiento bueno, con aspectos a mejorar en las diferentes afirmaciones. La información obtenida en este apartado indica que cierto grupo de usuarios presentaron cierta dificultad al momento de la búsqueda y clonación de viajes, por lo cual su nivel de satisfacción no es total. Al ver el detalle de las respuestas, se desprende que la funcionalidad para la clonación de viajes debiese ser encontrada de manera mas fácil.



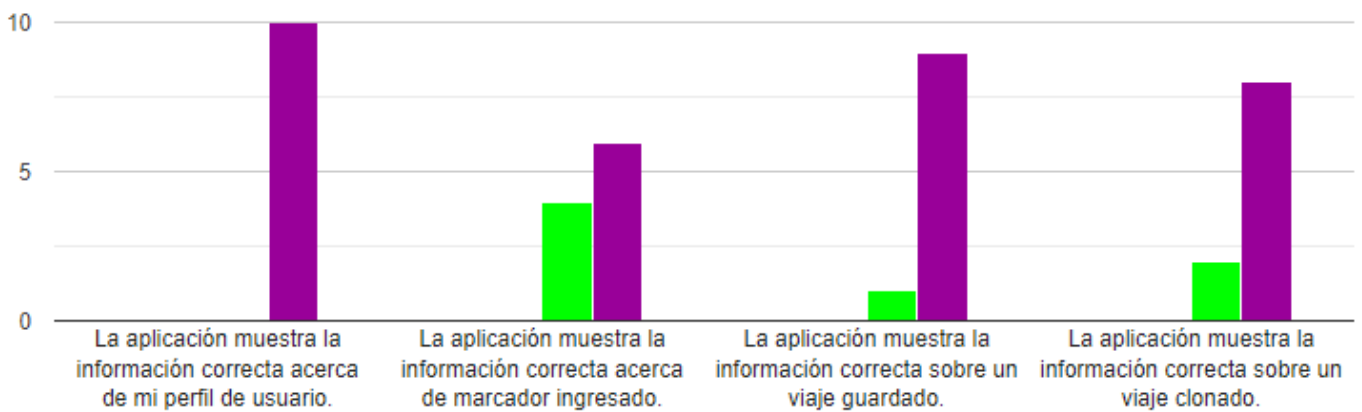
■ **Funcionalidad de información de eventos**

En este apartado se obtiene un buen nivel de cumplimiento aceptable, con un detalle el cual puede mejorar. Con respecto a las funciones de eliminación, estas funcionan con un nivel completo de satisfacción, mientras tanto, existe un usuario que tuvo problemas al buscar eventos en la ubicación en la cual se encontraba. Esto se debe a la falta de información ingresada en el sistema al momento de realizar la prueba, dado a que los eventos cuentan con una duración máxima de una semana.



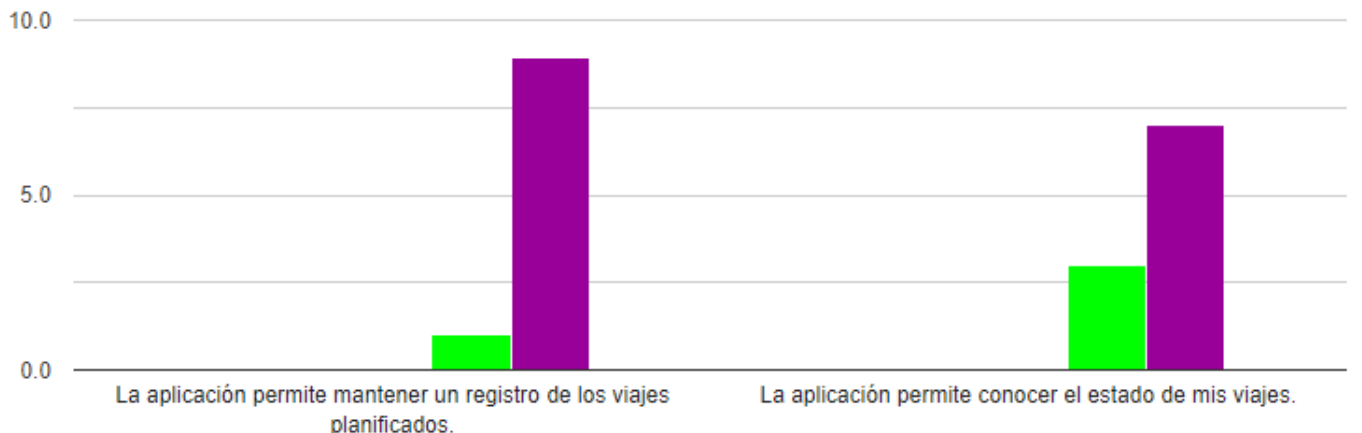
■ **Corrección del sistema**

En este apartado en general se obtiene un nivel aceptable de cumplimiento aceptable con la excepción de la información de un marcador, el cual presenta un nivel de aceptación mas bajo. Esto indica que en este aspecto se requiere un mayor detalle en el nivel de información mostrada. Este aspecto también indica que se debe revisar la funcionalidad que muestra la los detalles de un marcador en búsqueda de una posible falla en la información mostrada por la aplicación.



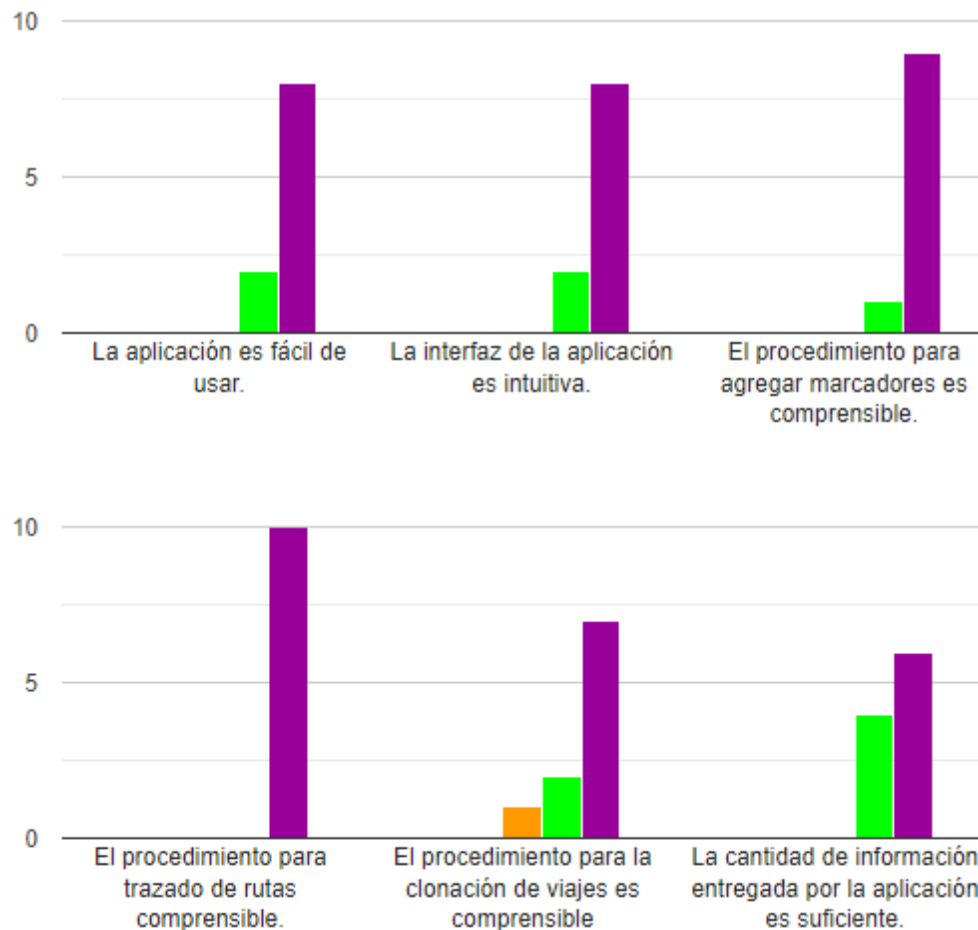
■ **Utilidad del sistema**

En este apartado se obtiene un buen nivel de cumplimiento aceptable, con aspectos a mejorar en el estado y la información de los viajes mostrada. Con respecto a la afirmación referente a la información del estado de los viajes es posible apreciar que a un grupo de usuarios les hubiese gustado tener mayor detalles en ellos, lo cual se complementa con los resultados obtenidos en la sección previa.



■ **Facilidad de uso del sistema**

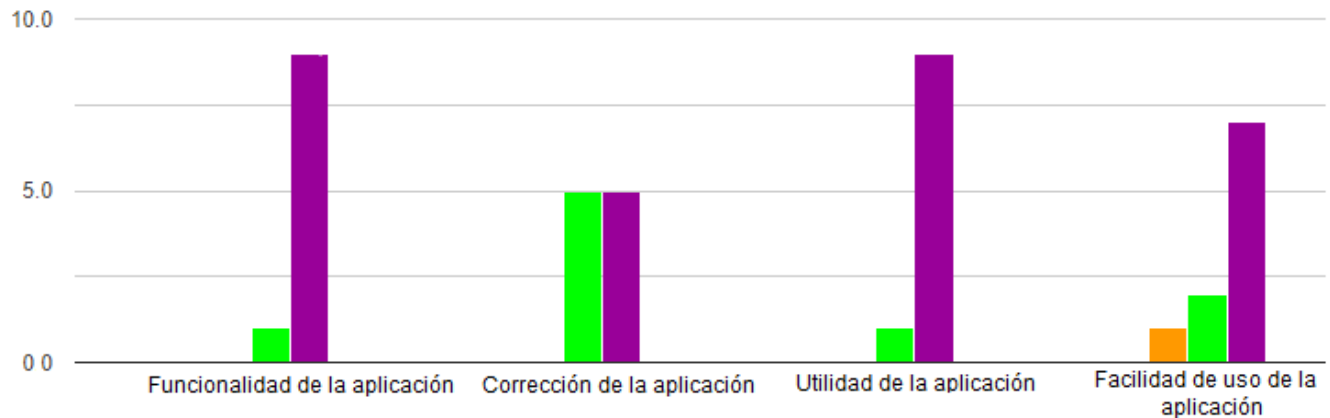
Esta sección presenta una respuesta favorable de parte de los usuarios, dejando un margen para la mejora de las funcionalidades y se logró un cumplimiento total en la comprensión para el trazado de rutas. Sin embargo, en el proceso de clonación de viajes se obtuvo una con menor grado de satisfacción, lo cual sugiere revisar la realización del proceso y las interacciones necesarias para realizar el procedimiento. De igual manera, en el ítem acerca de la suficiencia de la información mostrada, se observa una tendencia que indica que a una cantidad considerable de usuarios les gustaría tener mayor información en general.



■ **Cómo le ha parecido la aplicación con respecto a:**

En general la aplicación tuvo un desempeño aceptable en los aspectos evaluados. Las proporciones obtenidas en esta sección concuerdan con los resultados obtenidos

previamente, indicando un aspecto a mejorar es la cantidad de información entregada a los usuarios. Por otro lado en resultado asociado a la facilidad de uso guarda relación con el resultado obtenido sobre el procedimiento de clonación de viajes.



## 5.2. Revisión de resultados de experimentación

Como es posible observar en los gráficos presentados, existe una clara tendencia a mostrar el cumplimiento esperado de la aplicación, dejando la mayoría de respuestas inclinadas a estar totalmente de acuerdo con las sentencias propuestas.

Si bien, los resultados obtenidos son favorables, se puede apreciar que existen ámbitos que pueden ser mejorados. Lo que podemos encontrar en común entre estos, es que son asociados a los aspectos de la información presentada por el sistema, dejando a esta como un área se requiera de una revisión para agregar información complementaria o presentar la existente de diferente manera.

Con respecto a las funcionalidades como tal, se encuentra que se cumplieron con los objetivos propuestos para la experimentación. Con respecto al primero, los usuarios no tuvieron problemas con la instalación de la aplicación, por ende ese objetivo se cumplió de manera correcta. En segunda instancia el segundo y tercer objetivo también son satisfechos, permitiendo a los usuarios realizar las tareas propuestas y obteniendo la información deseada. Por ende, dados los resultados presentados, se considera que el prototipo desarrollado cumple con el propósito con el cual es desarrollado.

## 6. Conclusiones

---

En este capítulo se describen los resultados obtenidos por la realización del proyecto y sus resultados, además de aludir al trabajo futuro que es posible llevar a cabo con la propuesta.

### 6.1. Discusión

En el presente trabajo de proyecto de titulación es diseñada e implementada una aplicación para la planificación y el trazado de rutas turísticas urbanas las cuales pueden ser visualizadas por múltiples usuarios en diferentes contextos y dispositivos.

Haciendo uso de las metodologías y tecnologías seleccionadas para la implementación de la aplicación, las cuales sustentan sus funcionalidades, es posible la implementación de una aplicación con un modelo escalable a mayores características. El prototipo obtenido como resultado del desarrollo resulta útil para evaluar los objetivos propuestos por el proyecto.

Con respecto a los objetivos planteados como parte de este documento, se afirma que son satisfechos. A mayor detalle, siendo asociado con lo determinado el primer capítulo, dicta lo indicado a continuación:

- *Identificar en que consiste la planificación de una jornada de viaje urbano y sus tipos:* Esto es realizando en el **Capítulo 2**, al inicio de los antecedentes, en el cual se abordan los aspectos del dominio en el cual se enmarca el desarrollo.
- *Asistir al usuario mediante recomendaciones de ubicaciones y lugares para realizar un recorrido turístico:* Este objetivo forma parte del prototipo obtenido, por lo cual la información relacionada a ello se encuentra en el **Capítulo 4**.



- *Brindar una herramienta de geolocalización que permita al usuario conocer su posición al realizar su viaje:* De igual manera que el objetivo anterior, lo conseguido en relación a este objetivo se encuentra en el **Capítulo 4**.
- *Aplicar una metodología de desarrollo de software para construir la aplicación:* El detalle de este objetivo se encuentra en el **Capítulo 3 y 4**, en el cual se detalla como es adaptada la metodología utilizada durante el desarrollo.
- *Aplicar una metodología de evaluación a la aplicación para determinar la efectividad del proyecto:* Por último, el cumplimiento de este objetivo se encuentra en el **Capítulo 5**, en el cual se realiza la experimentación asociada al producto conseguido.

En base a lo anteriormente planteado, dado a que son cumplidos los objetivos específicos, por lo tanto se puede inferir que el objetivo general del sistema es satisfecho, el cual dicta de la siguiente manera: *Guiar el proceso de organización de una jornada de turismo mediante una aplicación móvil que brinde recomendaciones e información sobre aspectos asociados a la planificación.*

Por otro lado, con respecto al desarrollo realizado, el producto obtenido cuenta con las siguientes funcionalidades implementadas como resultado del prototipo:

- **Registro y gestión de un usuario:** Permite al sistema el registro de nuevos usuarios y el manejo de sus cuentas, así como también mantener la sesión y la persistencia de datos de la aplicación.
- **Seguimiento de ubicación:** Es el conjunto de funcionalidades que permiten al usuario conocer su ubicación actual en el mapa.
- **Creación y gestión de viajes:** Módulo el cual gestiona los aspectos relacionados a los viajes, tales como sus marcadores e información asociada a este.
- **Búsqueda de ubicaciones:** Son las funcionalidades de la aplicación que otorgan al usuario la alternativa de realizar una búsqueda de ubicaciones bajo su nombre o en una zona determinada.
- **Trazado de rutas:** Métodos asociados a la capacidad del software de comunicarse con el API de trazado de caminos dado un medio de transporte y enlazar una ruta

que, tomando en consideración un punto de partida, dibuje un camino pasando por los marcadores de un viaje utilizando como criterio de orden la distancia entre estos.

- **Búsqueda y clonación de viajes:** Es el módulo que provee la funcionalidad de realizar una búsqueda de viajes en una zona para conocer sus destinos y contar con la opción de clonarlos.
- **Creación y búsqueda de eventos:** Módulo que gestiona lo asociado con la búsqueda de eventos cercanos y su creación.

Como es ilustrado en la fase de experimentación, el prototipo que se obtiene como resultado del desarrollo presentó niveles de aceptación favorables en relación a los criterios evaluados. Por ende, es posible afirmar que el producto obtenido satisface los objetivos planteados inicialmente como parte de la memoria.

## **6.2. Lecciones aprendidas**

A pesar de la existencia de aplicaciones para la realización de viajes e información de trayectorias, los resultados obtenidos como parte de la experimentación permiten observar que existe la necesidad del desarrollo de herramientas con fines más específicos en relación al tema. Bajo esta misma lógica, en otros ámbitos es posible encontrar necesidades las cuales puedan derivar en la creación de aplicaciones o herramientas que resuelvan de manera mas eficiente la situación.

Por otra parte, con respecto al desarrollo, resulta importante la utilización de patrones de diseño y arquitectura propios de la ingeniería, al permitir el diseño de sistemas u aplicaciones con características modulares y escalables que permitan la inclusión de nuevas funcionalidades o características.

Con respecto a las metodologías aplicadas se considera que su elección es la adecuada para el tiempo disponible para el proyecto. También, asociado a la metodología de desarrollo, esta permitió que la implementación se haya podido adaptar a los cambios que se presentaron en el desarrollo, como los requisitos adicionales al final de las iteraciones. En relación a la metodología de evaluación, esta permitió conocer la apreciación de potenciales usuarios finales, lo cual no se hubiese podido obtener utilizando evaluaciones de código estático.

En el ámbito personal, se considera que el trabajo ha permitido validar los conocimientos adquiridos a lo largo del estudio, al tener que seleccionar y aplicar múltiples

conocimientos asociados al área de la Ingeniería en Computación. El resultado obtenido forma una buena base para la extensión del trabajo desarrollado y funcionalidades futuras.

### 6.3. Trabajo Futuro

Durante este proyecto son priorizados cierto conjunto de funcionalidades que sirven como base para la evaluación de la aplicación, sin embargo, es posible pensar en características adicionales que complementan el trabajo realizado. Dentro de estas funcionalidades, es posible extender la aplicación en los siguientes ámbitos:

- Herramientas de accesibilidad para usuarios con discapacidades visuales, tales como un modo de alto contraste o retroalimentación utilizando sonidos o vibraciones.
- Capacidad de subir fotos de los viajes para hacer que estos sean mas atractivos al momento de realizar una búsqueda.
- Creación de un registro de lugares propio de la aplicación, desligándola de la utilización del API de consulta de lugares.
- Migración a una tecnología de base de datos en tiempo real que permita ofrecer unos tiempos de respuesta mas rápidos.
- Almacenamiento de información en la memoria del dispositivo, con el fin de acceder a los viajes sin requerir de una conexión a la base de datos.
- Implementación de la funcionalidad que permita priorizar el orden de los marcadores en base al tiempo entre ellos.
- Implementación de una funcionalidad que permita al usuario elegir el orden de los marcadores
- Generación de múltiples rutas para ser presentadas al usuario.
- Implementación de un ranking de ubicaciones que permita a los usuarios recomendar hoteles, restaurantes, bares y otros, de manera personalizada.
- Utilización de los datos de genero y fecha de nacimiento para permitir a los usuarios la búsqueda de recorridos populares entre usuarios similares.

- Implementación de recomendación de lugares en base a la distancia del punto de búsqueda en el cual se busca.

# Bibliografía

- [1] Stack overflow developer survey 2020.
- [2] Martin Bean. *Laravel 5 essentials*. Packt Publishing Ltd, 2015.
- [3] Sufyan bin Uzayr, Nicholas Cloud, and Tim Ambler. Pm2. In *JavaScript Frameworks for Modern Web Development*, pages 61–87. Springer, 2019.
- [4] Pankti Doshi, Pooja Jain, and Abhishek Shakwala. Location based services and integration of google maps in android. *International Journal of Engineering and Computer Science*, 3(03), 2014.
- [5] Korry Douglas and Susan Douglas. *PostgreSQL: a comprehensive guide to building, programming, and administering PostgreSQL databases*. SAMS publishing, 2003.
- [6] Yani Dzhurov, Iva Krasteva, and Sylvia Ilieva. Personal extreme programming—an agile process for autonomous developers. 2009.
- [7] International Organization for Standardization and International Electrotechnical Commission. *Software Engineering-Product Quality*, volume 9126. ISO/IEC, 2001.
- [8] Omar Gómez, Juan Ucán, and Gerzon Gómez. Aplicación del proceso de experimentación a la ingeniería de software. *Abstraction and Application Magazine*, 8, 2014.
- [9] Ankur Joshi, Saket Kale, Satish Chandel, and D Kumar Pal. Likert scale: Explored and explained. *Current Journal of Applied Science and Technology*, pages 396–403, 2015.
- [10] Dennis R Judd. El turismo urbano y la geografía de la ciudad. *EURE (Santiago)*, 29(87):51–62, 2003.

- [11] Mika Kuitunen. Cross-platform mobile application development with react native. B.S. thesis, 2019.
- [12] Azat Mardan. Using express. js to create node. js web apps. In *Practical Node. js*, pages 51–87. Springer, 2018.
- [13] Dirk Merkel. Docker: lightweight linux containers for consistent development and deployment. *Linux journal*, 2014(239):2, 2014.
- [14] Laurence Moroney. The firebase realtime database. In *The Definitive Guide to Firebase*, pages 51–71. Springer, 2017.
- [15] Diego Navarro. Recursos turísticos y atractivos turísticos: conceptualización, clasificación y valoración. *Cuadernos de turismo*, (35):335–357, 2015.
- [16] Andy Opper and Robert Sheldon. *SQL: A BEGINNER'S GUIDE 3/E*. McGraw-Hill, Inc., 2008.
- [17] Prakash Patel. A guide to material design, a modern software design language. 04 2016.
- [18] Rap Payne. Developing in flutter. In *Beginning App Development with Flutter*, pages 9–27. Springer, 2019.
- [19] Carlos Andrés Pinassi and Patricia Susana Ercolani. El turismo urbano-metropolitano y su configuración espacial en las ciudades. análisis de la focalidad urbana en bahía blanca (argentina). 2012.
- [20] Javier Hernández Ramírez. Los caminos del patrimonio. rutas turísticas e itinerarios culturales. *Pasos. Revista de turismo y patrimonio cultural*, 9(2):225–236, 2011.
- [21] Paola Andrea Roa, Claribel Morales, and Patricia Gutiérrez. Norma iso/iec 25000. *Tecnología Investigación y Academia*, 3(2):27–33, 2015.
- [22] MJ Standard. Information technology-vocabulary-part 1: Fundamental terms (iso/iec 2382-1: 1993, idt). *Malaysian standard. Department of Standards Malaysia*, 2005.
- [23] Stefan Tilkov and Steve Vinoski. Node. js: Using javascript to build high-performance network programs. *IEEE Internet Computing*, 14(6):80–83, 2010.
- [24] Frank Zammetti. Server-side action: Node and npm. In *Modern Full-Stack Development*, pages 1–20. Springer, 2020.

# **ANEXOS**

## **A. Descripción de las tareas realizadas por cada iteración**

---

En este anexo se encuentra una breve descripción de las tareas realizadas en cada iteración. Como es descrito en el cuerpo del documento, su el conjunto de tareas correspondientes a cada iteración es realizado en un periodo de dos semanas.



## A.1. Iteración 1

Cuadro A.1: Descripción de las tareas de la Iteración 1.

Iteración 1	
Tarea	Descripción
Diseño del diagrama de clases	Creación de la primera versión del diagrama UML que dicta la estructura de clases en la aplicación.
Diseño del Modelo relacional	Creación de la primera versión del modelo relacional que llevará la base de datos.
Despliegue del servidor con CI/CD en Azure	Configuración del entorno de desarrollo de pruebas en el cual es desplegado el servidor de back-kend.
Despliegue de la base de datos en ElephantSql	Carga del script de base de datos en la plataforma para alojar base de datos ElephantSql.
Diseño de mock-ups	Diseño de las interfaces y la navegación básica.
Configuración plugin de Google Maps	Instalación del plugin de Google Maps para Flutter y realización de pruebas para probar sus funcionalidades.
Back-end registro de usuario	Implementación del módulo que registra usuarios en la base de datos.
Back-end ingreso de usuario	Implementación del módulo verifica los datos de ingreso de un usuario
Agregar un marcador	Implementación de la funcionalidad para agregar un marcador simple al mapa.
Eliminar un marcador	Implementación de la funcionalidad que permite eliminar un marcador seleccionado.
Cálculo de distancia entre marcadores	Implementación de la función que consulta por la distancia y la trayectoria entre dos marcadores.
Centrar la cámara en el usuario	Implementación de la funcionalidad que consulta por la ubicación del usuario y centra la vista del mapa.
Vista de registro de usuario	Implementación de la interfaz de registro de usuario.
Vista de inicio de sesión	Implementación de la interfaz de inicio de sesión.
Vista de guardar viaje	Implementación de la interfaz que permite ingresar la información de un viaje y guardarlo.
Back-end guardar viaje	Implementación del módulo que permite guardar un viaje asociado a un usuario y sus respectivos marcadores.
Buscar e ir a una ubicación	Implementación de la funcionalidad que permite buscar una ubicación por nombre y centrar la cámara en ella.
Persistencia de datos	Implementación de Redux para mantener el inicio de sesión y mantener los datos del usuario activos.

## A.2. Iteración 2

Cuadro A.2: Descripción de las tareas de la Iteración 2.

Iteración 2	
Tarea	Descripción
Listar ubicaciones cercanas	Implementación de la funcionalidad que consulta por puntos de interés cercanos dada una categoría seleccionada por el usuario.
Obtener detalles de una ubicación	Implementación de la consulta al API de Google Places que obtiene los detalles de un lugar dado su identificador.
Agregar el marcador de una búsqueda	Implementación de la vista y la funcionalidad que agregar un marcador al mapa dado el resultado de una búsqueda de lugar.
Selección del punto de partida	Implementación de la vista con un listado de marcadores en los cual se selecciona el punto de partida de la ruta.
Obtención de distancias y tiempos entre los marcadores	Comunicación con el API de Google Maps para obtener los tiempos y distancias entre un conjunto de puntos.
Selección del medio de transporte	Implementación de la ventana emergente que permite seleccionar el transporte del viaje
Implementación del algoritmo para generar la ruta de viaje	Codificación del algoritmo que determina un camino entre los marcadores en base a su distancia y posteriormente arma la ruta tomando en consideración el transporte elegido.
Cálculo del tiempo total de viaje	En base a la ruta generada se obtiene el tiempo que se tarda en recorrer los marcadores.

### A.3. Iteración 3

Cuadro A.3: Descripción de las tareas de la Iteración 3.

Iteración 3	
Tarea	Descripción
Back-end para agregar eventos	Implementación del módulo recibe los datos de un evento desde la aplicación y lo inserta en la base de datos.
Back-end para búsqueda de viajes	Implementación del módulo que realiza una búsqueda entre los viajes activos dada una coordenada y lista los resultados de la búsqueda en base a su valoración.
Agregar eventos al mapa	Implementación de la vista y la funcionalidad que permite agregar un evento al mapa.
Vista de agregar evento	Implementación de la vista en la cual se selecciona el tipo de evento y se agrega información relacionada a él.
Vista de resultados de búsqueda de viajes	Implementación de interfaz en la cual se muestran los resultados de la búsqueda de viajes.
Clonación de un viaje	Implementación de la funcionalidad que permite al usuario clonar un viaje de otro usuario, creando uno nuevo asociado a su cuenta.

**A.4. Iteración 4**

Cuadro A.4: Descripción de las tareas de la Iteración 4.

Iteración 4	
Tarea	Descripción
Back-end para edición de viajes y marcadores	Implementación del módulo permite modificar los datos de un viaje recibidos desde la aplicación.
Back-end calificación de eventos y viajes	Implementación del módulo que permite asignar la calificación por parte del usuario a un determinado viaje o evento.
Fijar la cámara al usuario	Implementación la funcionalidad que permite al usuario fijar la cámara en su posición.
Back-end para la eliminación de un viaje y un usuario	Implementación del módulo que permite eliminar viajes y usuarios.
Vista de edición de un viaje y un usuario	Implementación de las interfaces en las cuales se permiten la edición de un viaje o un usuario.
Widget de clasificación para viajes y eventos	Implementación de la funcionalidad que permite al usuario calificar un viaje o evento.

## **B. Resultados de la encuesta**

---

En este anexo se presentan los resultados individuales obtenidos al aplicar la encuesta de evaluación de la aplicación.

Para evitar tablas con texto redundante, las opciones son reemplazadas por valores numéricos, siendo la siguiente la correspondencia:

- 5 = Totalmente de acuerdo
- 4 = De acuerdo
- 3 = Ni en desacuerdo ni de acuerdo
- 2 = En desacuerdo
- 1 = Totalmente en desacuerdo

### **B.1. Formulario**

En esta sección se presenta el formulario utilizado para llevar a cabo la encuesta de evaluación.

#### **B.1.1. Datos generales**

- a.- Edad
- b.- Género
- c.- Ocupación

### **B.1.2. Sección 2**

#### **Funcionalidad de gestión de usuario**

- a.- La aplicación es capaz de registrar, editar y eliminar un perfil de usuario.
- b.- La aplicación permite visualizar el perfil del usuario.
- c.- La aplicación mantiene el perfil de usuario activo.

#### **Funcionalidad de planificación de viaje**

- a.- La aplicación es capaz de crear y eliminar un marcador.
- b.- La aplicación permite visualizar un viaje.
- c.- La aplicación permite crear, editar y eliminar un viaje.

#### **Funcionalidad de trazado e información de la ruta**

- a.- La aplicación permite trazar una ruta entre los marcadores
- b.- La ruta trazada pasa por todos los marcadores
- c.- La ruta trazada considera el medio de transporte seleccionado
- d.- El punto de partida escogido para la ruta es correcto

#### **Funcionalidad de búsqueda de viajes**

- a.- La aplicación permite buscar viajes.
- b.- La aplicación permite la clonación de viajes.
- c.- La aplicación permite editar los viajes clonados.
- d.- La aplicación permite calificar viajes.

#### **Funcionalidad de información de eventos**

- a.- La aplicación permite la creación y eliminación de eventos.
- b.- La aplicación permite buscar eventos cercanos.

### **B.1.3. Sección 3**

#### **Corrección del sistema**

- a.- La aplicación muestra la información correcta acerca de mi perfil de usuario.
- b.- La aplicación muestra la información correcta acerca de marcador ingresado.
- c.- La aplicación muestra la información correcta sobre un viaje guardado.
- b.- La aplicación muestra la información correcta sobre un viaje clonado.

#### **Utilidad del sistema**

- a.- La aplicación permite mantener un registro de los viajes planificados.
- b.- La aplicación permite conocer el estado de mis viajes.

#### **Facilidad de uso del sistema**

- a.- La aplicación es fácil de usar.
- b.- La interfaz de la aplicación es intuitiva.
- c.- El procedimiento para agregar marcadores es comprensible.
- d.- El procedimiento para trazado de rutas comprensible.
- e.- El procedimiento para la clonación de viajes es comprensible.
- d.- La cantidad de información entregada por la aplicación es suficiente.

### **B.1.4. Sección 4**

#### **Cómo le ha parecido la aplicación con respecto a:**

- a.- Funcionalidad de la aplicación.
- b.- Corrección de la aplicación.
- c.- Utilidad de la aplicación.
- d.- Facilidad de uso de la aplicación.

**B.1.5. Sección 5**

- a.- ¿La aplicación presentó algún error al momento de utilizarse?
- b.- ¿Qué cambios o mejoras le haría usted la aplicación?
- c.- ¿Utilizaría la aplicación para planificar un recorrido turístico a una ciudad?

**B.2. Respuestas****B.2.1. Sección 1**

Cuadro B.1: Datos generales

Participante	a	b	c
1	22 - 25 años	Masculino	Estudiante
2	22 - 25 años	Masculino	Estudiante
3	22 - 25 años	Masculino	Estudiante
4	22 - 25 años	Masculino	Estudiante
5	22 - 25 años	Masculino	Estudiante
6	22 - 25 años	Masculino	Estudiante
7	22 - 25 años	Femenino	Publicista
8	22 - 25 años	Masculino	Estudiante
9	22 - 25 años	Masculino	Estudiante
10	22 - 25 años	Masculino	Estudiante



**B.2.2. Sección 2**

Cuadro B.2: Funcionalidad de gestión de usuario

Participante	a	b	c
1	5	5	5
2	5	5	5
3	5	5	5
4	5	5	5
5	5	5	5
6	5	5	5
7	5	5	5
8	5	5	5
9	5	5	5
10	5	5	5

Cuadro B.3: Funcionalidad de planificación de viaje

Participante	a	b	c
1	5	5	5
2	5	4	4
3	4	5	5
4	5	5	5
5	5	5	5
6	5	5	5
7	5	5	5
8	5	5	5
9	5	5	5
10	5	5	5

Cuadro B.4: Funcionalidad de trazado e información de la ruta

Participante	a	b	c	d
1	5	5	5	5
2	4	4	5	5
3	5	5	5	5
4	5	5	5	5
5	5	4	5	4
6	5	4	5	5
7	5	5	5	5
8	4	5	5	5
9	5	5	5	5
10	5	5	5	5

Cuadro B.5: Funcionalidad de búsqueda de viajes

Participante	a	b	c	d
1	5	5	5	5
2	5	5	5	4
3	5	5	5	5
4	5	5	5	5
5	4	4	5	5
6	5	5	5	5
7	5	5	4	5
8	5	4	4	5
9	5	5	5	5
10	5	5	5	5

**B.2.3. Sección 3**

Cuadro B.6: Corrección del sistema

Participante	a	b	c	d
1	5	5	5	5
2	5	5	4	5
3	5	5	5	5
4	5	5	5	5
5	5	4	5	5
6	5	5	5	5
7	5	5	5	5
8	5	4	5	4
9	5	4	5	4
10	5	4	5	5

Cuadro B.7: Utilidad del sistema

Participante	a	b
1	5	5
2	4	4
3	5	5
4	5	5
5	5	4
6	5	5
7	5	5
8	5	5
9	5	4
10	5	5

Cuadro B.8: Facilidad de uso del sistema

Participante	a	b	c	d	e	f
1	5	5	5	5	5	5
2	4	4	5	5	5	4
3	5	5	5	5	5	5
4	4	4	5	5	4	4
5	5	5	4	5	4	5
6	5	5	5	5	5	5
7	5	5	5	5	5	4
8	5	5	5	5	3	5
9	5	5	5	5	5	4
10	5	5	5	5	5	5

**B.2.4. Sección 4**

Cuadro B.9: Cómo le ha parecido la aplicación con respecto a

Participante	a	b	c	d
1	5	5	5	5
2	5	4	5	4
3	4	4	5	5
4	5	4	5	3
5	5	4	5	5
6	5	5	5	5
7	5	5	5	5
8	5	5	4	4
9	5	4	5	5
10	5	5	5	5

**B.2.5. Sección 5**

Cuadro B.10: Cómo le ha parecido la aplicación con respecto a

#	a	b	c
1	Ninguno	Agregaría otros idiomas en la interfaz	Por supuesto
2	Solo un problema con el tiempo de una parte de un trazado.	Pondría el poder seleccionar un punto de llegada, como algún hotel u alojamiento. Dentro de ese contexto, también que pudiese recomendar los mejores alojamientos según la puntuación de los usuarios. Otra idea sería el poder agregar subrecorridos a los marcadores, por ejemplo, algún sendero que se encuentre dentro de un marcador. Por último, faltaría el poder agendar viajes y poder agregar una fecha específica para un evento.	Sí
3	Solo un pequeño error en los tiempos de guardado.	Hacer mas notorias algunas acciones como el agregar marcadores	si, por supuesto.
4	- Al buscar viajes, el el botón 'Use my location' no encontró ninguno.	Al buscar viajes, debería especificar que lo que se ingresa en la barra de búsqueda debe ser el nombre de la ciudad en lugar del nombre de un viaje. 'Use my location' podría renombrarse a 'Buscar en las cercanías' o algo parecido.	Definitivamente.
5	No	Diferentes colores	Si
6	No	Inclusión de colores para daltónicos	Por supuesto
7	No	Que al pulsar un lugar aparezca su información	Si
8	No	Una pantalla principal mas atrayente	Si
9	No	Mostrar mas imágenes de las ubicaciones	Si
10	No	Mostrar mas información de los lugares	Si

## **C. Interfaces de la aplicación**

---

En este apéndice se presentan las vistas que componen la interfaz de la aplicación, separadas por secciones.

### C.1. Vistas de inicio y búsqueda.

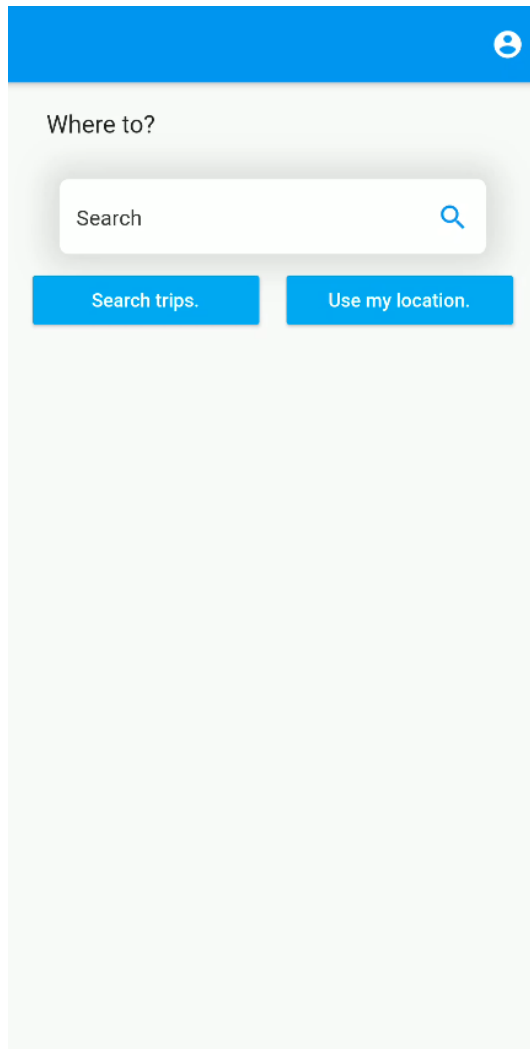


Figura C.1: Pantalla de inicio vacía.

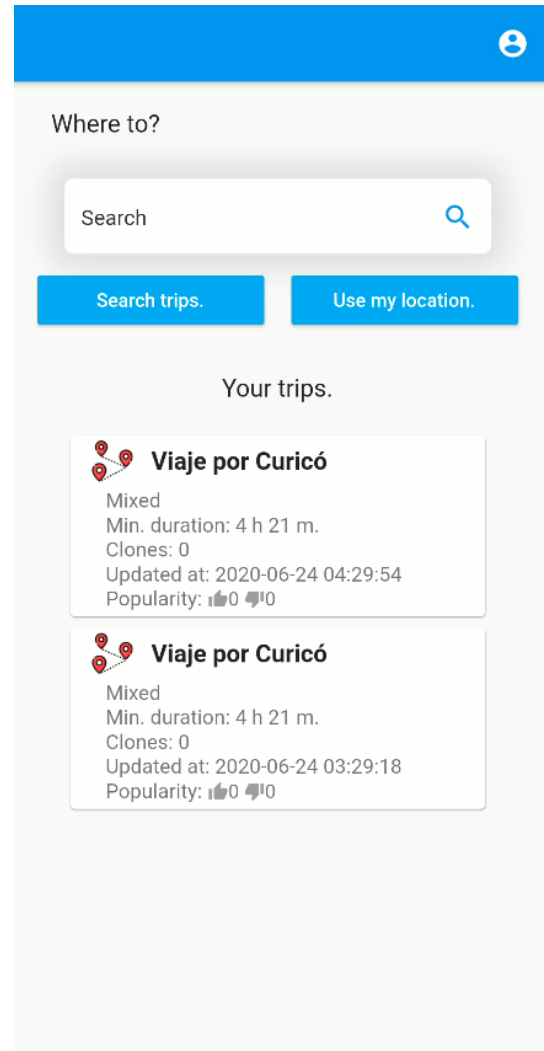


Figura C.2: Pantalla de inicio con viajes.

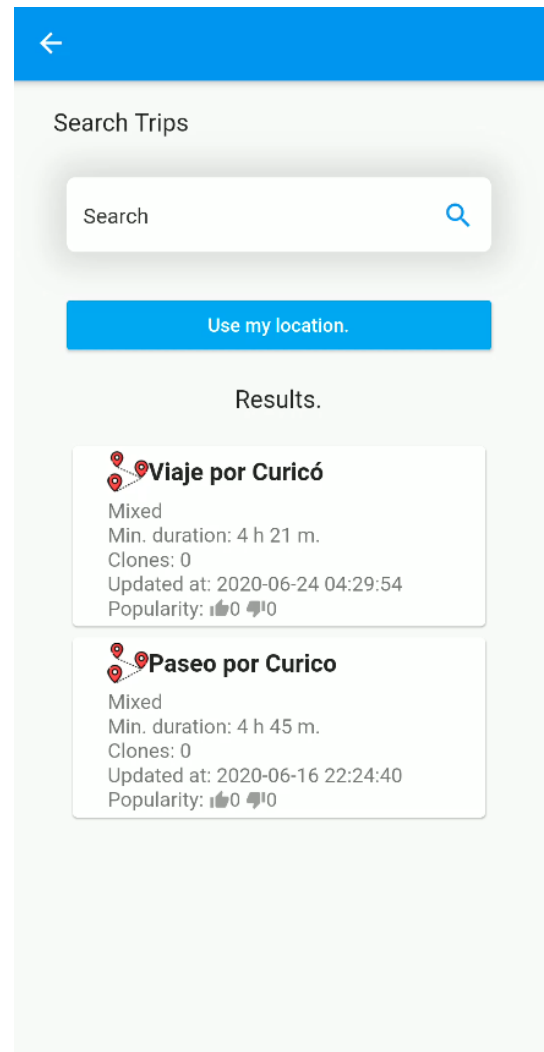
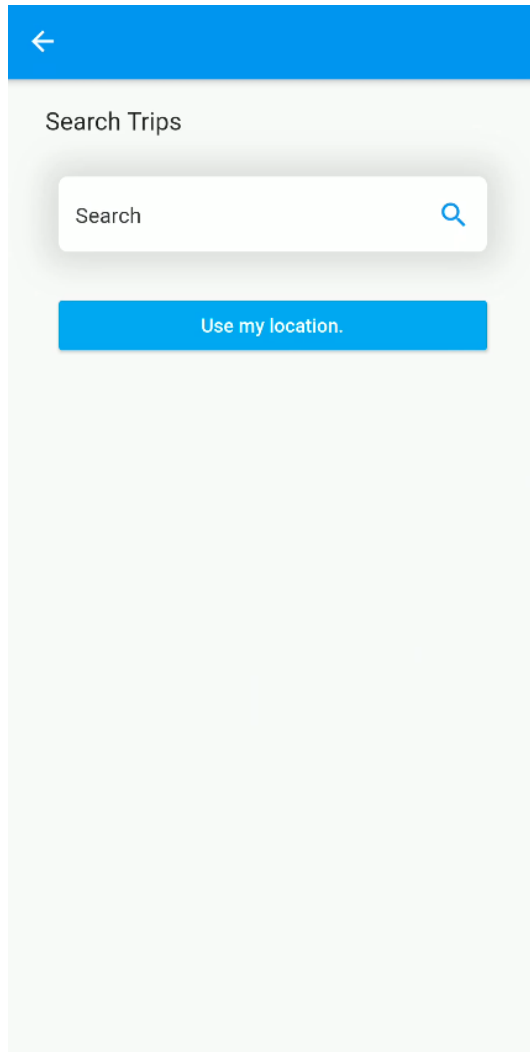


Figura C.3: Pantalla de búsqueda de viajes  
Figura C.4: Pantalla de búsqueda de viajes con resultados.



## C.2. Vistas de gestión de usuario.

email

password

Login

Register

Figura C.5: Pantalla de inicio de sesión.

email

password

confirm password

name

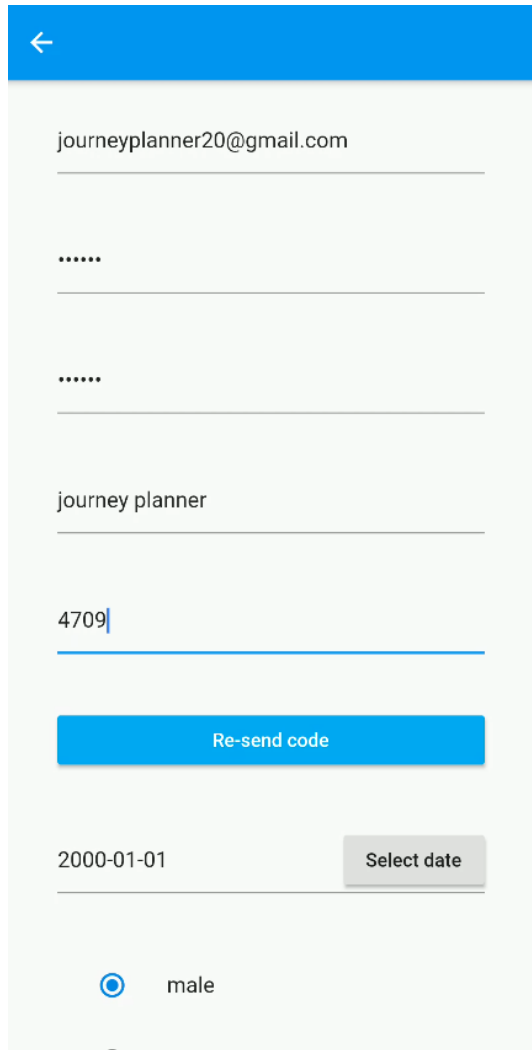
birthday Select date

male

female

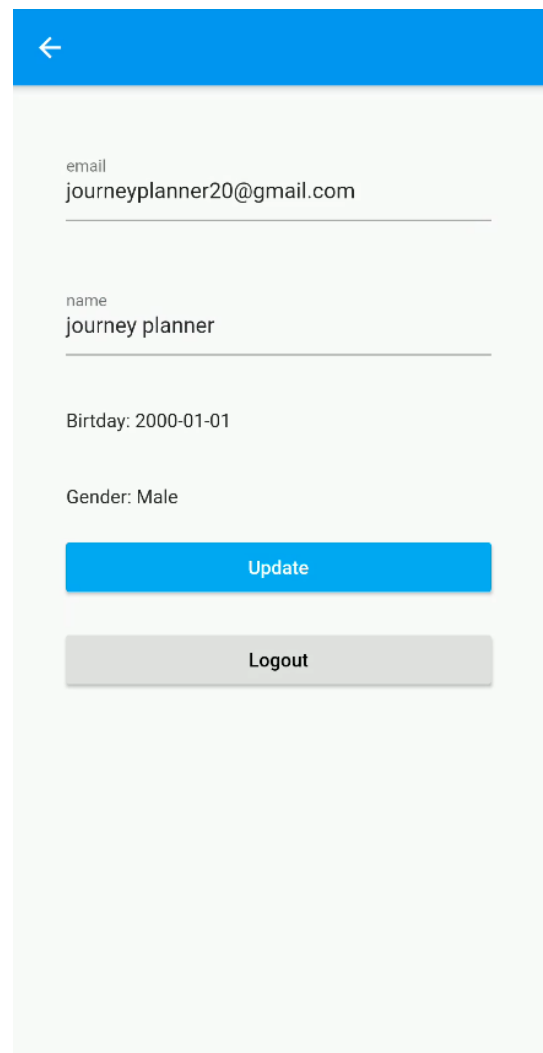
Send

Figura C.6: Pantalla de registro de usuario.



The registration screen features a blue header with a back arrow. It contains several input fields: an email field with 'journeyplanner20@gmail.com', two masked password fields (each with six dots), a name field with 'journey planner', a verification code field with '4709', and a date field with '2000-01-01' and a 'Select date' button. A 'Re-send code' button is positioned below the code field. At the bottom, there is a radio button selected for 'male'.

Figura C.7: Pantalla de registro con código de verificación.



The user data screen has a blue header with a back arrow. It displays the user's information: email 'journeyplanner20@gmail.com', name 'journey planner', birthday '2000-01-01', and gender 'Male'. There are two buttons at the bottom: a blue 'Update' button and a grey 'Logout' button.

Figura C.8: Pantalla de datos del usuario.

The screenshot shows a mobile application interface for editing a user profile. At the top, there is a blue header with a white back arrow. Below the header, the form contains the following elements: a text input field for 'name' with the value 'journey planner'; a text input field for 'birthday' with the value '1999-12-31' and a grey 'Select date' button to its right; two radio button options for gender: 'male' (selected) and 'female'; a blue 'Save Changes' button; a grey 'Change password' button; and a red 'Delete user' button.

Figura C.9: Pantalla de edición de usuario.

The screenshot shows a mobile application interface for changing a password. At the top, there is a blue header with a white back arrow. Below the header, the form contains three text input fields: 'old password', 'new password', and 'confirm new password'. At the bottom of the form, there is a blue 'Save Changes' button.

Figura C.10: Pantalla de cambio de contraseña.

### C.3. Mapa.

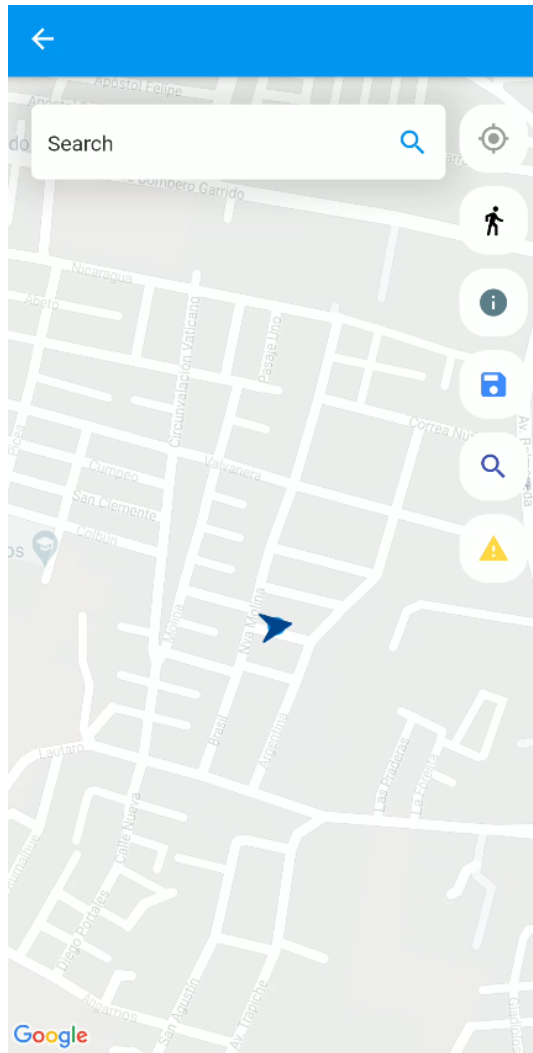


Figura C.11: Mapa centrado en la posición del usuario.

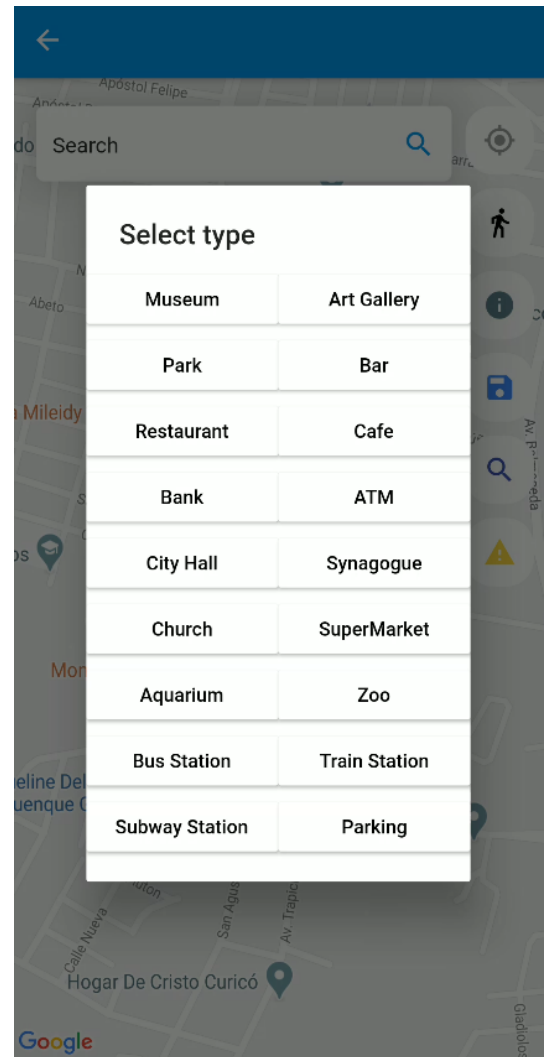


Figura C.12: Búsqueda por categoría.

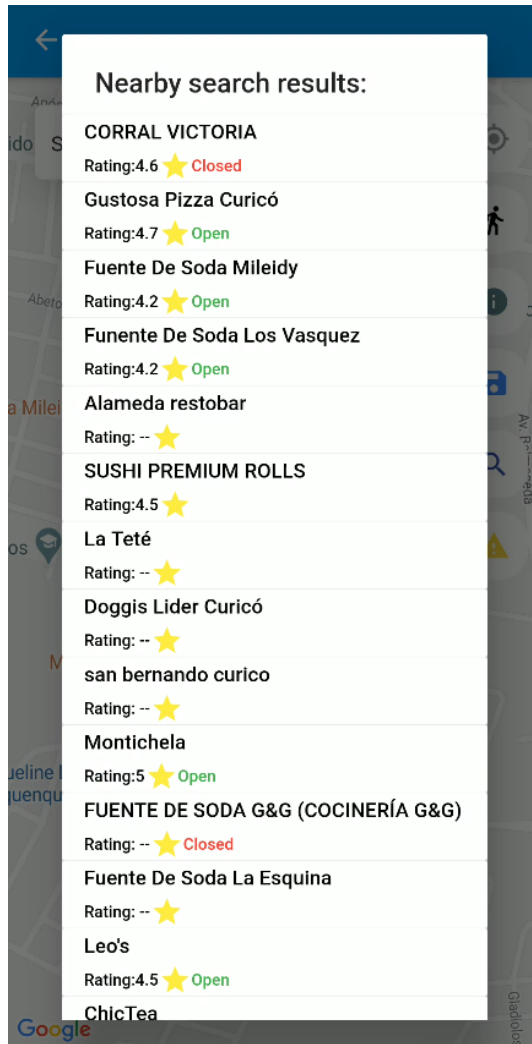


Figura C.13: Resultado de la búsqueda de ubicaciones.

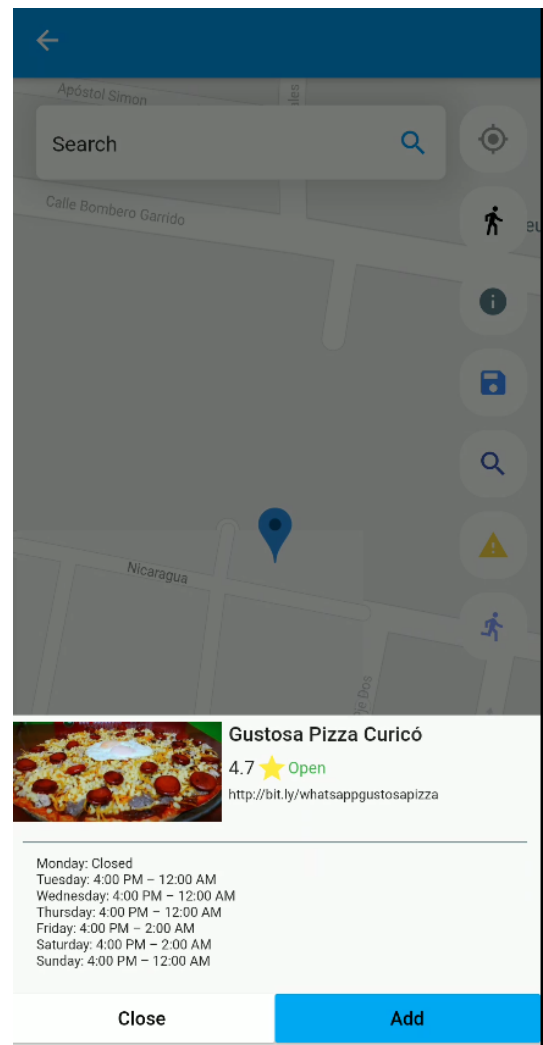


Figura C.14: Detalles de una ubicación seleccionada.

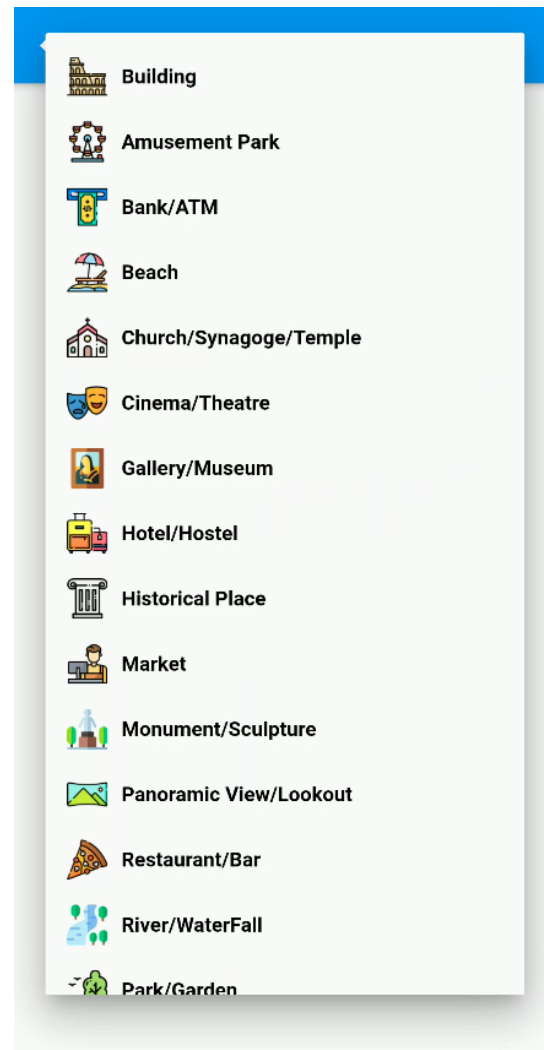
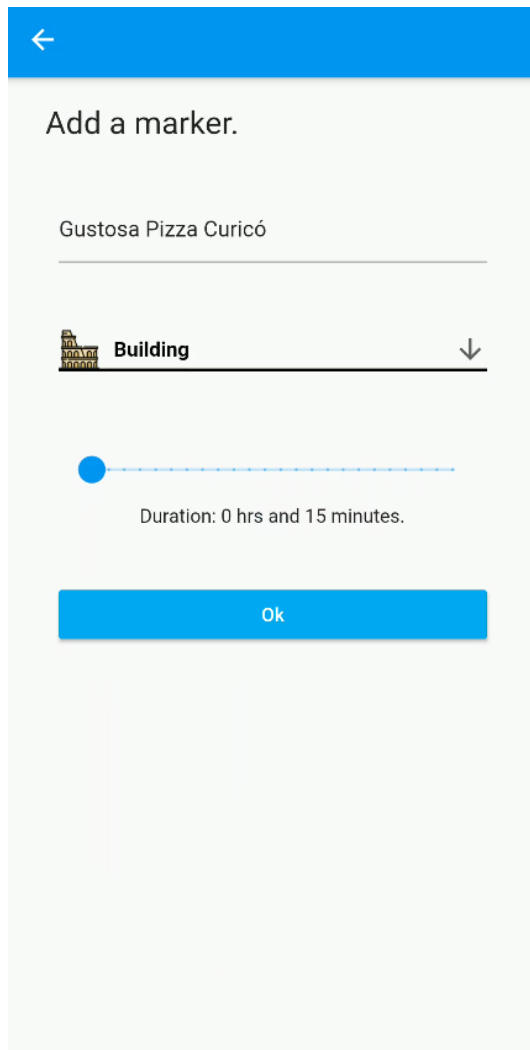


Figura C.15: Pantalla para añadir un marcador al mapa. Figura C.16: Despliegue de tipos de ubicaciones para el marcador.

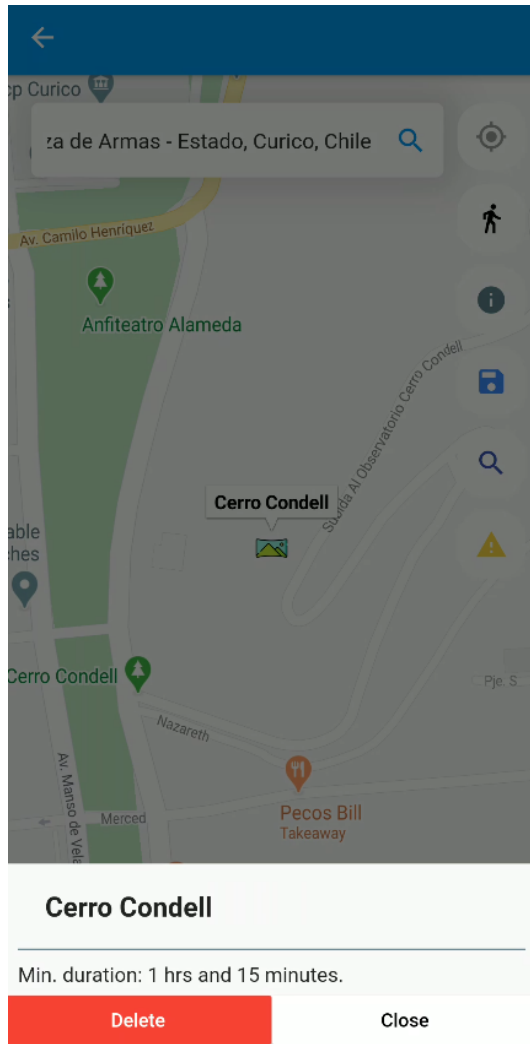


Figura C.17: Selección de un marcador en el mapa.

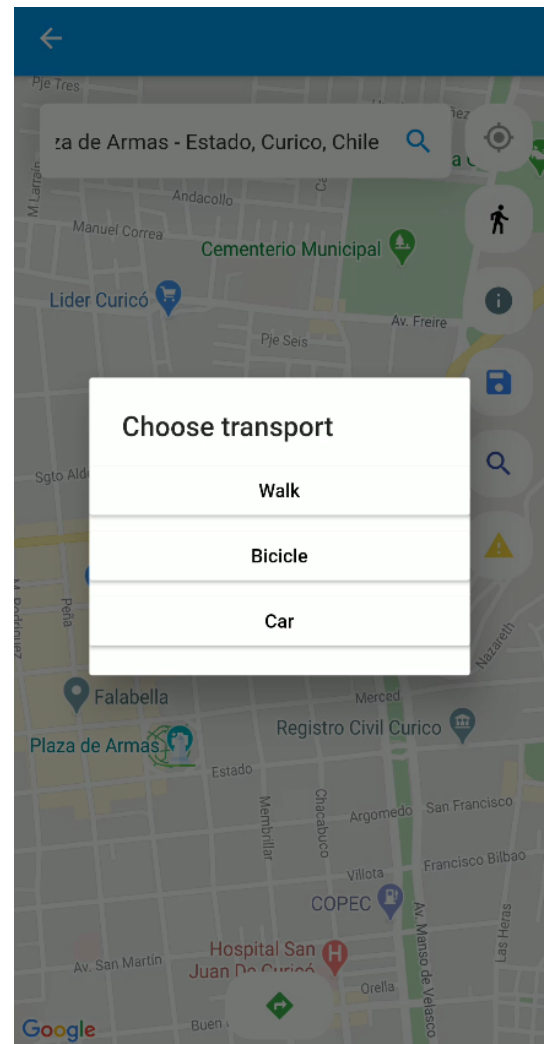


Figura C.18: Selección del medio de transporte.

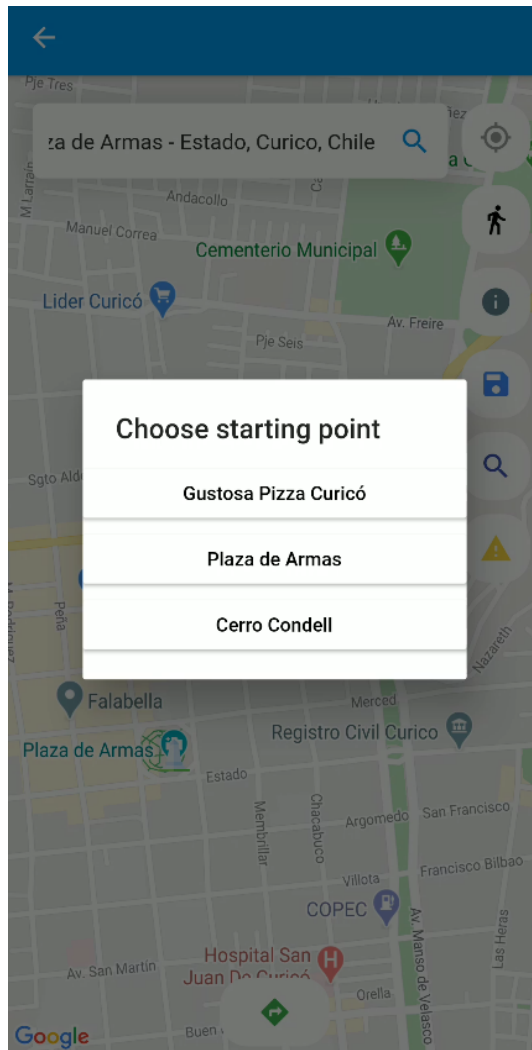


Figura C.19: Selección del punto de partida.

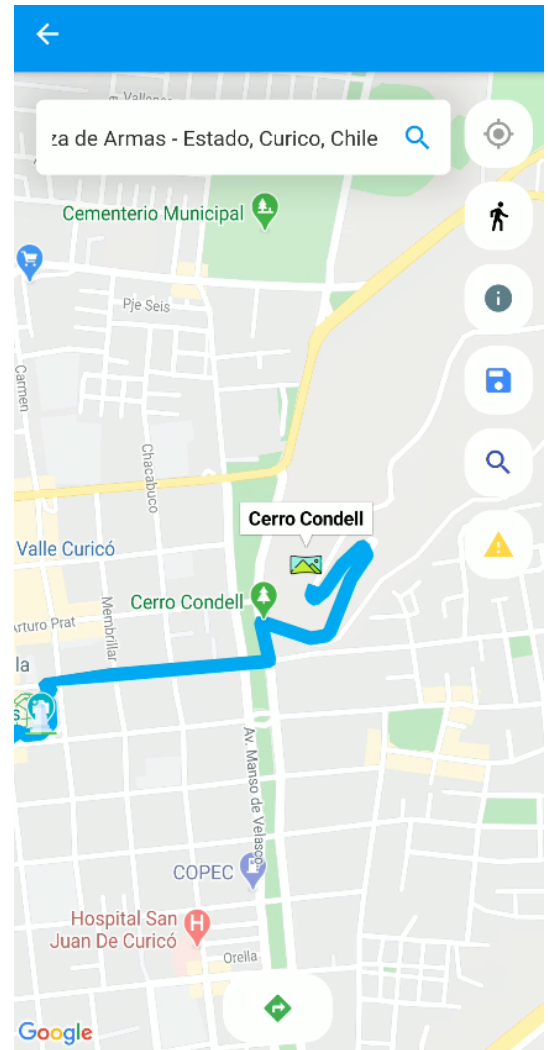


Figura C.20: Ruta trazada en el mapa.



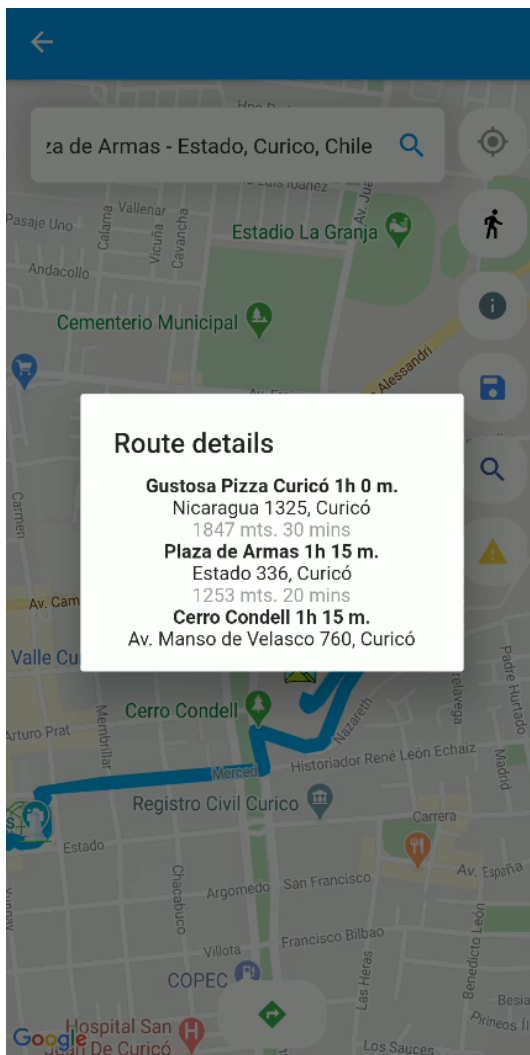


Figura C.21: Detalles de la ruta trazada.

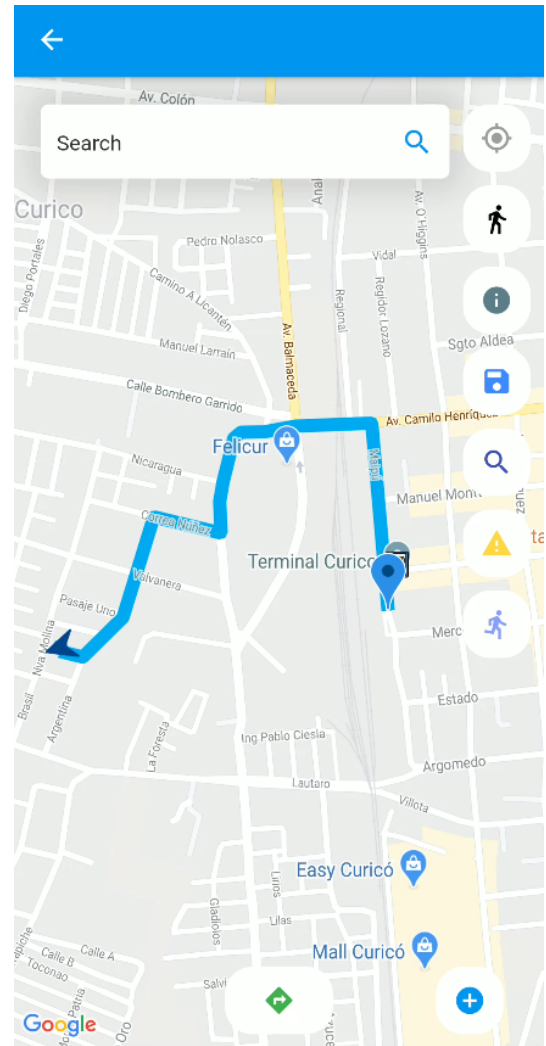
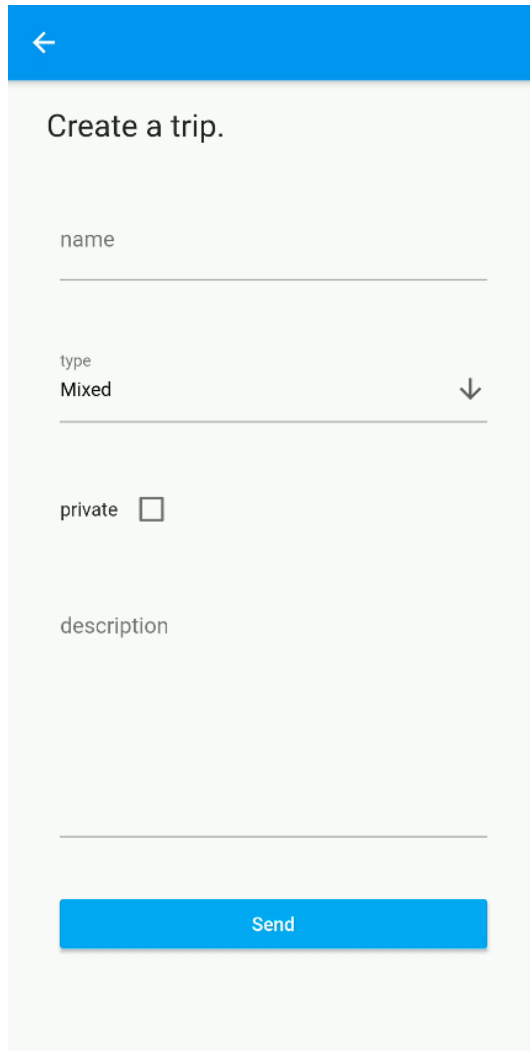


Figura C.22: Trazado de ruta desde la ubicación del usuario a un punto en el mapa.

#### C.4. Vistas de creación y edición de viaje.



←

Create a trip.

name

---

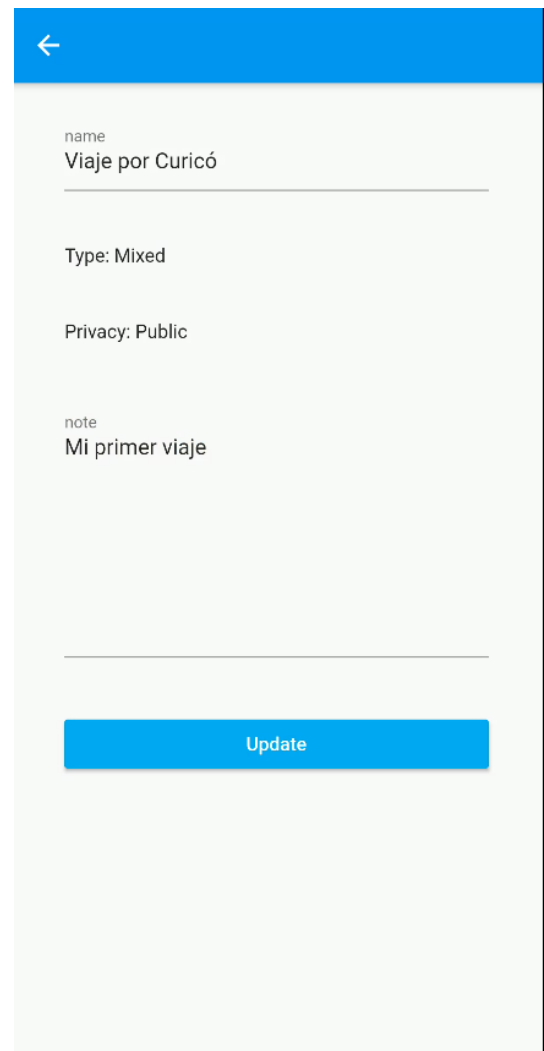
type  
Mixed ↓

private

description

---

Send



←

name  
Viaje por Curicó

---

Type: Mixed

Privacy: Public

note  
Mi primer viaje

---

Update

Figura C.23: Pantalla de creación de un viaje. Figura C.24: Pantalla de información de un viaje.

name  
Viaje por Curicó

note  
Mi primer viaje

private

type  
Mixed

Save

Delete trip

name  
Paseo por Curico

Type: Mixed

note

1 0

Add to my trips

Figura C.25: Pantalla de edición de un viaje. Figura C.26: Pantalla de información de un viaje no perteneciente al usuario.

### C.5. Vistas de creación de eventos.

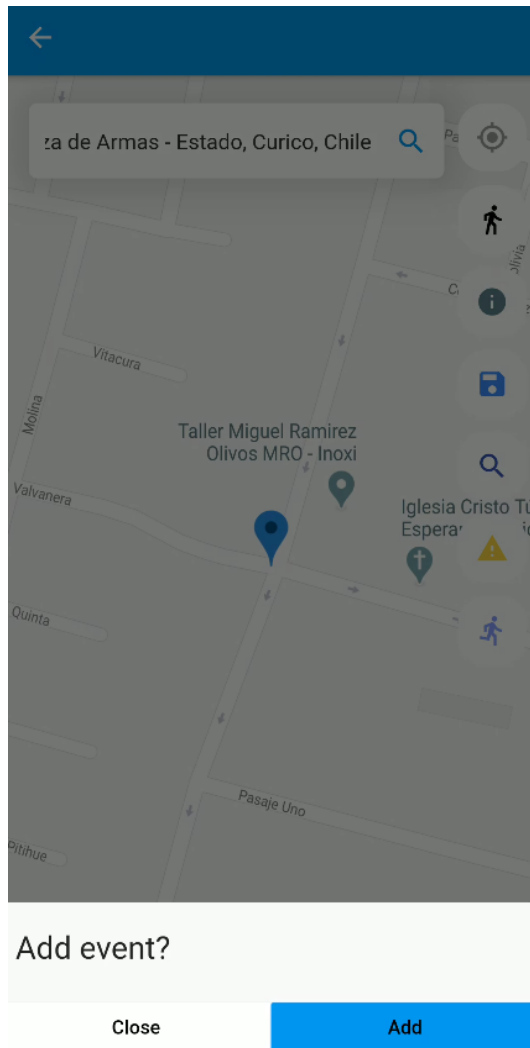


Figura C.27: Pantalla de acceso a creación de un evento.

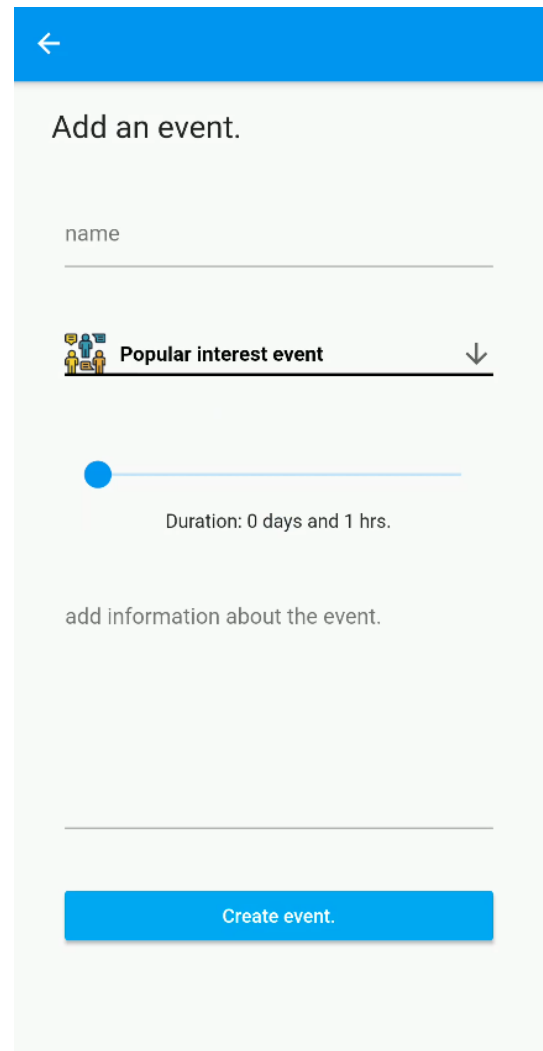


Figura C.28: Pantalla de creación de un evento.

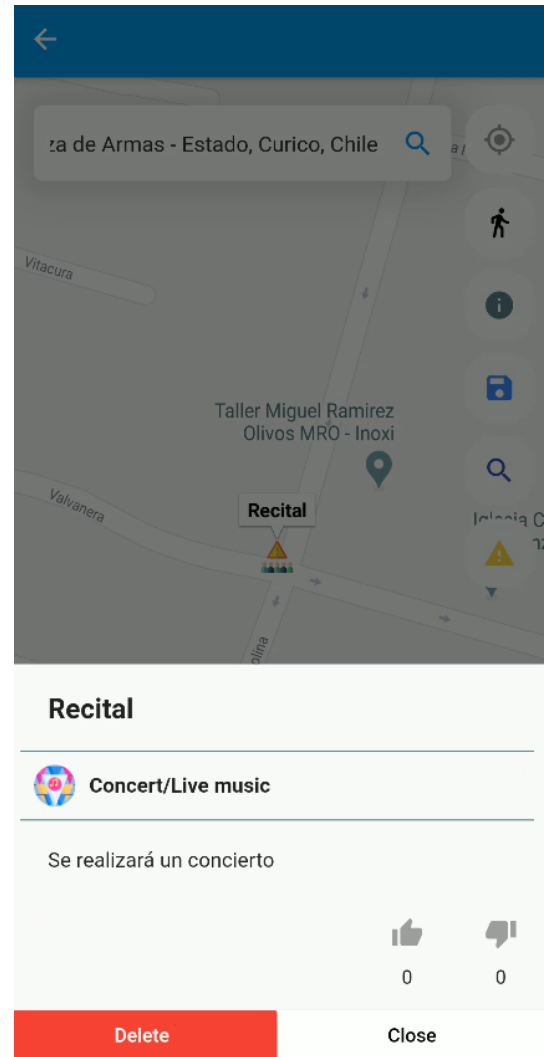
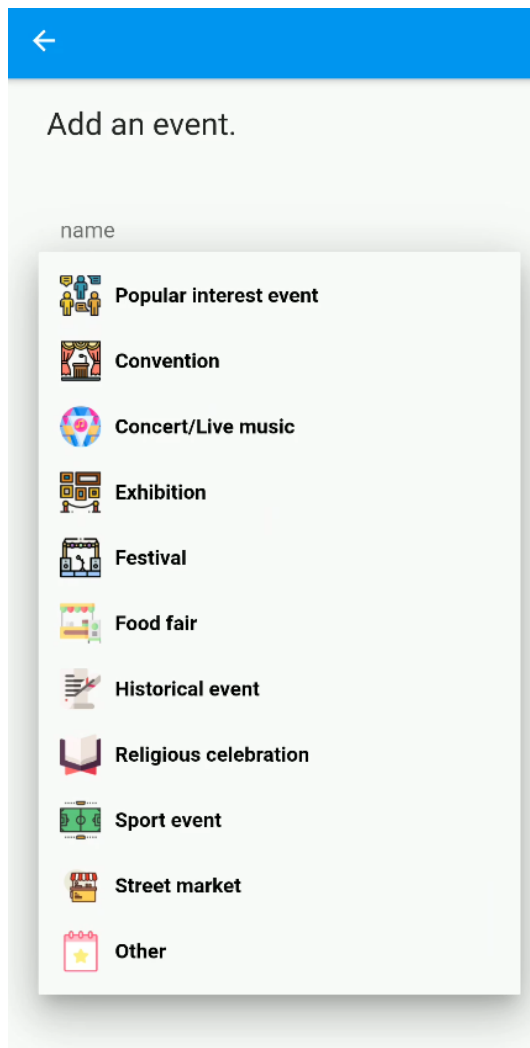


Figura C.29: Despliegue de tipos de eventos. Figura C.30: Visualización de eventos en el mapa.

## D. Historias de usuario expandidas

---

En este apéndice se presenta una extensión de las historias de usuario con información adicional.

### D.1. Historias de usuario.

Código	HU1
Historia	Como usuario quiero añadir marcadores personalizados a un mapa para definir los lugares que visitaré
Clasificación	Funcional
Prioridad	Alta
Fuente	Requerimientos iniciales

Código	HU2
Historia	Como usuario quiero conocer mi ubicación en el mapa para orientarme en este
Clasificación	Funcional
Prioridad	Alta
Fuente	Requerimientos iniciales

Código	HU3
Historia	Como usuario quiero guardar la información de mis viajes para llevar un registro de ellos
Clasificación	Funcional
Prioridad	Alta
Fuente	Requerimientos iniciales

Código	HU4
Historia	Como usuario quiero consultar una ruta y el tiempo para recorrer los lugares seleccionados
Clasificación	Funcional
Prioridad	Media
Fuente	Requerimientos iniciales

Código	HU5
Historia	Como usuario quiero seleccionar un medio de transporte para mi recorrido
Clasificación	Funcional
Prioridad	Media
Fuente	Profesor guía

Código	HU6
Historia	Como usuario quiero buscar lugares turísticos por categoría para agregarlos a mis marcadores
Clasificación	Funcional
Prioridad	Alta
Fuente	Requerimientos iniciales

Código	HU7
Historia	Como usuario quiero conocer la información de un lugar para planificar mi visita
Clasificación	Funcional
Prioridad	Alta
Fuente	Profesor guía

Código	HU8
Historia	Como usuario quiero conocer los eventos realizados en un lugar para poder complementar mi viaje
Clasificación	Funcional
Prioridad	Media
Fuente	Requerimientos iniciales

Código	HU9
Historia	Como usuario quiero buscar otros viajes y clonarlos para facilitar la planificación de mi viaje
Clasificación	Funcional
Prioridad	Alta
Fuente	Requerimientos iniciales

Código	HU10
Historia	Como usuario quiero calificar los viajes y eventos para entregar retroalimentación a otros usuarios
Clasificación	Funcional
Prioridad	Media
Fuente	Requerimientos iniciales



Código	HU11
Historia	Como usuario quiero poder determinar como llegar a ubicación para poder realizar un trayecto a esta
Clasificación	Funcional
Prioridad	Alta
Fuente	Requerimientos iniciales

Código	HU12
Historia	Como usuario quiero administrar mi información y viajes para llevar un control de mis datos
Clasificación	Funcional
Prioridad	Alta
Fuente	Profesor guía

Código	HU13
Historia	Como desarrollador quiero evaluar la aplicación para determinar errores o mejoras en el diseño
Clasificación	Funcional
Prioridad	Alta
Fuente	Desarrollador

## E. Diccionario de datos

---

En este apéndice se presenta una breve descripción de los atributos presentes en el modelo de datos de las tablas más significativas, excluyendo los índices, marcas de tiempo y las claves foráneas.

### E.1. users

- email: El correo electrónico del usuario, que utilizará para iniciar sesión.
- password: La contraseña cifrada utilizada para iniciar sesión.
- name: El nombre del usuario de la aplicación para ser presentado en la interfaz de esta.
- birth\_date: La fecha de nacimiento del usuario, pensada inicialmente para búsqueda de viajes por edad similar.
- gender: El género del usuario, pensado inicialmente para recomendaciones de viajes por esta característica.

### E.2. trips

- name: El nombre del viaje determinado por el usuario.
- note: Una descripción u texto para dejar asociado al viaje.
- privacy: Variable para determinar si un viaje se encuentra publico o privado.

- *transportation*: El método de transporte predeterminado por el usuario para realizar el recorrido.
- *duration*: La cantidad de tiempo pensada para realizar el viaje en relación al medio de transporte determinado por el usuario.
- *init\_lat*: La latitud del punto de inicio del recorrido.
- *init\_lng*: La longitud del punto de inicio del recorrido.
- *clone\_count*: La cantidad de veces que el viaje ha sido clonado.

### **E.3. points**

- *name*: El nombre del lugar que representa el punto.
- *lat*: La latitud del punto del marcador.
- *lng*: La longitud del punto del marcador.
- *duration*: El tiempo estimado por el usuario para permanecer en esta ubicación.

### **E.4. events**

- *name*: El nombre el evento definido por el usuario.
- *lat*: La latitud del punto donde se realiza el evento.
- *lng*: La longitud del punto donde se realiza el evento.
- *detail*: La descripción sobre el evento.
- *created\_at*: La fecha en la cual fue creado el evento.
- *ends\_at*: La fecha en la cual termina el evento y termina de ser mostrado en el mapa.