

**UNIVERSIDAD TÉCNICA FEDERICO SANTA MARÍA
SEDE VIÑA DEL MAR – JOSÉ MIGUEL CARRERA**

**TRAIL HUNTER, APLICACIÓN COLABORATIVA DE RUTAS DE MOUNTAIN
BIKE EN CHILE**

Trabajo de Titulación para optar al Título de
Ingeniero de Ejecución en SOFTWARE

Estudiante:
Aldo Luciano López

Profesor Guía:
Pamela Gatica Caballero

2025



CONSTANCIA DE VALIDACIÓN Y CONFIDENCIALIDAD DE MONOGRAFÍA A REPOSITORIO ACADÉMICO

1.- IDENTIFICACIÓN DEL TRABAJO ACADÉMICO

Tipo de monografía (marcar una opción): ☒ Memoria o trabajo de título ☐ Tesis de Postgrado

Título del trabajo: TRAIL HUNTER, APLICACIÓN COLABORATIVA DE RUTAS DE MOUNTAIN BIKE EN CHILE

Nombre del candidato(a): ALDO LUCIANO PATRICIO LÓPEZ VALLEJOS

Carrera / Grado: Ingeniería de Ejecución en SOFTWARE

Campus: VIÑA DEL MAR

Departamento: ELECTROTECNIA E INFORMÁTICA

2.- VALIDACIÓN DEL PROFESOR GUÍA/DIRECTOR DE TESIS

Yo, ÓSCAR FRANCISCO CARRASCO VERA, en mi calidad de profesor(a) guía/director(a) del trabajo académico mencionado anteriormente **DEJO CONSTANCIA** que:

- He revisado esta versión del documento y corresponde a la versión final aprobada del trabajo.
- El trabajo cumple con los requisitos académicos y de formato establecidos por la institución.

3.- EVALUACIÓN DE CONFIDENCIALIDAD POR PROPIEDAD INDUSTRIAL (marcar una opción)

☒ El trabajo **NO contiene** información que amerite confidencialidad y puede ser publicado de inmediato en repositorio con acceso abierto.

☐ El trabajo **CONTIENE** información con potenciales implicancias de propiedad industrial o intelectual y requiere un periodo de confidencialidad (**embargo**) por (marcar una opción):

☐ 6 meses ☐ 12 meses ☐ 2 años ☐ 3 años ☐ 5 años ☐ 10 años

Fundamentación de la necesidad de confidencialidad (obligatorio si se solicita embargo):

4.- FIRMAS

Profesor(a) guía o director(a) de memoria o tesis:

Fecha: 02/OCTUBRE/2026 Firma:

Estudiante o Candidato(a):

Fecha: 02/OCTUBRE/2026 Firma:

Este formulario debe ser insertado como página 2 de la memoria o tesis, completado y firmado por estudiante y profesor(a) antes de la entrega en portal PRISMA de Biblioteca USM.

ÍNDICE

INTRODUCCIÓN	1
1. ASPECTOS RELEVANTES DEL PROYECTO Y SU GESTIÓN.....	3
1.1. Descripción de la Empresa	3
1.2. Situación sin proyecto.....	4
1.2.1. Descripción de la Situación sin Proyecto	4
1.2.2. Problemas detectados	6
1.2.3. Requerimientos y Requisitos del Usuario.....	8
1.3. Objetivos.....	8
1.4. Alternativas globales de solución	9
1.4.1. Alternativa 1: Contratación de empresa de desarrollo de software externa	9
1.4.2. Alternativa 2: Adaptación de una Plataforma existente.....	10
1.4.3. Alternativa 3 (Alternativa propia): Desarrollo a la medida.....	11
1.5. Evaluación de Alternativas.....	12
1.5.1. Criterios de Evaluación	12
1.5.2. Evaluación.....	16
1.6. Alternativa Seleccionada	18
1.6.1. Beneficios Económicos:.....	19
1.6.2. Beneficios Sociales:	19
1.7. Planificación	20
1.7.1. Desarrollo del plan personal	20
1.7.2. Planificación temporal y asignación de recursos	22
1.7.3. Estimación de costos.....	23
1.7.4. Administración del riesgo	24
1.7.5. Riesgos detectados	25
2. ASPECTOS RELEVANTES DEL ANÁLISIS.	30
2.1. Descripción de la solución propuesta	30
2.2. Diagrama de actividades general UML	30
2.2.1. Flujo administrativo: Traza Sendero.....	30

2.2.2. Flujo administrativo: Compartir Sendero	32
2.3. Requerimientos del sistema	33
2.3.1. Requerimientos funcionales	33
2.3.2. Requerimientos no funcionales.....	35
2.4. Estructura funcional del sistema (Caso de uso general)	39
2.4.1. Diagrama de caso de uso general	40
2.5. Matriz de trazabilidad	41
2.6. Modelo conceptual.....	42
2.7. Modelo de casos de uso.	43
2.7.1. Caso de Uso: CU01 – Mantener Usuario.....	43
2.7.2. Caso de uso: CU02 – Descargar Mapa	45
2.7.3. Caso de uso: CU03 – Crear Sendero	47
2.7.4. Caso de uso: CU04 – Trazar Sendero	49
2.7.5. Caso de uso: CU05 – Compartir ubicación.....	51
2.7.6. Caso de uso: CU06 – Compartir Imagen.....	53
2.7.7. Caso de uso: CU07 – Crear Comunidad	55
2.7.8. Caso de uso: CU08 – Publicar Muro	57
2.7.9. Caso de uso: CU09 - Compartir sendero.....	59
2.7.10. Caso de uso: CU10 – Seguir Comunidad/Usuario.....	61
2.7.11. Caso de uso: CU11 – Consultar datos sendero	63
2.7.12. Caso de uso: CU12 – Enviar SOS.....	65
2.7.13. Caso de uso: CU13 – Validar Ruta	67
3. ASPECTOS RELEVANTES DEL DISEÑO	70
3.1. Arquitectura del Software	70
3.1.1. Sistema Operativo	71
3.1.2. Herramientas necesarias para el desarrollo del proyecto	71
3.1.3. Stack de tecnologías de la aplicación.....	72
3.1.4. Lenguajes de programación y tecnologías	73
3.2. Diseño de datos	74
3.2.1. Modelo de Clases.	74
3.2.2. Modelo Relacional.	76

3.2.3. Tipos de Datos.	77
3.2.4. Diccionario de datos.	78
3.3. Diagrama de Secuencia Extendido	82
3.3.1. CU01 - Mantener Usuario	82
3.3.2. CU02 - Descargar Mapa	83
3.3.3. CU03 - Crear sendero.....	84
3.3.4. CU04 - Trazar sendero	85
3.3.5. CU05 - Compartir ubicación.....	86
3.3.6. CU06 - Compartir imagen	87
3.3.7. CU07 - Crear comunidad.....	88
3.3.8. CU08 - Publicar en muro	89
3.3.9. CU09 - Compartir sendero.....	90
3.3.10. CU10 - Seguir usuario/comunidad	91
3.3.11. CU11 - Consultar datos sendero	92
3.3.12. CU12 - Enviar SOS	93
3.3.13. CU13 - Validar Ruta.....	94
3.4. Diseño de Interfaz.....	95
3.4.1. UI 01: Registro de Usuario	95
3.4.2. UI 02: Buscar Senderos.....	97
3.4.3. UI 03: Compartir Ubicación	98
3.4.4. UI 04: Enviar SOS	99
3.4.5. UI 03: Buscar Comunidades	101
CONCLUSIÓN	102
BIBLIOGRAFÍA.....	104
ANEXOS	105
4. ANEXOS	105
4.1. N° 1	105
4.2. Anexo 2.....	111

ÍNDICE DE FIGURAS

Figura 2-1. Flujo Administrativo Traza Sendero 01.....	31
Figura 2-2. Flujo Administrativo Traza Sendero 02.....	31
Figura 2-3. Flujo Alternativo Compartir Sendero.....	32
Figura 2-4. Caso de uso General.....	40
Figura 2-6. Modelo Conceptual	42
Figura 2-7. Diagrama de Secuencia: CU01 – Mantener Usuario.....	44
Figura 2-8. Diagrama de Secuencia: CU02 – Descargar Mapa.....	46
Figura 2-9. Diagrama de Secuencia: CU03 – Crear Sendero	48
Figura 2-10. Diagrama de Secuencia: CU04 – Trazar Sendero.....	50
Figura 2-11. Diagrama de Secuencia: CU05 – Compartir Ubicación	52
Figura 2-12. Diagrama de Secuencia: CU06 – Compartir Imagen.....	54
Figura 2-13. Diagrama de Secuencia: CU07 – Crear Comunidad	56
Figura 2-14. Diagrama de Secuencia: CU08 – Publicar Muro	58
Figura 2-15. Diagrama de Secuencia: CU09 – Compartir Sendero.....	60
Figura 2-16. Diagrama de Secuencia: CU10 – Seguir Comunidad/Usuario	62
Figura 2-17. Diagrama de Secuencia: CU11 – Consultar datos Sendero	64
Figura 2-18. Diagrama de Secuencia: CU12 – Enviar SOS	66
Figura 2-19. Diagrama de Secuencia: CU13 – Validar Ruta.....	68
Figura 3-1. Arquitectura de Capas Trail Hunter	71
Figura 3-2. Modelo de Clases	75
Figura 3-3. Modelo de Datos	76
Figura 3-4. Diagrama de secuencia extendido: CU01 – Mantener Usuario	82
Figura 3-5. Diagrama de secuencia extendido: CU02 – Descargar Mapa	83
Figura 3-6. Diagrama de secuencia extendido: CU03 – Crear sendero.....	84
Figura 3-7. Diagrama de secuencia extendido: CU04 – Trazar sendero	85
Figura 3-8. Diagrama de secuencia extendido: CU05 – Compartir ubicación.....	86
Figura 3-9. Diagrama de secuencia extendido: CU06 – Compartir imagen	87
Figura 3-10. Diagrama de secuencia extendido: CU07 – Crear comunidad	88
Figura 3-11. Diagrama de secuencia extendido: CU08 – Publicar Muro.....	89
Figura 3-12. Diagrama de secuencia extendido: CU09 – Compartir sendero	90
Figura 3-13. Diagrama de secuencia extendido: CU10 – Seguir usuario/comunidad	91
Figura 3-14. Diagrama de secuencia extendido: CU11 – Consultar datos sendero .	92
Figura 3-15. Diagrama de secuencia extendido: CU12 - Enviar SOS	93
Figura 3-16. Diagrama de secuencia extendido: CU13 - Validar Ruta	94
Figura 3-17. Diseño de Interfaz: Login.....	96
Figura 3-18. Diseño de Interfaz: Buscar senderos.....	97
Figura 3-19. Diseño de Interfaz: Compartir ubicación.....	98
Figura 3-20. Diseño de Interfaz: Buscar Comunidades	101

ÍNDICE DE TABLAS

Tabla 1-1. Upgrades de aplicación Trailfork mediante pagos	5
Tabla 1-2. Ponderaciones de Criterios de Evaluación	16
Tabla 1-3. Escala de evaluación, Elaboración propia	16
Tabla 1-4. Alternativa N° 1	17
Tabla 1-5. Alternativa N° 2	17
Tabla 1-6. Alternativa N° 3	17
Tabla 1-7. Ponderaciones	18
Tabla 1-8. Carta Gantt	22
Tabla 1-9. Distribución de Costos	23
Tabla 1-10. Costos totales de proyecto	24
Tabla 1-11. Tabla de Riesgos	26
Tabla 1-12. Plan de Mitigación	27
Tabla 2-1. Requerimientos no funcionales	35
Tabla 2-2. Descripción de Casos de Uso	39
Tabla 2-3. Matriz de Trazabilidad, Elaboración propia	41
Tabla 2-4. Caso de Uso Narrativo: CU01 - Mantener Usuario	43
Tabla 2-5. Caso de uso Narrativo: CU02 - Descargar Mapa	45
Tabla 2-6. Caso de uso Narrativo: CU03 - Crear Sendero	47
Tabla 2-7. Caso de uso Narrativo: CU04 - Trazar Sendero	49
Tabla 2-8. Contrato: CU04 - Trazar Sendero	50
Tabla 2-9. Caso de uso Narrativo: CU05 - Compartir Ubicación	51
Tabla 2-10. Caso de uso Narrativo: CU06 - Compartir imagen	53
Tabla 2-11. Caso de uso Narrativo: CU07 - Crear Comunidad	55
Tabla 2-12. Caso de uso narrativo: CU08 - Publicar Muro	57
Tabla 2-13. Contrato: CU08 - Publicar Muro	58
Tabla 2-14. Caso de uso narrativo: CU09 - Compartir Sendero	59
Tabla 2-15. Caso de uso Narrativo: CU10 - Seguir Comunidad/Usuario	61
Tabla 2-16. Caso de uso Narrativo: CU11 - Consultar datos sendero	63
Tabla 2-17. Caso de uso Narrativo: CU12 - Enviar SOS	65
Tabla 2-18. Caso de uso narrativo: CU13 - Validar Ruta	67
Tabla 3-1. Base de Datos: Tipos de Datos	77
Tabla 3-2. Diccionario de datos: Tabla Mapa	79
Tabla 3-3. Diccionario de datos: Tabla Clima	78
Tabla 3-4. Diccionario de datos: Tabla MapaSenedero	79
Tabla 3-5. Diccionario de datos: Tabla Sendero	80
Tabla 3-6. Diccionario de datos: Tabla Traza	80
Tabla 3-7. Diccionario de datos: Tabla Usuario	81
Tabla 3-8. Diccionario de datos: Tabla GpsRider	79
Tabla 3-9. Diccionario de datos: Tabla Ubicacion	81

Tabla 3-10. Diccionario de datos: Tabla Comunidad 78

Tabla 3-11. Diccionario de datos: Tabla UsuarioComunidad..... 81

Tabla 3-12. Diccionario de datos: Tabla Muro 80

DEDICATORIA

Quiero partir agradeciendo a las personas que me han apoyado en esta etapa de mi vida, ya sea amigos con un "Dale, tú puedes" o con solo escuchar cuando pasaba por épocas de estrés, compañeros de trabajo con los que comparto día a día que, sin ellos saberlo, fueron de ayuda para poder sacar adelante esta tarea de estudiar nuevamente.

Especiales agradecimientos a mi trabajo, BancoEstado, quien me apoyó económicamente y a mi jefa que sin quererlo me dio el empujón principal cuando me mandó a seguir estudiando cuando le dije que solamente quería hacer un diplomado.

Agradecer a mi amigo Christian Lamas, que también es parte de esta generación, sin él hoy ni siquiera estaría aquí, muchas gracias, Christian, por traerme a Santiago a trabajar hace más de 10 años, por confiar en mí y por convertirme en uno de mis mejores amigos. Si no me hubieras dicho que te habías inscrito a la carrera quizás no me hubiera entusiasmado tanto la idea de volver a estudiar, porque ver una cara familiar en las aulas claramente sirvió.

Agradecer a mi impulsividad, esa que siempre me hace hacer lo que mi mente decide sin darle una vuelta, mi gran amiga y enemiga, mi impulsividad. Cuando vi que me estaba quedando sin herramientas para poder ser de ayuda en mi trabajo y a mis compañeros y decidió volver a estudiar.

Quiero también no dejar pasar esta oportunidad para agradecer a todos los profesores, de plan común como los específicos de la carrera, cada uno de ellos me ayudó en este camino, sin ellos no me cabe la menor duda que hoy por hoy no estaría aquí entregando esta tesis para poder optar a un título profesional.

Agradecimientos especiales a German Wehrhah de MTB Enduro Chile, ya que él fue mi primer acercamiento al enduro y el descenso en bicicleta. Gracias a él me enamoré del Mountain Bike y de toda la comunidad que creó, de la cual hoy por hoy mantengo grandes amigos, sin él este proyecto nunca hubiera nacido.

SIGLA Y SIMBOLOGÍA

ANDROID, API, APP, AWS, BD, CI/CDCORE, CRUD, CU, DATA, ETC, GIS, GIST, GPS, IDE, IOS, KOTLIN, MTB, OFFLINE, OGC, PRO, QA, SWIFT, TRAILFORKS, UF, UI, UML, UX, XCODE.

A continuación, se entregará una breve descripción de cada SIGLA y SIMBOLOGÍA.

ANDROID: Sistema operativo móvil desarrollado por Google, basado en Linux.

API: Conjunto de reglas que permite a aplicaciones comunicarse entre sí.

APP: Programa o software diseñado para dispositivos móviles o computadoras.

AWS: Plataforma de servicios en la nube de Amazon para computación y almacenamiento.

BD: Base de Datos, sistema para almacenar y gestionar información estructurada.

CHECKMARKS: Marca de "visto bueno".

CI/CD: Métodos de integración y entrega continua para automatizar el desarrollo de software.

CORE: Parte esencial de un sistema o aplicación que realiza sus funciones principales.

CRUD: Operaciones básicas de Base de Datos: Crear, Leer, Actualizar, Eliminar.

CU: Caso de Uso, representación de una funcionalidad del sistema.

DATA: Conjunto de datos procesados o no, útiles para análisis.

DEBUGGEAR: Proceso de encontrar, aislar y corregir errores (bugs) en el código de un programa o sistema.

ETC: Abreviatura que significa "y otras cosas similares".

GIS: Sistema de Información Geográfica para manejar datos espaciales.

GIST: Resumen o representación simplificada de un tema o idea.

GPS: Sistema global de posicionamiento para determinar ubicaciones exactas.

IDE: Entorno de Desarrollo Integrado, herramienta para programar y depurar código.

IOS: Sistema operativo móvil de Apple para iPhones y iPads.

KOTLIN: Lenguaje de programación moderno utilizado principalmente para Android.

MTB: Mountain Bike, bicicleta diseñada para terrenos montañosos.

OFFLINE: Estado en el que un dispositivo no tiene conexión a internet.

OGC: Open Geospatial Consortium, estándares abiertos para datos geoespaciales.

PRO: Abreviatura de "Profesional" o versión avanzada de un producto.

QA: Aseguramiento de Calidad, procesos para garantizar software libre de errores.

SWIFT: Lenguaje de programación utilizado para desarrollar aplicaciones en iOS.

TRAILFORKS: Plataforma para mapear, buscar y compartir rutas de ciclismo.

UF: Unidad de Fomento, índice económico usado en Chile para ajustar valores.

UI: Interfaz de Usuario, diseño visual y funcional de cómo interactuar con un sistema.

UML: Lenguaje de modelado unificado para representar sistemas y procesos.

UPDATE: Actualización de sistemas/módulos/base de datos.

UX: Experiencia de Usuario, cómo percibe y utiliza alguien un producto o sistema.
XCODE: Entorno de desarrollo integrado (IDE) de Apple para aplicaciones iOS y macOS.

RESUMEN

KEYWORDS: Mountain Bike, Trail Hunter, Chile, Geoposicionamiento, Senderos, Desarrollo De Software, Comunidad Deportiva, Pen-Source, Aplicaciones Móviles.

Existe actualmente el problema de que no hay disponible una aplicación colaborativa, completamente nacional, que cubra la totalidad del territorio chileno en cuanto a senderos.

El objetivo principal del proyecto es desarrollar "Trail Hunter", una aplicación colaborativa dirigida específicamente a practicantes de Mountain Bike en Chile. Esta herramienta permite registrar, trazar y compartir senderos, proporcionando información en tiempo real sobre las condiciones climáticas y el estado actual de las rutas. Además, fomenta la creación y fortalecimiento de comunidades interactivas alrededor de esta actividad deportiva.

Se ha escogido la metodología de desarrollo **en** cascada, para ejecutar el proyecto de manera efectiva, eficiente y dentro del tiempo establecido, ya que se considera adecuado para un proyecto que no requiere metodologías más especializadas o complejas.

Este proyecto se desarrolla en el contexto académico de la Universidad Técnica Federico Santa María, en colaboración directa con "Trail Hunter", una empresa emergente de software fundada por el autor. Dicha empresa tiene como objetivo satisfacer necesidades específicas de la comunidad de ciclistas de montaña en Chile, ofreciendo una solución local, personalizada y gratuita para los usuarios, diferenciándose así de alternativas internacionales existentes.

Los beneficios potenciales de este desarrollo abarcan aspectos tanto económicos como sociales. En términos económicos, se prevé monetizar la aplicación mediante publicidad relacionada al ámbito del ciclismo. En cuanto a lo social, se proyecta captar rápidamente una audiencia significativa al ofrecer un software altamente personalizado, además de establecer colaboraciones estratégicas con equipos deportivos y facilitar la creación de una comunidad activa y una red de apoyo alrededor de la plataforma.

El documento presentado incluye desde la introducción, que detalla la motivación detrás de cada etapa del proyecto, hasta la fase de diseño, abordando también aspectos relevantes, alternativas consideradas y la correspondiente evaluación de estas.

INTRODUCCIÓN

El Mountain Bike (MTB) en Chile ha crecido significativamente en los últimos años, consolidándose como una actividad popular tanto para deportistas profesionales como aficionados. Sin embargo, actualmente existe una falta de herramientas específicas que permitan a los ciclistas planificar rutas personalizadas, conocer las condiciones climáticas en tiempo real y compartir experiencias de manera sencilla. Las soluciones internacionales disponibles presentan limitaciones, ya sea por costos elevados, falta de datos locales o ausencia de características adaptadas a las necesidades del usuario chileno.

Abordar esta problemática resulta esencial para fomentar el desarrollo de este deporte, promover un estilo de vida saludable y conectar a la comunidad MTB de manera más efectiva. Una solución tecnológica que combine funcionalidad, accesibilidad y personalización puede potenciar la experiencia de los ciclistas, mejorar la seguridad en los senderos y contribuir al crecimiento de la red deportiva en el país.

El proyecto tiene como objetivo principal diseñar e implementar una aplicación gratuita e integral que facilite el registro, la planificación y el uso compartido de rutas de Mountain Bike en Chile. La herramienta busca ofrecer información relevante sobre senderos, condiciones climáticas y actividades sociales, adaptándose a las características y necesidades específicas del contexto local.

La solución fue desarrollada a través de un enfoque de cascada que incluyó fases tales como investigación de tecnologías, integraciones, como y que usuarios finales harían las pruebas y un alcance del proyecto desde lo más básico a lo más complejo como lo es las mismas limitaciones de las tecnologías a utilizar.

El documento que se presenta contiene desde el marco teórico hasta la etapa de diseño.

CAPÍTULO 1: ASPECTOS RELEVANTES DEL PROYECTO Y SU GESTIÓN

1. ASPECTOS RELEVANTES DEL PROYECTO Y SU GESTIÓN.

1.1. Descripción de la Empresa

Trail Hunter es una empresa de software fundada en Santiago de Chile en 2023 por un estudiante de la Universidad Técnica Federico Santa María. Nacida de la pasión compartida por el Enduro y el Descenso en bicicleta, la empresa surge de la colaboración entre un grupo de amigos entusiastas del mountain bike y profesionales del área de Tecnologías de la Información (TI). Este equipo multidisciplinario incluye desde desarrolladores de software hasta ingenieros de datos y especialistas en análisis y explotación de información.

Motivados por la ausencia de herramientas tecnológicas que satisfagan las necesidades específicas de los ciclistas de montaña en Chile, este grupo de amantes del deporte y la tecnología decidió unir sus habilidades y conocimientos para crear una solución innovadora. Trail Hunter representa tanto un proyecto empresarial como la materialización de una visión compartida: combinar la experiencia técnica con la pasión por el mountain bike para enriquecer y conectar a la comunidad ciclista del país.

La empresa se centra en desarrollar aplicaciones y plataformas colaborativas que permiten a los usuarios registrar, trazar y compartir senderos, ofreciendo información en tiempo real sobre condiciones climáticas y el estado de las rutas. Además, busca fomentar la creación de comunidades interactivas, facilitando la comunicación y el intercambio de experiencias entre los aficionados al ciclismo de montaña.

Esta iniciativa refleja el compromiso de Trail Hunter por impulsar el deporte, aprovechar el talento local en TI y aportar valor a la sociedad mediante soluciones tecnológicas adaptadas al contexto chileno. Al integrar conocimientos especializados y una profunda comprensión de las necesidades de los ciclistas, la empresa se posiciona como un referente en la promoción de estilos de vida activos y la innovación en el ámbito deportivo.

Su visión es convertirse en uno de los referentes y líderes en innovación tecnológica para el ciclismo de montaña en Chile, inspirando a los ciclistas a explorar, conectar y compartir su pasión por los senderos mediante soluciones que integran tecnología avanzada y comunidad. La empresa visualiza un futuro donde cada ciclista, independientemente de su nivel de experiencia, tenga acceso a herramientas que potencien sus aventuras, fortalezcan su vínculo con la naturaleza y promuevan un estilo de vida activo y colaborativo.

La misión de Trail Hunter es promover el uso de la bicicleta ofreciendo herramientas tecnológicas adaptadas a los tiempos actuales. Estas herramientas proporcionan tranquilidad y certeza a los usuarios mediante geolocalización e integración con aplicaciones de clima en tiempo real, facilitando así la práctica deportiva. Este objetivo se fortalece gracias al fuerte compromiso de la empresa con la creación de una comunidad unida, capaz de apoyarse mutuamente y fomentar el crecimiento de nuevos deportistas.

1.2. Situación sin proyecto

A continuación, describiremos la situación actual para poder llevar a cabo el proyecto de Trail Hunter.

1.2.1. Descripción de la Situación sin Proyecto

Si bien existen aplicaciones exclusivas de Mountain Bike para usuarios, para tener todas las funcionalidades es a través de un pago y en Chile no existe una aplicación propia que esté personalizada para el país, en este caso se adecuan las aplicaciones internacionales para poder funcionar en el territorio nacional lo que conlleva a que no estén optimizadas para la realidad geográfica, precisión del GPS, interpretación de direcciones y categorización de senderos debido a que utilizan bases globales no actualizadas o propias incompletas. Esto se traduce en rutas mal trazadas, indicaciones incorrectas o poco claras, senderos privados mostrados como públicos y escasa información local.

Comentado [PG1]: 2 párrafos para explicar la situación actual no dejan claro cuál es el problema

Luego en la table haces un comparativo sobre una aplicación que se puede usar sin pagar y con pago... pero no queda claro cuál es el problema

Ahondar

Tabla 1-1. Upgrades de aplicación Trailfork mediante pagos

USUARIO GRATUITO	TRAILFORKS PRO EN LA WEB	TRAILFORKS PRO EN LA APLICACIÓN
• Ver tu ubicación en el mapa • Límite de 1 región descargada a la vez • Mapas base TF topo, satelital, osm • Ver rutas, puntos de interés, regiones y rutas compartidas en el mapa 2D • Ver 2 informes de rutas recientes en una ruta o región • Ver 10 informes cercanos • Enviar informes de seguimiento • Enviar fotos de la ruta • Enviar comentarios sobre la ruta • Rastros de tarifas • Guardar registros de actividad (registros de recorridos) • Ver tus actividades personales (registros de viaje) • Compartir enlaces al contenido • Ver insignias y concursos • Participación en la credencial • Ver eventos • Acceder a la pantalla de información de emergencia • Cambiar el tipo de actividad del mapa • Ver detalles básicos de la ruta • Ver 2 fotos y videos de senderos • Ver 2 comentarios más recientes sobre senderos	• Capas de mapa (incluido el mapa de calor, actividades, Strava, clima) • Mapas de estilos de senderos (incluida la popularidad) • Cambiar mapas base (Satélite, Gaia, OSM) • Mapa 3D • Resultados ilimitados de Route Finder (límite gratuito 3) • Estadísticas de uso de senderos • Artículos ilimitados de la lista de deseos personales • Descargar GPX • Mapas base de Garmin • Mapas de impresión personalizados de alta resolución • Guardar vistas de mapas y colecciones • Mapa del tiempo • Planificador de viajes por carretera • Segmentos de Strava • Mapa de informes de senderos • Prioridad en las colas de sincronización de Strava y Garmin • Icono Pro en tu avatar de usuario	• Descargas ilimitadas por región • Descargas de mapas sin conexión (mosaicos) • Mapas base premium (Gaia, ArcGis, Satélite) • Capas de mapa (incl. mapa de calor, actividades, strava, clima, fuego) • Mapas de estilos de senderos (incluida la popularidad) • Filtros de mapa • Mapa 3D • Gráfico de elevación de senderos y rutas con visualización de pendientes • Detalles completos de las páginas de rutas, regiones, puntos de interés y parques de habilidades • Función de ruta a sendero • Función Descubrir • Ver rutas en el mapa de Discover • Lista de deseos personal para senderos, rutas, regiones • Waypoints personales • Descargar archivos GPX • Ver información meteorológica • Búsqueda global fuera de línea • Ver todos los comentarios • Ver todos los informes de rutas • Ver todas las fotos y videos • Ver estadísticas personales de registro de viajes • Buscar registros de viajes personales • Tablero de estado • Direcciones de conducción de Google • Mapa "modo brújula" • Ordenar y filtrar listas de rutas y senderos • Traildar • Compartir ubicación en tiempo real

Fuente: Información oficial Trailforks.com

Se tienen varias aplicaciones hoy que están en las manos de personas que practican el Mountain bike en Chile.

Como se ve en la Tabla 1-1 adjunta, la única forma de acceder en una aplicación exclusiva de Mountain bike de todas las funcionalidades es a través de un pago.

Comentado [PG2]: En la imagen o en la tabla 1???
Me pierdo usar nombre de las tablas figuras o otros para referenciar

Situaciones como la explicadas anteriormente hacen que Trail Hunter tenga que intervenir y dar un paso adelante para poder dar solución a las exigencias actuales del mercado y del creciente número de practicantes de Mountain Bike en Chile.

1.2.2. Problemas detectados

En el análisis del contexto actual de la comunidad deportiva del Mountain Bike es necesario mencionar el escaso nivel de unión y comunicación, compuesta por ciclistas de distintos niveles de experiencia y trayectoria. Además, existen diversas categorías y estilos, tales como Cross Country (XC), Trail, Enduro y Downhill (DH), cada una con requerimientos técnicos, niveles de dificultad y tipos de senderos distintos, lo que genera una segmentación natural entre los usuarios. A eso hay que sumarle los practicantes de Trail running o senderismo tipo Trekking que muchas veces usan los mismos senderos exclusivos para Mountain Bike.

Esta diversidad, si bien contribuye a la práctica del deporte, genera desafíos relevantes en términos de coordinación, particularmente en el uso adecuado de las rutas, la correcta delimitación de zonas de ascenso y descenso, la mantención de los senderos y la difusión oportuna y confiable de su estado. Estas dificultades impactan directamente en la seguridad de los usuarios y la convivencia entre distintas disciplinas. Debido a lo anterior, se hace necesario fortalecer y sistematizar la organización y comunicación entre los distintos grupos de ciclistas, con el fin de promover una práctica más ordenada, segura y sostenible del Mountain Bike.

En Chile se tiene, según la última encuesta realizada por el Ministerio del Deporte en el año 2021, alrededor de 500.000 practicantes de Mountain Bike de manera regular. Esto incluye a tanto a ciclistas recreacionales como a aquellos que participan en competencias.

Las regiones con mayor número de practicantes incluyen:

- Región Metropolitana: Con un 30% del total nacional, es la zona con más ciclistas debido a la accesibilidad a parques y senderos.
- Región de La Araucanía y Los Lagos: Ambas regiones combinadas representan un 25% de los practicantes, aprovechando su geografía montañosa.
- Región de Valparaíso: Con aproximadamente un 15% de los practicantes, es otra zona popular debido a su variedad de terrenos.

El Mountain Bike en Chile ha experimentado un crecimiento promedio de entre un 8% y un 10%, especialmente en el rango etario de 18 a 35 años.

En el país se realizan eventos deportivos de nivel mundial, tales como Valparaíso Cerro Abajo, la World Enduro Series y Andes Pacífico Enduro. Estas competencias han tenido un aumento significativo en popularidad y participación en los últimos años, especialmente desde la llegada del primer evento masivo internacional. El número de participantes ha aumentado considerablemente desde 300 ciclistas en 2010 hasta más de 1500 participantes en 2023, evidenciando claramente el auge y la creciente popularidad de este deporte. Esto subraya la importancia de desarrollar una solución efectiva y definitiva que responda a las necesidades específicas de los deportistas y de la comunidad creciente que rodea esta actividad.

Los problemas más importantes detectados son los siguientes:

1. Falta de acceso a la totalidad de senderos creados por los deportistas.
2. Obligatoriedad de uso de aplicaciones de mapas, las cuales no están especializadas en el deporte MTB.
3. Necesidad de uso de aplicaciones de terceros para tener un acceso medio/bajo de información de los senderos.
4. Inexistencia de un canal de comunicación en tiempo real para informar de posibles problemas o accidentes en los senderos.
5. Imposibilidad de compartir imágenes de los senderos (tipo muro) para mantener la base de estos actualizada en aplicaciones de terceros.
6. Ausencia de una comunidad con un medio de comunicación para la coordinación y colaboración entre ciclistas.
7. Falta de aplicativos gratuitos para usuarios promedios y/o amateurs. Las aplicaciones existentes son de pago para poder tener la totalidad de funciones.

1.2.3. Requerimientos y Requisitos del Usuario.

A continuación, se listan los requerimientos que se han detectado en el cliente:

Comentado [PG3]: A continuación se listas los...

- Registrar rutas: Necesita que cada usuario pueda registrar las rutas que crea o en las que participa.
- Compartir ubicación: Compartir la ubicación con amigos dentro de la aplicación.
- Consultar clima : Permita consultar el tiempo climático en la localización donde el usuario quiera ir.
- Acceder a la totalidad de senderos: Que permita ver y utilizar senderos propios de la aplicación o los que han creado los usuarios dentro de ella.
- Permitir comunicación en tiempo real: Que permita poder dejar mensajes en perfiles de senderos o de usuarios para comunicarse .
- Compartir imágenes: Que usuarios puedan postear imágenes dentro de perfiles de usuarios, senderos y comunidades al interior de la app.
- Creación de comunidades: Que usuarios puedan crear comunidades para mantenerse en contacto, ayudarse y compartir información relevante de senderos y/o actividades.
- Descarga de mapas de manera local: Que el usuario pueda descargar mapas en la aplicación para poder ser utilizados en modo off-line.

1.3. Objetivos

Objetivo General:

Mejorar la experiencia de los practicantes de Mountain Bike en Chile, mediante el diseño e implementación de una solución informática especializada, que permita registrar, trazar y compartir senderos, visualizar condiciones climáticas en tiempo real y fomentar la creación de comunidades deportivas activas, promoviendo la práctica segura del deporte dentro del país. El tener todos los deportistas o practicantes dentro de una misma zona asegura que se apoyen entre ellos ya que las noticias o estados de los senderos se compartirán transversalmente a los usuarios y no solamente a unos pocos.

Comentado [PG4]: Se mejora la experiencia con la creación/arriendo de una aplicación??

Objetivos Específicos:

Dar solución a un problema que se viene presentando hace años en Chile.

1. Mejorar la obtención de información de senderos, información meteorológica y actividades de la comunidad del mountain bike, es decir las condiciones para la práctica de este deporte.
2. Fortalecer la comunicación entre miembros de la disciplina.
3. Optimizar la información de ubicación de cada participante.

1.4. Alternativas globales de solución

Para abordar la problemática identificada en la comunidad de mountain bike en Chile, se plantean las siguientes alternativas globales de solución, evaluadas en función de su viabilidad, costo, tiempo de implementación y alineación con los objetivos del proyecto.

1.4.1. Alternativa 1: Contratación de empresa de desarrollo de software externa

Una opción considerada es contratar a una empresa desarrolladora de software en Chile que pueda generar una solución alineada a los requerimientos establecidos, como se describe en la propuesta técnico-comercial (Véase Anexo N° 2), al analizar esta alternativa desde diversos factores relevantes, se pueden identificar los siguientes aspectos:

En términos de funcionalidad, la empresa propone una solución integral que contempla el desarrollo de una aplicación móvil compatible con iOS y Android, incorporando servicios de geolocalización, visualización de mapas, registro y compartición de senderos, así como integración con servicios externos como clima y notificaciones en tiempo real. Esto permite cubrir de manera robusta los requerimientos funcionales del sistema.

Respecto a la usabilidad, la propuesta destaca el desarrollo de una interfaz amigable, responsiva y centrada en la experiencia de usuario, considerando además la implementación de fases específicas de diseño UX/UI. Sin embargo, al tratarse de una solución desarrollada por un proveedor externo, podría existir una limitación en la personalización total de la interfaz según necesidades particulares del proyecto.

En relación con la mantención y soporte, el servicio contempla soporte técnico, mantenimiento y administración de la plataforma, lo que garantiza continuidad operativa. No obstante, esto implica una dependencia del proveedor y la existencia de costos adicionales recurrentes asociados al soporte mensual y a posibles mejoras futuras.

Por otro lado, en materia de escalabilidad, la arquitectura propuesta basada en servicios en la nube y APIs permite un crecimiento progresivo del sistema, asegurando alta disponibilidad y adaptabilidad. Sin embargo, esta escalabilidad dependerá de las condiciones contractuales establecidas, pudiendo generar restricciones o costos adicionales si no se definen adecuadamente desde el inicio.

Finalmente, en cuanto a la seguridad de la información, la propuesta considera el uso de protocolos de cifrado, infraestructura segura y cumplimiento de estándares tecnológicos, lo que garantiza un nivel adecuado de protección de datos. Aun así, el control sobre la información y su gestión dependerá del proveedor, lo que podría limitar la autonomía del proyecto en este ámbito.

No obstante, esta alternativa es descartada principalmente debido al alto costo asociado al desarrollo e implementación de la solución, el cual supera significativamente el presupuesto estimado para el proyecto. Asimismo, al compararse con un desarrollo propio, esta opción presenta desventajas en términos de flexibilidad, control y capacidad de adaptación a requerimientos específicos, lo que impide que compita de manera eficiente frente a una solución desarrollada internamente.

1.4.2. Alternativa 2: Adaptación de una Plataforma existente

Una segunda alternativa, consiste en la adaptación de una plataforma ya existente como TRAILFORK, aceptando las limitaciones inherentes respecto a su adaptación local.

Al considerar esta opción desde diversos factores clave, se observa lo siguiente:

Desde la perspectiva de funcionalidad, TRAILFORK ya ofrece características avanzadas para registro y el compartir senderos, pero podría estar limitada en cuanto a funcionalidades específicas requeridas para el contexto chileno, como integración profunda con servicios climáticos locales.

En cuanto a usabilidad, TRAILFORK es reconocida por ser intuitiva y eficiente, aunque nuevamente, la personalización al mercado chileno podría ser limitada.

Respecto a la mantención, utilizar TRAILFORK disminuiría considerablemente los costos asociados al desarrollo y mantenimiento de software, pero generaría dependencia del proveedor internacional respecto a futuras actualizaciones y mejoras.

En términos de escalabilidad, TRAILFORK presenta una estructura ya probada y establecida internacionalmente, facilitando una eventual expansión de usuarios. En este punto se debe considerar que las adaptaciones específicas que surjan podrían enfrentar limitaciones.

En materia de seguridad de la información, TRAILFORK ya posee estándares altos de seguridad y protocolos definidos internacionalmente, aunque el control directo sobre la gestión de datos locales sería limitado.

Finalmente, respecto a la arquitectura, TRAILFORK ofrece una solución sólida y ya establecida, pero cualquier requerimiento arquitectónico adicional o específico del contexto chileno requeriría negociaciones especiales con la empresa proveedora y se considerarían desarrollos a escala o específicos del país. En este sentido, sería necesario establecer acuerdos estratégicos con TRAILFORK y potenciales colaboradores locales para ampliar las funcionalidades específicas al contexto chileno mediante, por ejemplo, publicidad dentro de la aplicación y la liberación del mapa completo para usuarios nacionales.

1.4.3. Alternativa 3 (Alternativa propia): Desarrollo a la medida

Otra opción sería elegir directamente el desarrollo propuesto por la empresa Trail Hunter, partiendo desde cero. Al evaluar esta alternativa considerando distintos criterios relevantes, se observa lo siguiente:

En términos de funcionalidad, esta opción ofrece la posibilidad de adaptar y personalizar completamente cada requerimiento específico del cliente, lo que garantiza que todas las funcionalidades estén alineadas con las necesidades particulares del usuario.

Respecto a la usabilidad, al desarrollar desde cero, Trail Hunter podría asegurar una interfaz sencilla, intuitiva y totalmente adaptada a los hábitos y expectativas del usuario local.

En cuanto a la mantención, desarrollar internamente permite un control directo y simplifica futuras actualizaciones y mejoras, aunque implica una gestión activa y constante del equipo interno.

La escalabilidad es otro aspecto destacable, ya que, al diseñar la aplicación desde su inicio, se puede prever un crecimiento futuro, garantizando que la arquitectura soporte fácilmente incrementos de usuarios y funcionalidades adicionales.

En materia de seguridad de la información, al tener un desarrollo interno completo, se facilita enormemente la implementación de medidas específicas de protección y privacidad, adaptadas a las normativas locales y exigencias del cliente.

Finalmente, la arquitectura será diseñada específicamente para este proyecto, utilizando tecnologías modernas y prácticas recomendadas para facilitar el mantenimiento, escalabilidad y robustez de la solución final. Esto permitirá posicionar rápidamente a Trail Hunter en el mercado y generar confianza y seguridad en los usuarios.

1.5. Evaluación de Alternativas

En el marco del proceso de evaluación, se recibieron dos propuestas de desarrollo por parte de empresas privadas. Otra evaluación se basó en el costo de la obtención de la licencia del aplicativo TrailForks y cambios sobre este. Finalmente, se evaluó una alternativa interna considerando costos promedio de mercado.

1.5.1. Criterios de Evaluación

Para determinar la alternativa más adecuada, se establecieron los siguientes criterios de evaluación:

A. Costo de Desarrollo e Implementación (15%)

Este criterio evalúa la viabilidad económica del proyecto en relación con el alcance propuesto y los beneficios esperados. Debe considerar:

- Costos de desarrollo de software (frontend, backend, base de datos).
- Integración con servicios de mapas y APIs.
- Infraestructura tecnológica (servidores, nube, almacenamiento).
- Licencias necesarias.
- Costos de pruebas, despliegue y puesta en marcha.

- Capacitación inicial a administradores.
- Costos asociados a publicación en App Store y Google Play.

B. Propuesta Técnica y Arquitectura (15%)

Este criterio analiza la solidez tecnológica de la solución propuesta. Debe Incluir en la propuesta:

- Arquitectura del sistema (arquitectura en capas, microservicios o monolítica modular).
- Diseño de base de datos (estructuración de rutas, usuarios, reportes, multimedia).
- Integración con servicios de mapas con capacidad de personalización local.
- Escalabilidad para soportar crecimiento de usuarios.
- Rendimiento y tiempos de respuesta.
- Compatibilidad con dispositivos móviles actuales.
- Uso de estándares abiertos y buenas prácticas de desarrollo.

C. Plan de Soporte y Mantenimiento (25%)

Este criterio evalúa la continuidad operativa del sistema. Debe contemplar:

- Soporte técnico post-implementación.
- Canales de atención a usuarios.
- Tiempo de respuesta ante incidentes.
- Mantenimiento correctivo, preventivo y evolutivo.
- Monitoreo de servidores y rendimiento.
- Gestión de respaldo y recuperación ante fallos.
- Plan de contingencia ante caídas del sistema.

Se valorará la existencia de un plan formal con niveles de servicio (SLA) claramente definidos.

D. Propuesta de Garantía de Software (10%)

Este criterio evalúa el compromiso del desarrollador respecto a la calidad del producto. Debe incluir:

- Período de garantía post entrega.
- Corrección de errores sin costo adicional durante el período de garantía.
- Ajustes por fallas críticas.
- Compromiso de estabilidad operativa.
- Procedimiento formal de reporte y resolución de defectos.

E. Programa de Plazos (Inicio, Hitos y Cierre) – 10%

Este criterio evalúa la planificación y organización del proyecto. Debe Detallar:

- Fecha estimada de inicio.
- Cronograma detallado por fases: análisis, diseño, desarrollo, pruebas, implementación y puesta en marcha.
- Definición de hitos intermedios verificables.
- Fecha estimada de término.
- Identificación de riesgos que puedan afectar los plazos.

F. Seguridad de la Información (25%)

Este criterio evalúa la protección de datos personales y geolocalización de los usuarios. Debe considerar:

- Protección de datos personales conforme a normativa vigente.
- Encriptación de datos en tránsito (HTTPS, SSL/TLS).
- Encriptación de datos sensibles en base de datos.
- Autenticación segura de usuarios y control de accesos y perfiles.
- Protección contra vulnerabilidades comunes (OWASP).
- Política de privacidad clara.
- Manejo responsable de datos de ubicación en tiempo real.

Los criterios de evaluación definidos para el presente proyecto se fundamentan en el modelo de calidad de producto software establecido en la norma ISO/IEC 25010:2011, el cual define características esenciales como adecuación funcional, eficiencia del desempeño, compatibilidad, fiabilidad, seguridad, mantenibilidad y portabilidad. Asimismo, el criterio de Seguridad de la Información se complementa con los lineamientos de la norma ISO/IEC 27001:2022, asegurando la protección adecuada de los datos personales y de geolocalización gestionados por la aplicación. Fuente: ISO/IEC 25010:2011 y ISO/IEC 27001:2022

Tabla 1-2 Criterios de Evaluación

Criterio	Ponderación
Costo desarrollo e implementación	15
Propuesta técnica, arquitectura	15
Plan de soporte y mantenimiento	25
Propuesta Garantía Software	10
Programa de plazos(inicio, hitos y cierre) 10%	10
Seguridad de la Información	25
TOTAL	100

Las ponderaciones asignadas a cada criterio son las siguientes:

Tabla 1-3. Ponderaciones de Criterios de Evaluación

CRITERIO	PONDERACIÓN
Costo desarrollo e implementación	15%
Propuesta técnica, arquitectura	15%
Plan de soporte y mantenimiento	30%
Propuesta Garantía Software	10%
Programa de plazos(inicio, hitos y cierre) 10%	10%
Seguridad de la Información	20%
TOTAL:	100%

Fuente: Elaboración propia

1.5.2. Evaluación

Para evaluar cada alternativa según los criterios seleccionados, se empleará la siguiente escala de calificación:

Tabla 1-4. Escala de evaluación, Elaboración propia

CALIFICACIÓN	CONCEPTO
0	No Aplica
1	Muy Malo
2	Malo
3	Regular
4	Bueno
5	Muy Bueno

Fuente: Elaboración propia

Tabla 1-5. Alternativa N° 1

CRITERIO	CALIFICACIÓN
Costo desarrollo e implementación	4
Propuesta técnica, arquitectura	3
Plan de soporte y mantenimiento	4
Propuesta Garantía Software	4
Programa de plazos(inicio, hitos y cierre) 10%	3
Seguridad de la Información	3
Total	21

Fuente: Elaboración propia

Comentado [PG5]: Pq 2 si es un desarrollo externo a la medida??

Tabla 1-6. Alternativa N° 2

CRITERIO	CALIFICACIÓN
Costo desarrollo e implementación	4
Propuesta técnica, arquitectura	4
Plan de soporte y mantenimiento	4
Propuesta Garantía Software	3
Programa de plazos(inicio, hitos y cierre) 10%	3
Seguridad de la Información	0
Total	18

Fuente: Elaboración propia

Tabla 1-7. Alternativa N° 3

CRITERIO	CALIFICACIÓN
Costo desarrollo e implementación	4
Propuesta técnica, arquitectura	5
Plan de soporte y mantenimiento	3
Propuesta Garantía Software	4
Programa de plazos(inicio, hitos y cierre) 10%	5
Seguridad de la Información	5
Total	26

Fuente: Elaboración propia

Tabla 1-8. Ponderaciones

Criterio	Ponderación	Alternativa 1	Alternativa 2	Alternativa 3
Costo desarrollo e implementación	15%	0.6	0.6	0.6
Propuesta técnica, arquitectura	15%	0.45	0.6	0.75
Plan de soporte y mantenimiento	30%	1.2	1.2	0.9
Propuesta Garantía Software	10%	0.4	0.3	0.4
Programa de plazos(inicio, hitos y cierre) 10%	10%	0.3	0.3	0.5
Seguridad de la Información	20%	0.6	0	1
TOTAL	100%	3.55	3.0	4.15

Fuente: Elaboración propia

1.6. Alternativa Seleccionada

La Alternativa 3: Desarrollo interno del sistema obtuvo el puntaje más alto, destacándose en los criterios de funcionalidad, usabilidad y arquitectura. Esta opción permite ofrecer una solución completamente personalizada, adaptada a las necesidades locales y con un control total sobre el desarrollo y mantenimiento del producto.

El objetivo de la alternativa seleccionada, al contar con personas en el equipo que además de ser desarrolladores, son practicantes de MTB y usuarios de aplicaciones similares, se puede lograr una solución perfecta (razón por la cual se obtuvo un puntaje perfecto en las ponderaciones de Funcionalidad e Usabilidad) ya que conocen las deficiencias de las aplicaciones actuales y lo que en la comunidad de MTB se necesita y requiere en el futuro.

Comentado [PG6]: Todos tienen nota 5, pq estos son destacables??

1.6.1. Beneficios Económicos:

En este punto se enumeran algunos beneficios económicos :

1. **Generación de Ingresos Adicionales:** La monetización de la aplicación a través de publicidad, asociaciones y otras estrategias puede aumentar los ingresos de la empresa.
2. **Diferenciación Competitiva:** Ofrecer un software completo y de calidad puede diferenciar a la empresa en el mercado, atrayendo a más usuarios y generando ventajas competitivas.
3. **Atracción de Nuevos Usuarios:** La mejora de la aplicación puede atraer a nuevos usuarios, ampliando la base de clientes y aumentando el potencial de ingresos.
4. **Imagen de Marca Mejorada:** La entrega de una aplicación de calidad puede mejorar la percepción de la marca, lo que puede influir en la decisión de los usuarios de elegir productos o servicios de la empresa.
5. **Colaboración con Marcas de Equipos:** La aplicación puede servir como plataforma para colaboraciones con marcas de equipos de ciclismo, generando oportunidades de patrocinio y asociación.
6. **Análisis de Datos para la Toma de Decisiones:** La recopilación y análisis de datos de usuarios pueden proporcionar información valiosa para la toma de decisiones estratégicas, orientando el desarrollo futuro y la inversión.

1.6.2. Beneficios Sociales:

Beneficios sociales que serán satisfechos con este proyecto:

1. **Experiencia del Usuario Mejorada:** Una interfaz intuitiva y funciones útiles mejoran la experiencia de usuario, proporcionando a los ciclistas de montaña una herramienta valiosa.
2. **Fomento de la Comunidad:** La funcionalidad de comunicación y el intercambio de información pueden fomentar una comunidad activa de ciclistas, mejorando las relaciones sociales en el mundo del ciclismo de montaña.
3. **Acceso a Información Valiosa:** Proporcionar información detallada sobre senderos y condiciones climáticas ayuda a los ciclistas a tomar decisiones informadas y a mejorar su experiencia en la montaña.

4. **Colaboración con Comunidades Locales:** Facilitar la comunicación entre usuarios puede fomentar la colaboración con comunidades locales de ciclistas, promoviendo eventos y actividades conjuntas.
5. **Capacitación y Concienciación:** La aplicación puede ser una herramienta educativa, proporcionando a los usuarios información sobre seguridad, técnicas de ciclismo y conciencia ambiental.
6. **Facilitación de Interacción Social:** Funciones sociales como comentarios y valoraciones pueden mejorar la interacción entre los usuarios, fortaleciendo los lazos sociales en la comunidad de ciclismo de montaña.
7. **Promoción de un Estilo de Vida Activo:** Alentar el uso de la aplicación para la planificación de rutas puede promover un estilo de vida activo y saludable entre los usuarios.

Estos beneficios, tanto económicos como sociales, contribuyen al impacto positivo del proyecto tanto en términos empresariales como sociales en general.

1.7. Planificación

La planificación del proyecto Trail Hunter se organiza en etapas clave para garantizar una ejecución eficiente y el cumplimiento de los objetivos establecidos. Se incluyen el desarrollo del plan personal, la planificación temporal, la asignación de recursos, la estimación de costos y la gestión de riesgos.

1.7.1. Desarrollo del plan personal

Aquí se listan los distintos profesionales y técnicos que se necesitarán para el desarrollo de este proyecto de manera eficiente y eficaz, tratando de planificar para la mayor entrega de cada recurso:

- **Jefe de Proyecto:**
 - Funciones:
 - Supervisar y coordinar todas las actividades del proyecto.
 - Gestionar el presupuesto y los plazos.
 - Comunicarse con los stakeholders.
- **Analista de Negocios/Requisitos:**
 - Funciones:
 - Entender las necesidades del cliente y traducirlas en requisitos técnicos.

Comentado [PG7]: Dónde señalas cuántos de c/rol vas a necesitar??

- Colaborar con los stakeholders para definir los objetivos del proyecto.
- **Desarrollador de Backend:**
 - Funciones:
 - Desarrollar la lógica del servidor y la gestión de la base de datos.
 - Implementar las funciones del sistema.
- **Desarrollador de Frontend:**
 - Funciones:
 - Crear la interfaz de usuario.
 - Implementar las funcionalidades del cliente.
- **Diseñador de Interfaz de Usuario (UI)/Diseñador de Experiencia de Usuario (UX):**
 - Funciones:
 - Crear el diseño visual de la aplicación.
 - Mejorar la experiencia del usuario.
- **Ingeniero de Pruebas (Tester/QA):**
 - Funciones:
 - Desarrollar y ejecutar planes de prueba.
 - Identificar y reportar errores.
- **Especialista en Seguridad:**
 - Funciones:
 - Evaluar y mitigar riesgos de seguridad.
 - Garantizar la protección de datos sensibles.
- **Administrador de Base de Datos:**
 - Funciones:
 - Gestionar y optimizar la base de datos.
 - Garantizar la integridad de los datos.
- **Especialista en Redes y Servidores:**
 - Funciones:
 - Configurar y mantener la infraestructura de red y servidores.
 - Optimizar el rendimiento del sistema.

1.7.2. Planificación temporal y asignación de recursos

A continuación, se presenta la carta Gantt para el proyecto completo, desde la toma de requerimientos hasta la puesta en marcha y posteriores actualizaciones y mejoras.

Tabla 1-9. Carta Gantt

➤ Definición de Requisitos	5 días	40 horas	Analista de Negocios/Requisitos- Jefe de Proyectos
Identificar características clave	2 días	16 horas	Analista de Negocios/Requisitos- Jefe de Proyectos
definir requisitos de usuario	2 días	16 horas	Analista de Negocios/Requisitos- Jefe de Proyectos
revisar con stakeholders	1 día	8 horas	Analista de Negocios/Requisitos - Jefe de Proyectos
➤ Investigación	10 días	80 horas	Desarrollador de Backend- Jefe de Proyectos
Investigar APIs	3 días	24 horas	Desarrollador de Backend- Jefe de Proyectos (33%)
Evaluar sistemas de mapas	2 días	16 horas	Desarrollador de Backend- Jefe de Proyectos (50%)
Evaluar sistemas de climatología	2 días	16 horas	Desarrollador de Backend- Jefe de Proyectos (50%)
Estudiar integraciones entre APIs	3 días	24 horas	Desarrollador de Backend- Jefe de Proyectos (33%)
➤ Diseño de Base de Datos y Arquitectura	10 días	80 horas	Desarrollador de Backend - DBA- Jefe de Proyectos
Definir estructura de BD	3 días	24 horas	Desarrollador de Backend- Jefe de Proyectos (33%)
Diseñar la arquitectura del sistema	7 días	56 horas	DBA- Jefe de Proyectos (14%)
➤ Desarrollo del Backend	30 días	240 horas	Desarrollador de Backend - Ingeniero de Pruebas (QA)- Jefe de Proyectos
Implementar lógica del servidor	10 días	80 horas	Desarrollador de Backend - Ingeniero de Pruebas (QA)
Integrar APIs	10 días	80 horas	Desarrollador de Backend - Ingeniero de Pruebas (QA)
Desarrollar funciones del sistema	10 días	80 horas	Desarrollador de Backend - Ingeniero de Pruebas (QA)- Jefe de Proyectos (10%)
➤ Desarrollo del Frontend	30 días	240 horas	Desarrollador de Frontend - Diseñador UI/UX - Ingeniero de Pruebas (QA)- Jefe de Proyectos
Crear interfaz de usuario	20 días	160 horas	Desarrollador de Frontend - Diseñador UI/UX- Jefe de Proyectos (20%)
Implementar funciones del cliente	10 días	80 horas	Diseñador UI/UX - Ingeniero de Pruebas (QA)- Jefe de Proyectos (20%)
➤ Desarrollo de Funciones de Socialización y Comunicación	20 días	160 horas	Desarrollador de Backend - Desarrollador de Frontend- Jefe de Proyectos
Implementar funciones de mensajería	15 días	120 horas	Desarrollador de Backend- Jefe de Proyectos (33%)
Configurar Muro de Comunicación	5 días	40 horas	Desarrollador de Frontend- Jefe de Proyectos (20%)
➤ Integración de Mapas y Funciones Sociales	20 días	160 horas	Desarrollador de Backend - Desarrollador de Frontend - Jefe de Proyectos
Integrar servicios de mapas	10 días	80 horas	Backend- Jefe de Proyectos (10%)
Implementar funciones sociales	10 días	80 horas	Frontend- Jefe de Proyectos (10%)
➤ Implementación de Seguridad y Pruebas	14 días	104 horas	Especialista en Seguridad - Especialista Redes - QA- Jefe de Proyectos
Desarrollo de casos de pruebas	7 días	56 horas	Especialista de Seguridad - Especialista Redes- Jefe de Proyectos (14%)
Pruebas del sistema	6 días	48 horas	Ingeniero de Pruebas (QA)- Jefe de Proyectos (16%)
➤ Optimización y Pruebas Finales	15 días	120 horas	Ingeniero de Pruebas (QA) - Especialista en Redes y Servidores- Jefe de Proyectos
Optimizar rendimiento	5 días	40 horas	Especialista en Redes y Servidores
Pruebas finales	5 días	40 horas	Ingeniero de Pruebas (QA)
Pruebas de aceptación	5 días	40 horas	Ingeniero de Pruebas (QA) - Jefe de Proyectos (20%)
➤ Despliegue y Monitoreo	10 días	80 horas	Jefe de Proyectos - DBA - Especialista en Redes y Servidores- Jefe de Proyectos
Desplegar la aplicación	2 días	16 horas	Jefe de Proyectos - DBA - Especialista en Redes y Servidores - Jefe de Proyectos
Configurar herramientas de Monitoreo	8 días	64 horas	Especialista en Redes y Servidores - Jefe de Proyectos (13%)
➤ Mantenimiento Continuo y Mejoras	36 días	200 horas	DBA - FrontEnd - Backend- Jefe de Proyectos
Monitoreo Continuo	24 días	192 horas	DBA - FrontEnd - Backend
Implementación de Actualizaciones y mejoras	1 día	8 horas	DBA - FrontEnd - Backend -Jefe de Proyectos

Fuente: Elaboración propia

Comentado [PG8]: Ojo con la asignación del JP para el desarrollo del Backend, dura 30 días, y el JP está considerado 1 sólo día para realizar todas las labores descritas un poco más arriba en el documento... no me calza!!! Se queda corto de horas

1.7.3. Estimación de costos.

Según lo expresado en la carta Gantt (solo para los profesionales) y un salario promedio mensual para este caso, ya que en la carta se asumieron costos ceros por ser un proyecto de una sola persona que cubriría todos los puestos, serían los siguientes (Expresados en UF, sueldos promedios en Chile)

Tabla 1-10. Distribución de Costos

Rol	UF	% sueldos	distribución de costos por recurso	distribución por recurso/30 días	días trabajo en PY	costo por rol
Frontend	28,22	0,081	28,22	0,94	70,0	65,80
Negocios Requisitos	35,28	0,101	35,28	1,18	5,0	5,88
Jefe de Proyecto	63,50	0,182	63,50	2,12	33,0	69,96
Backend	35,28	0,101	35,28	1,18	90,0	105,84
Diseñador UX	35,28	0,101	35,28	1,18	30,0	35,28
Seguridad	52,92	0,152	52,92	1,76	20,0	35,28
DBA	45,86	0,131	45,86	1,53	20,0	30,58
Redes y Servidores	24,70	0,071	24,70	0,82	15,0	12,35
Ingeniero de Pruebas (QA)	28,22	0,081	28,22	0,94	15,0	14,10

Elaboración propia con datos desde Robert Half Guía salarial 2026

Los valores fueron ajustados proporcionalmente a partir de referencias de mercado (Guía Salarial Robert Half 2026), manteniendo la estructura relativa del modelo de costos original para asegurar consistencia metodológica.

Además, se muestra cómo se verá el costo final (Monto expresado en UF):

Comentado [PG9]: Acá la fuente debiera ser un medio que tenga información de cuánto le está pagando a los diferentes roles asociados a proyectos informáticos, pues con valores del mercado se podría estimar un costo más menos real del proyecto a desarrollar

Tabla 1-11. Costos totales de proyecto

Ítem	Valor (UF)
Costo proyecto para empresa	371,93
CV	20
Costo proyecto para cliente	391,93
Riesgo (5%)	19,60

Elaboración propia

Comentado [PG10]: Cómo llegas a estos valores?
Dónde lo explicas??
MEJORAR

No hay costos variables o fijos de Luz, internet, arriendo de oficina, entre otros???

Costos totales comprende trabajadores con teletrabajo, sin uso de oficinas ni costos fijos de arriendos ni costos operacionales propios de una.

1.7.4. Administración del riesgo

El desarrollo del proyecto Trail Hunter implica riesgos técnicos, operativos y organizacionales. Para garantizar una implementación exitosa, se identificaron los riesgos potenciales, evaluando su impacto y probabilidad, y se establecieron estrategias específicas de mitigación. A continuación, se presenta un resumen de los principales riesgos:

Para cálculo de riesgos :

1. Evaluar la posibilidad de que el riesgo se presente, la posibilidad se grafica indicando un valor entre 1 y 10, siendo 10 un indicador de que el riesgo es altamente posible que se presente, y 1 reflejando que el riesgo es poco probable que se presente.
2. Después se evalúa impacto en el sistema. Igual que el indicador anterior se determinar un valor que va de 1 a 10 para especificar el impacto del riesgo.
3. Luego valoriza de 1 a 10 que tan costoso seria retirar el riesgo del sistema si es que se presenta.
4. Y finalmente se pondera cada riesgo utilizando la siguiente formula

$$\text{Prioridad} = (11 - \text{Posibilidad del riesgo}) * (11 - \text{impacto}) * \text{Costo de retiro}$$

1.7.5. Riesgos detectados

- **Poco interés de stakeholders principales:** Este riesgo habla de cómo las personas interesadas en este proyecto pueden perder interés y de a poco se van desligando de este. Los principales stakeholders son sponsors y dueños/representantes de tiendas de artículos de bicicleta ya que este proyecto, para que sea de costo cero a usuarios, se basa en patrocinadores locales.
- **Falta de profesionales competentes:** Este riesgo se refiere a que las personas contratadas no manejen las tecnologías que se piensan utilizar en este proyecto.
- **Copyright:** En este riesgo surge el peligro de que alguna de las aplicaciones que ya existen y estén bien posicionadas en el mercado se sientan amenazadas por el proyecto y busquen bajar el proyecto de manera legal.
- **Cuello de botella de tráfico de DATA/GPS:** Es un riesgo que, ya en producción, entre el envío de data y la recepción de señal de internet móvil y GPS se genere un cuello de botella afectando el perfecto funcionamiento de la aplicación en terreno.
- **Interfaz poco clara o intuitiva:** Este riesgo se refiere a que, luego de la toma de requerimientos y diseño de interfaces, no se integren a los usuarios finales en las pruebas para dar el visto bueno al aplicativo.
- **Pruebas de QA se extienden más de la cuenta:** En este riesgo, que, si bien es Organizacional, afecta la calidad del *Producto de software* entregado ya que atrasa plazos y etapas del proyecto.
- **Objetivo poco claro por parte de Desarrolladores:** Este riesgo habla más claramente de que, como se plantea en un comienzo, sea una aplicación para deportistas de mountain bike, hecha por deportistas de mountain bike; por esta razón se buscará además de desarrolladores a personas que también practiquen este deporte para este proyecto.
- **Mal uso de aplicativo en zonas de reservas naturales o privadas:** Este riesgo se explica ya que, al ser una aplicación con senderos de montañas, no muchas veces estos senderos están demarcados de manera correcta haciendo que los usuarios puedan entrar a zonas restringidas y/o prohibidas causando problemas legales para los usuarios y los mismos creadores del aplicativo por incentivar el traspaso de propiedad privada o de reservas naturales protegidas por el Estado de Chile.(Ley 21600).
- **Licencias de Software y herramientas:** Este riesgo se puede materializar si los desarrolladores o cualquier persona involucrada en el proyecto no cuenta

con licencias de uso de los software o herramientas que sean necesarias para la realización del proyecto.

- **Falla en la integración de tecnologías openSource:** En este paso se vería muy afectados ya que atrasaría gran parte del proyecto, sobre todo en la etapa de desarrollo e investigación.

Tabla 1-12. Tabla de Riesgos

Ítem	Riesgo	Posibilidad	Impacto	Costo	Calculo Prioridad
Organización	Poco interés de stakeholders	8	10	10	30
Proyecto	Falta de Profesionales competentes	7	10	10	40
Proyecto	Copyright	5	10	10	60
Tecnología	Cuello de botella de trafico de data/GPS	6	9	9	90
Producto de Software	Interfaz poco clara o intuitiva	5	9	10	120
Producto de Software	Pruebas de QA se extienden más de la cuenta	4	8	7	147
Proyecto	Objetivo poco claro por parte de Desarrolladores	6	6	6	150
Organización/Ambienta	Mal uso de Aplicativo en zonas reservas naturales/Privadas	3	7	5	160
Tecnología	Licencias de Softwares y herramientas	3	4	4	224
Tecnología	Falla en la integración de tecnologías OpenSource	2	6	6	270

Elaboración propia

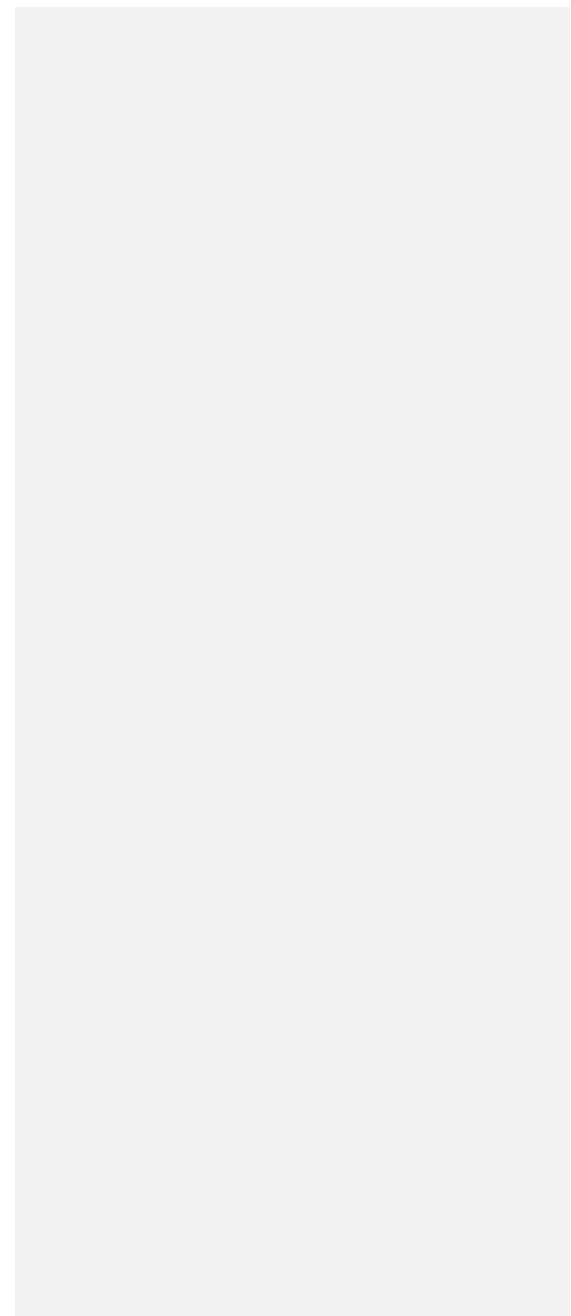
Comentado [PG11]: Dónde explicas los conceptos de esta tabla y la forma del cálculo de la prioridad?? Y cómo se debe interpretar?? El valor más alto es el mejor??

Tabla 1-13. Plan de Mitigación

Riesgo	Señal de Alerta	Responsable	Mitigación
Poco interés de stakeholders: - Patrocinadores del proyecto - Tiendas de MTB, posibles sponsors	Bajo interés debido a variedad de soluciones similares en el mercado. Usuarios no asisten a reuniones No responden correos	Jefe de Proyecto	Definir alcance y expectativas Involucrar a sponsors/patrocinadores
Falta de Profesionales competentes	Reuniones de equipo repetitivas Retraso en etapas de diseño	Jefe de Proyecto	Capacitar al profesional
Copyright	Posible riesgo debido a Usuarios/QA reportan similitud con aplicaciones en el mercado	Stakeholders PO Jefe de Proyecto	Estudiar mercado y licencias vigentes al momento de comenzar con la etapa de análisis y la toma de requerimientos evitando así un retraso en la etapa de desarrollo.
Cuello de botella de tráfico de data/GPS	Caídas de Señales (Satelitales, telefónicas, REDES 3G/4G/5G ETC)	Desarrollador Arquitecto	Realizar pruebas de estrés para tráfico de data
Interfaz poco clara o intuitiva	Usuario final poco interesado en el producto	Inteligencia de Negocios Analista Requerimientos	Investigación desde Inteligencia de Negocios y experiencia de Clientes
Pruebas de QA se extienden más de la cuenta	Atraso en etapa de Implementación	Jefe de Proyecto	Incluir usuarios finales en etapa de QA
Objetivo poco claro por parte de Desarrolladores	Reiteradas preguntas Aumento en Solicitud de Reuniones	Jefe de Proyecto Analista de Requerimientos	Seguir traza de requerimientos
Mal uso de Aplicativo en zonas reservas naturales/Privadas	Precedentes de otras aplicaciones	Desarrollo (backend) Jefe de Proyecto	Deshabilitar opciones de traza y senderos en Reservas naturales/terrenos Privados, solo dejar habilitado MAPAS descargados de manera local
Licencias de Softwares y herramientas	Reuniones sobre tecnologías recurrentes	Arquitecto JP	Buscar la mayor cantidad de tecnologías gratuitas y de código abierto
Falla en la integración de tecnologías OpenSource	No cumplimiento de plazos de etapa de diseño	Arquitecto Jefe de Proyecto	Investigar integración entre las tecnologías

Comentado [PG12]: O pones siempre JP o siempre jefe de proyecto, pero no la mezcla

Elaboración propia



**CAPÍTULO 2: ASPECTOS RELEVANTES DEL ANÁLISIS PARA LA
ALTERNATIVA SELECCIONADA**

2. ASPECTOS RELEVANTES DEL ANÁLISIS.

Comentado [PG13]: Falta poner un párrafo de introducción a lo que se verá en este capítulo

En este capítulo se ahondará en la propuesta escogida en el capítulo anterior, además de revisar aspectos relevantes de la solución propuesta como lo son sus diagramas, Modelos conceptuales, casos de uso, entre otros.

2.1. Descripción de la solución propuesta

La solución propuesta consta de una aplicación, que, mediante geolocalización, permite grabar, mostrar, buscar y llegar a senderos alrededor del territorio nacional.

Esta aplicación combina funcionalidades avanzadas de mapeo y planificación de rutas con características de integración de datos climáticos en tiempo real y una plataforma de red social dedicada.

Esta aplicación tiene como objetivo mejorar la experiencia de los ciclistas de montaña al proporcionarles las herramientas y la comunidad necesarias para explorar, disfrutar y compartir su pasión por el ciclismo de montaña

2.2. Diagrama de actividades general UML

Se **presenta** algunos de los diagramas de actividades más importantes dentro del proyecto, los que le darán un valor agregado extra y que permitirá que, nuestro producto, se diferencie del resto del mercado.

2.2.1. Flujo administrativo: Traza Sendero

La Figura 2-1 muestra el diagrama de actividades del proceso de Traza de Sendero, donde interactúan el Sistema y el Usuario.

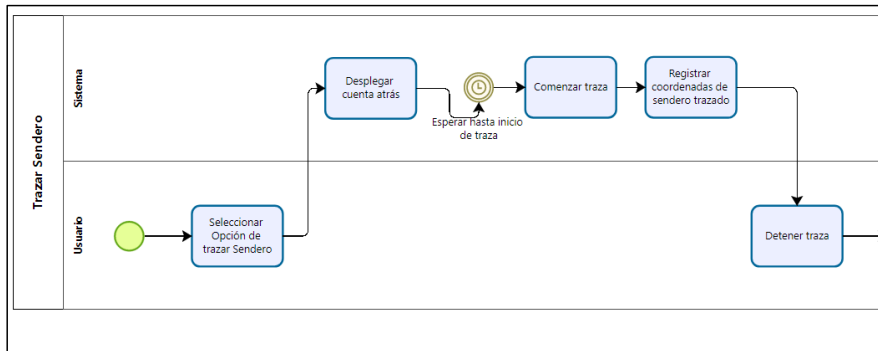


Figura 2-1. Flujo Administrativo Traza Sendero 01

Fuente: Elaboración propia

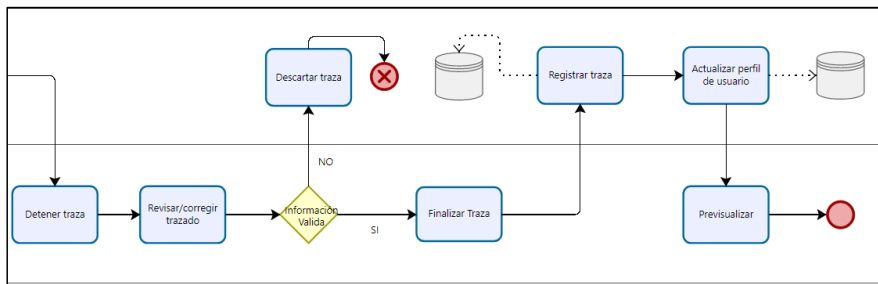


Figura 2-2. Flujo Administrativo Traza Sendero 02

Fuente: Elaboración propia

2.2.2. Flujo administrativo: Compartir Sendero

La Figura 2-3 muestra el diagrama de actividades del proceso de Compartir Sendero.

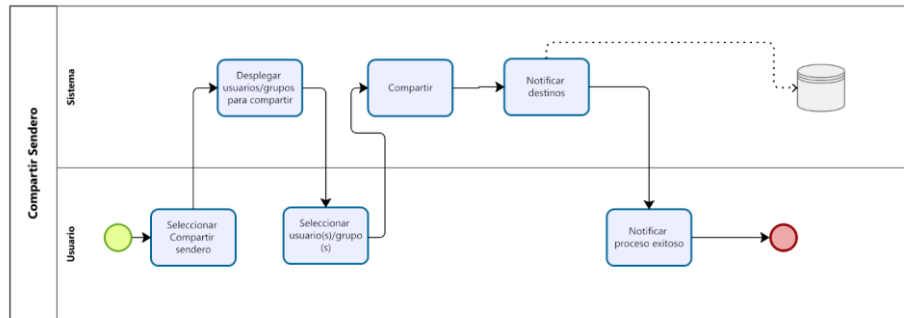


Figura 2-3. Flujo Alternativo Compartir Sendero

Fuente: Elaboración propia

Comentado [PG14]: En la actividad de notificar destinos, qué significa ese flujo a una especie de BD?? Se registran las notificaciones?? Es lo único que se registra

Comentado [AV15R14]: Si, porque el sendero queda solo en el perfil del creador, con permisos para que otros los vean, pero no se guarda en ningún otro perfil.

2.3. Requerimientos del sistema

El sistema Trail Hunter debe cumplir con un conjunto de requerimientos funcionales y no funcionales diseñados para garantizar su operatividad, accesibilidad y seguridad. Estos requerimientos han sido identificados a través del análisis de las necesidades de los usuarios y la naturaleza técnica del proyecto.

2.3.1. Requerimientos funcionales

Los requerimientos funcionales para este proyecto son los siguientes:

RF-01 Mantener Usuarios

Descripción: El sistema debe permitir al usuario la creación/edición y eliminación de usuarios.

RF-02: Guardar senderos en el perfil del usuario

Descripción: El sistema debe permitir al usuario guardar los senderos seleccionados en su perfil personal para facilitar su acceso posterior.

RF-03: Compartir ubicación con amigos

Descripción: El sistema debe permitir al usuario compartir su ubicación actual con otros usuarios marcados como "amigos" en la plataforma.

RF-04: Consulta de clima

Descripción: El sistema debe proporcionar al usuario información actualizada sobre las condiciones climáticas en los senderos disponibles.

RF-05: Acceso a senderos creados por deportistas

Descripción: El sistema debe permitir al usuario acceder a una lista completa de senderos registrados por otros deportistas en la plataforma.

RF-06 Comunicación en tiempo real para emergencias

Descripción: El sistema debe permitir al usuario el habilitar un canal de comunicación en tiempo real para que otros usuarios reporten problemas o accidentes ocurridos en los senderos.

RF-07: Compartir imágenes de senderos

Descripción: El sistema debe permitir que los usuarios suban y compartan imágenes de los senderos en un muro público, con el objetivo de mantener actualizada la base de datos.

RF-08: Creación de comunidades

Descripción: El sistema debe permitir que los usuarios creen comunidades organizadas por región o comuna, con un medio de comunicación para compartir información en tiempo real sobre senderos.

RF-09: Descargar mapas para uso sin conexión

Descripción: El sistema debe permitir que los usuarios descarguen mapas de senderos, para utilizarlos sin conexión cuando no haya acceso a redes móviles (4G/5G) o GPS.

RF-10: Acceso al perfil en modo online y offline

Descripción: El sistema debe garantizar que los usuarios puedan acceder a su perfil personal tanto en modo online como offline, sin afectar la sincronización de datos al recuperar la conexión.

RF-11: Visualización del último punto de geolocalización

Descripción: El sistema debe permitir que otros usuarios visualicen el último punto de geolocalización registrado de un usuario, únicamente si este autoriza dicha visibilidad.

RF-12: Validación de rutas ante entidades oficiales

Descripción: El sistema deberá validar, previo al almacenamiento de un sendero de tipo MTB, si la ruta propuesta está permitida para su uso, según fuentes oficiales. Esta validación se realizará consultando una API externa que contenga información de restricciones o permisos territoriales, , además de otros usuarios que puedan autorizar la creación de senderos dentro de zonas no autorizadas.

2.3.2. Requerimientos no funcionales

Los requerimientos no funcionales para este proyecto son los siguientes:

Tabla 2-1. Requerimientos no funcionales

ID	Descripción	Validación	Cómo medirlo	Valores aceptables
RNF-01	El sistema debe garantizar la seguridad y privacidad de la información almacenada en los perfiles de usuario y senderos.	Realizar pruebas de seguridad para identificar posibles vulnerabilidades en el aplicativo, como la exposición de datos sensibles o la falta de cifrado en la comunicación.	Pruebas de seguridad, cifrado de datos en tránsito/reposo	Cifrado TLS/HTTPS, sin exposición de datos personales
RNF-02	Eficiencia en el Consumo de Recursos: El sistema debe optimizar el consumo de recursos (CPU, memoria, red) para garantizar un rendimiento óptimo incluso en dispositivos con recursos limitados.	Especificar umbrales de uso de recursos, añadiéndole estas mediciones en las pruebas en terreno y de laboratorio. No superar porcentajes de uso de batería o de procesador en tareas (Bajo 5% por hora para uso de batería y entre 5 a 15% de procesador). El sistema debe pasar a segundo plano y solo registrar coordenadas si la acción fue gatillada.	Medición de CPU y batería en uso prolongado	Batería <5%/hora, CPU <15% uso en dispositivos estándar
RNF-03	El sistema debe ser capaz de manejar un crecimiento en el número de usuarios, senderos y comunidades sin comprometer su rendimiento.	Tener garantizado un mínimo de usuarios simultáneos. Simulaciones de carga con 10,000 usuarios concurrentes garantizarían que el tiempo de respuesta promedio se mantenga por debajo de 3 segundos aun con masividad de usuarios simultáneos.	Pruebas de carga con usuarios concurrentes.	Tiempo de respuesta <3s con 10,000 usuarios simultáneos.
RNF-04	La interfaz de usuario debe ser fácil de usar, intuitiva y	Realización de pruebas de usabilidad con al menos 20 usuarios de perfiles diversos para evaluar la claridad y	Pruebas de usuarios con encuesta SUS	Puntaje SUS ≥ 80 en usuarios diversos

	accesible para usuarios de diferentes niveles de habilidad.	accesibilidad del diseño. Cada interfaz debe obtener sobre 80 puntos en promedio en encuesta SUS.		
RNF-05	El sistema debe ser capaz de recuperarse rápidamente de fallos inesperados, garantizando la integridad de los datos y minimizando las interrupciones en la experiencia del usuario.	Se agregarán pruebas de fallos en las pruebas estándar de funcionamiento.	Simulación de errores y recuperación	Sistema se recupera sin pérdida de datos en <10s
RNF-06	El sistema debe minimizar el consumo de energía en dispositivos para prolongar la duración de la batería y reducir el impacto ambiental.	Especificar umbrales de consumo de energía, como El sistema debe consumir menos de 5% de batería por hora calculables y medibles en uso continuo en dispositivos móviles estándar.	Medición de batería en dispositivos móviles	Consumo <5% de batería por hora de uso continuo
RNF-07	El sistema debe permitir la implementación de actualizaciones, parches y mejoras en componentes individuales sin afectar el funcionamiento general del sistema, manteniendo la disponibilidad del servicio durante el proceso.	Ejecutar pruebas de despliegue controlado de actualizaciones y parches en ambiente de pruebas/preproducción.	• Porcentaje de actualizaciones realizadas sin caída del servicio • Tiempo de indisponibilidad durante una actualización • Cantidad de módulos no afectados que continúan operativos tras el parche • Tiempo de reversión ante falla de actualización	• Minutos de indisponibilidad no planificada durante la aplicación de parches en ambiente de prueba = 0. • Todos los módulos no intervenidos deben mantener su operación normal después del update. • Tiempo máximo de reversión ante fallo: 5 minutos.

Comentado [PG16]: Esto permite validar este requerimiento no funcional?? REVISAR

RNF-08	El sistema debe ser compatible con tecnologías y servicios existentes, como APIs de clima, servicios de mapas y redes sociales, para facilitar la interoperabilidad y la integración con otras plataformas.	Integraciones deben ser con aplicaciones/APIs que sean OPEN-SOURCE para garantizar el funcionamiento sin costo para los usuarios.	Integración con APIs y servicios externos	Funciona con APIs públicas y formatos estándar (REST, JSON)
RNF-09	El sistema debe minimizar el uso de ancho de banda durante la transferencia de datos, especialmente en áreas con acceso limitado a Internet, mediante técnicas como compresión de datos y caché local.	Probar la aplicación en diferentes condiciones de red, incluyendo redes móviles (4G, 5G), Wi-Fi y conexiones de datos intermitentes o de baja calidad.	Simulación de red baja/intermitente	Transferencia con compresión, buen funcionamiento con 3G mínimo
RNF-10	El sistema debe ser compatible con una amplia gama de dispositivos móviles, tabletas y navegadores web, así como con diferentes sistemas operativos (iOS, Android, etc.), garantizando una experiencia consistente para todos los usuarios.	Verificar la interoperabilidad del aplicativo con diferentes versiones de sistemas operativos móviles (iOS, Android) y dispositivos (teléfonos, tabletas) populares. Probar el aplicativo en una variedad de dispositivos para garantizar una experiencia consistente para todos los usuarios.	Pruebas en múltiples OS y dispositivos	Operatividad en Android/iOS, móviles y tabletas populares

Comentado [PG17]: Tienen algún formato o protocolo determinado??

RNF-11	El sistema debe tener mismas funcionalidades operativas tanto en modo ONLINE y OFFLINE	Testear en terreno y en ambiente controlado aplicativo con y sin redes conectadas a los dispositivos con los controles previos como haber descargado los mapas para poder funcionar de manera offline y haber iniciado sesión.	Simulación de uso sin conexión	Funcionalidad completa (salvo sincronización) con mapas descargados
RNF-12	El sistema debe funcionar solo con GPS en zonas de reservas naturales o propiedades privadas.	Testear en terreno que se deshabiliten las funcionalidades de traza de senderos en estas zonas y que el usuario solo pueda tener acceso al mapa (de manera online o descargado de manera local).	Pruebas en zonas prohibidas	Trazado deshabilitado en reservas, acceso solo a mapa local

Fuente: Elaboración propia

2.4. Estructura funcional del sistema (Caso de uso general)

La estructura funcional del sistema Trail Hunter se define mediante un conjunto de casos de uso que representan las principales interacciones entre los usuarios y las funcionalidades del sistema. Estos casos de uso se han diseñado para garantizar que la solución aborde las necesidades específicas de los practicantes de mountain bike en Chile, maximizando la eficiencia, usabilidad y accesibilidad.

Tabla 2-2. Descripción de Casos de Uso

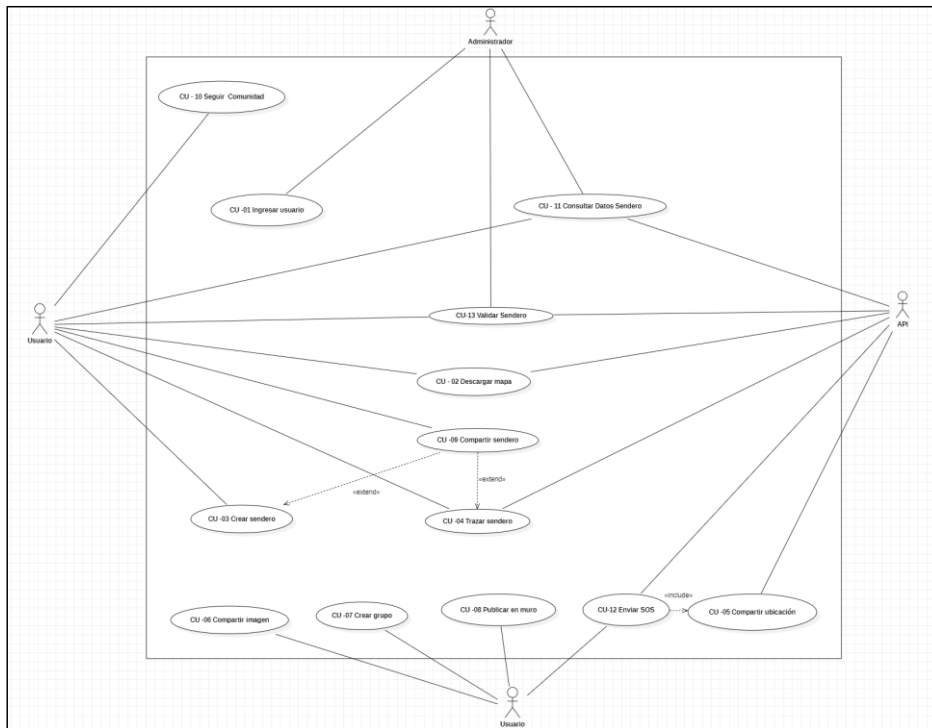
CASO DE USO	DESCRIPCIÓN
CU-01 Ingresar usuario	Este caso de uso se encarga básicamente del CRUD del usuario. Están los 4 mantenedores dentro del mismo caso de uso.
CU-02 Descargar mapa	Este caso de uso se encarga principalmente de, permitir la descarga del mapa de la región de tu preferencia para poder ser utilizado en modo OFFLINE.
CU-03 Crear sendero	Este caso de uso permite la creación de un sendero: Crear nombre, guardarlo y asignarlo al usuario. Sin haberlo trazado aún. Deshabilitado si detecta que está dentro de alguna reserva natural.
CU-04 Trazar sendero	Este caso de uso es el encargado de la traza, el dibujado, del sendero mediante las coordenadas en el mapa ya sea online o guardado y lo dibuja dentro de este. Deshabilitado si detecta que está dentro de alguna reserva natural.
CU-05 Compartir ubicación	Este caso de uso se encarga del requerimiento de poder compartir la ubicación con "amigos" en tiempo real dentro de la aplicación.
CU-06 Compartir imagen	Este caso de uso, de igual manera que el CU-08 Publicar Muro, permite subir una imagen ya sea dentro de un sendero, un grupo/comunidad o al muro de un amigo.
CU-07 Crear Grupo/Comunidad	Este caso de uso permite crear la funcionalidad que de atribuciones al usuario de poder crear una comunidad o grupo donde varios usuarios puedan juntarse y organizarse.
CU-08 Publicar Muro	Este caso de uso permite poder publicar en un muro ya sea de un amigo, algún grupo o en un sendero para dejar un mensaje genérico o para alertar a otros usuarios de algún problema con un sendero o emergencias dentro de este.
CU-09 Compartir sendero	Este caso de uso permite, al igual que el anterior, poder compartir un sendero creado y trazado por un usuario en particular con grupos o amigos dentro o fuera de su propio perfil.
CU-10 Seguir grupo / Comunidad / usuario	Este caso de uso permite hacer amigos dentro de la aplicación como también poder unirse a comunidades o grupos ya creados por otros usuarios.
CU-11 Consultar Datos sendero	Este caso de uso permite consultar datos del sendero como el autor, la longitud, elevación, estado de este dentro de otros datos.
CU-12 Enviar SOS	Este caso de uso permite enviar mensaje a contactos del celular junto con la ubicación del usuario en caso de emergencia.
CU-13 Validar Ruta	Este caso de uso es el que se encarga de hacer las validaciones, tantos de usuarios como desde la misma API pasando por el administrador del sistema.

Fuente: Elaboración propia

2.4.1. Diagrama de caso de uso general

El siguiente diagrama representa los casos de uso generales del sistema, mostrando las interacciones entre los actores y las funcionalidades principales:

Figura 2-4. Caso de uso General



Fuente: Elaboración propia

Comentado [PG18]: Revisar Extend del CU12 al CU05, pq es extend y no incluye??

2.5. Matriz de trazabilidad

A continuación, se presenta la matriz de trazabilidad, que utilizaremos para hacer seguimiento a cada requerimiento y que caso de uso da solución a este.

Matriz de trazabilidad para los Casos de uso explicados en secciones anteriores.

Comentado [PG19]: Para qué sirve, cuál es su uso??
Explicar antes de presentar

Tabla 2-3. Matriz de Trazabilidad, Elaboración propia

	Casos de Uso												
	01	02	03	04	05	06	07	08	09	10	11	12	13
RF-01	X												
RF-02			X	X									
RF-03					X							X	
RF-04											X		
RF-05									X		X		
RF-06					X			X				X	
RF-07						X		X	X				
RF-08							X			X			
RF-09		X									X		
RF-10		X	X	X							X	X	
RF-11					X					X		X	
RF-12													X

Fuente: Elaboración propia

2.6. Modelo conceptual

El modelo conceptual de Trail Hunter describe las entidades principales y sus relaciones, necesarias para implementar las funcionalidades clave del sistema. Este modelo se basa en un análisis detallado de los requerimientos funcionales y no funcionales, asegurando que se satisfagan las necesidades de los usuarios de manera eficiente.

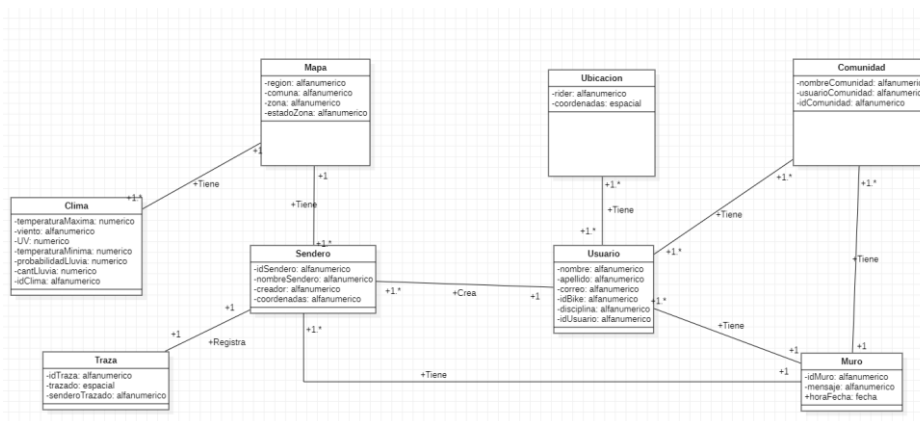


Figura 2-5. Modelo Conceptual

Fuente: Elaboración propia

2.7. Modelo de casos de uso.

A continuación, se presentan los casos de uso narrativos del sistema Trail Hunter, que describen en detalle las interacciones entre los usuarios y las funcionalidades clave del sistema. Se incluyen los respectivos diagramas de secuencia además de los contratos para casos de uso que representan el corazón del proyecto.

2.7.1. Caso de Uso: CU01 – Mantener Usuario

Tabla 2-4. Caso de Uso Narrativo: CU01 - Mantener Usuario

Caso de uso:	Mantener Usuario
Código:	CU01
Resumen:	Permite al usuario ingresar, editar y eliminar un perfil de usuario.
Tipo:	Evidente
Referencias Cruzadas:	Ninguna.
Actor:	Usuario.
Precondiciones:	El usuario debe contar con una dirección de correo electrónico válida. El sistema debe estar en funcionamiento y accesible.
Postcondición:	El usuario quedó registrado en la base de datos. El usuario recibió un correo de confirmación con las credenciales de acceso
Flujos Normal: El sistema completa con éxito la traza del sendero.	
Usuario	Sistema
1 El usuario accede a la opción "Registrar Usuario" en la pantalla de inicio.	2 El sistema muestra un formulario de registro
3 El usuario completa los campos requeridos (nombre, correo, contraseña, etc.)	4 El sistema valida los datos ingresados
6 El usuario recibe una notificación de confirmación.	5 El sistema registra al usuario en la base de datos.
Flujo Alternativo: Datos inválidos	
Usuario	Sistema
4.a.2 El usuario puede corregir la información y reenviar el formulario	4.a.1 El sistema muestra un mensaje de error indicando los campos incorrectos. 4.a.3 Vuelve al punto 5 del Flujo normal.

Fuente: Elaboración propia

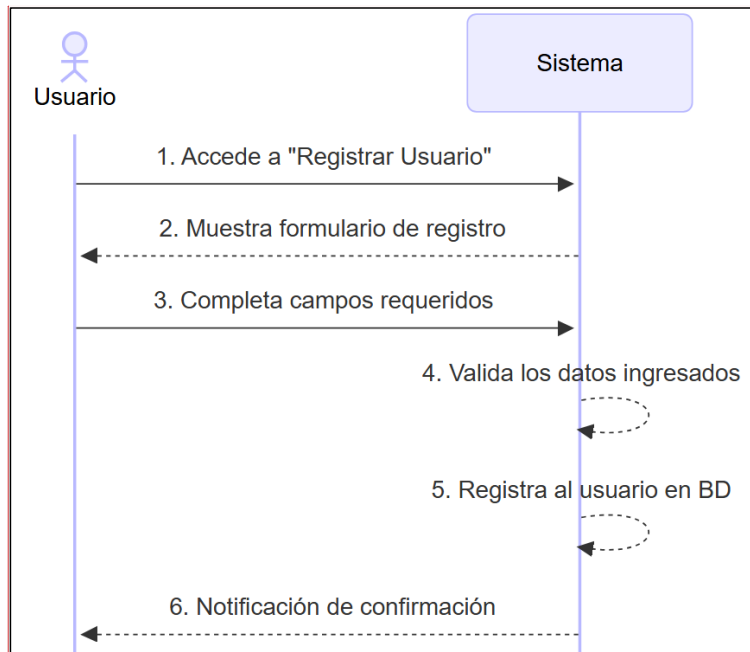


Figura 2-6. Diagrama de Secuencia: CU01 – Mantener Usuario

Fuente: Elaboración propia

Comentado [PG20]: Acá se busca mostrar más
menos la interacción entre los
mensajes/métodos

Por tanto quitar Accede a
Dejar
Seleccionar opción registrar_usuario (o solo
registrar_usuario)
Mostrar formulario
Completar campos formulario
Validar datos
Registrar...

2.7.2. Caso de uso: CU02 – Descargar Mapa

Tabla 2-5. Caso de uso Narrativo: CU02 – Descargar Mapa

Caso de uso:	Descargar Mapa
Código:	CU02
Resumen:	Permite al usuario descargar un mapa de manera local en su dispositivo.
Tipo:	Evidente
Referencias Cruzadas:	Ninguna
Actor:	Usuario; Open Street Map API
Precondiciones:	El usuario debe estar autenticado en el sistema. El dispositivo debe contar con espacio de almacenamiento suficiente.
Postcondición:	El mapa quedó almacenado localmente en el dispositivo del usuario.
Flujos Normal: El sistema completa con éxito la descarga del mapa.	
Usuario	Sistema
1 El usuario selecciona la opción "Descargar Mapa" en el menú principal	2 El sistema muestra una lista de mapas disponibles.
3 El usuario selecciona el mapa deseado.	4 El sistema descarga el mapa y lo guarda en el dispositivo. 5 El sistema notifica al usuario que la descarga se ha completado.
Flujos Alternativo: Mapa no disponible	
Usuario	Sistema
	4.a.1 El sistema muestra un mensaje de error. 4.a.2 El flujo termina sin volver al flujo normal.

Fuente: Elaboración propia

Comentado [PG21]: Esta lista en base a qué filtros opera??
Es por la región, por la ubicación GPS? O muestra todos??

Comentado [AV22R21]: Muestra mapas de la zona donde se encuentre o global, según el usuario elija.

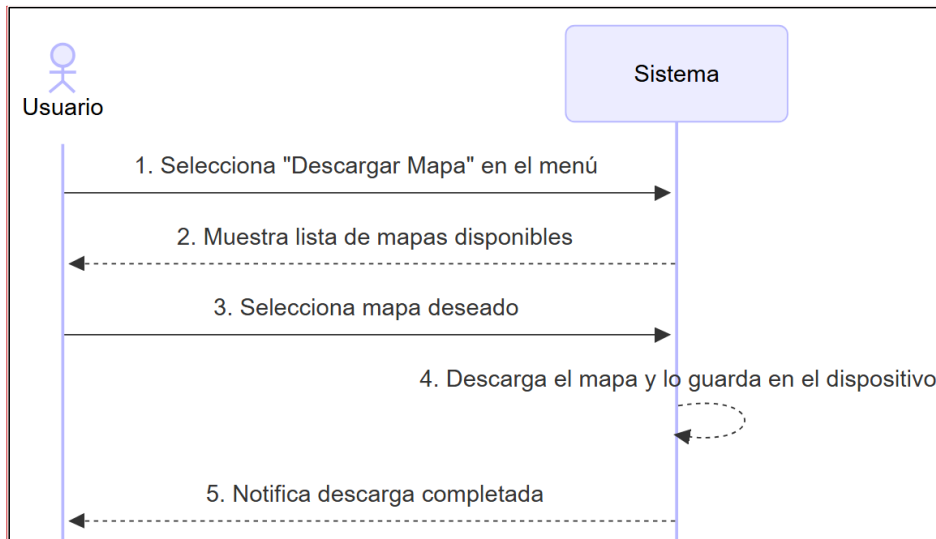


Figura 2-7. Diagrama de Secuencia: CU02 – Descargar Mapa

Fuente: Elaboración propia

Comentado [PG23]: Los comentarios del DSS 1 aplican para todos, no los pondré nuevamente

2.7.3. Caso de uso: CU03 – Crear Sendero

Tabla 2-6. Caso de uso Narrativo: CU03 - Crear Sendero

Caso de uso:	Crear Sendero
Código:	CU03
Resumen:	Permite al usuario crear y administrar un sendero dentro de su perfil.
Tipo:	Evidente
Referencias Cruzadas:	CU10 - Actualizar Data
Actor:	Usuario
Precondiciones:	El usuario debe estar registrado y autenticado en el sistema. Debe contar con permisos para agregar nuevos senderos.
Postcondición:	El sendero creado quedó registrado en la base de datos.
Flujo Normal: El sistema completa con éxito la creación de un sendero.	
Usuario	Sistema
1 El usuario selecciona la opción que desea efectuar: Crear/Editar/Eliminar Sendero" en el sistema.	2 El sistema muestra un formulario según opción seleccionada.
3 El usuario completa el formulario y confirma.	4 El sistema valida la información ingresada.
	5 El sistema determina acción solicitada por usuario y ejecuta cambios. El sistema notifica al usuario que
	6 acciones han sido realizadas.
Flujo Alternativo: Datos inválidos	
Usuario	Sistema
4.a.2 El usuario puede corregir la información y reenviar el formulario	4.a.1 El sistema muestra un mensaje de error indicando los campos incorrectos.
	4.a.3 Vuelve al punto 4 del Flujo normal.

Fuente: Elaboración propia

Comentado [PG24]: Acá tbn se valida el hecho de que el sendero ya esta registrado? O son registros locales para el usuario?

Comentado [AV25R24]: Puede haber un sendero ya creado, pero el usuario puede trazarlo para añadirle variantes nuevas lo que lo haría "un nuevo sendero"

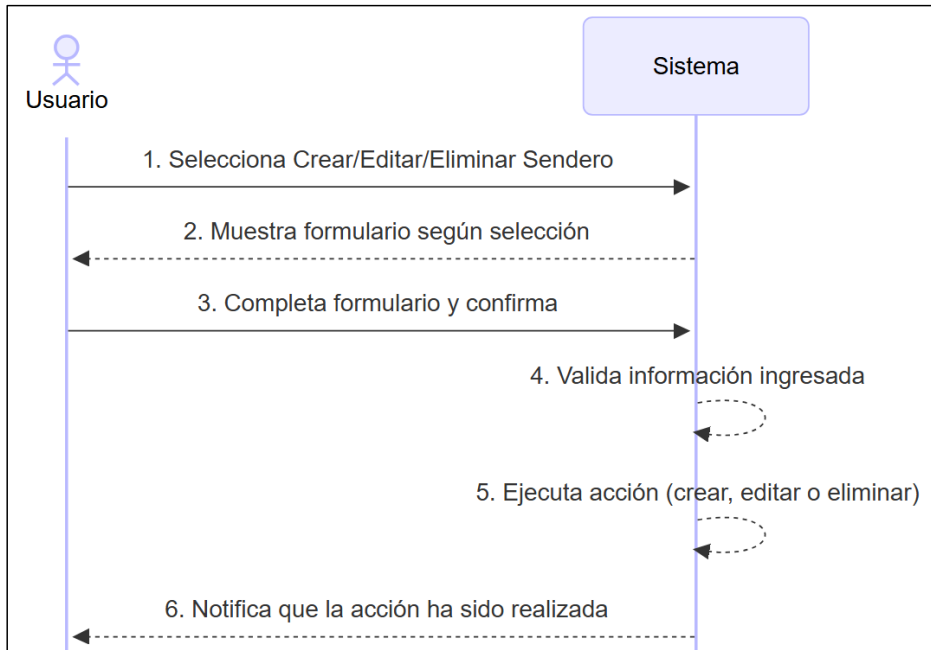


Figura 2-8. Diagrama de Secuencia: CU03 – Crear Sendero

Fuente: Elaboración propia

2.7.4. Caso de uso: CU04 – Trazar Sendero

Tabla 2-7. Caso de uso Narrativo: CU04 - Trazar Sendero

Caso de uso:	Trazar sendero
Código:	CU04
Resumen:	Permite al usuario trazar mediante geolocalización un sendero.
Tipo:	Evidente
Referencias Cruzadas:	CU03 - Crear Sendero, CU01 - Mantener Usuario
Actor:	Usuario; Open Street Map API
Precondiciones:	Debe existir un usuario. El usuario debe haber creado un sendero previamente y haber seleccionado "trazar sendero".
Postcondición:	Se trazó correctamente el sendero.
Flujo Normal: El sistema completa con éxito la traza del sendero	
Usuario	Sistema
1 El usuario selecciona la opción de "trazar sendero".	2 El despliega una cuenta atrás para el usuario que advierte el tiempo que queda para empezar a trazar la ruta usando el GPS del equipo.
4 El usuario detiene la traza del sendero.	3 Sistema registra coordenadas del sendero trazado.
5 El usuario revisa/corriga el trazado.	
6 El usuario da por finalizado el sendero.	7 El sistema registra la trazada a la BD.
	8 El sistema despliega la confirmación de la traza y previsualización del sendero.
Flujo Alternativo: Red no disponible	
Usuario	Sistema
7.a.2 El usuario selecciona la opción "Guardar local"	7.a.1 El sistema entrega la opción de guardar sendero de manera local.
	7.a.3 Vuelve al punto 10 del Flujo normal.

Fuente: Elaboración Propia

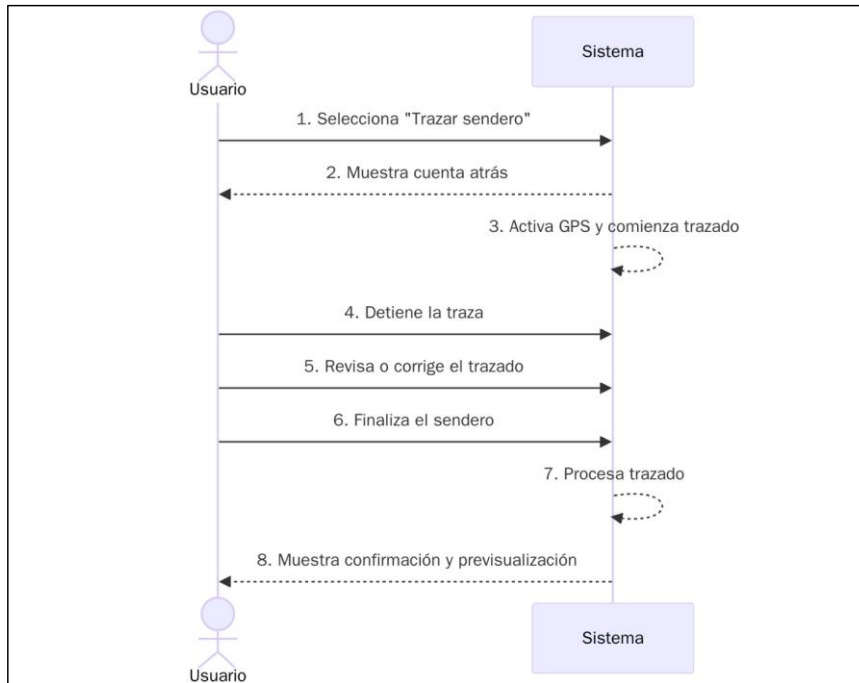


Figura 2-9. Diagrama de Secuencia: CU04 - Trazar Sendero

Fuente: Elaboración propia

Tabla 2-8. Contrato: CU04 – Trazar Sendero

Operación:	Confirmar trazado
Referencias cruzadas:	CU_04 Trazar Sendero.
Precondiciones:	Existe un usuario. Se creo previamente un sendero.
Postcondiciones:	Se creó una instancia de trazarSendero trsn. • trsn se asoció a usuario • trsn.usuario pasó a ser usuario • trsn.nombre pasó a ser nombre • trsn.fechaCreacion pasó a ser fecha creación • trsn.traza pasó a ser traza • trsn.mapa pasó a ser mapa

Fuente: Elaboración propia

2.7.5. Caso de uso: CU05 – Compartir ubicación

Tabla 2-9. Caso de uso Narrativo: CU05 - Compartir Ubicación

Caso de uso:	Compartir Ubicación
Código:	CU05
Resumen:	Permite al usuario publicar en el Muro de un usuario o de un grupo/comunidad.
Tipo:	Evidente.
Referencias Cruzadas:	Ninguna.
Actor:	Usuario; Open Street Map API
Precondiciones:	El usuario debe estar registrado y autenticado en el sistema. El dispositivo del usuario debe contar con un sensor GPS habilitado. Debe existir conexión a Internet para la sincronización de datos.
Postcondición:	La ubicación compartida quedó registrada en la base de datos con un identificador asociado al usuario. Los usuarios autorizados pueden visualizar la ubicación compartida en tiempo real.
Flujo Normal: El sistema permite compartir ubicación del usuario	
Usuario	Sistema
1 Accede a la opción "Compartir Ubicación" desde el menú principal.	2 El sistema solicita permisos para acceder al GPS del dispositivo.
3 Otorga los permisos necesarios.	4 El sistema obtiene las coordenadas actuales del usuario.
5 Selecciona los contactos o comunidades con las que desea compartir su ubicación.	6 El sistema envía la información al servidor, donde queda registrada.
Flujos Alternativos: Red no disponible	
Usuario	Sistema
	3.a.1 El sistema envía mensaje al usuario de sistema sin conexión.
	3.a.2 El flujo termina.

Fuente: Elaboración Propia

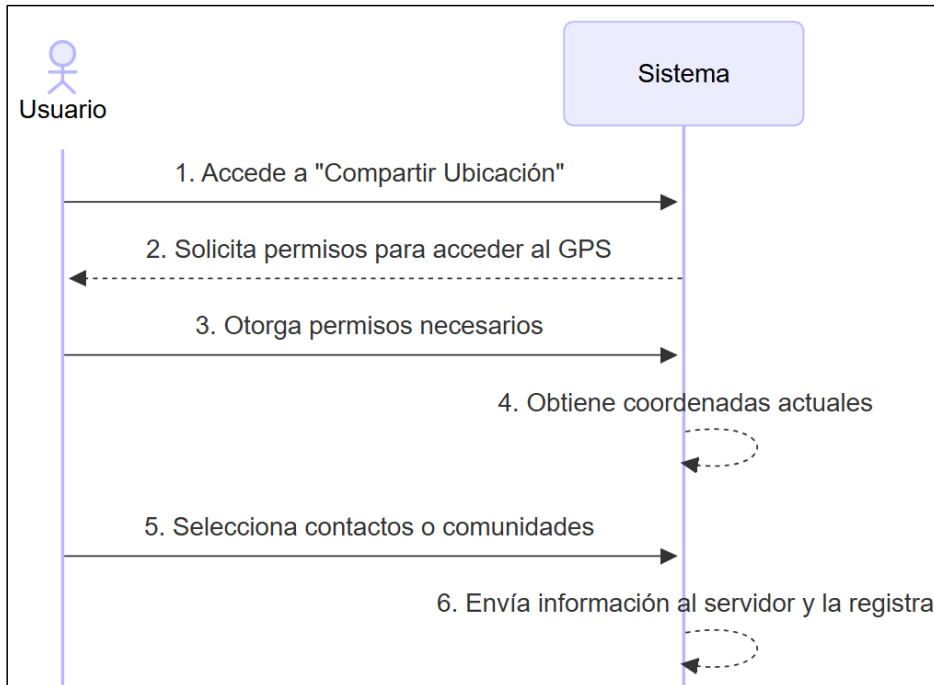


Figura 2-10. Diagrama de Secuencia: CU05 - Compartir Ubicación

Fuente: Elaboración Propia

2.7.6. Caso de uso: CU06 – Compartir Imagen

Tabla 2-10. Caso de uso Narrativo: CU06 - Compartir imagen

Caso de uso:	Compartir Imagen
Código:	CU06
Resumen:	Permite al usuario publicar en el Muro de un usuario o de un grupo/comunidad.
Tipo:	Evidente.
Referencias Cruzadas:	No tiene
Actor:	Usuario.
Precondiciones:	El usuario debe estar registrado y autenticado en el sistema. El dispositivo debe contar con una cámara o acceso a una galería de imágenes. Debe existir conexión a Internet.
Postcondición:	La imagen compartida quedó almacenada en el servidor con un identificador único.
Flujos Básico	El sistema permite compartir imagen a un usuario.
Flujo Normal	
Usuario	Sistema
1 Selecciona la opción "Compartir Imagen" en el sistema.	2 Muestra opciones para capturar una foto o seleccionar una imagen de la galería.
3 Elige una de las opciones y selecciona o captura la imagen.	4 Permite al usuario agregar una descripción opcional a la imagen.
5 Selecciona los destinatarios (usuarios o comunidades).	6 Carga la imagen al servidor y la vincula al usuario que la envió. 7 Los destinatarios reciben una notificación y pueden visualizar la imagen.
Flujo Alternativo: Si el usuario cancela la selección de la imagen	
Usuario	Sistema
	5.a.1 El sistema muestra un mensaje de cancelación
	5.a.2 El caso de uso finaliza sin realizar ninguna acción.

Fuente : Elaboración Propia

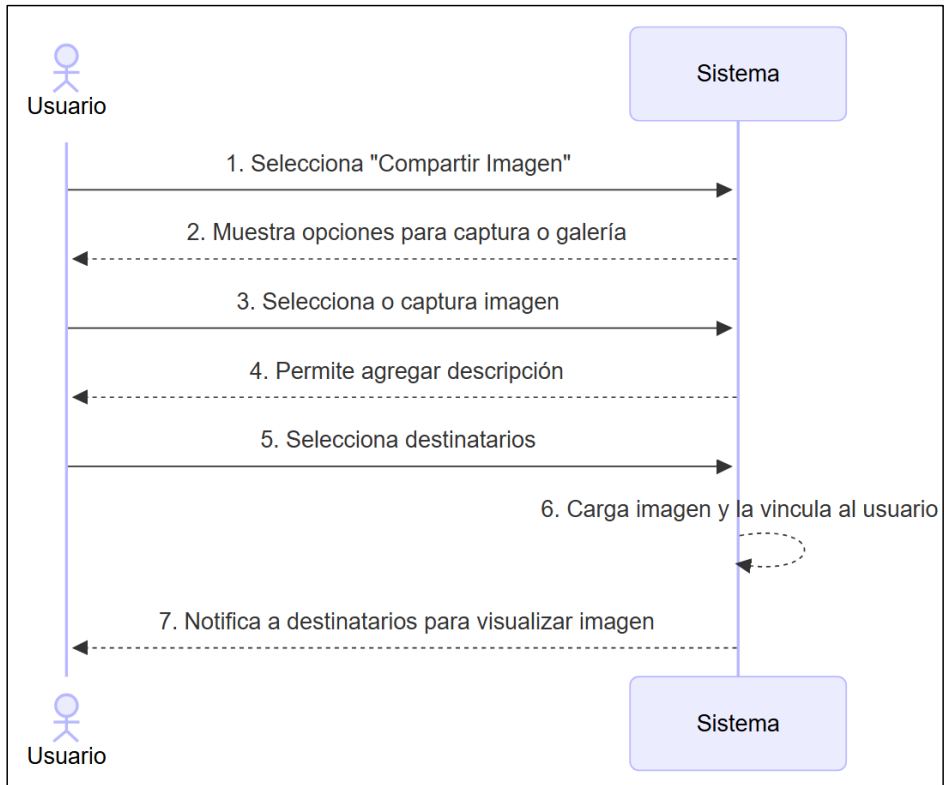


Figura 2-11. Diagrama de Secuencia: CU06 - Compartir Imagen

Fuente: Elaboración Propia

2.7.7. Caso de uso: CU07 – Crear Comunidad

Tabla 2-11. Caso de uso Narrativo: CU07 - Crear Comunidad

Caso de uso:	Crear Comunidad
Código:	CU07
Resumen:	Permitir que un usuario cree una comunidad dentro del sistema para compartir información y comunicarse con otros usuarios.
Tipo:	Evidente.
Referencias Cruzadas:	No tiene
Actor:	Usuario.
Precondiciones:	El usuario debe estar autenticado en el sistema.
Postcondición:	El usuario creó una comunidad
Flujo Normal: El sistema permite compartir imagen a un usuario	
Usuario	Sistema
1 El usuario selecciona la opción "Crear comunidad" en su perfil de usuario.	2 El sistema despliega la plantilla para poder crear la comunidad con los datos solicitados.
3 El usuario completa la información del formulario y selecciona "CREAR" dentro del menú.	4 El sistema permite al usuario crear la comunidad luego de validar que los datos son correctos. 5 El sistema envía mensaje de creación satisfactoria al usuario.
Flujo Alternativo: Datos inválidos	
Usuario	Sistema
4.a.3 El usuario nuevamente manda formulario con datos corregidos	4.a.1 El sistema muestra un mensaje de error de datos ingresados 4.a.2 El sistemas solicita nuevamente datos correctos. 4.a.4 El sistema valida correctamente los datos ingresados. Vuelve al punto 5 del flujo normal.

Fuente: Elaboración Propia

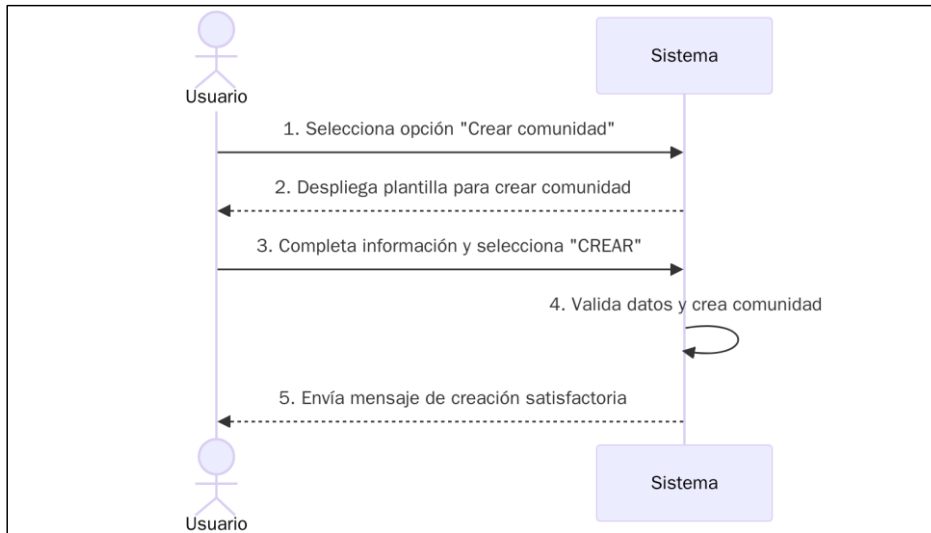


Figura 2-12. Diagrama de Secuencia: CU07 - Crear Comunidad

Fuente: Elaboración Propia

2.7.8. Caso de uso: CU08 – Publicar Muro

Tabla 2-12. Caso de uso narrativo: CU08 - Publicar Muro

Caso de uso:	Publicar Muro
Código:	CU08
Resumen:	Permite al usuario publicar en el Muro de un usuario o de un grupo/comunidad.
Tipo:	Evidente
Referencias Cruzadas:	No tiene
Actor:	Usuario
Precondiciones:	Debe ser amigo del usuario o pertenecer al grupo donde se desea publicar.
Postcondición:	Se publicó en el muro de un usuario o grupo.
Flujo Normal: El sistema permite publicar con éxito en un muro	
Usuario	Sistema
1 El usuario ingresa al perfil del usuario y/o grupo. 2 El usuario escribe mensaje en el muro de otro usuario o algún grupo.	3 El sistema captura mensaje. 4 El sistema publica mensaje en el Muro de usuario/grupo. 5 El sistema actualiza información de usuario. 6 El sistema registra publicación 7 El sistema notifica a usuario de publicación exitosa.
Flujo Alternativo: Permisos insuficientes (en caso de que usuario quiera compartir con algún usuario que no es "amigo" o algún grupo al que no pertenece)	
Usuario	Sistema
	3.a.1 El sistema envía notificación a usuario que no tiene los permisos suficientes para compartir con el/los 3.a.2 usuario/s o grupo/s . El sistema da por terminado el flujo.

Fuente: Elaboración Propia

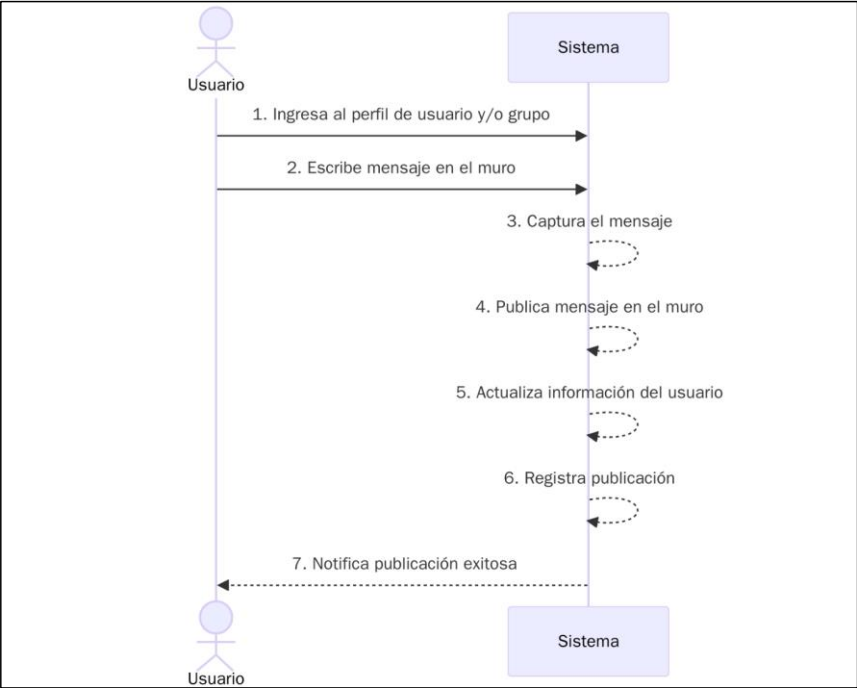


Figura 2-13. Diagrama de Secuencia: CU08 - Publicar Muro
Fuente: Elaboración Propia

Tabla 2-13. Contrato: CU08 - Publicar Muro

Operación:	Confirmar Publicar Muro
Referencias cruzadas:	CU08 - Publicar Muro.
Precondiciones:	- Existe un usuario. - Existen permisos de publicación en muro.
Postcondiciones:	Se creó una instancia de publicarMuro pmuro. - pmuro se asoció a usuario. - pmuro.mensaje pasó a ser mensaje. - pmuro.usuario pasó a ser usuario. - pmuro.fechaCreacion pasó a ser fecha creación. - pmuro.notificacion pasó a ser notificación.

Fuente: Elaboración Propia

2.7.9. Caso de uso: CU09 - Compartir sendero

Tabla 2-14. Caso de uso narrativo: CU09 - Compartir Sendero

Caso de uso:	Compartir sendero	
Código:	CU09	
Resumen:	Permite al usuario compartir un sendero recién trazado o uno que se haya trazado con anterioridad.	
Tipo:	Evidente	
Referencias Cruzadas:	CU08 - Publicar en Muro	
Actor:	Usuario	
Precondiciones:	Debe existir un sendero trazado previamente.	
Postcondición:	Se compartió correctamente el sendero.	
Flujos Básico: El sistema comparte con éxito un sendero.		
Usuario		Sistema
1	El usuario selecciona la opción de “compartir sendero”.	2 El sistema despliega un listado de opciones de usuarios/grupos donde el desee compartir.
3	El usuario selecciona el/los usuario/s y el/los grupo/s.	4 El Sistema comparte en el muro de el/los usuario/os y el/los grupo/s que el usuario selecciona
		5 El sistema notifica al usuario que el sendero ha sido compartido.
		6 El sistema envía notificación a los usuarios que se les ha compartido un sendero.
		7 El sistema actualiza datos de usuarios y grupos El sistema notifica a usuarios.
Flujo Alternativo: Permisos insuficientes (en caso de que usuario quiera compartir con algún usuario que no es “amigo” o algún grupo al que no pertenece)		
Usuario		Sistema
	4.a.1	El sistema envía notificación a usuario que no tiene los permisos suficientes para compartir con el/los usuario/s o grupo/s .
	4.a.2	El sistema da por terminado el flujo.

Fuente: Elaboración Propia

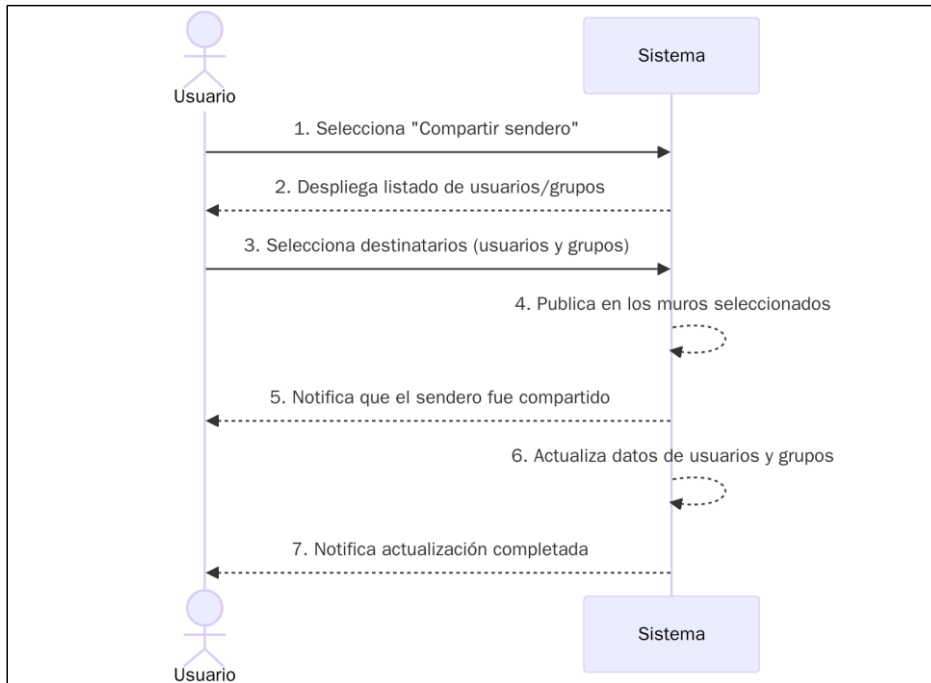


Figura 2-14. Diagrama de Secuencia: CU09 - Compartir Sendero

Fuente: Elaboración Propia

2.7.10. Caso de uso: CU10 – Seguir Comunidad/Usuario

Tabla 2-15. Caso de uso Narrativo: CU10 - Seguir Comunidad/Usuario

Caso de uso:	Seguir Comunidad/Usuario
Código:	CU10
Resumen:	Permite al usuario compartir un sendero recién trazado o uno que se haya trazado con anterioridad.
Tipo:	Evidente
Referencias Cruzadas:	CU08 - Publicar en Muro
Actor:	Usuario
Precondiciones:	Debe existir un sendero trazado previamente.
Postcondición:	Se compartió correctamente el sendero.
Flujos Básico: El usuario sigue a un usuario o comunidad con éxito.	
Usuario	Sistema
1 El usuario selecciona la opción de "compartir sendero".	2 El sistema despliega un listado de opciones de usuarios/grupos donde el desee compartir.
3 El usuario selecciona el/los usuario/s y el/los grupo/s.	4 El Sistema comparte en el muro de el/los usuario/os y el/los grupo/s que el usuario selecciona
	5 El sistema notifica al usuario que el sendero ha sido compartido.
	6 El sistema envía notificación a los usuarios que se les ha compartido un sendero.
	7 El sistema actualiza datos de usuarios y grupos
	8 El sistema notifica a usuarios.

Fuente: Elaboración Propia

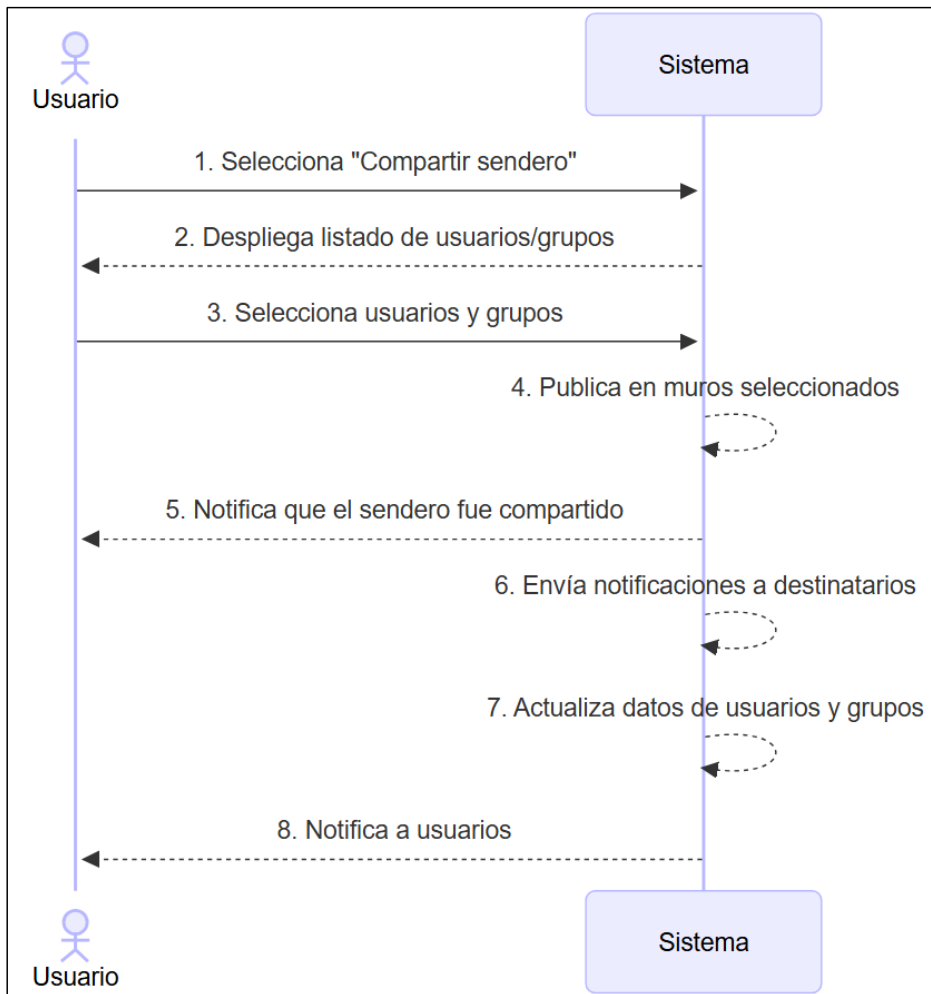


Figura 2-15. Diagrama de Secuencia: CU10 - Seguir Comunidad/Usuario

Fuente: Elaboración Propia

2.7.11. Caso de uso: CU11 – Consultar datos sendero

Tabla 2-16. Caso de uso Narrativo: CU11 - Consultar datos sendero

Caso de uso:	Consultar datos sendero
Código:	CU11
Resumen:	Permitir que un usuario consulte información detallada sobre un sendero registrado en el sistema.
Tipo:	Evidente
Referencias Cruzadas:	Ninguno
Actor:	Usuario; Open Street Map API
Precondiciones:	El usuario debe estar autenticado en el sistema. Debe existir al menos un sendero registrado.
Postcondición:	El usuario visualiza los datos del sendero seleccionado.
Flujos Básico: El sistema comparte con éxito un sendero.	
Usuario	Sistema
1 El usuario accede a la opción "Consultar Sendero" en el sistema.	2 El sistema muestra una lista de senderos disponibles.
3 El usuario selecciona un sendero.	4 El sistema muestra los detalles del sendero seleccionado (nombre, descripción, longitud, dificultad, etc.).
Flujo Alternativo: Sin senderos disponibles	
Usuario	Sistema
	2.a.1 El sistema muestra un mensaje indicando que no hay datos para consultar.
	2.a.2 El sistema da por terminado el flujo.

Fuente: Elaboración Propia

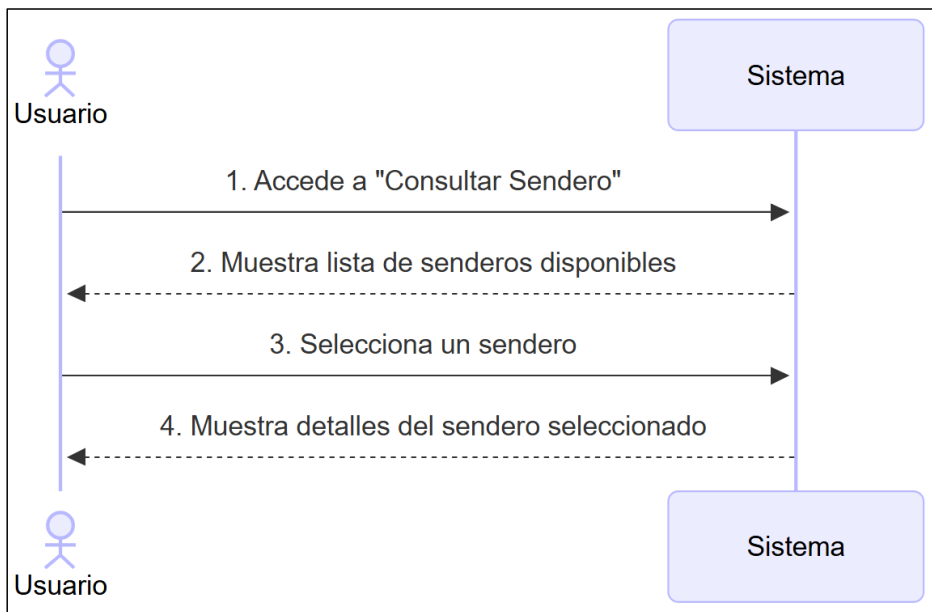


Figura 2-16. Diagrama de Secuencia: CU11 - Consultar datos Sendero

Fuente: Elaboración Propia

2.7.12. Caso de uso: CU12 – Enviar SOS

Tabla 2-17. Caso de uso Narrativo: CU12 - Enviar SOS

Caso de uso:	Enviar SOS
Código:	CU12
Resumen:	Permitir que un usuario envíe un mensaje de emergencia (SOS) a sus contactos junto con su ubicación actual en caso de una situación crítica.
Tipo:	Evidente
Referencias Cruzadas:	CU05 – Compartir Ubicación
Actor:	Usuario; Open Street Map API
Precondiciones:	El usuario debe estar registrado y autenticado en el sistema. El dispositivo debe contar con GPS habilitado. El usuario debe haber configurado previamente sus contactos de emergencia.
Postcondición:	El mensaje de SOS se envió exitosamente a los contactos seleccionados. La ubicación del usuario quedó registrada en el servidor.
Flujo Básico: El sistema envía con éxito un SOS.	
Usuario	Sistema
1 El usuario selecciona la opción "Enviar SOS" desde el menú principal.	2 Solicita confirmación para activar la función SOS.
3 El usuario confirma la acción.	4 Obtiene las coordenadas actuales del usuario.CU_05 5 Genera un mensaje de emergencia con la ubicación del usuario y lo envía a los contactos configurados. 6 Notifica al usuario que el mensaje se envió con éxito.
Flujo Alternativo: Sin contactos configurados	
Usuario	Sistema
2.a.2 El usuario configura los contactos y reintenta la acción.	2.a.1 Solicita al usuario agregar contactos de emergencia

Comentado [PG26]: Esto lo pueden hacer offline??

Comentado [AV27R26]: Si.

Flujo Alternativo: GPS no está habilitado	
Usuario	Sistema
2.a.2 El usuario puede habilitar el GPS e intentar nuevamente.	2.a.1 El sistema muestra un mensaje indicando que la ubicación no puede ser obtenida

Elaboración propia

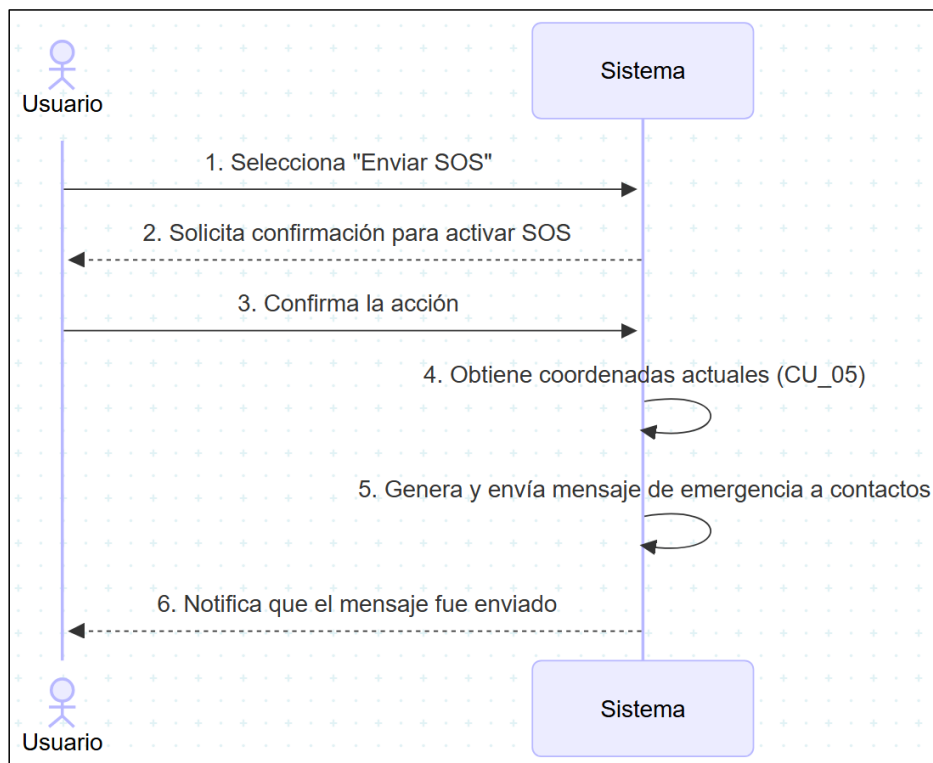


Figura 2-17. Diagrama de Secuencia: CU12 - Enviar SOS

Fuente: Elaboración Propia

2.7.13. Caso de uso: CU13 – Validar Ruta

Tabla 2-18. Caso de uso narrativo: CU13 – Validar Ruta

Caso de uso:	Validar Ruta
Código:	CU13
Resumen:	Permitir que el sistema valide si una ruta ingresada por el usuario está permitida para su uso según criterios oficiales proporcionados por una entidad validante (API) o usuarios.
Tipo:	Evidente
Referencias Cruzadas:	CU04 - Trazar Sendero
Actor:	Usuario; Open Street Map API
Precondiciones:	El usuario debe haber ingresado, trazado o editado una ruta. La API de validación debe estar operativa y accesible.
Postcondición:	El sistema validó la ruta y permite publicarlo y compartirlo.
Flujos Básico: El sistema valida con éxito la ruta según criterios oficiales.	
Usuario	Sistema
1 El usuario traza o ingresa una nueva ruta en el sistema	2 El sistema envía la ruta a la API de validación oficial. 3 La API responde con el estado legal de la ruta: permitido. 4 El sistema muestra el resultado al usuario con el motivo de validación. 5 El sistema guarda la ruta. 6 El sistema informa a cliente que la ruta es válida y se ha guardado correctamente.
Flujo Alternativo: API no disponible	
Usuario	Sistema
	2.a.1 El sistema intenta conectarse con la API de validación. 2.a.2 La API no responde o está inactiva. 2.a.3 El sistema muestra mensaje de error al usuario e indica que intente más tarde. 2.a.4 El sistema finaliza el proceso.

Fuente: Elaboración Propia

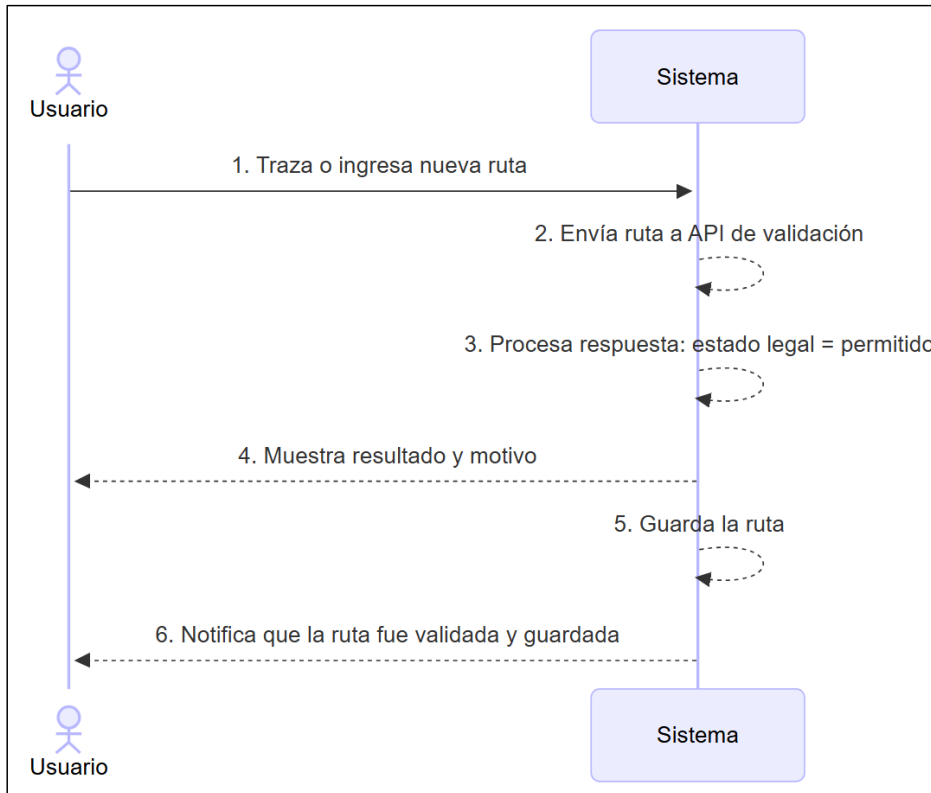


Figura 2-18. Diagrama de Secuencia: CU13 - Validar Ruta

Fuente: Elaboración Propia

Comentado [PG28]: Creo necesario al inicio de esta unidad, se explique tu criterio para crear o no los contratos, dado que no todos los CU tienen sus contratos. Ayudará a entender al lector

CAPÍTULO 3. ASPECTOS RELEVANTES DEL DISEÑO

3. ASPECTOS RELEVANTES DEL DISEÑO

A continuación, se presentarán los aspectos relevantes del diseño, como lo son arquitectura, diseño de datos, diagrama de secuencias, entre otros.

Comentado [PG29]: Acá introducías los aspectos a tratar en este capítulo, este párrafo lo trasladaría al punto 3.1

3.1. Arquitectura del Software

El diseño del sistema Trail Hunter se basa en una arquitectura modular y escalable que asegura la eficiencia, seguridad y mantenimiento a largo plazo. A continuación, se describen los aspectos más relevantes del diseño del sistema.

Como se detalla más adelante las herramientas necesarias, en este punto se definirá la arquitectura a utilizar.

La arquitectura seleccionada es una arquitectura por capas, que organiza el sistema en componentes funcionales bien definidos, permitiendo un desarrollo más estructurado y una fácil integración de futuras mejoras. Las capas principales son:

1. Capa de Presentación:
 - Maneja la interacción con el usuario, incluyendo las interfaces gráficas (UI) desarrolladas con tecnologías modernas.
 - Herramientas: React Native para aplicaciones móviles multiplataforma.
2. Capa de Lógica de Negocio:
 - Contiene las reglas y procesos que gestionan las funcionalidades clave, como el trazado de senderos y la integración de datos climáticos.
 - Lenguaje: Node.js.
3. Capa de Acceso a Datos:
 - Gestiona el almacenamiento y recuperación de información desde la base de datos. Incluye el manejo de datos geoespaciales mediante PostgreSQL y PostGIS.
 - Base de datos: PostgreSQL con extensiones geoespaciales.

Ahora se presenta el diagrama de nuestra arquitectura con las herramientas que se utilizarán y que se explican en el punto 3.1.2 :

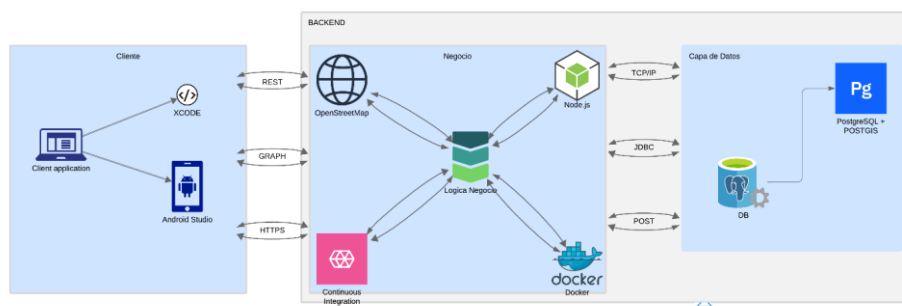


Figura 3-1. Arquitectura de Capas Trail Hunter

Fuente: Elaboración Propia

3.1.1. Sistema Operativo

Para asegurar que la aplicación funcione como se espera se detallarán algunas recomendaciones para los usuarios.

En cuanto a compatibilidad recomendada: la aplicación Trail Hunter para que sea compatible con las versiones de Android y la de iOS, deberá tener comprendido al menos desde la versión Android 10 (lanzada en septiembre de 2019) y posteriores, y con iOS 15 (lanzada en septiembre de 2021) y posteriores. Esto garantiza que la mayoría de los usuarios con dispositivos relativamente recientes puedan utilizar la aplicación sin inconvenientes.

Existe una compatibilidad limitada: las versiones anteriores pueden presentar limitaciones en cuanto a rendimiento y soporte de funcionalidades modernas.

Al desarrollar Trail Hunter, es esencial realizar pruebas en dispositivos que ejecuten las versiones recomendadas de los sistemas operativos para asegurar una experiencia de usuario óptima y minimizar posibles incompatibilidades.

3.1.2. Herramientas necesarias para el desarrollo del proyecto

A continuación, se presentan las herramientas utilizadas para el desarrollo y gestión del proyecto Trail Hunter:

- PostgreSQL con PostGIS: Base de datos relacional de código abierto con extensión PostGIS, que permite el almacenamiento y manejo de datos geoespaciales, fundamentales para el registro de rutas y ubicaciones.
- StarUML: Herramienta de modelado que permite crear diagramas UML (como casos de uso, diagramas de clases y de secuencia) para planificar y estructurar el sistema.
- Balsamiq: Software para crear prototipos de baja fidelidad, utilizado para diseñar y validar las interfaces de usuario.
- CI/CD Tools (GitHub Actions, GitLab CI/CD): Herramientas de integración y entrega continua que automatizan pruebas, despliegue y actualización del software.
- Docker: Plataforma para contenerización, utilizada para garantizar consistencia entre entornos de desarrollo, pruebas y producción.
- OpenStreetMap: Plataforma colaborativa de mapas que provee datos geográficos detallados para el desarrollo de funcionalidades relacionadas con rutas y ubicaciones.
- Android Studio: Entorno de desarrollo integrado (IDE) oficial para el desarrollo de aplicaciones Android.
- Xcode: IDE oficial para el desarrollo de aplicaciones en el ecosistema iOS.

3.1.3. Stack de tecnologías de la aplicación

El stack tecnológico específico para la aplicación Trail Hunter se organiza por capas, reflejando las tecnologías que forman parte del desarrollo del producto final:

Capa Cliente (Interfaces)

Enfocada en el diseño y la interacción con el usuario final.

- React Native: Framework de desarrollo móvil multiplataforma, utilizado para crear aplicaciones nativas tanto en Android como en iOS con un único código base.
- Material UI: Biblioteca de componentes que facilita la creación de interfaces consistentes y atractivas.
- CSS-in-JS: Técnica para aplicar estilos directamente en los componentes, mejorando la modularidad del código.

Capa Negocio (Backend)

Se encarga de gestionar la lógica de negocio, servicios y procesamiento de datos.

- Node.js: Entorno de ejecución basado en JavaScript, utilizado para construir un backend escalable y eficiente.
- Express.js: Framework para Node.js que simplifica la creación de APIs RESTful y la gestión de rutas.
- Socket.IO: Biblioteca para habilitar comunicación en tiempo real entre cliente y servidor, útil para notificaciones y actualizaciones instantáneas.
- OpenStreetMap API: Proporciona datos cartográficos detallados y gratuitos para funcionalidades como cálculo de rutas.
- Google Maps API: Se utiliza para funcionalidades avanzadas como geocodificación, cálculo de distancias y rutas optimizadas.

Capa de Datos

Responsable del almacenamiento y manejo de la información del sistema.

- PostgreSQL con PostGIS: Base de datos principal que almacena datos transaccionales y geoespaciales. PostGIS permite realizar cálculos avanzados y gestionar rutas con precisión.
- GeoJSON: Formato estándar para el intercambio de datos geográficos entre el backend y el cliente.
- Redis (opcional): Base de datos en memoria utilizada para almacenamiento en caché y mejorar el rendimiento en consultas frecuentes.

3.1.4. Lenguajes de programación y tecnologías

Para el sistema de base de datos del proyecto se utilizará PostgreSQL debido a la facilidad y robustez de su infraestructura al momento de realizar consultas con datos geográficos.

Algunas de sus características son:

- **Tipos de Datos Espaciales:** PostGIS añade nuevos tipos de datos a PostgreSQL, como geometry y geography, que pueden almacenar puntos, líneas, polígonos, y otros tipos de geometrías.

- **Funciones Espaciales:** PostGIS proporciona una amplia gama de funciones para manipular y analizar datos espaciales, incluyendo funciones para calcular distancias, áreas, intersecciones, uniones y transformaciones de coordenadas.
- **Índices Espaciales:** PostGIS soporta el uso de índices espaciales R-tree basados en GIST, que mejoran significativamente el rendimiento de las consultas espaciales.
- **Compatibilidad con Estándares:** PostGIS es compatible con los estándares OGC (Open Geospatial Consortium), lo que garantiza interoperabilidad con otros sistemas y herramientas GIS.

Como lenguajes de programación se utilizarán los lenguajes de cada ambiente de desarrollo, KOTLIN para Android y SWIFT para iOS, a menos que al momento de desarrollar este proyecto no sean los más actualizados.

Para desarrollar se utilizaría un IDE que soporten ambos lenguajes por separado, como lo son ANDROID STUDIO y XCODE para Android y iOS respectivamente.

Si bien se pueden optar por IDE que soporte ambos lenguajes no es lo recomendado cuando se tienen desarrollos multiplataformas, lo que quita algunas especificaciones propias del lenguaje a la hora de debuggear además de , muchas veces, obligar al desarrollador a añadir algunos plug-ins para poder tener el 100% de una funcionalidad en el desarrollo.

Comentado [PG30]: Buscar los errores en el código!!! No existe esta palabra

3.2. Diseño de datos

El diseño de datos para Trail Hunter se centra en estructurar de manera eficiente la información relacionada con los usuarios, senderos, comunidades y mapas. A continuación, se describen los elementos principales del diseño de datos:

3.2.1. Modelo de Clases.

El modelo de clases representa las entidades principales del sistema y sus atributos y métodos. Este modelo se utiliza como base para la implementación del backend y la base de datos.

En esta sección se dan a conocer el modelo de clases, con sus atributos y métodos que están asociados a la implementación de la aplicación.

Como nota adicional en este modelo de clases no se incluyen las tablas adicionales (senderoMapa y usuarioComunidad), como si se hace en el Modelo Conceptual y en el Modelo Relacional, ya que no aportan campos adicionales más que los que se usan para romper la redundancia.

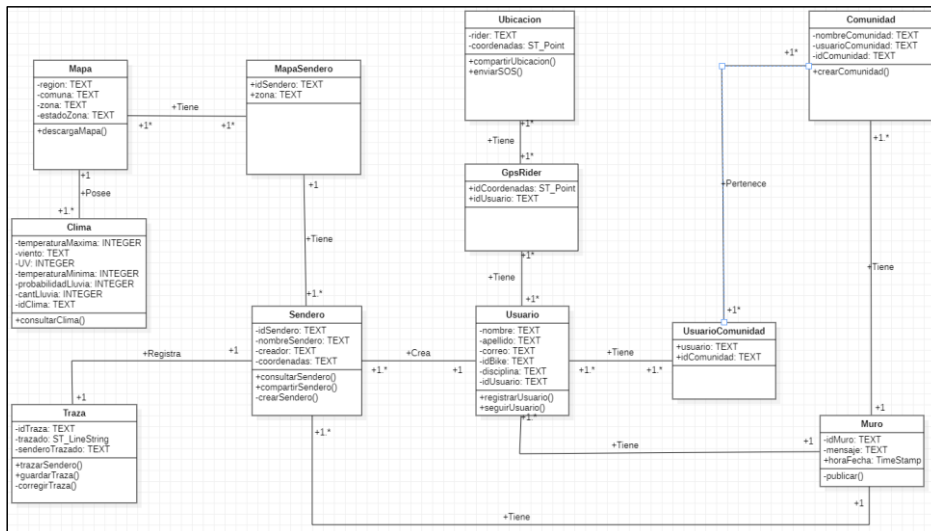


Figura 3-2. Modelo de Clases

Fuente: Elaboración Propia

3.2.2. Modelo Relacional.

El modelo relacional define las tablas y relaciones necesarias para estructurar y gestionar los datos del sistema de manera eficiente. En el caso de Trail Hunter, este modelo garantiza que la información de usuarios, senderos, comunidades y mapas se almacene y recupere de forma óptima, respetando los principios de normalización y evitando redundancias.

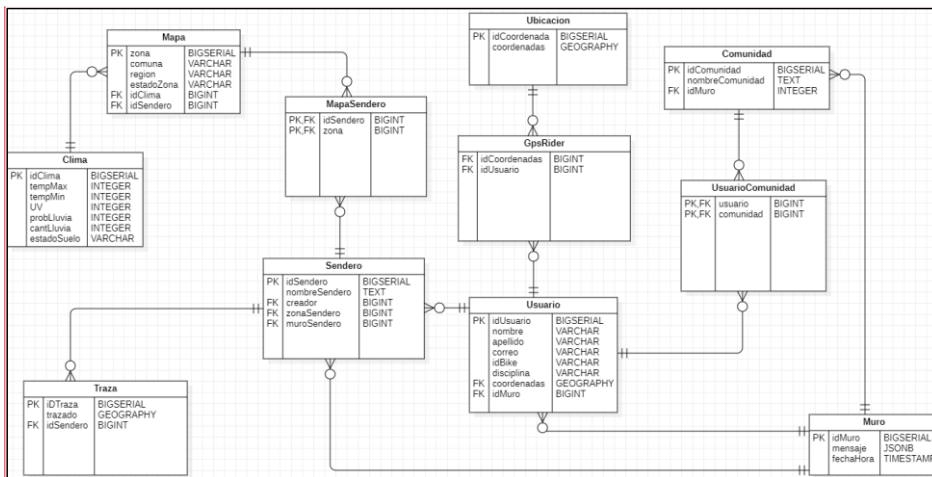


Figura 3-3. Modelo de Datos

Fuente: Elaboración Propia

Comentado [PG31]: Esta misma estructura debiera verse más menos reflejada en el modelo de clases. (relaciones mucho a mucho)

Vuelvo a consultar acá, dónde se almacenaría el llamado de SOS (la ubicación del usuario)

3.2.3. Tipos de Datos.

En el diseño de la base de datos relacional, se emplearán los siguientes tipos de datos:

Tabla 3-1. Base de Datos: Tipos de Datos

Tipo de Dato	Descripción
INTEGER	Número entero. Útil para cantidades pequeñas, como un índice UV o número de reacciones.
VARCHAR	Cadena de texto de longitud limitada a n caracteres. Útil si deseas validar que un campo no exceda cierto número de caracteres.
BIGSERIAL	Entero grande auto incremental. Ideal para claves primarias con muchos registros. Internamente asigna automáticamente valores únicos.
TEXT	Cadena de texto de longitud ilimitada. Se usa para almacenar mensajes, descripciones u otros textos extensos.
TIMESTAMP	Fecha y hora. Se utiliza para registrar cuándo se hizo una publicación o una modificación. Puede incluir zona horaria con TIMESTAMPTZ
JSONB	Objeto JSON binarizado. Permite almacenar datos estructurados (clave:valor) de forma eficiente. Ideal para guardar datos con estructura flexible como publicaciones con texto, imágenes y metadata.
GEOGRAPHY	Similar a GEOMETRY, pero usa un modelo esférico (basado en el globo terrestre) en lugar de un plano. Ideal para distancias y áreas en coordenadas geográficas (lat/lon).
BIGINT	Tipo de dato numérico entero de 8 bytes (64 bits). Permite almacenar valores entre -9,223,372,036,854,775,808 y 9,223,372,036,854,775,807. Es ideal para claves primarias o conteos que pueden crecer a escalas muy grandes. Suele usarse como tipo base para claves generadas automáticamente con BIGSERIAL

Fuente: Elaboración Propia

3.2.4. Diccionario de datos.

A continuación, se presenta el diccionario de datos del sistema Trail Hunter, que describe cada campo en las tablas del modelo relacional, su tipo de datos, restricciones y una breve descripción de su propósito.

Comentado [PG32]: Al ser un diccionario de Datos, las tablas se presentan en orden alfabético

Tabla 3-2. Diccionario de datos: Tabla Clima

Nombre:	Clima	
Descripción:	Contiene los registros de las condiciones climáticas	
Clave Principal:	idClima	
Claves Foráneas:	No aplica.	
Campo	Tipo	Descripción
idClima	Bigserial	Identificador único de clima .
tempMaxima	Integer	Temperatura máxima registrada (°C).
tempMinima	Integer	Temperatura mínima registrada (°C).
UV	Integer	Índice UV registrado (0–11+).
probLluvia	Integer	Probabilidad de lluvia (en %).
cantLluvia	Integer	Cantidad estimada de lluvia (mm).
estadoSuelo	Varchar(10)	Estado del suelo o terreno (ej: seco, mojado, barroso).

Fuente: Elaboración Propia

Tabla 3-3. Diccionario de datos: Tabla Comunidad

Nombre:	Comunidad	
Descripción:	Registra las comunidades creadas por usuarios	
Clave Principal:	idComunidad	
Claves Foráneas:	idMuro→Muro	
Campo	Tipo	Descripción
idComunidad	Bigserial	Identificador único de Comunidad
nombreComunidad	Text	Nombre de la comunidad
idMuro	Bigint	Clave foránea de Muro

Fuente: Elaboración Propia

Tabla 3-4. Diccionario de datos: Tabla GpsRider

Nombre:	GpsRider	
Descripción:	Tabla intermedia entre Usuario y Ubicación para controlar m:m	
Clave Principal:	coordenadas + idUsuario	
Claves Foráneas:	idCoordenada→Ubicación idUsuario→Usuario	
Campo	Tipo	Descripción
idCoordenada	Bigint	Clave foránea de Ubicación
idUsuario	Bigint	Clave foránea de Usuario

Fuente: Elaboración Propia

Tabla 3-5. Diccionario de datos: Tabla Mapa

Nombre:	Mapa	
Descripción:	Contiene los registros de los mapas disponibles	
Clave Principal:	zona	
Claves Foráneas:	idClima → Clima idSendero → MapaSendero	
Campo	Tipo	Descripción
zona	Bigserial	Zona específica dentro de la comuna.
region	Varchar(20)	Región del mapa.
comuna	Varchar(20)	Comuna del mapa.
estadoZona	Varchar(15)	Estado actual de la zona (Abierta, cerrada, Precaución).
idClima	Bigint	Clave foránea de Clima
idSendero	Bigint	Clave foránea de Sendero

Fuente: Elaboración Propia

Tabla 3-6. Diccionario de datos: Tabla MapaSendero

Nombre:	MapaSendero	
Descripción:	Tabla intermedia entre Zona y Sendero para controlar m:m	
Clave Principal:	idSendero + zona	
Claves Foráneas:	idSendero→Sendero zona→Mapa	
Campo	Tipo	Descripción
idSendero	Bigint	Clave foránea de sendero.
zona	Bigint	Clave foránea de Mapa.

Fuente: Elaboración Propia

Tabla 3-7. Diccionario de datos: Tabla Muro

Nombre:	Muro	
Descripción:	Guarda publicaciones o mensajes a usuarios , grupos y/o senderos	
Clave Principal:	idMuro	
Claves Foráneas:	No Aplica	
Campo	Tipo	Descripción
idMuro	Bigserial	Identificador único de muro
mensaje	JSONB	Guarda mensaje del muro
fechaHora	timeStamp	Fecha y hora de publicación en muro

Fuente: Elaboración Propia

Tabla 3-8. Diccionario de datos: Tabla Sendero

Nombre:	Sendero	
Descripción:	Contiene los senderos registrador por usuarios	
Clave Principal:	idSendero	
Claves Foráneas:	creador→ Usuario zonaSendero→ MapaSendero muroSendero→ Muro trazaSendero→ Traza	
Campo	Tipo	Descripción
idSendero	Bigserial	Identificador único de sendero .
nombreSendero	Text	Nombre válido del sendero dado por el usuario.
creador	Bigint	Clave foránea de Usuario.
zonaSendero	Bigint	Clave foránea de MapaSendero.
muroSendero	Bigint	Clave foránea de Muro.

Fuente: Elaboración Propia

Tabla 3-9. Diccionario de datos: Tabla Traza

Nombre:	Traza	
Descripción:	Guarda las coordenadas de los senderos trazados por el usuario	
Clave Principal:	idTraza	
Claves Foráneas:	idSendero→Sendero	
Campo	Tipo	Descripción
idTraza	Bigserial	Identificador único de traza
trazado	Geography	Coordenadas de traza de sendero
idSendero	Bigint	Clave foránea de Sendero

Fuente: Elaboración Propia

Tabla 3-10. Diccionario de datos: Tabla Ubicacion

Nombre:	Ubicacion	
Descripción:	Registra coordenadas de ubicación del usuario	
Clave Principal:	Idcoordenadas	
Claves Foráneas:	NO Aplica	
Campo	Tipo	Descripción
IdCoordenada	Bigserial	Identificador único de coordenadas
coordenadas	Geography	Coordenadas de ubicación de usuarios

Fuente: Elaboración Propia

Comentado [PG33]: No es ID_coordenadas??

Tabla 3-11. Diccionario de datos: Tabla Usuario

Nombre:	Usuario	
Descripción:	Contiene los usuarios registrador en la aplicación	
Clave Principal:	idUsuario	
Claves Foráneas:	coordenadas→GpsRider idMuro→Muro	
Campo	Tipo	Descripción
idUsuario	Bigint	Identificador único de usuario
nombre	Varchar(20)	Nombre del usuario
apellido	Varchar(20)	Apellido del usuario
correo	Varchar(50)	Correo electrónico de usuario
idBike	Varchar(20)	Código Alfanumérico de bicicleta
disciplina	Varchar(10)	Disciplina que practica (Enduro,XC, DH)
coordenadas	Geography	Clave foránea de GpsRider
idMuro	Bigint	Clave foránea de Muro

Fuente: Elaboración Propia

Tabla 3-12. Diccionario de datos: Tabla UsuarioComunidad

Nombre:	UsuarioComunidad	
Descripción:	Tabla intermedia entre Usuario y Comunidad para controlar m:m	
Clave Principal:	usuario + comunidad	
Claves Foráneas:	usuario→Usuario comunidad→Comunidad	
Campo	Tipo	Descripción
usuario	Bigint	Clave foránea de Usuario
comunidad	Bigint	Clave foránea de Comunidad

Fuente: Elaboración Propia

Comentado [PG34]: En la tabla usuario lo declaras como tipo BIGSERIAL?? Pq cambia de tipo de dato??

3.3. Diagrama de Secuencia Extendido

Los diagramas de secuencia extendido ilustran las interacciones detalladas necesarias para implementar los casos de uso principales del sistema Trail Hunter. Este diagrama asegura una comprensión clara del flujo de datos y las responsabilidades de cada actor y componente técnico involucrado.

3.3.1. CU01 - Mantener Usuario

Diagrama de secuencia extendido del caso de uso Mantener Usuario, que como se ve interactúa el usuario, el sistema (GUI) y el servidor:

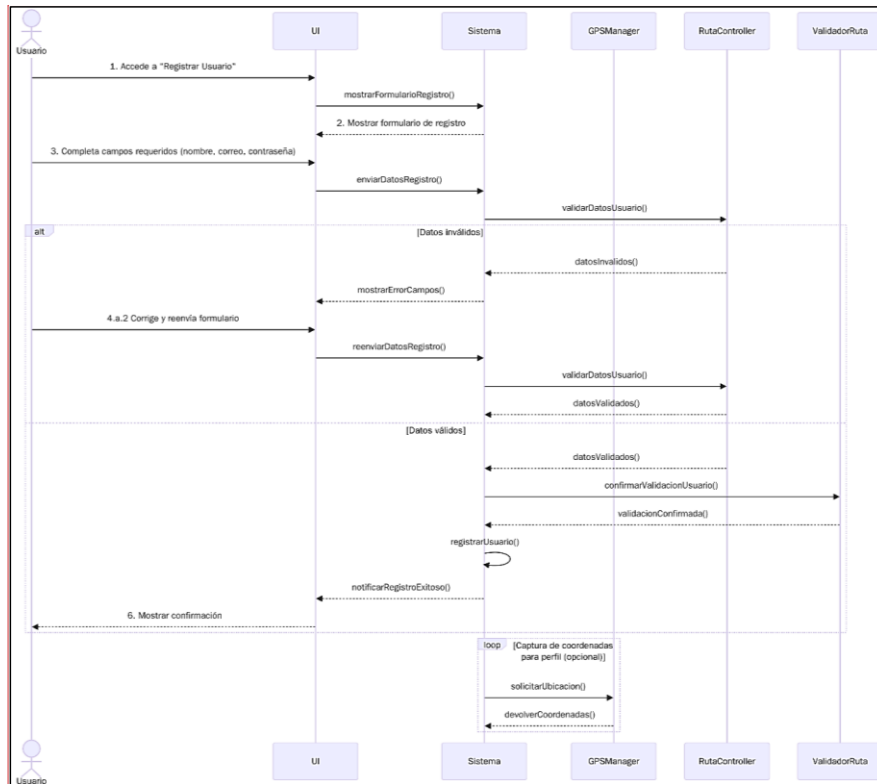


Figura 3-4. Diagrama de secuencia extendido: CU01 – Mantener Usuario

Fuente: Elaboración propia

Comentado [PG35]: Cuál es la diferencia entre UI y sistema??

No se supone que acá al sistema (entidad que existía en los DSS) se le hace lupa y se ve lo interno?? Es decir la UI y el resto de los componentes del sistema (Nego+backend)

Creo que sistema está demás en todos los diagramas

3.3.2. CU02 - Descargar Mapa

Diagrama de secuencia extendido del caso de uso descargar mapa, que interactúa el usuario, el sistema (GUI) y el servidor para poder generar un mapa en el dispositivo:

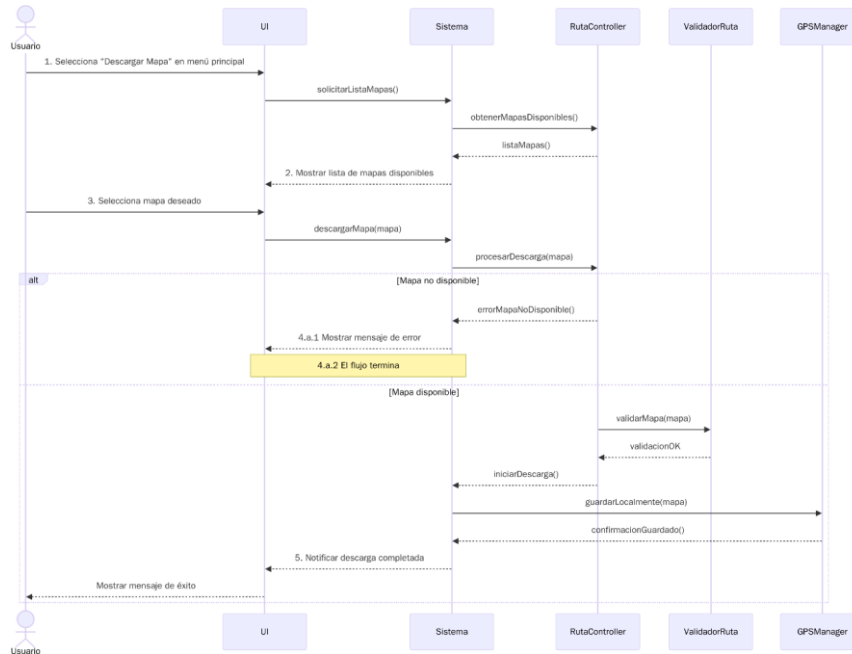


Figura 3-5. Diagrama de secuencia extendido: CU02 – Descargar Mapa

Fuente: Elaboración propia

3.3.3. CU03 - Crear sendero

Diagrama de secuencia extendido del caso de uso crear sendero. En él se puede ver cómo hay una mayor interacción entre sistema y usuario:

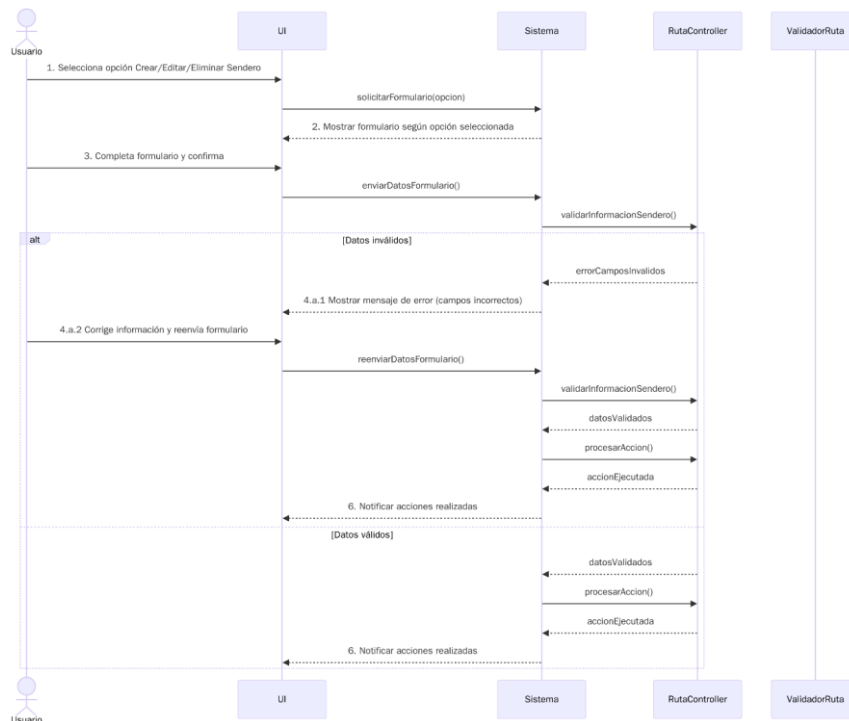


Figura 3-6. Diagrama de secuencia extendido: CU03 – Crear sendero

Fuente: Elaboración propia

3.3.4. CU04 - Trazar sendero

Se presenta el diagrama de secuencia extendido del caso de uso trazar sendero, aquí actúan las funcionalidades de captura de datos de geolocalización del usuario:

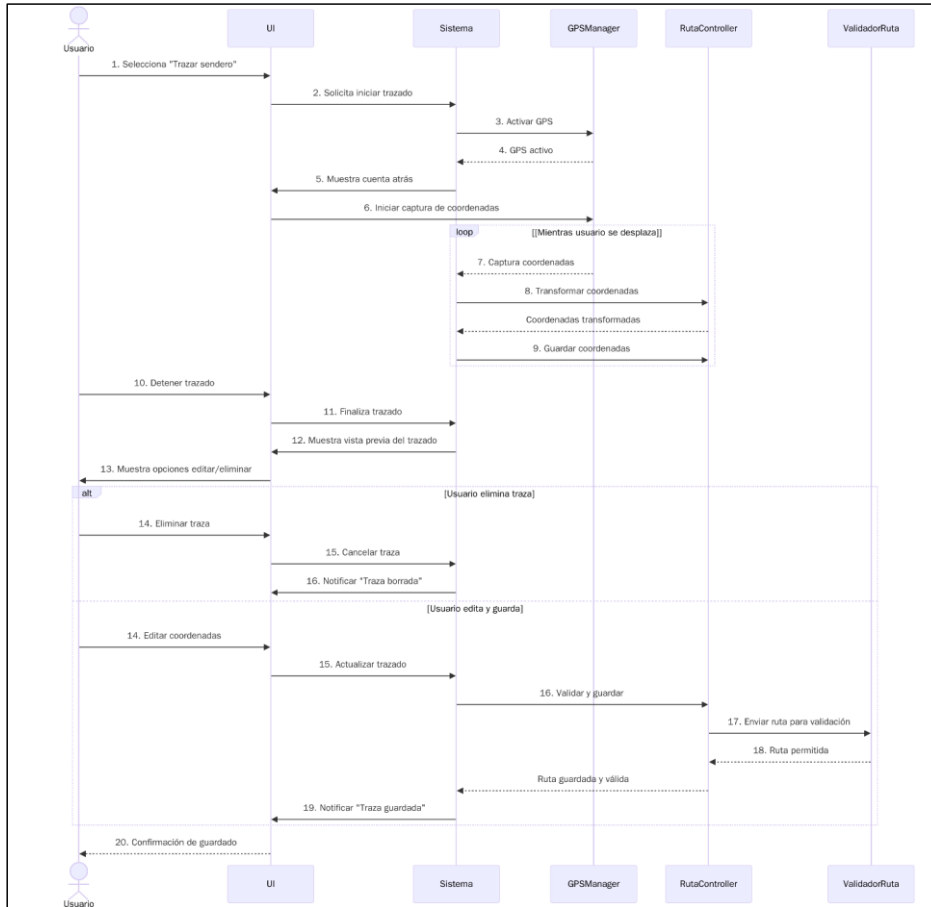


Figura 3-7. Diagrama de secuencia extendido: CU04 – Trazar sendero

Fuente: Elaboración propia

3.3.5. CU05 - Compartir ubicación

El diagrama de secuencia extendido del caso de uso compartir ubicación es el primero del listado que no tiene interacción con el servidor ya que no actúan datos guardados con anterioridad:

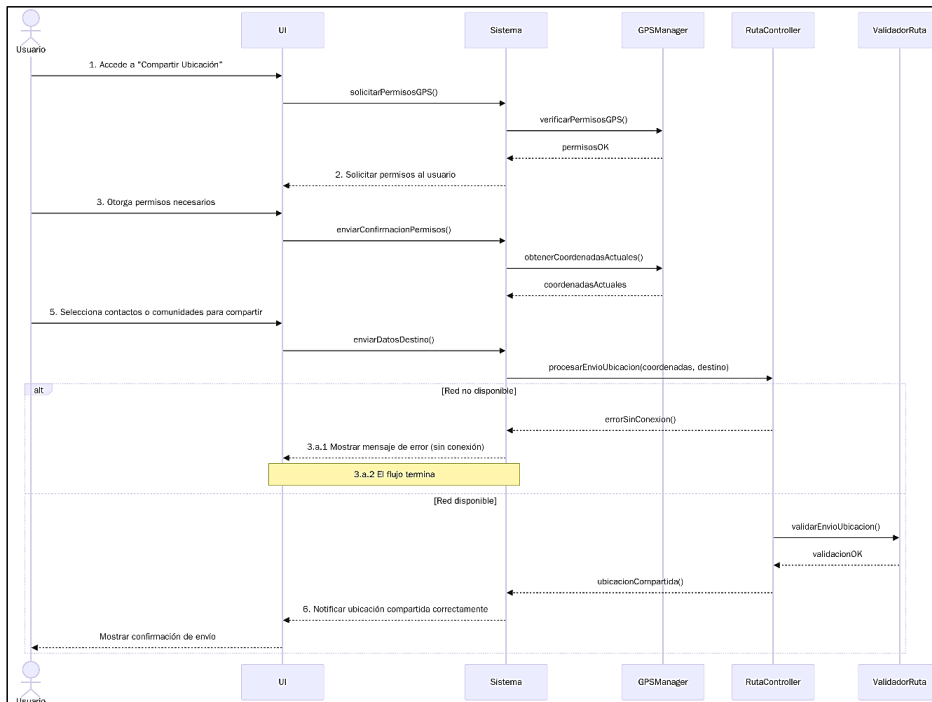


Figura 3-8. Diagrama de secuencia extendido: CU05 – Compartir ubicación

Fuente: Elaboración propia

3.3.6. CU06 - Compartir imagen

En el diagrama de secuencia extendido del caso de uso compartir imagen, se puede ver un aumento en la interacción de actores ya que actúan más funcionalidades sociales, lo que incrementa la comunicación entre usuario y sistema:

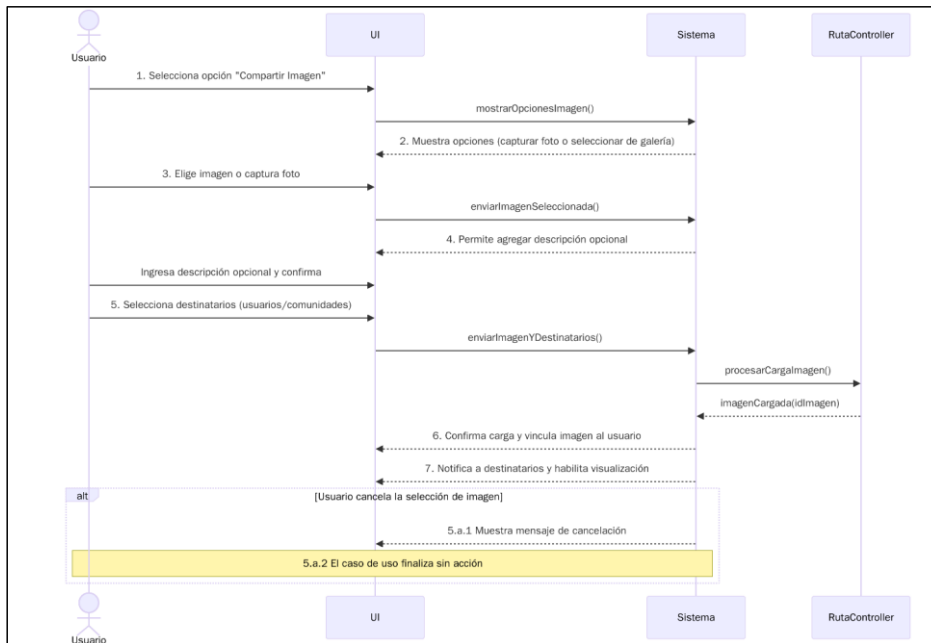


Figura 3-9. Diagrama de secuencia extendido: CU06 – Compartir imagen

Fuente: Elaboración propia

3.3.7. CU07 - Crear comunidad

Se presenta el diagrama de secuencia extendido del caso de uso crear comunidad:

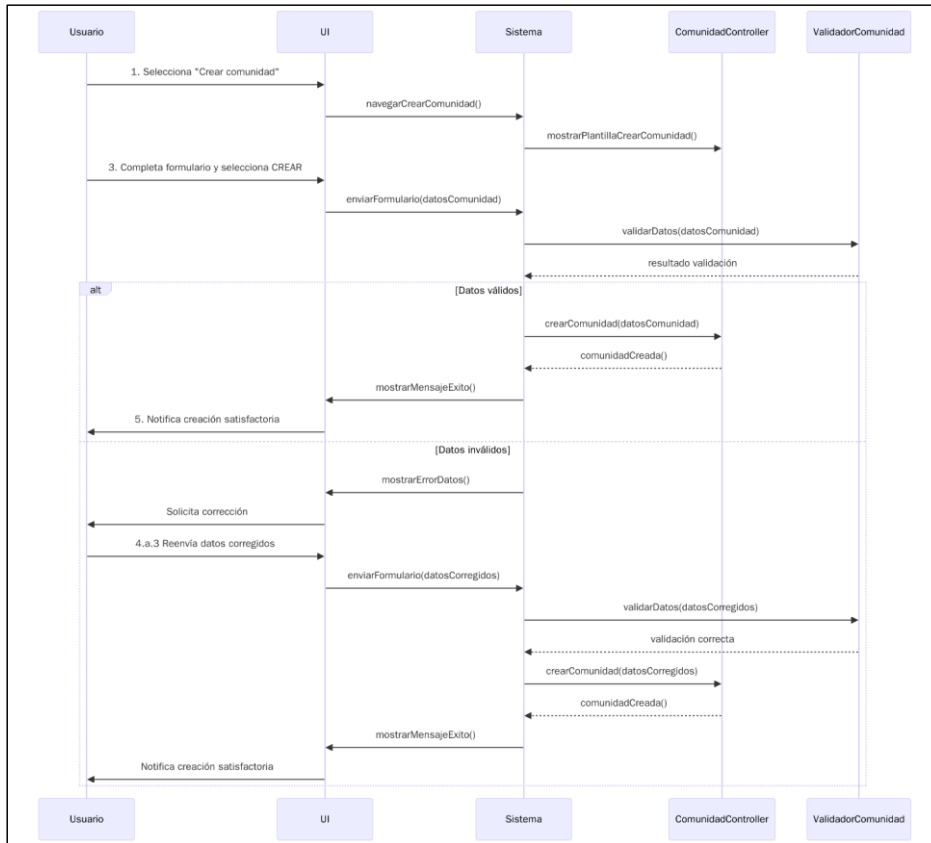


Figura 3-10. Diagrama de secuencia extendido: CU07 – Crear comunidad

Fuente: Elaboración propia

3.3.8. CU08 - Publicar en muro

A continuación, se presenta el diagrama de secuencia extendido del caso de uso publicar muro; en este se puede ver con claridad que es la primera vez que la interacción mayormente es entre el sistema y la base de datos:

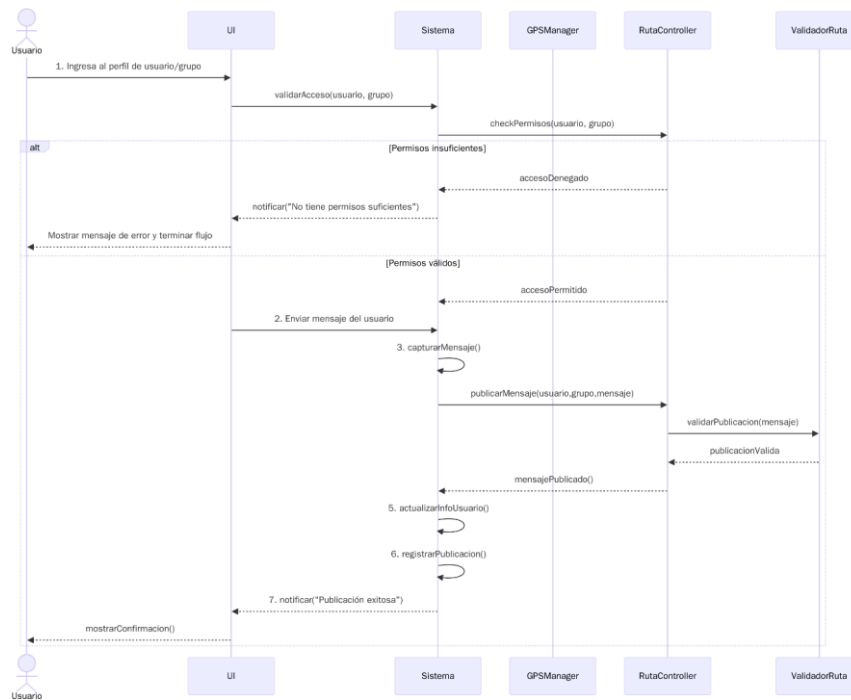


Figura 3-11. Diagrama de secuencia extendido: CU08 – Publicar Muro

Fuente: Elaboración propia

3.3.9. CU09 - Compartir sendero

El diagrama de secuencia extendido para el caso de uso compartir Sendero:

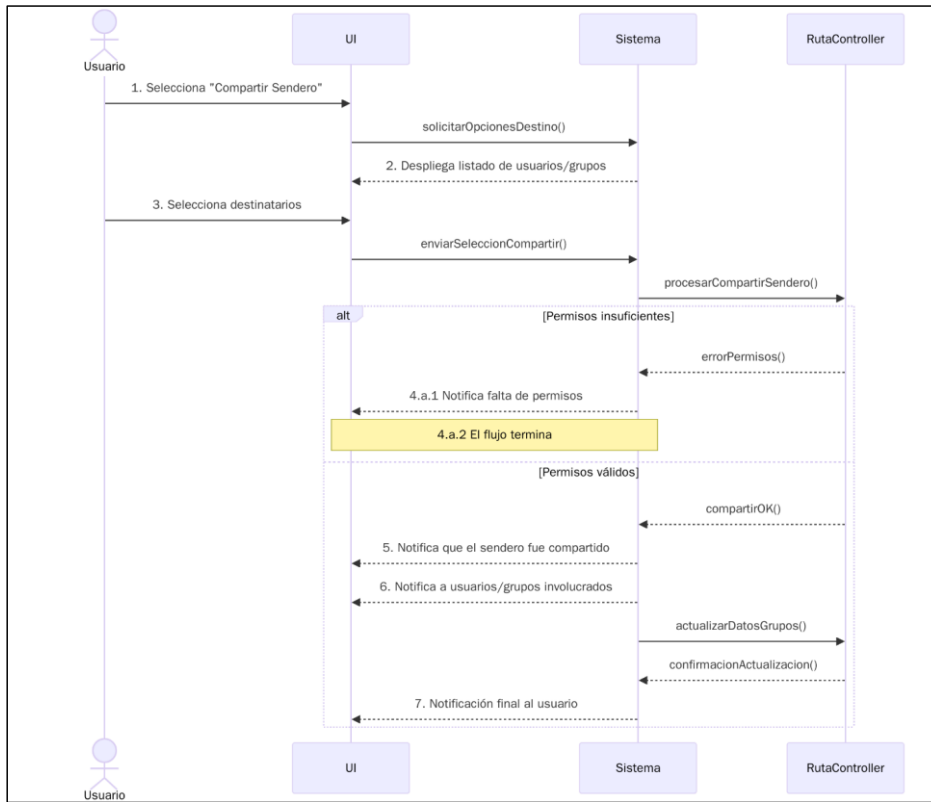


Figura 3-12. Diagrama de secuencia extendido: CU09 - Compartir sendero

Fuente: Elaboración propia

3.3.10. CU10 - Seguir usuario/comunidad

El siguiente diagrama de secuencia extendido para el caso de uso número 10: seguir usuario y comunidad:

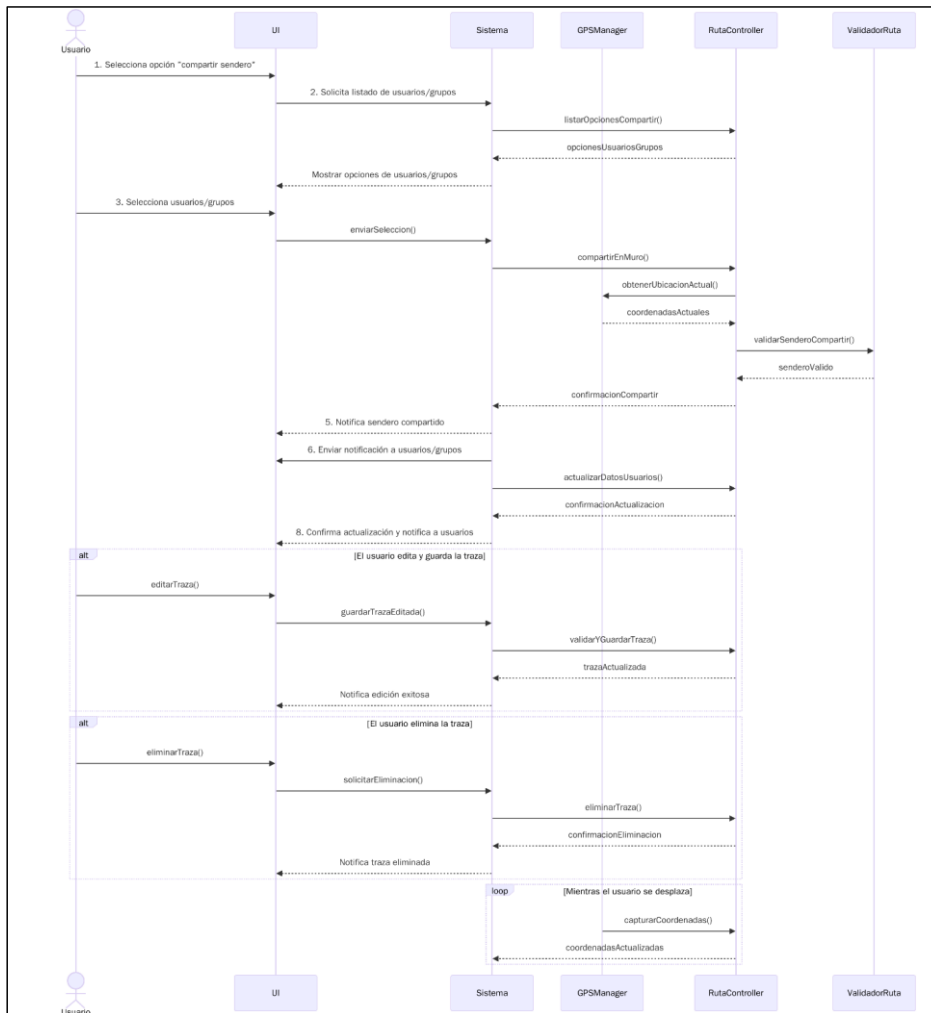


Figura 3-13. Diagrama de secuencia extendido: CU10 – Seguir usuario/comunidad

Fuente: Elaboración propia

3.3.11. CU11 - Consultar datos sendero

Consultar datos sendero es un caso de uso donde la interacción es mínima entre usuario, sistema y Servidor de datos:

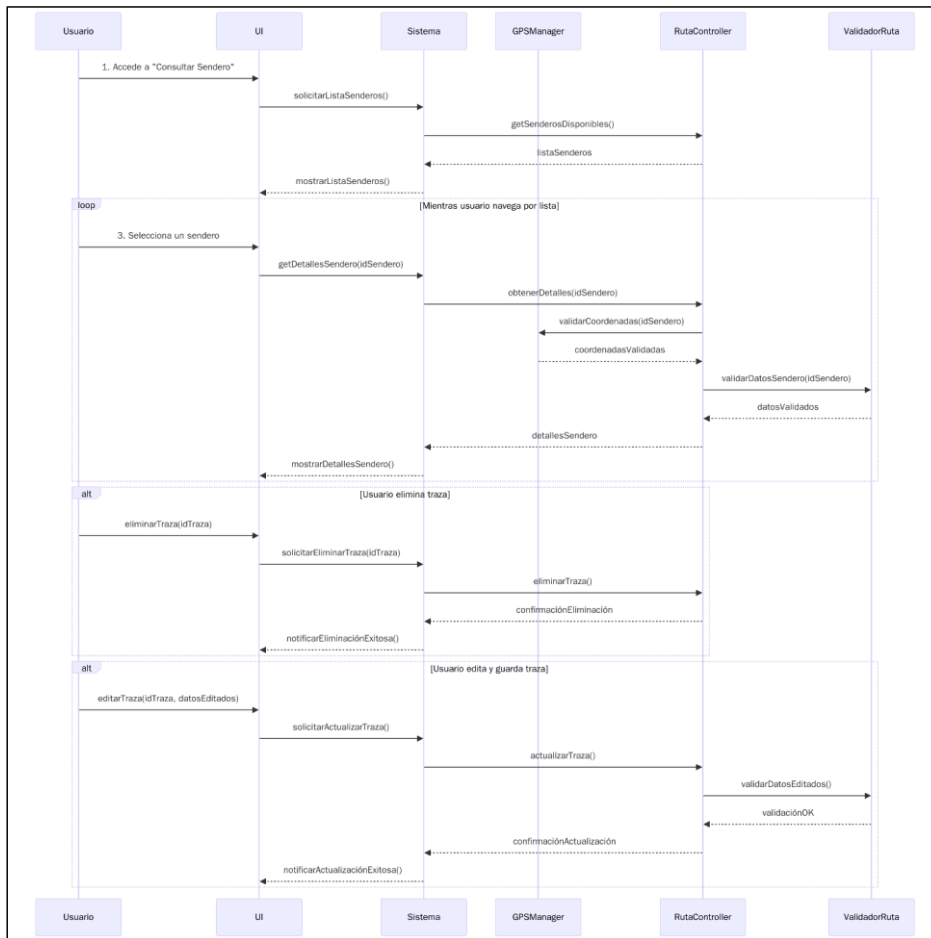


Figura 3-14. Diagrama de secuencia extendido: CU11 – Consultar datos sendero

Fuente: Elaboración propia

3.3.12. CU12 - Enviar SOS

Enviar SOS es el caso de uso que se encarga de avisar a autoridades y a contactos del usuario predeterminados un mensaje de ayuda con sus coordenadas.

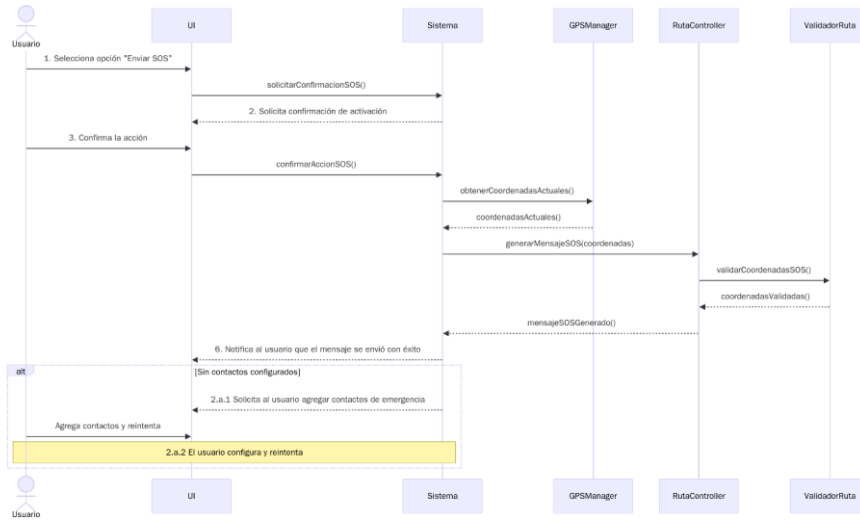


Figura 3-15. Diagrama de secuencia extendido: CU12 - Enviar SOS

Fuente: Elaboración propia

3.3.13. CU13 - Validar Ruta

Validar Ruta es uno de los casos de uso del core del proyecto, ya que es el que se encarga de que un sendero no incumpla normas legales ni de la propia comunidad:

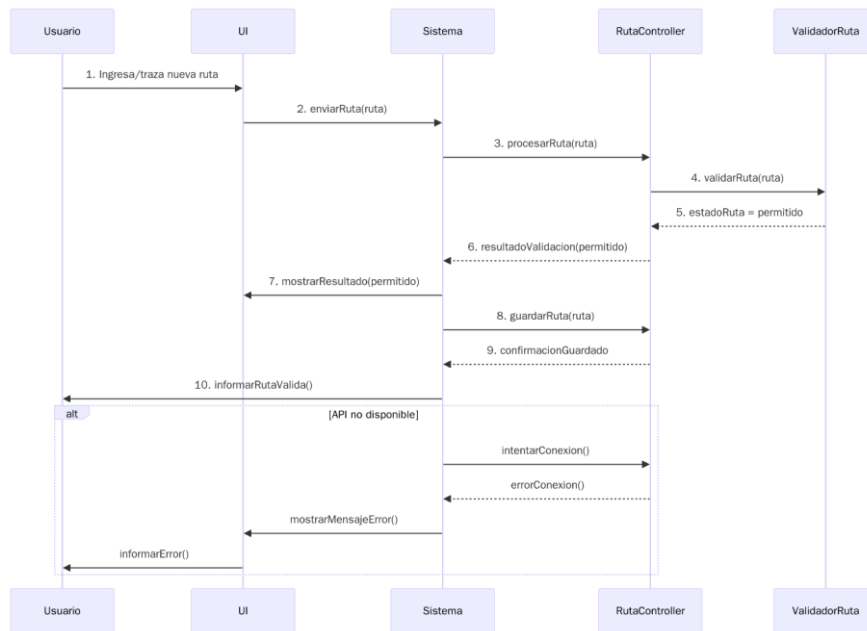


Figura 3-16. Diagrama de secuencia extendido: CU13 - Validar Ruta

Fuente: Elaboración propia

3.4. Diseño de Interfaz

A continuación, se presenta el diseño de las interfaces más relevantes e importantes para el proyecto, como lo **son el login**, la búsqueda de sendero y el compartir ubicación a terceros.

Comentado [PG36]: Pq sería relevante?? Desde el punto de vista de negocios, no lo es, al ser una funcionalidad que ya todos la han desarrollado

En estas interfaces se trata de cumplir con algunos de los puntos de las heurísticas de Nielsen tales como el minimizar la carga de memoria y la flexibilidad y eficiencia de uso de los usuarios creando un diseño minimalista y sobrio que cumpla con lo que promete, pero no atiborré la pantalla del usuario con información innecesaria.

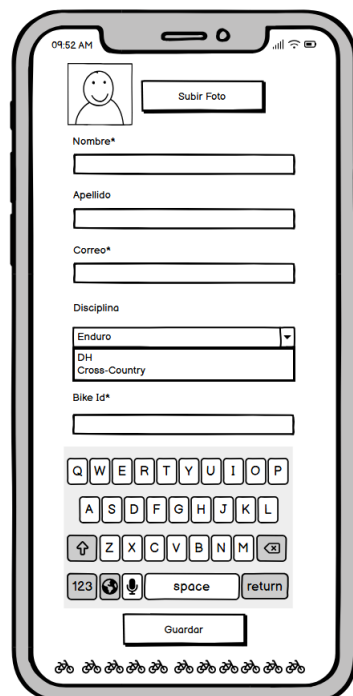
A continuación, se entrega una breve descripción de cada una:

3.4.1. UI 01: Registro de Usuario

El Objetivo es poder permitir a los nuevos usuarios registrarse en el sistema ingresando su información personal y configurando credenciales de acceso.

- Casos de Uso Relacionados: CU01 - Mantener Usuario.
- Requisitos Funcionales Relacionados: RF01 Registrar usuarios en el sistema.
- Aspectos de Usabilidad:
 - Diseño claro y minimalista para facilitar el ingreso de datos.
 - Mensajes de validación en tiempo real para evitar errores.
 - Indicadores de progreso y confirmación de registro.

Comentado [PG37]: Estos elementos, debieran estar marcados de alguna forma sobre la interfaz, ya sea con un número o símbolo que ayude al lector a entender el mensaje y lo que se pretende lograr, sino, obligas al lector a querer interpretar o buscar estos elementos dentro de la imagen



A mobile app login screen design. At the top, the status bar shows '09:52 AM', signal strength, and battery level. Below the status bar is a header area with a placeholder for a profile picture and a button labeled 'Subir Foto'. The main form contains several input fields: 'Nombre*' (Name), 'Apellido' (Last Name), 'Correo*' (Email), 'Disciplina' (Discipline) with a dropdown menu showing 'Enduro', 'DH', and 'Cross-Country', and 'Bike Id*'. Below the form is a virtual keyboard with keys for letters, numbers, and special characters. At the bottom of the screen is a button labeled 'Guardar' (Save).

Figura 3-17. Diseño de Interfaz: Login

Fuente: Elaboración propia

3.4.2. UI 02: Buscar Senderos

El Objetivo es facilitar la búsqueda de senderos registrados en el sistema, ya sea por nombre, dificultad o ubicación geográfica.

- Casos de Uso Relacionados: CU_12 Consultar Datos Sendero.
- Requisitos Funcionales Relacionados: RF_11 Permitir la consulta de senderos registrados.
- Aspectos de Usabilidad:
 - Barra de búsqueda intuitiva con filtros avanzados.
 - Mapa interactivo para mostrar los resultados visualmente.
 - Indicadores de estado del sendero (disponibilidad, mantenimiento).

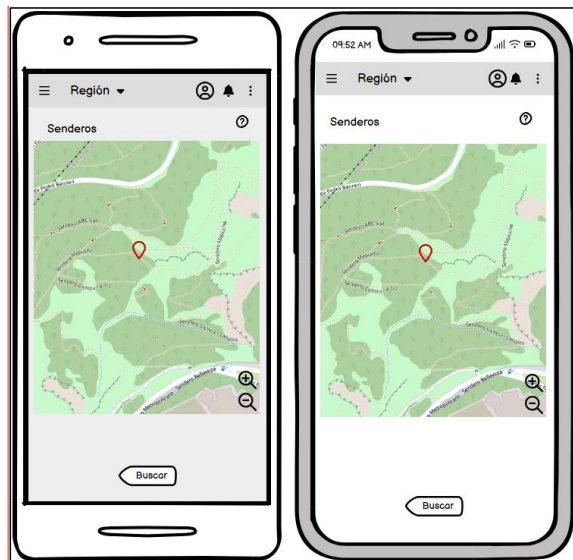


Figura 3-18. Diseño de Interfaz: Buscar senderos

Fuente: Elaboración propia

Comentado [PG38]: Veo sólo la opción buscar, cómo podría interrumpir esta tarea??

3.4.3. UI 03: Compartir Ubicación

El Objetivo permitir que un usuario comparta su ubicación en tiempo real con otros usuarios o comunidades.

- Casos de Uso Relacionados: CU_05 Compartir Ubicación
- Requisitos Funcionales Relacionados: RF-03: Compartir ubicación con amigos
- Aspectos de Usabilidad:
 - **Íconos** claros para indicar el estado de la ubicación compartida.
 - Ícono de zoom en mapa para acercar o alejar la vista del mapa.
 - Opciones rápidas para habilitar o deshabilitar el envío de datos.
 - Indicadores visuales de confirmación de envío.

Comentado [PG39]: Íconos (con tilde)

Comentado [PG40]: Qué significa eso de (+) y (-) en verde y rojo que aparece en los submenús?



Figura 3-19. Diseño de Interfaz: Compartir ubicación

Fuente: Elaboración propia

3.4.4. UI 04: Enviar SOS

El Objetivo permitir que un usuario comparta su ubicación y envíe un SOS (mensaje de emergencia predeterminado con anterioridad en la aplicación) a sus contactos de emergencia y a servicios de SALUD.

- Casos de Uso Relacionados: CU_12 Enviar SOS
- Requisitos Funcionales Relacionados: RF-03: Compartir ubicación con amigos y RF-06 Comunicación en tiempo real para emergencias
- Aspectos de Usabilidad:
 - Botón de acción destacado (CTA crítico): El botón "ENVIAR SOS" está resaltado en color rojo, con tamaño mayor y posición accesible para facilitar su uso en situaciones de emergencia.
 - Confirmación previa a acción crítica: Se presenta un cuadro de confirmación antes de enviar el SOS, evitando activaciones accidentales.
 - Feedback inmediato del sistema: El sistema muestra estados como "Obteniendo ubicación..." y "Enviando alerta..." , lo que reduce incertidumbre del usuario.
 - Indicadores visuales de éxito: Uso de checkmarks (✓) para confirmar:
 - Ubicación compartida
 - Contactos notificados
 - Servicios alertados
 - Información contextual relevante: Se muestran datos clave en tiempo real:
 - Precisión GPS
 - Altitud
 - Velocidad
 - Batería
 - Jerarquía visual clara: La información crítica (SOS enviado, estado) está en primer plano, mientras que el mapa queda como contexto de fondo.
 - Reducción de carga cognitiva
 - Opciones limitadas (Cancelar / Enviar SOS) para evitar confusión en situaciones de estrés.
 - Uso de iconografía significativa: Iconos de ambulancia 🚑 y helicóptero 🚁 refuerzan visualmente el tipo de ayuda esperada.
 - Tiempo estimado de respuesta visible: Mostrar "15-20 minutos" ayuda a gestionar expectativas del usuario.
 - Consistencia visual con mapas: Uso de mapa y marcador de ubicación facilita la comprensión espacial inmediata.



3.4.5. UI 03: Buscar Comunidades

El objetivo es permitir a los usuarios localizar comunidades relacionadas con sus intereses o ubicación geográfica.

- Casos de Uso Relacionados: CU_10 Seguir Usuario y Comunidad.
- Requisitos Funcionales Relacionados: RF_10 Permitir la búsqueda y seguimiento de comunidades.
- Aspectos de Usabilidad:
 - Listado de comunidades con descripciones claras.
 - Filtros por intereses o ubicación.
 - Indicadores visuales para identificar comunidades activas.



Figura 3-20. Diseño de Interfaz: Buscar Comunidades

Fuente: Elaboración propia

CONCLUSIÓN

A lo largo de estos más de 18 meses, se ha trabajado muy duro en tratar de entregar una solución a un problema real. A comienzos este trabajo partió como una página de comparación de equipo y componentes de mountain bike, luego pasó a ser una página ecommerce para una empresa de componentes de Mountain bike que vendía a través de redes sociales sus productos, pero luego terminó siendo este proyecto de senderos de mountain bike y añadiendo un valor extra con funcionalidades sociales para poder distinguirse del resto de aplicativos que tienen hoy por hoy en el mercado.

Si bien se comenzó juntando ideas y viendo factibilidad de esta idea. Se busca como poder distinguirnos entre el resto y como poder dar valor agregado a la idea, al desarrollo, al diseño y al producto final. Se investigó un poco sobre el usuario final y que es lo que ayudaría a usar nuestro producto por sobre los demás. Se necesitaría de un equipo especializado no solo en lo técnico sino en el producto y el usuario finales, así es como nació Trail Hunter, la empresa de usuarios y riders de Mountain Bike creando la solución definitiva para los riders... una empresa de riders para riders.

Había trabajado en proyectos anteriormente en el trabajo, como PO, como especialista en materias específicas, en mesas de trabajo multi disciplinarias para proyectos grandes a nivel país, pero nunca me había sentido tan puesto a prueba como haciendo este proyecto desde cero, viendo alcances y estimaciones de esfuerzo junto con planificar todo desde cero. Quizás a ese vaivén de mi parte es que este trabajo comenzó como una idea y se fue transformando en el camino ya que, el trabajo que hoy entrego y presento no es ni parecido a la idea de origen.

Es curioso como partes viendo todo borroso pero cada vez que pasas una materia específica de la carrera ves cada vez más claro hasta que por fin, luego de pasar por toda la malla de la carrera tienes una fotografía perfectamente clara de que es lo que hay que hacerse y el cómo.

Así mismo viendo clase a clase metodologías y sus formas de trabajo y como emplearlas es que cuando tocó pensar en que metodología había que usar decidí irme a lo que más conocía y más hemos trabajado dentro de la carrera, como lo es Cascada, algo familiar y donde me sentía en mi zona de confort luego de salirme de ella por completo para poder hacer este trabajo.

¿ Si me sirvió hacerlo con Cascada ? Claro que sí, dentro de todo el desafío fue grato aplicar una metodología que me era familiar.

¿ Me hubiera gustado quizás hacerlo con alguna metodología más nueva? También sí, pero nuevamente como todo era un desafío y nunca había empezado desde cero un proyecto con una metodología más nueva preferí inclinarme por algo familiar para hacer el proceso más cómodo.

Comentado [PG41]: Cuáles son tus conclusiones sobre la metodología aplicada para el análisis y diseño de la propuesta de solución??

Te sirvió??

Qué es lo que más resaltas??

Crees que te aportó, hay cosas por mejorar??

Hay muchas cosas tal vez que mejorar, quizás si en un comienzo la toma de requerimientos hubiera sido realizada de mejor forma o mejor pensada habría tenido la película más clara... servirá lo aprendido para futuras mejoras de este o nuevos proyectos.

BIBLIOGRAFÍA

Ley 21600 - <https://www.bcn.cl/leychile/navegar?idNorma=1195666>

Versiones Trailfork - https://help.trailforks.com/hc/en-us/articles/5430153546135-Free-vs-Paid-Trailforks-Pro-Membership?utm_source=chatgpt.com

ANEXOS**4. ANEXOS****4.1. Nº 1**

PROPUESTA TÉCNICA Y ECONÓMICA
Desarrollo Aplicación Para usuarios MTB
Empresa: RFS Chile

Contamos con experiencia en el desarrollo de soluciones tecnológicas orientadas a la optimización de procesos y gestión de información, destacando su participación en proyectos para empresas del rubro logístico con uso de radiofrecuencia. En dichos proyectos se ha demostrado una alta capacidad para adaptar los requerimientos específicos de cada cliente, manteniendo una comunicación constante y efectiva durante todo el ciclo de desarrollo.

Asimismo, poseemos experiencia en el diseño e implementación de plataformas que integran servicios de geolocalización, lo que resulta relevante para el desarrollo de aplicaciones basadas en rutas y posicionamiento geográfico.

Propuesta Técnica

La propuesta contempla el desarrollo de una aplicación móvil colaborativa orientada a la práctica del Mountain Bike, cuyo objetivo principal es centralizar la información de rutas, facilitar la interacción entre usuarios y permitir la actualización en tiempo real del estado de los senderos.

La solución será diseñada bajo un enfoque escalable y modular, permitiendo futuras ampliaciones sin afectar la operación del sistema. Además, se considerará una arquitectura que permita una adecuada integración con servicios de mapas y almacenamiento de información georreferenciada.

1. Lenguaje de Programación

- a. Para garantizar un rendimiento nativo con un solo código base, utilizaremos Flutter (Dart).
- b. Esto permite una interfaz fluida (60 FPS) esencial para mapas interactivos, reduciendo
- c. costos de mantenimiento en un 30% al cubrir iOS y Android simultáneamente.

2. Metodología

- a. Implementaremos Extreme Programming (XP). El proyecto se dividirá en iteraciones de 2
- b. semanas, donde el cliente actúa como "On-site Customer".
- c. Pair Programming: Dos desarrolladores trabajarán en un solo equipo para reducir

- d. errores en el algoritmo de rutas.
- e. Small Releases: Entrega de versiones funcionales cada 14 días.

3. Prácticas de Desarrollo

- a. TDD (Test Driven Development): Escribiremos las pruebas antes que el código,
- b. asegurando que la geolocalización sea exacta antes de integrarla.
- c. Integración Continua (CI): El código se integra y prueba automáticamente varias
- d. veces al día.
- e. Propiedad Colectiva: Todo el equipo conoce la lógica de las rutas y disciplinas.

4. Seguridad

- a. Autenticación: Implementación de Auth0 con soporte para OAuth 2.0
- b. (Google/Apple ID).
- c. Cifrado: Datos en tránsito vía TLS 1.3 y en reposo (AES-256).
- d. Privacidad: Cumplimiento con la Ley 19.628 de Protección de Datos Personales en
- e. Chile.

5. Infraestructura

Aprovecharemos la nueva Región de AWS en Chile (Santiago), operativa en 2026, para garantizar una latencia mínima en la carga de mapas.

- a. Servicios: AWS Lambda (backend serverless), Amazon S3 (fotos de rutas) y PostgreSQL

(PostGIS) para el manejo de datos geográficos complejos.

Plan de Trabajo

1. Cronograma (Duración estimada: 16 semanas / 8 Iteraciones)

Cada Sprint tiene una duración de 80 horas hombre.

Sprint	Enfoque Técnico	Entregable Funcional
Sprint 1	Arquitectura & Geo-Core	Setup de mapa base Chile, GPS activo y registro.
Sprint 2	Capa Colaborativa	Creación de rutas, trazado sobre mapa y tags de dificultad.
Sprint 3	Crowdsourcing en Terreno	Sistema de reportes: Fotos de cierres, clima y construcción.
Sprint 4	Comunidad & Disciplinas	Filtros por disciplina (Enduro, DH, XC) y perfiles sociales.

Tabla 4-1 Sprint Empresa N° 1

2. Plan de Capacitación

- a. Capacitación: Taller de 2 horas al finalizar cada Sprint para el uso de las nuevas herramientas, además de 3 sesiones presenciales o vía Meet una vez entregado el proyecto.

- i. Administradores: Gestión de reportes de usuarios y moderación de contenido. Uso de tablas y relaciones entre éstas.
- ii. Embajadores: Uso avanzado de la herramienta de trazado de rutas.

3. Criterios y Conceptos de Desarrollo

El diseño será Offline-First. Sabemos que en los cerros de Chile la señal es inestable; la app permitirá guardar mapas y rutas localmente (caché) y sincronizar cuando detecte señal además de integrar los siguientes conceptos:

- a. Precisión GPS (Filtering): Uso de Kalman Filters para evitar que la ruta “salte” violentamente cuando el ciclista pasa por bosques densos o quebradas.
- b. Eficiencia Energética: La app debe optimizar el uso del sensor GPS para no agotar la batería del celular en salidas largas (4+ horas).
- c. Jerarquía de Información: En ruta, los botones deben ser grandes y fáciles de presionar con guantes de ciclismo.

4. Gestión del Cambio

Se utilizará un ambiente de Staging (espejo) donde el cliente podrá validar las nuevas funcionalidades antes de que lleguen a los ciclistas finales, evitando interrupciones en el servicio.

5. Gestión de Riesgos

- a. Baja precisión GPS en quebradas: Mitigado mediante algoritmos de suavizado de ruta (Snap-to-road/trail).
- b. Desactualización de cartografía: Mitigado por el sistema de reportes comunitarios con fotos georreferenciadas.

6. Requisitos de Organización

El cliente deberá designar un Product Owner con disponibilidad de 4 horas semanales para la revisión de iteraciones y priorización del “Backlog”.

7. Detalles de la Post-implementación

- a. Garantía: 6 meses contra errores de código (Bugs).
- b. Soporte: Mesa de ayuda nivel 1 y 2, con tiempo de respuesta de 4 horas para incidencias críticas según detalle

8. Soporte

El servicio contempla la administración integral de la plataforma tecnológica asociada a la aplicación de Mountain Bike, incluyendo la gestión de usuarios, monitoreo del sistema y mantenimiento de la información georreferenciada. Incluye administración de rutas, monitoreo, respaldos, soporte técnico y supervisión de mapas.

Se establece una mesa de ayuda como punto centralizado de contacto para los usuarios de la aplicación, con gestión de tickets para registro, seguimiento y resolución de incidencias.

Tiempos SLA:

- Soporte usuarios: respuesta 2h, solución 24h
- Reportes rutas: respuesta 4h, solución 48h
- Incidencias críticas: respuesta 1h

9. Detalles de la Post-implementación: Soporte y Continuidad Operativa

Una vez finalizada la etapa de desarrollo de los Sprints, se establece un plan de Soporte, Mantenimiento e Infraestructura para garantizar la alta disponibilidad de la plataforma en las rutas de MTB.

Componente	Descripción de Cobertura
Infraestructura Cloud	Alojamiento en la región AWS Santiago , base de datos geoespacial (PostGIS) y almacenamiento S3 para registros fotográficos.
Gestión de APIs	Administración y optimización de cuotas de Mapbox/Google Maps para asegurar la visualización de mapas topográficos.
Mantenimiento Preventivo	Actualización periódica de librerías de seguridad y compatibilidad con nuevas versiones de iOS y Android.
Monitoreo 24/7	Vigilancia proactiva de servidores para asegurar el funcionamiento de la App durante los fines de semana (periodo de mayor tráfico).
Soporte Técnico L2	Resolución de incidencias técnicas críticas que afecten la geolocalización o el acceso de usuarios.

Tabla 4-2 Estructura de Servicios Mensuales

Propuesta Económica Mensual

El costo por la continuidad operativa, administración de nube y soporte técnico se fija en una tarifa mensual reajutable:

Fee Mensual de Continuidad: 12 UF + IVA

Consideraciones:

- a. Facturación: El cobro se realizará de forma mensual anticipada, dentro de los primeros 5 días de cada mes.
- b. Cálculo de UF: Se utilizará el valor de la Unidad de Fomento (UF) correspondiente al día de la emisión de la factura. Para el caso de AWS se detallará la conversión de USD a UF en la facturación.
- c. Escalabilidad: El fee de 12 UF cubre hasta un volumen de 25.000 Usuarios Activos Mensuales (MAU). En caso de superar este tráfico, se

realizará una renegociación basada en el consumo excedente de las APIs de mapas.

- d. Vigencia: Este servicio entra en vigor inmediatamente después de la entrega del último Sprint de desarrollo.

Propuesta Económica

Valores expresados en pesos chilenos (CLP).

Ítem	Descripción	Valor Estimado
Desarrollo (Fase 1-8)	Equipo de 3 desarrolladores + Diseñador UX	236UF por Sprint
Infraestructura (Anual)	Hosting AWS Chile + APIs Google Maps	144UF Anual
Soporte y Mantenimiento	Mensual (opcional después de garantía)	12UF Mensual
TOTAL INVERSIÓN INICIAL	Incluye publicación en Stores	2.176UF + IVA

Tabla 4-3 Propuesta Económica

El valor final podrá ajustarse en función del alcance definitivo del proyecto, las integraciones requeridas y las funcionalidades específicas que se definan durante la etapa de levantamiento de requerimientos.

4.2. Anexo 2

PROPUESTA TÉCNICA COMERCIAL SISTEMA APLICACIÓN MOUNTAIN BIKE

Cotización N° 0000160

Santiago, 17 de febrero del 2026

Atención a: Aldo López Vallejos

Preparado por: Diego Alten – Sol Alten

1. Información General

El presente documento tiene por objetivo entregar una propuesta Técnica-Comercial para el diseño, desarrollo e implementación de una aplicación móvil orientada a la comunidad de Mountain Bike en Chile.

La solución propuesta busca abordar las problemáticas actuales detectadas en el ecosistema MTB, tales como la falta de centralización de información de senderos, escasa comunicación entre usuarios, uso de herramientas no especializadas y limitaciones en el acceso a información relevante en tiempo real.

La plataforma permitirá registrar, visualizar y compartir rutas, así como fomentar la interacción entre usuarios, promoviendo una práctica más segura, organizada y sostenible del deporte.

Características técnicas de la plataforma:

- Aplicación móvil compatible con iOS y Android
- Interfaz amigable y responsiva
- Sistema de mapas georreferenciados especializados en deportes
- Registro y visualización de senderos creados por usuarios
- Descarga de mapas para uso offline
- Integración con servicios de clima en tiempo real
- Sistema de mensajería entre usuarios
- Creación de comunidades
- Publicación de imágenes y contenido
- Notificaciones push
- Panel de administración web
- Seguridad mediante cifrado de datos (SSL)

2. Definición del Proyecto

2.1 Objetivos del Proyecto

1. **Mejorar la experiencia de los usuarios de Mountain Bike**
Facilitar el acceso a información relevante, rutas y herramientas digitales especializadas en la localización de Chile.
2. **Centralizar la información de senderos a nivel nacional**
Generar una base de datos colaborativa y actualizada.
3. **Mejorar la seguridad en la práctica del deporte**
Mediante comunicación en tiempo real y alertas.
4. **Fomentar la creación de comunidad entre ciclistas**
A través de funcionalidades sociales y de interacción.
5. **Optimizar el uso y gestión de rutas**
Promoviendo buenas prácticas y orden en los senderos.
6. **Reemplazar soluciones no especializadas**
Mediante una aplicación enfocada exclusivamente en MTB.

2.2 Metodología de Trabajo

Se utilizará una metodología basada en modelo Agile, permitiendo iteraciones continuas y adaptación a requerimientos del cliente.

Fase 0: Planificación (2 semanas)

- Reunión Kick-Off
- Definición de alcance
- Identificación de usuarios
- Definición de backlog
- Diseño inicial de arquitectura
- Generación Gantt detallada Fase 1

Fase 1: Desarrollo MVP funcional (12 semanas)

- Diseño UX/UI - Implementar componentes de la solución para desarrollar el MVP del producto (Front End, Back end, API's, base de datos, componentes gráficos, etc)
- Generación piloto de la aplicación con mockups
- Implementar flujo de desarrollo (desarrollo-qa-deployment)
- Incluir visualización de mapas
- Registro de rutas
- Capacitación
- Liberación para pruebas de usuario
- Actualizar Backlog, Gantt y definir fase 2

Fase 2: Funcionalidades sociales (8 semanas)

- Agregar funcionalidad de comunidades, publicaciones, mensajería, notificaciones
- Generación de Wireframes para esta fase
- Agregar funcionalidad a componentes (Front End, Back end, API's, base de datos, componentes gráficos, etc)

- QA interno
- Liberación para pruebas de usuario
- Actualizar Backlog, Gantt y definir fase 3

Fase 3: Funcionalidades avanzadas (8 semanas)

- Agregar funcionalidades locales, mapas offline, integración clima, optimización de conexiones a servicios.
- Generación de Wireframes para esta fase
- Agregar funcionalidad a componentes (Front End, Back end, API's, base de datos, componentes gráficos, etc)
- QA interno
- Liberación para pruebas de usuario
- Actualizar Backlog, Gantt y definir fase 4

Fase 4: Mejoras (6 semanas)

- Ajustes UX/UI
- Seguridad
- Escalabilidad
- Paso a productivo

Fase 5: Cierre (2 semanas)

- Pruebas finales
- Documentación – firma de actas de entrega
- Paso a productivo
- Cierre del proyecto
- Inicio garantía

Tabla 4-4 Fase Tiempo estimado

Fase	Tiempo estimado
0	2 semanas
1	12 semanas
2	8 semanas
3	8 semanas
4	6 semanas
5	2 semanas
Total	38 semanas

Duración total estimada: 38 semanas

2.2.1 Esquema General de la Solución

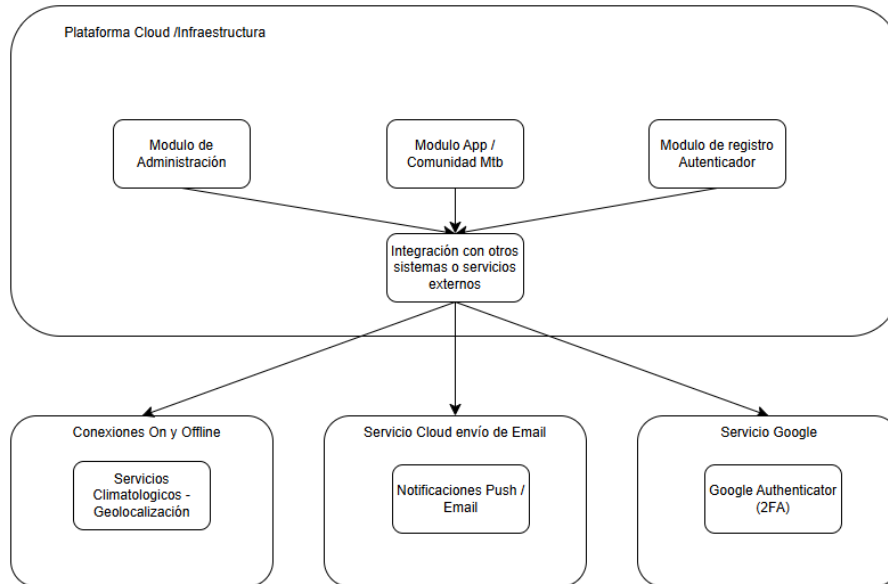


Figura 21 Esquema General de la Solución

La solución propuesta considera una arquitectura modular compuesta por una aplicación móvil para usuarios, un módulo de administración, un componente de autenticación y una capa de integración con servicios externos, tales como mapas, geolocalización, clima, notificaciones y autenticación de doble factor. Esta arquitectura permitirá escalabilidad, seguridad y una adecuada integración entre los distintos componentes funcionales de la plataforma.

La solución estará compuesta por:

- Aplicación móvil (Front-End)
- Backend / API
- Base de datos geoespacial
- Servicios externos (mapas, clima, notificaciones)
- Panel de administración web

2.2.2 Características Funcionales de la Solución

- Registro y autenticación de usuarios
- Creación y visualización de rutas
- Geolocalización en tiempo real
- Comunicación entre usuarios

- Publicación de contenido multimedia
- Creación de comunidades
- Consulta de clima
- Mapas offline

2.2.3 Características Técnicas de la Solución

- Arquitectura basada en API REST
- Infraestructura en la nube (AWS / Azure / GCP)
- Base de datos con soporte geoespacial
- Seguridad SSL
- Escalabilidad horizontal

2.2.4 Características Generales

- Plataforma disponible 24/7
- Alta disponibilidad
- Escalabilidad
- Enfoque en experiencia de usuario

2.3 Garantía

Se considera un periodo de garantía de 12 meses posterior a la puesta en producción, durante el cual se corregirán fallas sin costo adicional.

2.4 Entregables

- Aplicación móvil operativa
- Código fuente
- Manuales de usuario y técnicos
- Capacitación a usuarios

2.5 No incluidos en el proyecto

- Licencias de software o servicios externos
- Infraestructura cloud
- Hardware
- Servicios de terceros

3. Responsabilidades del Cliente

- Designar un responsable del proyecto
- Entregar información necesaria
- Proveer ambientes de prueba y producción
- Validar entregables

4. Equipo de Trabajo

Tabla 4-5 Equipo de Trabajo

Rol	Cantidad	Perfil
Jefe de Proyecto	1	Experiencia en gestión TI
Desarrollador Senior	1	Especialista backend
Desarrolladores	2	Desarrollo mobile/web

Diseñador UX/UI	1	Experiencia usuario
QA	1	Testing

5. Plan de Trabajo

El plan de trabajo se definirá mediante carta Gantt en base a las fases descritas en la metodología, una vez firmada la aprobación del entregable de la fase se continuará con la siguiente.

6. Confidencialidad

Toda la información tratada durante el proyecto será considerada confidencial.

7. Propuesta Económica

7.1 Valores

Tabla 4-6 Valores

Ítem	Cantidad	Valor (UF)
Desarrollo e implementación	1	2620

Total: 2620 UF

7.2 Servicio de Soporte

Tabla 4-7 Servicio de Soporte

Ítem	Valor (UF/Mes)
Soporte Aplicativo	10
Administración Plataforma	10

Total mensual: 20 UF

Nota: Trabajos extras que se consideren como mejora al sistema entregado serán considerados fuera de este punto, siendo evaluados como proyecto aparte.

7.3 Consideraciones

- Reuniones periódicas
- Valores netos + IVA
- Forma de pago por fases
- Soporte inicia en productivo

8. ANEXO A: Plan de Soporte y Mantenimiento

8.1 Objetivo

Asegurar continuidad y rendimiento de la aplicación.

8.2 Alcance

- Soporte técnico
- Mantenimiento correctiva
- Mantenimiento preventiva
- Mantenimiento evolutiva

