

UNIVERSIDAD PRIVADA FRANZ TAMAYO

CARRERA DE INGENIERÍA DE SISTEMAS



PROYECTO INTEGRADOR II

**“PLATAFORMA WEB CON CHATBOT GUIA DE RUTAS PARA
TRANSEÚNTES DE MINIBUSES EN LA CIUDAD DE EL ALTO”**

CASO: SINDICATO PEDRO DOMINGO MURILLO

Postulante: Ajoruro Condori Jhon Deyvid

Tutor: Ing. Juan Gabriel Lazcano Balanza

EL ALTO - BOLIVIA

2025

DECLARACIÓN JURADA DE AUTENTICIDAD DE PERFIL DE TRABAJO DE GRADO

Yo, Ajoruro Condori Jhon Deyvid, estudiante de Proyecto Intermedio Integrador I, de la Carrera de Ingeniería de Sistemas de la Universidad Privada Franz Tamayo, identificado con CI. 13494441 LP y Registro Universitario:

Declaro bajo juramento que:

1. Soy autor del Proyecto Integrador:
“PLANIFICADOR DE RUTAS DE MINIBUSES EN EL ALTO”
El mismo que presentamos bajo la modalidad de Proyecto Integrador.
2. El texto de mi perfil de Proyecto Integrador respeta y no vulnera los derechos de terceros, incluidos los derechos de propiedad intelectual. En tal sentido, declaro que este Proyecto Integrador no ha sido plagiado total ni parcialmente, para la cual he respetado las normas internacionales de citas y referencias de las fuentes consultadas
3. El texto del Proyecto Integrador que presento no ha sido publicado ni presentado antes en cualquier medio electrónico o físico.
4. Por tanto, declaro que mi perfil de Proyecto Integrador cumple con todas las normas de la carrera de Ingeniería de Sistemas de la Universidad Privada Franz Tamayo.
5. El incumplimiento de lo declarado da lugar a responsabilidad del declarante, en consecuencia; a través del presente documento asumo frente a terceros, la carrera de Ingeniería de Sistemas de la Universidad Privada Franz Tamayo toda responsabilidad que pueda derivarse por el Proyecto Integrador presentado.

Fecha: 20 de marzo de 2025

.....

Univ. Jhon Deyvid Ajoruro Condori
CI.

DEDICATORIA

A mi madre, por su amor incondicional, por haberme criado con esfuerzo, valores y dedicación, siendo mi mayor ejemplo de fortaleza y constancia.

A mi familia, que siempre me ha brindado su apoyo, paciencia y motivación para seguir adelante, incluso en los momentos más difíciles.

Este logro es tan mío como suyo.

AGRADECIMIENTOS

Quiero expresar mi sincero agradecimiento a la **Universidad Privada Franz Tamayo – UNIFRANZ**, por brindarme la oportunidad de formarme académicamente y por el acceso a los conocimientos que han sido clave en mi desarrollo profesional.

De igual manera, agradezco a mis **docentes**, quienes con su enseñanza, compromiso y dedicación han contribuido de manera significativa a mi formación.

Gracias a todos por la confianza y apoyo, que me motivan a seguir esforzándome para alcanzar nuevas metas.

ÍNDICE

| | |
|--|-----------|
| GENERALIDADES..... | 1 |
| 1.1. INTRODUCCIÓN..... | 1 |
| 1.2. ANTECEDENTES..... | 2 |
| 1.2.1 Antecedentes afines al proyecto de grado..... | 2 |
| 1.3. DESCRIPCIÓN DEL OBJETO DE ESTUDIO..... | 2 |
| 1.4. PLANTEAMIENTO DEL PROBLEMA..... | 3 |
| 1.4.1 Identificación de la situación problemática..... | 4 |
| 1.4.2 Problemas secundarios..... | 4 |
| 1.4.3 Formulación del problema..... | 5 |
| 1.5. OBJETIVOS..... | 5 |
| 1.5.1 Objetivo General..... | 5 |
| 1.5.2 Objetivos Específicos..... | 5 |
| 1.6. JUSTIFICACIÓN..... | 6 |
| 1.6.1 Justificación Técnica..... | 6 |
| 1.6.2 Justificación Social..... | 6 |
| 1.7. METODOLOGÍA..... | 7 |
| 1.7.1 Enfoque del Proyecto..... | 7 |
| 1.7.2 Tipo de Diseño..... | 7 |
| 1.7.3 Método de Desarrollo..... | 7 |
| 1.7.4 Herramientas..... | 8 |
| 1.8. ALCANCES Y APORTES..... | 10 |
| 1.8.1 Alcance temático..... | 10 |
| 1.8.2 Alcance geográfico..... | 11 |
| 1.8.3 Aportes del proyecto..... | 11 |
| 1.8.4 Aportes académicos..... | 11 |
| 1.8.5 Ingenieriles / tecnológicos..... | 12 |
| MARCO TEÓRICO..... | 13 |
| 2.1. Introducción..... | 14 |
| 2.2 Sistema de Información..... | 14 |
| 2.3. Transporte urbano de minibuses..... | 15 |
| 2.3.1. Transporte Urbano..... | 15 |
| 2.3.2. Minibús..... | 15 |
| 2.3.3 Rutas de transporte público..... | 16 |
| 2.3.4. Ciudad de El Alto..... | 16 |
| 2.4. Inteligencia Artificial..... | 17 |
| 2.4.1 ChatBot..... | 17 |
| 2.4.2 Agentes inteligentes..... | 17 |
| 2.5 Arquitectura de software..... | 18 |
| 2.5.1 Concepto de arquitectura de software..... | 18 |
| 2.5.2 Arquitectura cliente-servidor..... | 18 |

| | |
|--|-----------|
| 2.5.3 Arquitectura web modular..... | 19 |
| 2.6 Plataforma Web..... | 20 |
| 2.7. Metodología Scrum..... | 20 |
| 2.8 Herramientas de Desarrollo y Lenguajes..... | 21 |
| 2.8.1 HTML y CSS..... | 21 |
| 2.8.2 JavaScript..... | 21 |
| 2.8.3 Node.js y Express.js..... | 22 |
| 2.8.4 Bases de datos (MariaDB)..... | 22 |
| 2.8.5 Mapas y visualización geográfica (OpenStreetMap, MapLibre GL JS, Leaflet)..... | 22 |
| MARCO PRÁCTICO..... | 24 |
| 3.1 Introducción..... | 25 |
| 3.2 Tipo de Investigación..... | 25 |
| 3.3 Diseño de la Investigación..... | 26 |
| 3.4 Métodos, Técnicas e Instrumentos..... | 26 |
| 3.5. Requerimientos funcionales..... | 28 |
| 3.6. Requerimientos No Funcionales..... | 33 |
| 3.7 Diseño de la base de datos..... | 34 |
| 3.7.1 Estructura general..... | 34 |
| 3.7.2 Relaciones principales..... | 35 |
| 3.7.3 Modelo Entidad–Relación..... | 36 |
| 3.8 SCRUM..... | 36 |
| 3.8.1 Product Backlog..... | 36 |
| 3.8.2 Planificación de sprints..... | 45 |
| 3.8.3 Cronograma..... | 49 |
| - Anexos..... | 54 |
| Pdf con las correcciones..... | 55 |

Índice de figuras

| | |
|---|-----------|
| Figura 1 Diagrama de situación actual..... | 21 |
| Figura 2 Diagrama de contexto..... | 21 |
| Figura 3 Árbol de Problemas..... | 22 |

CAPÍTULO

I

GENERALIDADES

1.1. INTRODUCCIÓN

Actualmente, el acceso a la información de rutas de transporte público es fundamental para mejorar la movilidad urbana. Una buena planificación de los recorridos permite a los usuarios ahorrar tiempo, evitar confusiones y optimizar sus desplazamientos.

En la ciudad de El Alto, la mayoría de las personas utiliza minibuses como medio principal de transporte. Sin embargo, existe una ausencia de información clara y accesible sobre qué rutas seguir, provocando que los usuarios dependan de su experiencia previa o de preguntar a terceros. Esto genera demoras y puede aumentar los costos de traslado.

El presente proyecto propone el desarrollo de una plataforma web que permita a los usuarios consultar de manera rápida y sencilla las rutas de minibuses disponibles cerca de su ubicación en la ciudad de El Alto. Además, integra un chatbot para facilitar la interacción, permitiendo a los usuarios obtener recomendaciones sobre lugares y datos extras de las rutas.

Para el desarrollo del proyecto se utiliza la metodología Scrum. Se trabajará con tecnologías como Node.js para el servidor, y HTML, CSS y JavaScript para el desarrollo del frontend. Además, se utilizará la API de OpenStreetMap para visualizar los recorridos de los minibuses en el mapa. El sistema se organizará bajo la estructura Modelo-Vista-Controlador (MVC) para asegurar un desarrollo ordenado y escalable.

1.2. ANTECEDENTES

1.2.1 Antecedentes afines al proyecto de grado

A nivel internacional, diversas ciudades han implementado plataformas digitales para la gestión del transporte público. Aplicaciones como Moovit y Google Maps han integrado información de buses y trenes en múltiples ciudades, permitiendo a los usuarios planificar sus trayectos con datos en tiempo real. Sin embargo, en muchos países en desarrollo, los sistemas de minibuses no están debidamente incorporados en estas plataformas debido a la falta de bases de datos estructuradas y colaboración con operadores locales.

En países como Colombia y México, se han desarrollado iniciativas comunitarias que buscan mapear rutas de transporte informal utilizando aplicaciones móviles y sistemas de información geográfica (GIS). En otros casos, gobiernos han trabajado en conjunto con empresas tecnológicas para integrar datos del transporte público en tiempo real, mejorando la eficiencia del sistema y facilitando el acceso a la información para los ciudadanos. Estas experiencias pueden servir como referencia para el desarrollo del presente proyecto, adaptándolo a la realidad local y las necesidades específicas de los usuarios.

1.3. DESCRIPCIÓN DEL OBJETO DE ESTUDIO

En la ciudad de El Alto, el proceso que siguen los usuarios para utilizar el servicio de minibuses es informal, poco estructurado y basado principalmente en la experiencia personal o en recomendaciones de terceros. Cuando una persona necesita

Ilegar a un destino específico, suele recurrir a preguntas a transeúntes o a su conocimiento previo para elegir una ruta. Sin embargo, la información que recibe es, en muchos casos, imprecisa o basada en referencias locales ambiguas, lo que complica ubicar correctamente las paradas. Esto provoca que el usuario aborde un vehículo con incertidumbre sobre si lo llevará efectivamente a su destino, y dependa del criterio de otros para saber en qué punto exacto debe descender. Esta situación genera desorientación, errores en el trayecto y pérdida de tiempo (ver figura 1).

1.4. PLANTEAMIENTO DEL PROBLEMA

El transporte público constituye un elemento fundamental en la dinámica urbana, pues facilita el desplazamiento de las personas hacia sus actividades laborales, educativas y sociales. No obstante, cuando la información sobre rutas, recorridos, horarios y puntos de transbordo no está disponible de forma clara y accesible, se generan dificultades en la planificación de los viajes, lo que conlleva pérdida de tiempo, costos adicionales e incluso frustración en los usuarios.

Esta situación se hace particularmente evidente en la ciudad de El Alto, caracterizada por un crecimiento urbano acelerado y una fuerte dependencia del transporte en minibuses como principal medio de movilidad. A pesar de su relevancia, actualmente no existe una herramienta digital que brinde a los usuarios orientación precisa sobre qué rutas tomar para llegar a su destino, especialmente cuando se dirigen a zonas desconocidas.

En este contexto, el problema radica en la dificultad que enfrentan los usuarios del transporte público para obtener información clara, confiable y en tiempo real sobre

las rutas de minibuses que los lleven a un destino específico, lo cual genera incertidumbre, pérdida de tiempo y decisiones ineficientes en sus desplazamientos diarios.

1.4.1 Identificación de la situación problemática

La principal dificultad identificada radica en la falta de acceso a información organizada y actualizada sobre las rutas de minibuses que operan en la ciudad de El Alto. Los usuarios, especialmente aquellos que se desplazan por primera vez o que deben llegar a zonas poco conocidas, no cuentan con una guía que les indique de manera precisa qué rutas tomar, en qué puntos realizar transbordos ni cuánto tiempo o costo implicará su trayecto.

Esta carencia de información genera incertidumbre, pérdida de tiempo, costos adicionales y descontento generalizado entre los pasajeros. Además, la ausencia de una plataforma centralizada que unifique los datos sobre recorridos, tarifas y horarios impide la planificación eficiente de los desplazamientos urbanos. (ver figura 3).

1.4.2 Problemas secundarios

- **Carencia de información sistematizada** sobre rutas y recorridos de minibuses, lo que limita la orientación adecuada de los usuarios al momento de planificar sus desplazamientos dentro de la ciudad de El Alto.
- **Inexistencia de herramientas digitales** que faciliten la consulta de trayectos y puntos de transbordo, derivando en procesos ineficientes de toma de decisiones por parte de los pasajeros.

- **Insuficiencia de mecanismos tecnológicos** que integren datos sobre horarios, costos y condiciones del servicio, generando incertidumbre y dificultando la planificación anticipada de los viajes.
- **Ausencia de sistemas de actualización dinámica** ante cambios o desvíos operativos en las rutas, provocando que los pasajeros enfrenten imprevistos y retrasos durante sus desplazamientos.

1.4.3 Formulación del problema

¿Cómo optimizar el acceso y la planificación de rutas de minibuses para los usuarios del transporte público en la ciudad de El Alto, permitiéndoles conocer las opciones de recorrido disponibles y la mejor ruta según su origen y destino?

1.5. OBJETIVOS

1.5.1 Objetivo General

Implementar una plataforma web que facilite la planificación de rutas de minibuses en la ciudad de El Alto, permitiendo a los usuarios conocer las opciones de recorrido disponibles y la mejor ruta según su origen y destino.

1.5.2 Objetivos Específicos

- **Analizar** la información sobre las rutas de minibuses en la ciudad de El Alto para comprender su distribución y funcionamiento.
- **Diseñar** una interfaz de usuario intuitiva y accesible que permita a los usuarios ingresar su origen y destino para visualizar las rutas disponibles.

- **Desarrollar** un módulo de inteligencia artificial que optimice la planificación de rutas de minibuses en función de la información recopilada.
- **Validar** la plataforma web mediante pruebas con usuarios del transporte público, asegurando su funcionalidad y usabilidad en la planificación de rutas.

1.6. JUSTIFICACIÓN

1.6.1 Justificación Técnica

Este proyecto se realiza porque existe la necesidad de contar con una herramienta práctica y visual que permita trazar rutas de minibuses de forma manual. Con esta aplicación, se pueden marcar los caminos por donde pasan los vehículos, guardar paradas importantes y tener un control más claro del recorrido. Además, al estar basada en tecnologías web y mapas abiertos, es fácil de usar desde cualquier dispositivo con acceso a internet, sin necesidad de instalar programas especiales.

1.6.2 Justificación Social

La aplicación busca ayudar a las personas que usan el transporte público, especialmente a quienes no conocen bien las rutas o quieren planificar mejor su viaje. También puede ser útil para autoridades o conductores que quieran organizar sus recorridos. Al tener un sistema donde se muestren las paradas y el camino exacto, se mejora el acceso a la información y se apoya a la comunidad con una solución clara y útil.

1.7. METODOLOGÍA

1.7.1 Enfoque del Proyecto

El presente proyecto adopta un **enfoque cuantitativo**, ya que se basa en la recolección y análisis de datos objetivos mediante encuestas dirigidas a usuarios del transporte público en El Alto. Estos datos permitirán identificar patrones de movilidad, necesidades específicas y frecuencia de uso de rutas de minibuses, lo cual servirá de base para el diseño funcional y lógico del sistema.

1.7.2 Tipo de Diseño

El diseño del proyecto es **no experimental, descriptivo y de corte transversal**. No experimental porque no se manipulan variables, sino que se observa la realidad tal como ocurre en el uso del transporte público. Es descriptivo, ya que se busca identificar aspectos clave del servicio de minibuses, como horarios de funcionamiento, número de vehículos activos por ruta y condiciones de operación. Además, es transversal porque la información se recolectará en un solo momento del tiempo, permitiendo obtener una visión clara y puntual de la situación actual sin necesidad de un seguimiento prolongado(ver encuestas: [PI2_Encuestas_EasyWay](#)).

1.7.3 Método de Desarrollo

Para la implementación de la plataforma, se utilizará **Scrum**, permitiendo dividir el desarrollo en iteraciones llamadas *sprints*. Esta facilita una evolución continua del sistema con retroalimentación frecuente.

1.7.4 Herramientas

El desarrollo de la plataforma web guía de rutas para usuarios de minibuses en El Alto requiere una combinación de herramientas que permiten manejar tanto el frontend como el backend, la visualización de mapas y el almacenamiento de datos. A continuación, se describen las principales tecnologías que se utilizarán:

➤ **HTML y CSS**

HTML (Lenguaje de Marcado de Hipertexto) proporciona la estructura básica de la plataforma, mientras que CSS (Hojas de Estilo en Cascada) define su apariencia visual. Juntas permiten construir interfaces accesibles, organizadas y responsivas para los usuarios (Duckett, 2011, p. 29).

➤ **JavaScript**

JavaScript es un lenguaje de programación esencial para agregar interactividad al sitio web, permitiendo la ejecución de funciones dinámicas como la carga de rutas, interacción con el mapa y validaciones en el navegador.

➤ **NodeJS**

Node.js permite ejecutar JavaScript del lado del servidor, facilitando el manejo de solicitudes web, conexiones con bases de datos y procesamiento de lógica en tiempo real. Su modelo asincrónico y orientado a eventos lo hace ideal para aplicaciones con muchos usuarios concurrentes (Gascón, 2024, p. 15).

➤ **Express.js**

Framework minimalista para Node.js que permite crear APIs de forma rápida y sencilla. Facilita el desarrollo de rutas, gestión de peticiones HTTP y conexión con bases de datos, siendo clave para estructurar el backend del sistema.

➤ **MongoDB**

MongoDB es una base de datos NoSQL orientada a documentos que permite almacenar información de manera flexible y escalable. Es útil para gestionar datos como rutas, comentarios de usuarios y puntos de interés sin una estructura rígida (Chodorow, 2013, p. 42).

➤ **OpenStreetMap (OSM)**

OpenStreetMap ofrece mapas de código abierto y editables, ideales para integrar visualización geográfica de rutas en tiempo real. Su uso permite representar los trayectos de minibuses sobre un entorno interactivo y constantemente actualizado (Ramm, 2010, p. 10).

➤ **MapLibre GL JS o Leaflet**

Estas librerías JavaScript se utilizan para mostrar mapas interactivos en el navegador basados en datos de OSM. Permiten trazar rutas, colocar marcadores y personalizar la visualización cartográfica del sistema.

➤ **Chatbot API (por definir)**

Se integrará un chatbot que asistirá al usuario en la consulta de rutas. Puede implementarse mediante herramientas como Dialogflow o librerías personalizadas con NLP, con el fin de guiar al usuario mediante conversación natural.

1.8. ALCANCES Y APORTES

1.8.1 Alcance temático

El alcance del proyecto abarcara el desarrollo de una plataforma web orientada a la optimización del uso del transporte público en minibuses en la ciudad de El Alto. El sistema incluirá los siguientes módulos principales:

- **Módulo de consulta de rutas:** permitirá al usuario ingresar su ubicación actual y destino para visualizar las rutas de minibuses disponibles que se ajusten a su trayecto.
- **Chatbot interactivo:** un asistente virtual que brindará apoyo en la búsqueda de rutas, destinos, transbordos y recomendaciones de manera conversacional.
- **Módulo de gestión de rutas:** destinado a registrar, actualizar y verificar la información de las rutas ingresadas en la plataforma.
- **Módulo de administración de usuarios y roles:** permitirá gestionar los perfiles de las personas encargadas de administrar el sistema.
- **Módulo de autenticación (login):** control de acceso para garantizar la seguridad y uso adecuado de la plataforma.
- **Módulo de notificaciones y avisos:** servirá para informar a los usuarios sobre actualizaciones de rutas, novedades de la plataforma o incidencias relevantes.

De esta manera, el proyecto no solo se enfoca en la experiencia del usuario final, sino también en las herramientas necesarias para la gestión y administración del sistema.

1.8.2 Alcance geográfico

El proyecto se delimita al **Sindicato de Minibuses Pedro Domingo Murillo**, tomando como punto de referencia principal la **Avenida 6 de Marzo**, en la ciudad de El Alto. Se trabajará únicamente con las rutas afiliadas a este sindicato que circulen dentro de la ciudad, excluyendo aquellas que descienden hacia la ciudad de La Paz.

De este modo, el sistema se enfocará en digitalizar y optimizar la información de las rutas internas de El Alto que parten desde la Av. 6 de Marzo. En una etapa futura, se podría considerar ampliar la cobertura hacia otras rutas o sindicatos de transporte, según el avance del proyecto.

1.8.3 Aportes del proyecto

Esta plataforma ayuda a que las personas en El Alto encuentren fácilmente qué minibús tomar para llegar a su destino. Antes, tenían que preguntar o adivinar qué línea les sirve, ahora tendrán una herramienta confiable que les muestra rutas claras, paradas exactas y la mejor opción para moverse por la ciudad.

1.8.4 Aportes académicos

La plataforma permite poner en práctica conocimientos de desarrollo web, geolocalización y bases de datos, aplicándolos en un proyecto real. Además, contribuye al aprendizaje sobre la interacción entre el usuario y la computadora, el

diseño de interfaces y el manejo de sistemas de información aplicados al transporte urbano. De igual manera, sirve como un caso de estudio sobre el transporte público en El Alto, que puede ser útil como referencia para futuras investigaciones o trabajos académicos.

1.8.5 Ingenieriles / tecnológicos

Desde el punto de vista tecnológico, el proyecto incorpora agentes conversacionales inteligentes que facilitan la interacción del usuario con la plataforma y representan un aporte innovador frente a los sistemas tradicionales de consulta de rutas. También integra la digitalización de rutas y la gestión de información en tiempo real, lo que ayuda a planificar trayectos de manera más eficiente y optimizar el uso del transporte público. Además, el sistema está diseñado de manera modular y escalable, permitiendo la gestión de usuarios, roles y rutas, y ofreciendo una base sólida para futuras mejoras, como la incorporación de notificaciones inteligentes o la integración con aplicaciones móviles.

CAPÍTULO

II

MARCO TEÓRICO

2.1. Introducción

En este capítulo se presentan los conceptos fundamentales relacionados con el desarrollo de la plataforma web “Guía de rutas para usuarios de minibuses en El Alto aplicando chatbot”. Se abordan conceptos teóricos sobre trazado de rutas, plataformas web y el uso de chatbots como herramienta de asistencia al usuario.

2.2 Sistema de Información

Un **sistema de información** es un conjunto organizado de componentes interrelacionados que trabajan en conjunto para recolectar, procesar, almacenar y distribuir datos con el propósito de apoyar la toma de decisiones, coordinar actividades y controlar procesos dentro de una organización. Este sistema puede estar compuesto por personas, procedimientos, datos, hardware y software, y su objetivo principal es proporcionar información precisa y oportuna para respaldar las operaciones y la gestión organizacional.

Según Laudon y Laudon (2012), un sistema de información se define como "un conjunto de componentes interrelacionados que recolectan, procesan, almacenan y distribuyen información para apoyar la toma de decisiones, coordinación, control, análisis y visualización en una organización" (p. 4).

En el contexto de la **Plataforma Web con Chatbot Guía de Rutas para Transeúntes de Minibuses en la Ciudad de El Alto**, este sistema de información permite gestionar y procesar datos relacionados con las rutas de minibuses, horarios, ubicaciones y consultas de los usuarios. La implementación de un sistema de

información eficaz en este proyecto facilita la toma de decisiones informadas por parte de los usuarios y optimiza la gestión del transporte urbano en la ciudad.

2.3. Transporte urbano de minibuses

2.3.1. Transporte Urbano

Cervero (2000) señala que el transporte urbano cumple una función esencial en la movilidad de las ciudades, permitiendo el traslado eficiente de personas y mercancías. En contextos urbanos de países en desarrollo, el transporte informal, como los minibuses, emerge como una alternativa que cubre las deficiencias del sistema formal, brindando opciones más flexibles y adaptadas a las necesidades de la población local.

Actualmente, el transporte urbano se considera un componente clave para el funcionamiento dinámico de las ciudades. En especial, los medios informales como los minibuses cumplen un rol fundamental al complementar o reemplazar al transporte público tradicional, ofreciendo una respuesta práctica a los desafíos de accesibilidad y cobertura en áreas donde los servicios convencionales no llegan.

2.3.2. Minibús

Cervero (2000) describe al minibús como un medio de transporte público empleado principalmente en áreas urbanas y suburbanas para movilizar a una cantidad moderada de pasajeros. Este tipo de transporte opera comúnmente en rutas no estructuradas, con flexibilidad en sus paradas, lo cual le permite ajustarse a las demandas de movilidad de sectores urbanos diversos, en especial en ciudades con

rápido crecimiento poblacional.

En contextos urbanos como el de la ciudad de El Alto, el minibús se ha convertido en un elemento clave del sistema de transporte, ya que cubre áreas donde el transporte tradicional no llega. Gracias a su capacidad de adaptación y su servicio flexible, representa una solución eficaz frente a los desafíos de movilidad en ciudades con expansión acelerada y necesidades cambiantes.

2.3.3 Rutas de transporte público

De acuerdo con Vuchic (2007), las rutas de transporte público constituyen la estructura básica de los sistemas de movilidad, ya que definen los trayectos fijos o variables que los vehículos siguen para conectar distintos puntos de la ciudad. Una ruta determina no solo el origen y destino, sino también las paradas intermedias y la frecuencia del servicio, lo cual influye directamente en la accesibilidad, eficiencia y cobertura del sistema. En contextos urbanos con rápido crecimiento, como El Alto, la falta de información clara sobre las rutas genera dificultades para los usuarios, quienes deben recurrir a prueba y error o al conocimiento empírico para movilizarse. Por ello, digitalizar y organizar esta información es clave para mejorar la experiencia de viaje y optimizar el uso del transporte.

2.3.4. Ciudad de El Alto

Según el Centro de Promoción de la Mujer Gregoria Apaza (2011), la ciudad de El Alto, situada en Bolivia, ha atravesado un acelerado proceso de crecimiento urbano desde su creación en 1985. Este desarrollo ha implicado importantes desafíos en cuanto a la planificación urbana, la infraestructura y la provisión de servicios públicos.

El Alto es una de las ciudades más dinámicas de Bolivia, caracterizada por su crecimiento acelerado y la constante expansión de sus zonas urbanas. Este desarrollo ha generado diversas problemáticas relacionadas con la organización del territorio, el acceso a servicios básicos y la adecuada estructuración del transporte y la vivienda.

2.4. Inteligencia Artificial

2.4.1 ChatBot

Shevat (2017) define a los chatbots como programas informáticos diseñados para simular conversaciones humanas, facilitando interacciones eficientes entre los usuarios y los sistemas digitales. Estos asistentes virtuales pueden funcionar mediante reglas preestablecidas o inteligencia artificial, permitiendo interpretar y responder a las consultas de los usuarios para optimizar su experiencia en plataformas digitales.

En la actualidad, los chatbots se han convertido en una herramienta clave en la atención automatizada, al permitir una comunicación fluida entre los usuarios y los sistemas sin intervención humana directa. Su integración en plataformas web mejora significativamente la accesibilidad, el soporte y la personalización del servicio.

2.4.2 Agentes inteligentes

Según Corvalán y Sánchez Caparrós (2025), los agentes inteligentes son sistemas o programas de inteligencia artificial que tienen capacidad de percibir su entorno, procesar dicha información y tomar decisiones autónomas, adaptándose a distintos escenarios. Estos agentes no sólo responden a reglas predefinidas, sino que

pueden incorporar aprendizaje, adaptabilidad y capacidades de reacción frente a cambios en el entorno, lo que los hace más flexibles que los chatbots tradicionales.

En el contexto de una plataforma como la que desarrollo, un agente inteligente podría ayudar no sólo a responder consultas mediante diálogo, sino anticipar rutas útiles, sugerir transbordos eficientes, o adaptarse a horarios dinámicos para dar recomendaciones más oportunas.

2.5 Arquitectura de software

2.5.1 Concepto de arquitectura de software

Bass, Clements y Kazman (2013) definen la arquitectura de software como la estructura de un sistema compuesta por componentes de software, sus relaciones y las propiedades que estos poseen. Según los autores, la arquitectura establece un marco que guía tanto el diseño como la evolución del sistema, asegurando que cumpla con los requisitos funcionales y no funcionales.

En el caso del presente proyecto, la arquitectura de software constituye la base sobre la cual se organiza la plataforma de rutas de minibuses. Gracias a esta estructura, es posible definir cómo interactúan los distintos módulos, garantizar que el sistema sea escalable y asegurar que su desarrollo se realice de manera ordenada, facilitando futuras actualizaciones o ampliaciones.

2.5.2 Arquitectura cliente-servidor

Tanenbaum y Van Steen (2007) señalan que la arquitectura cliente-servidor es uno de los modelos más utilizados en sistemas distribuidos, caracterizándose por la interacción entre un cliente, que realiza solicitudes, y un servidor, que procesa dichas solicitudes y devuelve los resultados. Este enfoque permite una clara separación de responsabilidades, lo que lo convierte en un estándar en el desarrollo de aplicaciones web.

Aplicada al proyecto, la arquitectura cliente-servidor facilita la comunicación entre el navegador del usuario (cliente), que muestra las interfaces en HTML, CSS y JavaScript, y el servidor en Node.js, encargado de la lógica de negocio y la conexión con la base de datos MariaDB. Gracias a este modelo, los usuarios pueden consultar rutas en tiempo real de manera eficiente, mientras que el servidor gestiona los datos de forma centralizada y segura.

2.5.3 Arquitectura web modular

Shaw y Garlan (1996) explican que la modularidad en software consiste en dividir un sistema en componentes independientes que cumplen funciones específicas. Esta separación reduce la complejidad, facilita el mantenimiento y permite que el sistema pueda evolucionar incorporando nuevas funcionalidades sin alterar las ya existentes.

En el contexto de la plataforma, la arquitectura web modular permite organizar el backend en módulos especializados, como la consulta de rutas, el chatbot, la gestión de usuarios o las notificaciones. Esta organización asegura que el sistema sea flexible y escalable: por ejemplo, se podrá añadir un nuevo módulo de seguimiento en tiempo

real sin necesidad de modificar el resto de los componentes. De este modo, se garantiza un desarrollo más ordenado y sostenible en el tiempo.

2.6 Plataforma Web

Según (White 2015, p. 215) define la plataforma web como un entorno digital que permite a los usuarios realizar diversas tareas a través de navegadores, sin la necesidad de instalar aplicaciones en sus dispositivos. Estas plataformas se apoyan en tecnologías como HTML, CSS y JavaScript para facilitar la interacción, el procesamiento de datos y la presentación eficiente de información en línea.

En la actualidad, las plataformas web son fundamentales para el desarrollo de aplicaciones accesibles desde cualquier lugar con conexión a internet, optimizando procesos tanto en el ámbito empresarial como en el personal.

2.7. Metodología Scrum

Scrum es un marco de trabajo ágil que permite gestionar proyectos de manera eficiente mediante iteraciones cortas llamadas sprints. Este enfoque promueve la **transparencia, la adaptación continua y la colaboración** entre todos los miembros del equipo (Sutherland, 2014, p. 45). Al asignar roles específicos como **Product Owner, Scrum Master y Equipo de Desarrollo**, Scrum garantiza una organización clara del trabajo y mejora la capacidad de respuesta ante cambios en los requerimientos del proyecto. Gracias a estas características, Scrum resulta especialmente adecuado para el desarrollo de la plataforma web guía de rutas para

usuarios de minibuses, donde las necesidades y prioridades pueden evolucionar con el tiempo.

2.8 Herramientas de Desarrollo y Lenguajes

El desarrollo de sistemas web modernos requiere una combinación de herramientas y lenguajes que permitan construir aplicaciones eficientes, escalables e interactivas. Cada componente cumple un rol específico dentro de la arquitectura del sistema y asegura que la plataforma funcione correctamente para los usuarios finales.

2.8.1 HTML y CSS

HTML (Hypertext Markup Language) define la estructura de las páginas web, mientras que CSS (Cascading Style Sheets) permite controlar su presentación visual, incluyendo colores, tipografías y disposición de elementos. Según Duckett (2011), el uso conjunto de HTML y CSS facilita la creación de interfaces accesibles y responsivas, adaptables a diferentes dispositivos y tamaños de pantalla, lo que es fundamental para plataformas de transporte que requieren un acceso rápido y claro a la información.

2.8.2 JavaScript

JavaScript es un lenguaje de programación interpretado que permite agregar interactividad a las páginas web, como la actualización dinámica de contenidos, validación de formularios y manejo de eventos del usuario (Flanagan, 2011). En aplicaciones de transporte, su uso es clave para interactuar con mapas, mostrar rutas en tiempo real y responder de manera inmediata a las consultas del usuario.

2.8.3 Node.js y Express.js

Node.js permite ejecutar JavaScript en el servidor, habilitando el manejo de solicitudes concurrentes y la conexión con bases de datos. Express.js, como framework de Node.js, facilita la creación de APIs y la gestión de rutas HTTP (Gascón, 2024). Este conjunto permite estructurar el backend de forma eficiente y escalable, algo crucial para aplicaciones que reciben múltiples consultas simultáneas de rutas y horarios.

2.8.4 Bases de datos (MariaDB)

Las bases de datos son el núcleo de almacenamiento de cualquier sistema de información. **MariaDB**, como sistema de gestión de bases de datos relacional (RDBMS), permite almacenar información estructurada de manera segura y consistente. Su modelo relacional facilita la organización de datos en tablas relacionadas, garantizando integridad y evitando redundancias. Además, MariaDB soporta consultas complejas y transacciones, lo que lo hace ideal para aplicaciones que requieren manejar rutas, horarios y datos de usuarios de manera confiable (Widenius, Axmark & DuBois, 2002).

2.8.5 Mapas y visualización geográfica (OpenStreetMap, MapLibre GL JS, Leaflet)

La integración de mapas interactivos permite a los usuarios visualizar rutas y ubicaciones en tiempo real. OpenStreetMap proporciona datos cartográficos abiertos y editables (Ramm, 2010), mientras que librerías como MapLibre GL JS y Leaflet

permiten renderizar esos datos en el navegador y añadir funcionalidades como trazado de rutas, marcadores y capas personalizadas.

CAPÍTULO

III

MARCO PRÁCTICO

3.1 Introducción

En este capítulo se presenta el marco práctico del proyecto titulado “**Plataforma web con chatbot guía de rutas para transeúntes de minibuses en la ciudad de El Alto**”, el cual describe el proceso metodológico que orienta la investigación y el desarrollo del sistema. El objetivo de esta sección es establecer el camino a seguir para aplicar los conceptos teóricos previamente estudiados, garantizando que la solución propuesta se construya de manera estructurada, científica y alineada con los objetivos planteados.

Asimismo, el marco práctico permite organizar la recolección de información, el análisis de datos y la aplicación de herramientas tecnológicas necesarias para el diseño e implementación de la plataforma. De esta forma, se busca ofrecer una solución funcional que facilite a los usuarios encontrar rutas de transporte urbano de manera rápida, precisa y accesible.

3.2 Tipo de Investigación

La presente investigación adopta un **enfoque mixto**, al integrar elementos **cuantitativos y cualitativos** en el análisis y desarrollo de la plataforma web.

El enfoque **cuantitativo** se refleja en el manejo de datos numéricos dentro de la base de datos, tales como los costos de pasaje, la cantidad de minibuses por ruta, los horarios de operación, la frecuencia de uso y las coordenadas geográficas de las paradas y trazados. Estos datos permiten realizar mediciones objetivas sobre el comportamiento del transporte urbano y evaluar el funcionamiento de la plataforma.

Por otro lado, el enfoque **cualitativo** se manifiesta en la organización y descripción de información no numérica, como los nombres de rutas, sindicatos, lugares de referencia y observaciones sobre el servicio, lo que contribuye a contextualizar y caracterizar el sistema de transporte en la ciudad de El Alto.

Asimismo, la investigación es de tipo **descriptivo y aplicado**, ya que busca detallar las características del transporte público urbano y desarrollar una herramienta tecnológica que facilite el acceso a la información de rutas de minibuses, sin pretender generar nuevas teorías, sino ofrecer una solución práctica a una necesidad social.

3.3 Diseño de la Investigación

El diseño de la investigación es **no experimental y transversal**.

- Es **no experimental** porque no se manipulan deliberadamente las variables de estudio, sino que se observan y analizan los fenómenos tal como ocurren en su entorno natural, especialmente en el contexto del transporte público urbano en la ciudad de El Alto.
- Es **transversal** porque la recolección de información se realizará en un único periodo de tiempo, permitiendo obtener una visión general del comportamiento de los usuarios, las rutas de minibuses y el funcionamiento inicial de la plataforma durante una etapa específica del proyecto.

3.4 Métodos, Técnicas e Instrumentos

Para el desarrollo de esta investigación se emplean métodos y herramientas que permiten obtener y analizar la información necesaria para diseñar y construir la plataforma web guía de rutas de minibuses en la ciudad de El Alto.

Método:

Se utiliza el **método deductivo**, el cual parte de conocimientos generales sobre movilidad urbana, sistemas web y chatbots, para aplicarlos en la creación de una solución práctica adaptada a la realidad local. Este enfoque permite trasladar conceptos teóricos a un entorno real, con el objetivo de desarrollar una plataforma que facilite la búsqueda y planificación de rutas de transporte público.

Técnicas:

Las principales técnicas empleadas en la investigación son:

- **Encuestas:** orientadas a conocer las necesidades, costumbres y dificultades de los usuarios al buscar rutas de minibuses.
- **Observación directa:** realizada en diferentes puntos de la ciudad de El Alto para identificar el flujo de minibuses, los lugares de transbordo y los problemas más comunes que enfrentan los transeúntes.
- **Análisis de datos:** utilizado para interpretar la información recopilada sobre el uso de la plataforma, como la cantidad de consultas, los horarios de mayor actividad y las rutas más solicitadas.

Instrumentos:

Para la aplicación de las técnicas mencionadas se emplean los siguientes instrumentos:

- **Cuestionarios y formularios digitales** para recopilar las respuestas de los usuarios durante las encuestas.
- **Hojas de observación** para registrar la información obtenida en campo sobre el comportamiento del transporte y los usuarios.
- **Registros automáticos del sistema**, que almacenan datos estadísticos del uso de la plataforma, permitiendo evaluar su funcionamiento y aceptación.

En conjunto, estos métodos, técnicas e instrumentos facilitan la obtención de información confiable y útil, permitiendo validar el diseño y funcionamiento de la plataforma web de rutas de minibuses propuesta para la ciudad de El Alto.

3.5. Requerimientos funcionales

Funcionalidad: Consulta de rutas

Descripción: Permitir que los usuarios ingresen su ubicación y destino para ver rutas disponibles.

Prioridad: Alta.

Acciones iniciadoras y comportamiento esperado: El usuario ingresa ubicación actual y destino, el sistema procesa la solicitud, devuelve rutas posibles con detalles.

Requerimientos:

- RF1: El sistema debe permitir ingresar la ubicación de inicio y destino.

- RF2: El sistema debe calcular rutas posibles.
- RF3: El sistema debe mostrar puntos de bajada cuando corresponda tomar dos o mas rutas de minibuses.
- RF4: El sistema debe desplegar la ruta sobre un mapa interactivo.
- RF5: El sistema debe mostrar información detallada de cada ruta, incluyendo número de ruta, sindicato, horarios de inicio y fin, costos (regular, nocturno, extra), frecuencia, lugares de referencia, imágenes de inicio y destino

Condición de error: si no se encuentra ruta disponible, mostrar mensaje claro al usuario.

Funcionalidad: Gestión de rutas por administradores

Descripción: Crear, modificar o eliminar rutas dentro del sistema.

Prioridad: Alta.

Acciones iniciadoras y comportamiento esperado: El administrador inicia sesión, selecciona módulo de rutas, puede agregar, modificar o dar de baja rutas.

Requerimientos:

- RF6: Solo usuarios con rol editor/administrador pueden crear rutas.
- RF7: El sistema debe permitir trazar rutas sobre un mapa en la UI.
- RF8: El sistema debe permitir modificar datos de rutas.
- RF9: La eliminación debe ser lógica, no física.
- RF10: El sistema debe registrar todas las acciones realizadas sobre rutas (logs).

Funcionalidad: Gestión de roles y usuarios

Descripción: Permitir administración de usuarios con diferentes permisos (usuario, editor, administrador).

Prioridad: Media.

Acciones iniciadoras y comportamiento esperado: El administrador accede al panel de usuarios, asigna roles y permisos, el sistema actualiza privilegios.

Requerimientos:

- RF11: El sistema debe permitir registrar usuarios y diferentes roles.
- RF12: El sistema debe permitir asignar y revocar permisos.
- RF13: El sistema debe permitir al administrador consultar logs de acciones.

Condición de error: si un usuario intenta acceder a un módulo sin permisos, el sistema debe denegar acceso con mensaje.

Funcionalidad: Chatbot informativo

Descripción: Proporcionar asistencia automática sobre rutas, horarios, precios y calles por donde pasa una línea.

Prioridad: Alta.

Acciones iniciadoras y comportamiento esperado: El usuario abre el chatbot, escribe consulta, el sistema responde en lenguaje natural con información obtenida de la base de datos.

Requerimientos:

- RF14: El sistema debe permitir consultas sobre rutas por texto.
- RF15: El sistema debe responder con horarios, costos de pasaje y calles.
- RF16: El sistema debe responder en caso de consultas inválidas (“No entendí tu pregunta, intenta de nuevo”).
- RF17: El chatbot debe estar disponible desde cualquier pantalla de la plataforma.

Funcionalidad: Notificaciones y anuncios temporales

Descripción: Informar a los usuarios sobre cambios en rutas o disponibilidad del servicio.

Prioridad: Alta.

Acciones iniciadoras y comportamiento esperado: El administrador crea un anuncio temporal asociado a una ruta, los usuarios lo ven al consultar esa ruta.

Requerimientos:

- RF18: El sistema debe permitir crear anuncios con fecha de inicio y fin.
- RF19: Los anuncios deben mostrarse en la página de la ruta afectada.
- RF20: El sistema debe permitir a los administradores gestionar (crear, editar, eliminar) anuncios.

Condición de error: si se ingresa un anuncio sin fechas, el sistema debe rechazar el registro.

Funcionalidad: Comentarios en rutas

Descripción: Permitir que los usuarios registrados dejen comentarios en las rutas disponibles.

Prioridad: Media.

Acciones iniciadoras y comportamiento esperado: El usuario inicia sesión → accede a la sección de comentarios de una ruta → escribe su mensaje → el sistema lo valida y lo publica.

Requerimientos:

- RF21: El sistema debe permitir a los usuarios autenticados publicar comentarios en las rutas.
- RF22: El sistema debe validar que el comentario no esté vacío ni contenga caracteres inválidos.
- RF23: El sistema debe permitir al usuario editar o eliminar sus propios comentarios.
- RF24: El sistema debe mostrar los comentarios asociados a cada ruta comenzando con el mas reciente.
- RF25: El sistema debe permitir a los administradores eliminar comentarios inapropiados.

Condición de error: si un usuario no ha iniciado sesión e intenta comentar, el sistema debe mostrar un mensaje indicando que debe autenticarse primero.

3.6. Requerimientos No Funcionales

Rendimiento

- RNF1: El sistema debe responder a las consultas de rutas en un tiempo máximo de **3 segundos** bajo carga normal.
- RNF2: El sistema debe soportar al menos **100 usuarios concurrentes** en la primera versión.

Disponibilidad

- RNF3: La plataforma debe estar disponible al menos el **95% del tiempo** durante el horario de servicio (6:00 a 23:00).
- RNF4: El chatbot debe estar accesible en todas las pantallas del sistema.

Seguridad

- RNF5: Toda comunicación entre cliente y servidor debe realizarse bajo el protocolo **HTTPS** con cifrado SSL.
- RNF6: Las contraseñas de los usuarios deben almacenarse encriptadas (ejemplo: usando bcrypt).
- RNF7: El sistema debe registrar en logs todas las acciones administrativas (alta, baja, modificación de rutas o usuarios).

Usabilidad

- RNF8: La interfaz gráfica debe ser **responsiva**, adaptándose a dispositivos móviles y computadoras.

- RNF9: Todos los botones principales deben estar claramente identificados con texto e íconos.
- RNF10: El sistema debe mostrar mensajes de error claros y comprensibles para el usuario.

Mantenibilidad

- RNF11: El sistema debe estar desarrollado bajo una arquitectura modular que permita añadir nuevas funcionalidades sin afectar las existentes.
- RNF12: El código debe documentarse siguiendo buenas prácticas de estilo en Node.js y SQL.

3.7 Diseño de la base de datos

La base de datos del sistema “EasyWay” fue diseñada bajo un modelo relacional, con el objetivo de garantizar integridad, consistencia y fácil acceso a la información.

Su función principal es almacenar los datos relacionados con las rutas de minibuses, usuarios, paradas, trazados, comentarios, destinos turísticos y anuncios.

3.7.1 Estructura general

La base de datos se compone de las siguientes tablas principales:

- **roles:** Contiene los tipos de roles del sistema (por ejemplo, administrador).
Atributos principales: id, nombre.

- **usuarios:** Registra la información de los administradores o encargados de agregar rutas.

Atributos principales: id, nombre, email, password_hash, rol_id.

- **rutas:** Almacena la información completa de cada ruta registrada.

Atributos principales: id, número de ruta, sindicato, costos (regular, nocturno, extra), frecuencia, horarios, cantidad de minibuses, lugares de referencia, imágenes, id_usuario.

- **paradas:** Guarda las paradas pertenecientes a cada ruta, con su ubicación geográfica.

Atributos principales: id, id_ruta, nombre, latitud, longitud, orden.

- **trazado_ruta:** Representa el recorrido exacto de cada ruta mediante coordenadas.

Atributos principales: id, id_ruta, latitud, longitud, orden.

- **comentarios:** Permite registrar opiniones o valoraciones sobre una ruta.

Atributos principales: id, id_ruta, id_usuario, comentario, fecha.

- **destinos:** Contiene información sobre puntos turísticos de la ciudad.

Atributos principales: id, nombre, descripción, latitud, longitud, imagen, id_usuario.

- **anuncios:** Registra mensajes o avisos temporales vinculados a una ruta.

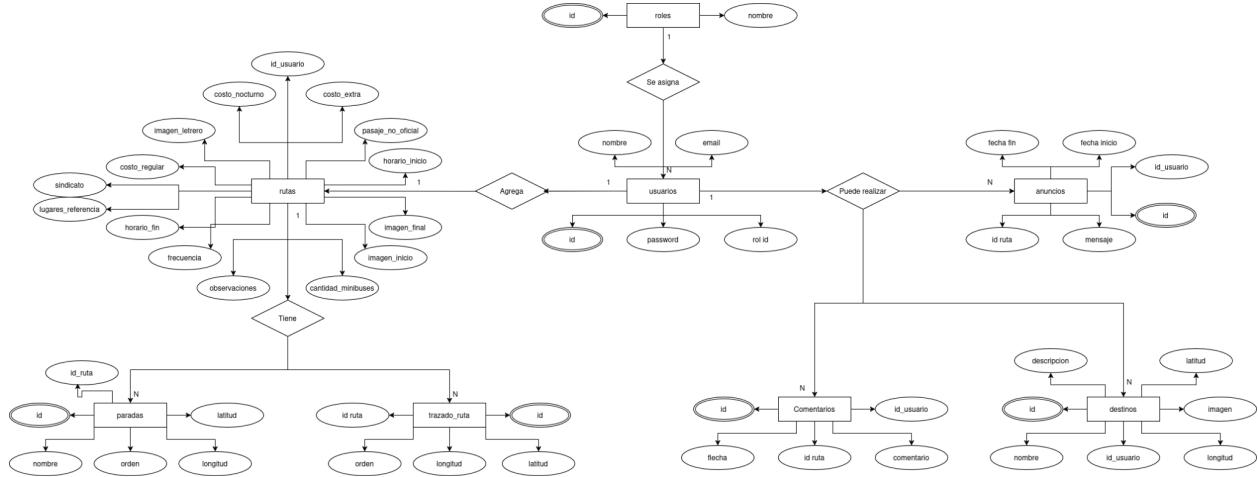
Atributos principales: id, id_ruta, mensaje, fecha_inicio, fecha_fin, id_usuario.

3.7.2 Relaciones principales

- Un **usuario** puede registrar varias **rutas**.
- Cada **ruta** tiene múltiples **paradas** y puntos de **trazado**.

- Los **comentarios** están asociados a una **ruta** y un **usuario**.
- Los **destinos** y **anuncios** también están vinculados a los **usuarios**.

3.7.3 Modelo Entidad–Relación



3.8 SCRUM

3.8.1 Product Backlog

ÉPICA 1: Investigación y Definición del Problema

Prioridad: Alta | Story Points: 13

Historia 1.1: Análisis de necesidades del usuario

- **Como:** Product Owner
- **Quiero:** Definir requerimientos de los usuarios sobre rutas de minibuses
- **Para que:** La plataforma entregue información útil y clara
- **Criterios de Aceptación:**
 - Entrevistas a 50+ usuarios sobre hábitos de transporte

- Identificación de principales problemas en movilidad
 - Documento de requerimientos aprobado
 - **Tareas técnicas:**
 - Encuestas y entrevistas
 - Análisis de datos obtenidos
 - Documento de requerimientos
- Story Points: 5 | Sprint: 1**

Historia 1.2: Investigación de soluciones técnicas existentes

- **Como:** Backend / GIS Developer
 - **Quiero:** Analizar APIs y frameworks de mapas, rutas y geolocalización
 - **Para que:** Seleccionar la mejor solución técnica
 - **Criterios de Aceptación:**
 - Comparativa de OpenStreetMap, Google Maps API, Mapbox
 - Identificación de librerías JS para rutas y geolocalización
 - Documento técnico con recomendación
 - **Tareas técnicas:**
 - Pruebas de APIs
 - Análisis de escalabilidad
 - Presentación al equipo
- Story Points: 8 | Sprint: 1**

ÉPICA 2: Preparación de Datos y Autenticación

Prioridad: Alta | **Story Points:** 21

Historia 2.1: Recolección de rutas de minibuses

- **Como:** GIS / Backend Developer
- **Quiero:** Obtener información de rutas, paradas y horarios
- **Para:** Mostrar datos confiables en la plataforma
- **Criterios de Aceptación:**
 - 100% de rutas mapeadas en El Alto
 - Información de paradas, horarios y recorridos georreferenciados
 - Datos validados con transportistas o fuentes oficiales
- **Tareas técnicas:**
 - Scraping o recolección de datasets abiertos
 - Normalización de datos
 - Almacenamiento en base de datos

Story Points: 8 | **Sprint:** 1-2

Historia 2.2: Limpieza y estructuración de datos

- **Como:** Backend Developer
- **Quiero:** Estandarizar rutas y horarios
- **Para:** Garantizar precisión y consistencia
- **Criterios de Aceptación:**
 - Datos estructurados en JSON/SQL
 - Eliminación de inconsistencias y duplicados
 - Validación cruzada de rutas
- **Tareas técnicas:**

- Scripts de limpieza de datos
- Validación con mapas interactivos
- Documentación del dataset final

Story Points: 13 | **Sprint:** 2

Historia 2.3: Sistema de login y roles de usuario

- **Como:** Backend Developer
- **Quiero:** Implementar login y control de roles (usuario, admin, operador)
- **Para:** Restringir accesos y funciones según tipo de usuario
- **Criterios de Aceptación:**
 - Login funcional con email/contraseña
 - Roles: Usuario final, Administrador, Operador
 - Validación de accesos en todas las secciones
- **Tareas técnicas:**
 - Implementación de JWT / sesiones
 - Integración con base de datos de usuarios
 - Pruebas unitarias de roles

Story Points: 8 | **Sprint:** 2

ÉPICA 3: Desarrollo del Pipeline Web y Chatbot

Prioridad: Alta | **Story Points:** 42

Historia 3.1: Configuración de infraestructura

- **Como:** Backend Developer

- **Quiero:** Configurar servidor y base de datos
 - **Para:** Tener un entorno estable de desarrollo y pruebas
 - **Criterios de Aceptación:**
 - Servidor Node.js funcionando
 - Base de datos MySQL/PostgreSQL configurada
 - Sistema de control de versiones activo
 - **Tareas técnicas:**
 - Deployment de servidor local y cloud
 - Configuración de repositorios
- Story Points: 8 | Sprint: 2**

Historia 3.2: Desarrollo del frontend y mapas interactivos avanzados

- **Como:** Frontend Developer
- **Quiero:** Implementar interfaz web con mapas de rutas
- **Para:** Que los usuarios consulten rutas fácilmente
- **Criterios de Aceptación:**
 - Visualización de rutas en mapa (OpenStreetMap)
 - Filtrado por tipo de minibús, horarios y rutas rápidas
 - Zoom dinámico y detalles de cada parada
 - Responsive para móviles y desktops
- **Tareas técnicas:**
 - Implementación con HTML, CSS, JS
 - Integración de API de mapas

- Validación UI/UX

Story Points: 8 | Sprint: 3

Historia 3.3: Desarrollo del chatbot guía de rutas

- **Como:** Product Owner
- **Quiero:** Implementar asistente conversacional
- **Para:** Ayudar a los usuarios a planificar rutas
- **Criterios de Aceptación:**
 - Chatbot responde preguntas sobre rutas y horarios
 - Integración con base de datos de rutas
 - Recomendaciones contextuales y personalizadas
- **Tareas técnicas:**
 - Integración con Node.js
 - Desarrollo de intents y respuestas
 - Test de interacción

Story Points: 8 | Sprint: 3-4

Historia 3.4: Integración frontend-backend-chatbot

- **Como:** Dev Team
- **Quiero:** Integrar todos los módulos del sistema
- **Para:** Tener la plataforma lista para pruebas
- **Criterios de Aceptación:**
 - Interfaz funcional con mapas y chatbot
 - Respuestas en tiempo real

- Manejo de errores y logs
- **Tareas técnicas:**
 - Integración de API REST con frontend
 - Pruebas unitarias e integración
 - Documentación técnica

Story Points: 10 | Sprint: 4

Historia 3.5: Sección de ingreso y administración de rutas

- **Como:** Product Owner
- **Quiero:** Crear, editar, eliminar rutas con información detallada
- **Para:** Mantener la base de datos de rutas actualizada
- **Criterios de Aceptación:**
 - CRUD completo de rutas
 - Subida de imágenes y horarios
 - Validación de datos obligatorios
- **Tareas técnicas:**
 - Formularios de administración
 - Backend para CRUD de rutas
 - Validación y pruebas

Story Points: 8 | Sprint: 4

Historia 3.6: Sección de notificaciones

- **Como:** Product Owner
- **Quiero:** Configurar notificaciones activas para usuarios

- **Para:** Informar rutas no disponibles o avisos importantes

- **Criterios de Aceptación:**

- Crear, editar, eliminar notificaciones
- Definir tiempo de vigencia de notificación
- Mostrar alertas en plataforma usuario

- **Tareas técnicas:**

- Implementación en frontend y backend
- Pruebas de visualización
- Documentación

Story Points: 5 | Sprint: 4

Historia 3.7: Sección de feedback de usuarios

- **Como:** Usuario final

- **Quiero:** Reportar problemas, errores o sugerencias sobre rutas

- **Para:** Mejorar información y servicio

- **Criterios de Aceptación:**

- Formulario de feedback funcional
- Envío de reportes a administradores
- Registro en base de datos

- **Tareas técnicas:**

- Desarrollo de formulario y backend
- Almacenamiento y notificación a admins

Story Points: 5 | Sprint: 5

Historia 3.8: Analytics / dashboard administrativo

- **Como:** Product Owner
 - **Quiero:** Visualizar métricas de uso, rutas más consultadas y efectividad de notificaciones
 - **Para:** Tomar decisiones sobre rutas y sistema
 - **Criterios de Aceptación:**
 - Contador mensual de rutas más consultadas
 - Panel de métricas de uso del chatbot
 - Visualización de notificaciones efectivas
 - **Tareas técnicas:**
 - Logging de eventos
 - Integración con dashboard tipo Grafana
 - Documentación de métricas
- Story Points: 8 | Sprint: 6**

Historia 3.9: Módulo de mantenimiento del sistema

- **Como:** Product Owner
- **Quiero:** Gestionar backups y logs de actividad
- **Para:** Mantener estabilidad y registro de la plataforma
- **Criterios de Aceptación:**
 - Backup automático de rutas y base de datos
 - Registro de actividad de usuarios y administradores
 - Restauración de datos validada

- **Tareas técnicas:**

- Scripts de backup y restore
- Logging y auditoría
- Documentación

Story Points: 5 | Sprint: 6

3.8.2 Planificación de sprints

SPRINT 1 – Research & Setup (2 semanas)

Historias de usuario:

- Historia 1.1: Análisis de necesidades del usuario
- Historia 1.2: Investigación de soluciones técnicas
- Historia 3.1: Configuración de infraestructura inicial (servidor, base de datos)

Objetivo:

Definir roadmap técnico, requerimientos de usuario y preparar un ambiente de desarrollo básico para comenzar módulos posteriores.

Requerimientos funcionales cubiertos:

- Ninguno directamente funcional aún.
- Este sprint prepara la base técnica y de análisis para implementar todos los RF posteriores.

SPRINT 2 – User & Admin Modules (2 semanas)

Historias de usuario:

- Historia 2.3: Sistema de login y roles de usuario
- Historia 3.5: Sección de administración de usuarios
- Historia 3.6: Sección de notificaciones

Objetivo:

Tener un sistema de gestión de usuarios y notificaciones funcionando, para que administradores puedan controlar accesos y comunicar alertas.

Requerimientos funcionales cubiertos:

- **RF11, RF12, RF13** → Gestión de roles y usuarios (registro, asignación, revocación, logs)
- **RF18, RF19, RF20** → Notificaciones y anuncios temporales (crear, mostrar, gestionar)

SPRINT 3 – Routes Management (2 semanas)**Historias de usuario:**

- Historia 3.5: Sección de ingreso y administración de rutas (CRUD completo)
- Historia 3.4: Integración inicial frontend-backend (sin mapas aún)

Objetivo:

Permitir que los administradores creen, editen y eliminen rutas, incluyendo

información de paradas, horarios e imágenes, preparando la plataforma para recibir rutas reales.

Requerimientos funcionales cubiertos:

- **RF6, RF7, RF8, RF9, RF10** → Gestión de rutas por administradores (roles de edición, trazado en mapa, modificación, eliminación lógica, logs)

SPRINT 4 – Web & Chatbot Development (2 semanas)

Historias de usuario:

- Historia 3.2: Desarrollo del frontend con mapas interactivos
- Historia 3.3: Desarrollo del chatbot guía de rutas

Objetivo:

Implementar interfaz web funcional con mapas interactivos y chatbot básico para guiar a los usuarios. La plataforma puede mostrar rutas de prueba y permitir interacción con el chatbot.

Requerimientos funcionales cubiertos:

- **RF1, RF2, RF3, RF4, RF5** → Consulta de rutas (ingreso de ubicación, cálculo de rutas, puntos de bajada, mapa interactivo, info detallada)
- **RF14, RF15, RF16, RF17** → Chatbot informativo (consultas de rutas, horarios, costos, calles, manejo de consultas inválidas, disponibilidad en toda la plataforma)

SPRINT 5 – Data Collection & Testing (2 semanas)

Historias de usuario:

- Historia 2.1: Recolección de rutas de minibuses
- Historia 2.2: Limpieza y estructuración de datos
- Historia 3.7: Sección de feedback de usuarios
- Historia 4.1: Testing funcional avanzado

Objetivo:

Recolectar y limpiar datos reales de rutas, validar funcionalidad de búsqueda de rutas, feedback y chatbot con usuarios reales.

Requerimientos funcionales cubiertos:

- **RF1, RF2, RF3, RF4, RF5** → Validación de consulta de rutas y puntos de bajada con datos reales
- **RF21, RF22, RF23, RF24, RF25** → Comentarios en rutas (publicación, validación, edición/eliminación, visualización, moderación por admin)

SPRINT 6 – Monitoring, Analytics & Documentation (2 semanas)

Historias de usuario:

- Historia 3.8: Analytics / dashboard administrativo
- Historia 3.9: Módulo de mantenimiento del sistema (backup y logs)
- Historia 5.1: Dashboard de monitoreo de actividad y errores
- Historia 5.2: Documentación técnica y de usuario

Objetivo:

Sistema monitoreado, con métricas, logs y backups funcionando; documentación completa lista para entrega y mantenimiento continuo.

Requerimientos funcionales cubiertos:

- Complementa todos los RF mediante monitoreo, trazabilidad y respaldo del sistema.
- Garantiza estabilidad, logs, backups y soporte para todas las funcionalidades desarrolladas previamente.

3.8.3 Cronograma

| Actividad / Tarea | Duración (Inicio – Fin) | Responsable | Sprint 1 30/09– 13/10 | Sprint 2 14/10– 27/10 | Sprint 3 28/10– 10/11 | Sprint 4 11/11– 24/11 | Sprint 5 25/11– 08/12 | Sprint 6 09/12– 15/12 |
|--|-------------------------|------------------|-----------------------------|-----------------------------|-----------------------------|-----------------------------|-----------------------------|-----------------------------|
| SPRINT 1 – Research & Setup | | | | | | | | |
| Reunión inicial y levantamiento de requerimientos (HU 1.1) | 30/09 – 03/10 | Líder + Analista | ✓ | | | | | |
| Análisis de necesidades del usuario | 04/10 – 06/10 | Analista | ✓ | | | | | |

| | | | | | | | | | |
|--|---------------|--------------------|---|---|--|--|--|--|--|
| Investigación de soluciones técnicas (HU 1.2) | 07/10 – 09/10 | Líder Técnico | ✓ | | | | | | |
| Configuración de infraestructura (HU 3.1) | 10/10 – 13/10 | DevOps | ✓ | | | | | | |
| SPRINT 2 – User & Admin Modules | | | | ⌚ | | | | | |
| Diseño e implementación del modelo de usuarios y roles (RF11–RF12) | 14/10 – 17/10 | Backend Dev | | ✓ | | | | | |
| Desarrollo del sistema de registro y login (HU 2.3) | 18/10 – 21/10 | Backend + Frontend | | ⌚ | | | | | |
| Integración de autenticación con base de datos | 22/10 – 23/10 | Backend Dev | | ⌚ | | | | | |
| Desarrollo de módulo de administración de usuarios (HU 3.5) | 24/10 – 26/10 | Backend Dev | | ⌚ | | | | | |
| Implementación del sistema de notificaciones (HU 3.6, RF18–RF20) | 25/10 – 27/10 | Backend + QA | | ⌚ | | | | | |

| SPRINT 3 – Routes Management | | | | | | | | |
|--|---------------|--------------------|--|--|---|---|--|--|
| Diseño del modelo de datos de rutas y paradas (RF6–RF7) | 28/10 – 31/10 | BD Admin | | | ✓ | | | |
| Implementación del CRUD de rutas (crear, editar, eliminar lógicamente) | 01/11 – 04/11 | Backend Dev | | | ✓ | | | |
| Integración inicial con el frontend (visualización básica sin mapa) | 05/11 – 07/11 | Full Stack | | | ✓ | | | |
| Registro de logs administrativos (RF10) | 08/11 – 10/11 | Backend Dev | | | ✓ | | | |
| SPRINT 4 – Web & Chatbot Development | | | | | | | | |
| Implementación del mapa interactivo (RF1–RF4) | 11/11 – 16/11 | Frontend Dev | | | | ✓ | | |
| Visualización detallada de rutas (RF5) | 17/11 – 19/11 | Backend + Frontend | | | | ✓ | | |

| | | | | | | | | | |
|--|---------------|----------------|--|--|--|--|---|---|--|
| Integración y desarrollo del chatbot informativo (RF14–RF17) | 20/11 – 24/11 | Backend Dev | | | | | ✓ | | |
| SPRINT 5 – Data Collection & Testing | | | | | | | | | |
| Recolección de rutas reales de minibuses (HU 2.1) | 25/11 – 28/11 | Data Team | | | | | ✓ | | |
| Limpieza y estructuración de datos (HU 2.2) | 29/11 – 30/11 | Data + Backend | | | | | ✓ | | |
| Implementación de comentarios en rutas (RF21–RF25) | 01/12 – 05/12 | Full Stack | | | | | ✓ | | |
| Testing funcional avanzado y corrección de errores | 06/12 – 08/12 | QA + Devs | | | | | ✓ | | |
| SPRINT 6 – Monitoring, Analytics & Documentation | | | | | | | | | |
| Implementación del dashboard administrativo y métricas | 09/12 – 11/12 | Backend Dev | | | | | | ✓ | |

| | | | | | | | | | |
|--|---------------|------------|--|--|--|--|--|--|---|
| Configuración de backups automáticos y logs globales | 12/12 – 13/12 | DevOps | | | | | | | ✓ |
| Redacción de documentación técnica y manual de usuario | 14/12 – 15/12 | Líder + QA | | | | | | | ✓ |

- Anexos

Figura 1

Diagrama de situación actual

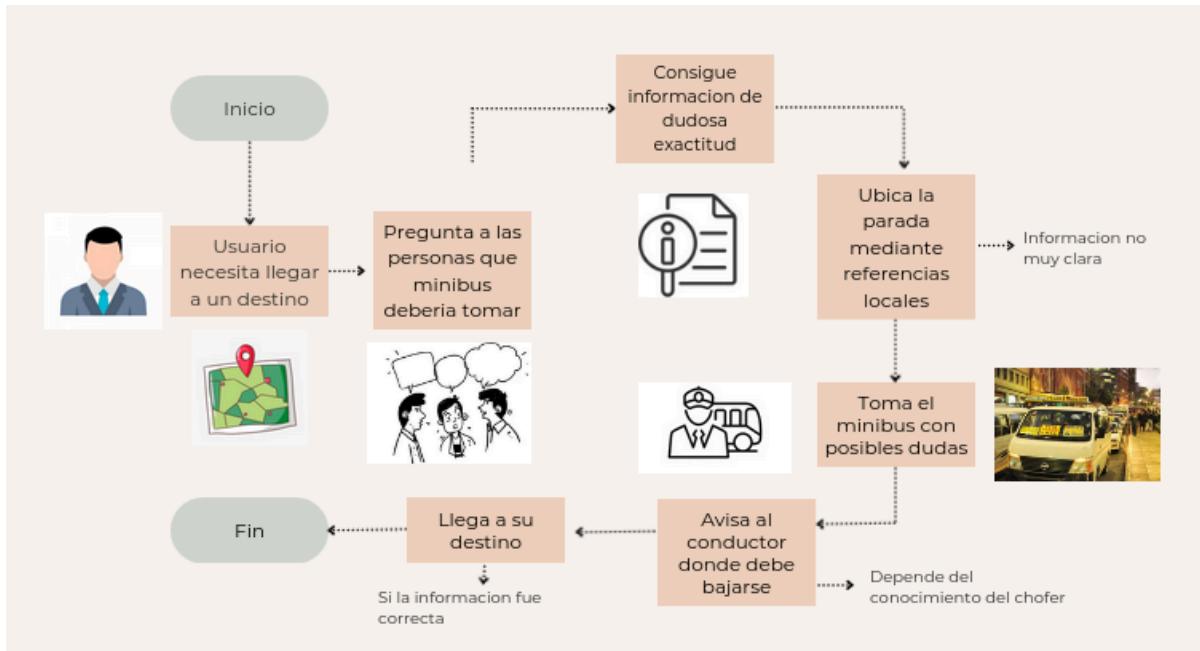


Figura 2

Diagrama de contexto

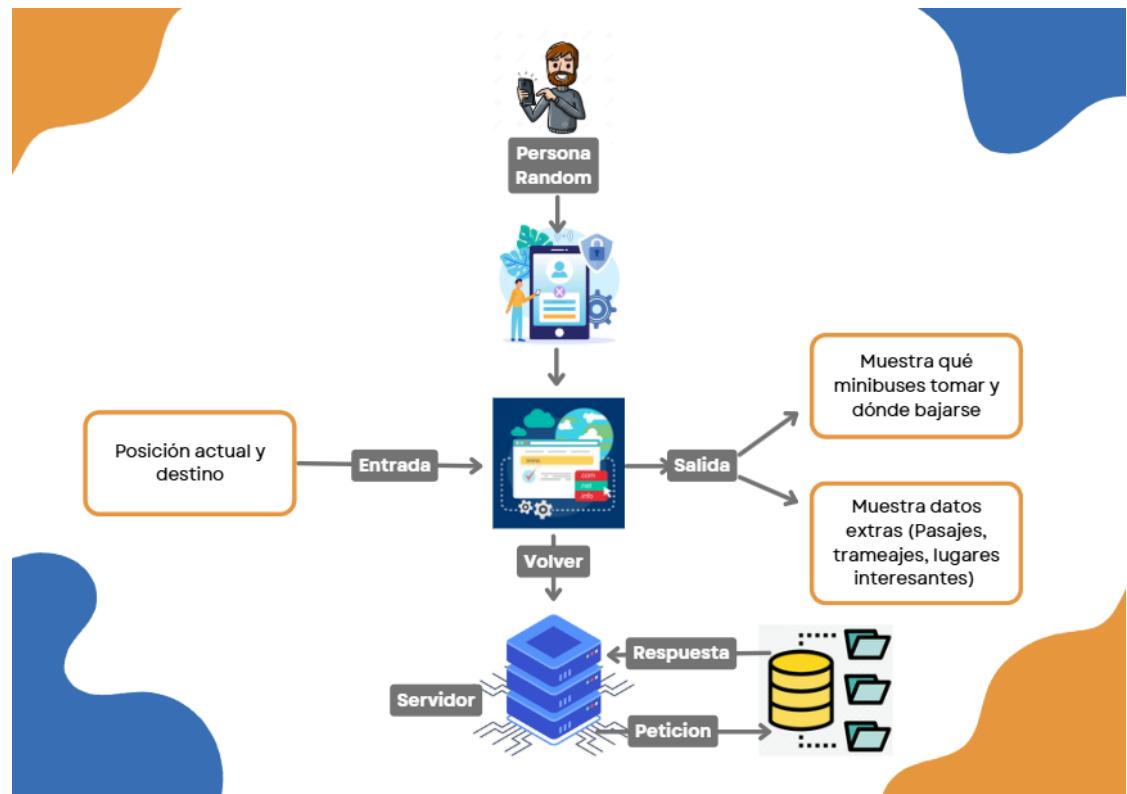
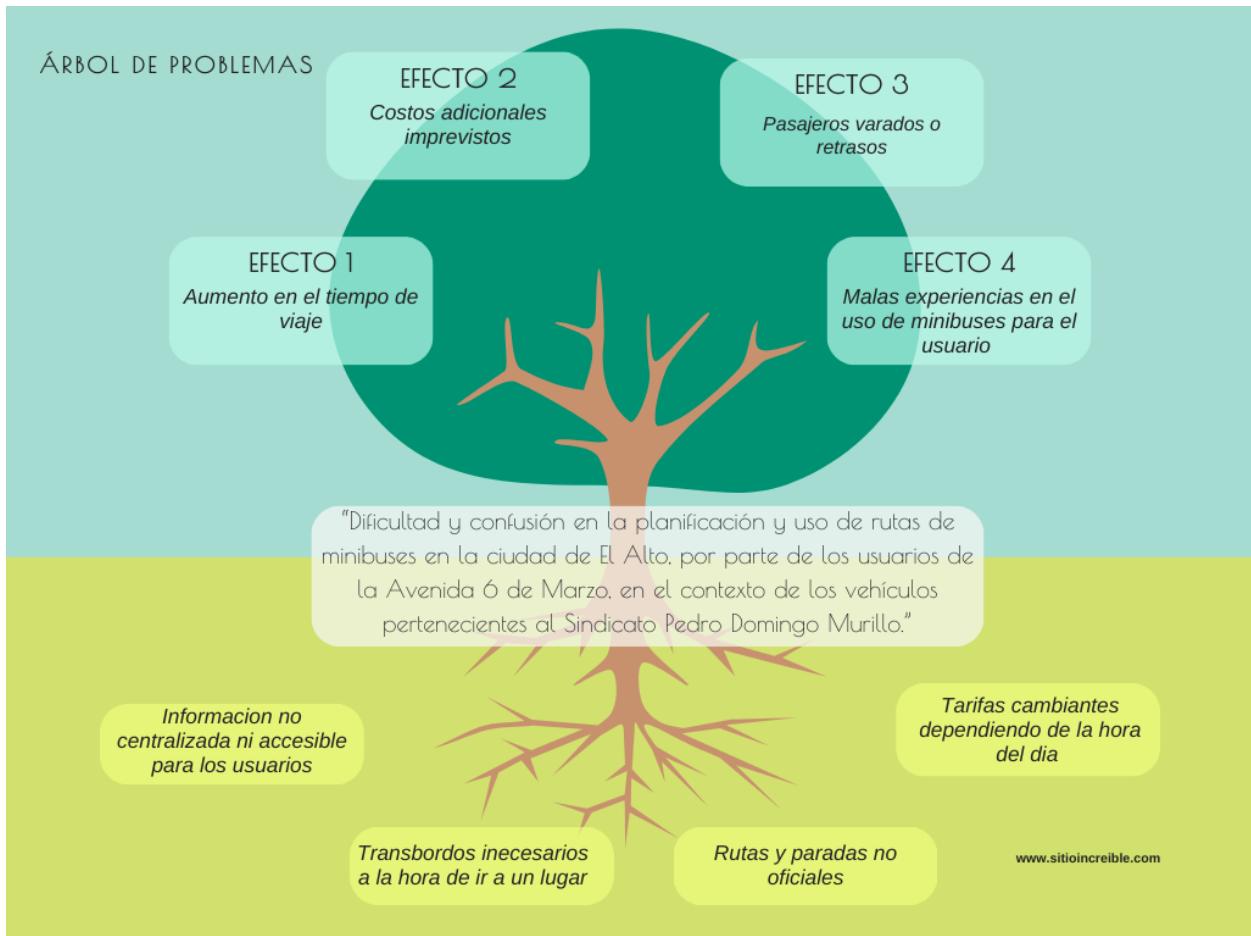


Figura 3**Árbol de Problemas**

Negativo: Dificultad y confusión

De qué: planificación y uso de rutas de minibuses

Dónde: ciudad de El Alto

Sujetos: usuarios de la Avenida 6 de Marzo

Contexto: vehículos del Sindicato Pedro Domingo Murillo

Pdf con las correcciones

Enlace: [Correcciones](#)

BIBLIOGRAFIA:

- Cervero, Robert. *Informal transport in the developing world*. United Nations Centre for Human Settlements (Habitat), 2000.
- Duckett, Jon. *HTML & CSS: Design and Build Web Sites*. Wiley, 2011.
- Gascón, Ulises. *El gran libro de Node.js: una guía moderna y completa para crear aplicaciones web eficientes con Node.js*. Marcombo, 2024.
- Laudon, K. C., & Laudon, J. P. (2012). *Sistemas de información gerencial* (12.^a ed.). Pearson Educación.
- Mamani, Rolando. *Atlas de El Alto: estudio con información estadística, descriptiva y analítica, sobre las condiciones, oportunidades e institucionalidad de la población y la ciudad*. Centro de Promoción de la Mujer "Gregoria Apaza", 2011.
- Ramm, Frederik, and Jochen Topf. *OpenStreetMap: die freie Weltkarte nutzen und mitgestalten*. Lehmanns Media, 2010.
- Shevat, Amir. *Designing Bots: Creating Conversational Experiences*. O'Reilly, 2017.
- Sutherland, Jeff, and J.J. Sutherland. *Scrum: The Art of Doing Twice the Work in Half the Time*. Crown, 2014.
- White, Ron. *How Computers Work: The Evolution of Technology*. Que, 2015.
- Chodorow, K. (2013). *MongoDB: The Definitive Guide*. O'Reilly Media.
- Duckett, J. (2011). *HTML and CSS: Design and Build Websites*. Wiley.
- Flanagan, D. (2011). *JavaScript: The Definitive Guide* (6th ed.). O'Reilly Media.
- Gascón, J. (2024). *Desarrollo de aplicaciones con Node.js y Express*. Editorial Académica.

Ramm, F., Topf, J., & Chilton, S. (2010). *OpenStreetMap: Using and Enhancing the Free Map of the World*. UIT Cambridge.