



PONTIFICIA UNIVERSIDAD CATÓLICA DEL ECUADOR

ESCUELA HÁBITAT, INFRAESTRUCTURA Y CREATIVIDAD

**TRABAJO DE INTEGRACIÓN CURRICULAR PREVIO A LA OBTENCIÓN
DEL TÍTULO DE INGENIERO EN TECNOLOGÍAS DE LA INFORMACIÓN**

**PLATAFORMA DIGITAL PARA EL PROCESAMIENTO Y ANÁLISIS DE
DATOS DE GEOLOCALIZACIÓN DEL TRANSPORTE URBANO EN LA
COOPERATIVA DE BUSES 28 DE SEPTIEMBRE CON INTEGRACIÓN DE
GOOGLE MAPS**

AUTOR: JEAN PIERRE RAMÍREZ CEVALLOS

TUTOR: MGS. RICARDO PATRICIO RUIZ QUIRANZA

IBARRA – ECUADOR

JULIO, 2026

CERTIFICACIÓN TUTOR

En mi calidad de Tutor del Trabajo de JEAN PIERRE RAMÍREZ CEVALLOS titulado:
Plataforma digital para el procesamiento y análisis de datos de geolocalización del transporte urbano en la cooperativa de buses 28 de septiembre con integración de Google maps, presentado por el estudiante JEAN PIERRE RAMÍREZ CEVALLOS con cédula de ciudadanía N°1003688791, para obtener el Título de Ingeniero en Tecnologías de la Información.

Certifico que el trabajo cumple con todos los parámetros establecidos, mediante el cual el estudiante demuestra el desarrollo de competencias en el campo de conocimiento de su profesión con un nivel de argumentación coherente, para ser sometido a la evaluación por parte de los lectores.

Adicionalmente, se adjunta el certificado de porcentaje de originalidad de TURNITIN.

Turnitin Originality Report

Processed on: 28-Jul-2026 16:50 -05
ID: 2995577272
Word Count: 27735
Submitted: 4

Trabajo de titulación By JEAN PIERRE RAMIREZ CEVALLOS

[Document Viewer](#)

Similarity Index	Similarity by Source
4%	Internet Sources: 4% Publications: 1% Student Papers: 2%

include quoted

include bibliography

excluding matches < 20 words

mode: quickview (classic) report

print

refresh

download

<1% match (Internet from 11-Sep-2025)
<https://repositorio.puce.edu.ec/server/api/core/bitstreams/ee352795-8dd9-42e9-9121-b7ac58484486/content>

<1% match (Internet from 11-Sep-2025)
<https://repositorio.puce.edu.ec/server/api/core/bitstreams/a8d3c2b4-db4f-485f-8d80-ff50e1488093/content>

<1% match (Internet from 11-Sep-2025)
<https://repositorio.puce.edu.ec/server/api/core/bitstreams/3b52ab16-c801-4a84-b621-a0ecdb49c394/content>

<1% match (Internet from 18-Mar-2025)
<https://repositorio.puce.edu.ec/server/api/core/bitstreams/d2636891-2e97-421e-aeba-23a9dafb07ce/content>

<1% match (Internet from 08-Apr-2025)
<https://repositorio.puce.edu.ec/server/api/core/bitstreams/95c309a9-ed2e-4734-a5ad-9bd3acbfccf1/content>

<1% match (Internet from 11-Sep-2025)
<https://repositorio.puce.edu.ec/server/api/core/bitstreams/ac7b0a60-6c27-4f24-919e-c377e818d5ce/content>

<1% match (Internet from 11-Sep-2025)
<https://repositorio.puce.edu.ec/server/api/core/bitstreams/23fc05c5-ca7b-4703-b046-d5af059c9a99/content>

RUIZ
QUIRANZA
RICARDO
(f): PATRICIO

Firmado digitalmente por
RUIZ QUIRANZA
RICARDO PATRICIO
Fecha: 2026.07.29
09:38:15 -05'00'

Mgs. Ricardo Patricio Ruiz Quiranza

TUTOR DE TRABAJO

C.C.: 1002836524

PÁGINA DE APROBACIÓN DEL TRIBUNAL

El tribunal examinador, aprueba el presente trabajo en nombre de la Pontificia Universidad Católica del Ecuador Ibarra:

RUIZ
QUIRANZA
RICARDO
PATRICIO

Firmado
digitalmente por
RUIZ QUIRANZA
RICARDO PATRICIO
Fecha: 2026.07.29
09:38:58 -05'00'

(f):

Mgs. Ricardo Patricio Ruiz Quiranza

C.C.: 1002836524



Alvaro Mauricio
Cevallos Ramirez



(f):

Mgs. Álvaro Mauricio Cevallos Ramírez

C.C.: 1002494019



Validar únicamente en FirmaEC.
Firmado electrónicamente por:
SEGUNDO ELICEO
PUSDA CHULDE

(f):

Mgs. Segundo Eliceo Pusdá Chulde

C.C.: 0401567938

ACTA DE CESIÓN DE DERECHOS

Yo, Jean Pierre Ramírez Cevallos, declaro conocer y aceptar la disposición del Art. 165 del Código Orgánico de Economía Social de los Conocimientos, Creatividad e Innovación, que manifiesta textualmente: “Se reconoce facultad de los autores y demás titulares de derechos de disponer de sus derechos o autorizar las utilidades de sus obras o prestaciones a título gratuito y oneroso, según las condiciones que determinen. Esta facultad podrá ejercerse mediante licencias libres, abiertas y otros modelos alternativos de licenciamiento o la renuncia”.

Ibarra, 21 de julio del 2026

JEAN
PIERRE
RAMIREZ
CEVALLOS

Digitally signed
by JEAN PIERRE
RAMIREZ
CEVALLOS
Date: 2026.07.29
09:33:58 -05'00'

(f): _____

Jean Pierre Ramírez Cevallos

C.C.:1003688791

AUTORÍA

Yo, Jean Pierre Ramírez Cevallos, portador de la cédula de ciudadanía N°1003688791, declaro que el presente trabajo de investigación es de total responsabilidad del autor, y eximo expresamente a la Pontificia Universidad Católica del Ecuador Ibarra de posibles reclamos o acciones legales.

JEAN PIERRE RAMIREZ CEVALLOS
(f):.....

Digitally signed by
JEAN PIERRE
RAMIREZ
CEVALLOS
Date: 2026.07.29
09:34:18 -05'00'

Jean Pierre Ramírez Cevallos

C.C.:1003688791

DEDICATORIA

Dedico este trabajo a Dios, por darme la vida, la fortaleza y la oportunidad de culminar una etapa tan importante de mi formación personal y profesional. Su guía me permitió mantenerme firme incluso en los momentos más exigentes de este proceso.

A mi familia, por su amor, apoyo y confianza incondicional a lo largo de este camino. Gracias por acompañarme en cada etapa, por impulsarme a seguir adelante y por recordarme siempre el valor del esfuerzo, la constancia y la fe en uno mismo.

A todas las personas que estuvieron presentes durante este proceso, por su compañía, sus palabras de aliento y su apoyo sincero. Cada gesto, consejo y muestra de afecto formó parte de este logro y dejó una huella importante en esta meta alcanzada.

AGRADECIMIENTOS

Agradezco a Dios por acompañarme a lo largo de este camino y darme la fortaleza necesaria para superar cada desafío que se presentó durante esta etapa. Su guía fue fundamental para mantener la perseverancia y concluir este proceso con responsabilidad y dedicación.

Expreso mi más sincero agradecimiento a mi familia, por su amor, comprensión y respaldo constante. Su confianza en mí fue una de las mayores motivaciones para seguir adelante y convertir este esfuerzo en una meta cumplida.

A mi tutor, por su orientación, tiempo y valiosos aportes durante el desarrollo de esta investigación. Sus observaciones y recomendaciones contribuyeron de manera significativa al fortalecimiento de este trabajo y a la mejora de sus distintos apartados.

A la Pontificia Universidad Católica del Ecuador, sede Ibarra, por la formación académica y humana brindada durante mi carrera, y a la Cooperativa de Transportes “28 de Septiembre”, por la apertura y disposición para facilitar el desarrollo de este estudio dentro de su contexto operativo.

Finalmente, agradezco a todas las personas que, de una u otra manera, formaron parte de este proceso. Su apoyo, sus consejos y su presencia hicieron más llevadero este camino y contribuyeron a la culminación de una etapa muy significativa en mi vida.

ÍNDICE DE CONTENIDOS

CERTIFICACIÓN TUTOR.....	ii
PÁGINA DE APROBACIÓN DEL TRIBUNAL.....	iii
ACTA DE CESIÓN DE DERECHOS.....	iv
AUTORÍA	v
DEDICATORIA	vi
AGRADECIMIENTOS	vii
ÍNDICE DE CONTENIDOS.....	viii
ÍNDICE DE TABLAS	x
ÍNDICE DE FIGURAS.....	xi
RESUMEN	1
ABSTRACT	2
INTRODUCCIÓN	3
CAPÍTULO I. ESTADO DEL ARTE.....	6
1.1 Transformación digital y movilidad inteligente	6
1.2 Sistemas Inteligentes de Transporte (ITS).....	7
1.3 Geolocalización en el transporte urbano.....	9
1.4 Tiempo estimado de llegada (ETA)	12
1.5 Estándares de información para transporte público	15
1.6 Plataformas digitales de movilidad	17
1.7 Gestión operativa del transporte.....	19
1.8 Arquitecturas de software para sistemas de transporte	21
CAPÍTULO II. MATERIALES Y MÉTODOS.....	24
2.1 Generalidades de la investigación.....	24
2.2 Metodología general del estudio	25
2.3 Desarrollo del prototipo con Scrum.....	26
2.4 Fase 1 – Identificación del problema y motivación.....	27
2.5 Fase 2 – Planteamiento de la solución	28
2.6 Fase 3 – Diseño y desarrollo del artefacto	28
2.7 Fase 4 – Demostración del prototipo.....	65
2.8 Fase 5 – Evaluación y validación	67
2.9 Fase 6 – Comunicación de la investigación	73
CAPÍTULO III. RESULTADOS Y DISCUSIÓN.....	75
3.1 Resultados de la implementación del prototipo.....	75

3.2 Cumplimiento de los requisitos funcionales	87
3.3 Cumplimiento de requisitos no funcionales	89
3.4 Validación consolidada del prototipo	90
3.5 Resultados de las pruebas de caja negra	92
3.6 Discusión de resultados	97
CONCLUSIONES	99
RECOMENDACIONES.....	100
REFERENCIAS BIBLIOGRÁFICAS.....	101
ANEXOS.....	104
Carta de recepción de la propuesta tecnológica por parte del cliente.....	104

ÍNDICE DE TABLAS

Tabla 1 Requisitos funcionales.....	300
Tabla 2 Requisitos no funcionales.....	311
Tabla 3 Historias de usuario de la plataforma propuesta	322
Tabla 4 Tareas derivadas de las historias de usuario, organizadas por prioridad.....	366
Tabla 5 Planificación de tareas distribuidas por sprint.....	400
Tabla 6 Casos de prueba aplicados al prototipo	69
Tabla 7 Matriz de cumplimiento de requisitos funcionales.....	88
Tabla 8 Matriz consolidada de resultados de validación del prototipo	911
Tabla 9 Matriz de resultados de pruebas de caja negra	92

ÍNDICE DE FIGURAS

Figura 1 Arquitectura general del sistema.....	433
Figura 2 Diagrama de casos de uso de la plataforma propuesta	45
Figura 3 Modelo entidad–relación de la base de datos	47
Figura 4 Diagrama de componentes de la plataforma digital.....	500
Figura 5 Diagrama de secuencia de recepción, validación y almacenamiento de datos GPS....	57
Figura 6 Diagrama de secuencia de procesamiento operativo y cálculo de ETA	58
Figura 7 Diagrama de secuencia de generación de GTFS-Realtime.....	59
Figura 8 Diseño del panel principal del sistema	611
Figura 9 Diseño del panel administrativo	622
Figura 10 Interfaz de inicio de sesión	76
Figura 11 Interfaz de registro de usuarios y asignación de perfiles	76
Figura 12 Entorno ciudadano interactivo del mapa público.....	77
Figura 13 Componentes de búsqueda geoespacial con autocompletado.....	77
Figura 14 Visualización del itinerario recomendado y desglose de tramos	78
Figura 15 Visualización de tramos detallados y conexiones peatonales.....	79
Figura 16 Visualización del trazado de ruta y tiempos de arribo por parada	79
Figura 17 Registro de incidencias mediante selección directa sobre el mapa.....	811
Figura 18 Registro de incidencias asociado a una unidad activa en circulación.....	811
Figura 19 KPIs de eficiencia individual y productividad de la flota.....	822
Figura 20 Interfaz de sincronización externa de telemetría	822
Figura 21 Tabla de monitoreo y estado de transmisión de la flota.....	833
Figura 22 Interfaz de reconstrucción cartográfica y alineación de trazados	83
Figura 23 Comparativa estadística de tiempos de tránsito por línea.....	84
Figura 24 Detalle de auditoría de paradas omitidas y desvíos espaciales	84
Figura 25 Análisis de velocidad por tramos y prioridad de infraestructura	85
Figura 26 Consola de moderación e incidentes ciudadanos correlacionados.....	86
Figura 27 Herramienta de auditoría GTFS y exportación de archivos comprimidos	87
Figura 28 Configuración del validador GTFS-Realtime con el feed GTFS estático y los feeds dinámicos del sistema	96
Figura 29 Resultados de la validación de los feeds GTFS-Realtime sin errores críticos.....	96

RESUMEN

La presente investigación se desarrolló con el objetivo de diseñar y construir un prototipo de plataforma digital para el procesamiento y análisis de datos de geolocalización del transporte urbano de la Cooperativa de Transportes “28 de Septiembre”. La propuesta surgió ante la necesidad de aprovechar de mejor manera la información GPS generada por las unidades y convertirla en datos útiles para la supervisión operativa, la consulta ciudadana y la interoperabilidad con otros sistemas. Para ello, el estudio se abordó bajo el enfoque de Design Science Research Methodology (DSRM) y utilizó Scrum como metodología complementaria para la construcción iterativa del artefacto.

El prototipo integró la recepción de telemetría GPS, el almacenamiento estructurado en PostgreSQL, el procesamiento de trayectorias, el cálculo de tiempos estimados de llegada, la planificación de itinerarios y la generación de salidas en formatos GTFS y GTFS-Realtime. Además, incorporó una capa de visualización compuesta por un mapa público orientado al usuario y un panel administrativo destinado al monitoreo y control operativo.

Como resultado, se obtuvo una solución funcional capaz de convertir datos de geolocalización en información útil para el seguimiento de la flota y la consulta del servicio. La validación permitió comprobar la coherencia de los módulos implementados, la continuidad del flujo de procesamiento y la generación de salidas compatibles con estándares abiertos de transporte. En conjunto, los resultados muestran que el aprovechamiento de datos GPS institucionales puede contribuir de manera concreta al fortalecimiento de la gestión del transporte urbano y a su posible integración con plataformas como Google Maps.

Palabras clave: geolocalización, transporte urbano, interoperabilidad, GTFS, Google Maps.

ABSTRACT

This research was carried out with the aim of designing and building a digital platform prototype for processing and analyzing urban transport geolocation data from the “28 de Septiembre” Transport Cooperative. The proposal emerged from the need to make better use of the GPS information generated by the vehicles and turn it into useful data for operational supervision, public consultation, and interoperability with other systems. To address this objective, the study followed the Design Science Research Methodology (DSRM) and used Scrum as a complementary approach for the iterative development of the technological artifact.

The prototype integrated GPS telemetry reception, structured storage in PostgreSQL, trajectory processing, estimated arrival time calculation, route planning, and the generation of GTFS and GTFS-Realtime outputs. In addition, it incorporated a visualization layer composed of a public map for users and an administrative dashboard intended for operational monitoring and control.

As a result, a functional solution was obtained, capable of converting geolocation data into useful information for fleet monitoring and service consultation. The validation process confirmed the coherence of the implemented modules, the continuity of the processing flow, and the generation of outputs compatible with open transport standards. Overall, the results show that the use of institutional GPS data can contribute in a practical way to strengthening urban transport management and to its potential integration with platforms such as Google Maps.

Keywords: geolocation, urban transport, interoperability, GTFS, Google Maps.

INTRODUCCIÓN

En los últimos años, la digitalización del transporte urbano ha abierto nuevas posibilidades para mejorar la gestión del servicio, el seguimiento de las unidades y la atención al usuario. En este contexto, herramientas como la geolocalización vehicular, la estimación de tiempos de llegada, los estándares de intercambio de datos y las plataformas de navegación digital han pasado a ocupar un lugar cada vez más importante dentro de las soluciones tecnológicas aplicadas a la movilidad. Sin embargo, su utilidad no depende solo de contar con dispositivos o con datos disponibles, sino de disponer de una estructura que permita capturarlos, procesarlos, organizarlos y convertirlos en información realmente útil.

Esta situación puede observarse en la Cooperativa de Buses 28 de Septiembre de la ciudad de Ibarra, Ecuador. Aunque la cooperativa ya dispone de GPS institucionales instalados en sus unidades, la información generada por estos equipos se utiliza principalmente para consultas posteriores y no dentro de una plataforma que permita producir tiempos estimados de llegada, análisis operativo o integración con servicios orientados al usuario. En otras palabras, el dato existe, pero todavía no se aprovecha de manera funcional dentro del servicio. A partir de esa realidad surge esta investigación: convertir la información de geolocalización ya disponible en un recurso útil para el monitoreo de rutas, el análisis del comportamiento de las unidades y la generación de información accesible tanto para usuarios como para administradores.

Con ese punto de partida, se propuso el desarrollo de un prototipo sustentado en una arquitectura compuesta por GPS institucional, API del proveedor, backend, base de datos PostgreSQL, generación de GTFS y GTFS-Realtime, integración con Google Maps y un panel web administrativo, trabajando inicialmente con tres rutas y tres buses. Bajo esta lógica, el estado del arte no se plantea solo como una revisión conceptual, sino como un apoyo para reconocer qué aportes metodológicos, técnicos y funcionales pueden recuperarse de estudios previos para orientar el diseño del sistema propuesto. Por ello, la revisión bibliográfica resulta necesaria para examinar cómo se han abordado en otras investigaciones temas como la digitalización del transporte, los sistemas inteligentes de transporte, la geolocalización en tiempo real, la estimación de llegada, la estructuración de datos mediante estándares abiertos, la integración con plataformas de movilidad y la visualización operativa de la información.

El objetivo general de esta investigación es diseñar una plataforma digital orientada al procesamiento y análisis de datos de geolocalización del transporte urbano de la Cooperativa de Buses 28 de Septiembre, con el fin de generar información sobre rutas y tiempos estimados de llegada mediante su integración con Google Maps y, al mismo tiempo, fortalecer la gestión operativa del servicio. Para alcanzar este propósito, se plantearon los siguientes objetivos específicos:

- Analizar el funcionamiento actual del sistema de transporte urbano de la Cooperativa de Buses 28 de Septiembre, identificando las necesidades de información tanto para los usuarios como para la gestión operativa del servicio.
- Diseñar la arquitectura tecnológica de la plataforma digital, considerando la integración de datos de geolocalización, su procesamiento y su estructuración para su posterior utilización en sistemas de información de transporte.
- Implementar un sistema de procesamiento de datos de geolocalización que permita generar información sobre rutas, recorridos y tiempos estimados de llegada de las unidades.
- Integrar los datos del sistema de transporte con plataformas de navegación digital como Google Maps, mediante el uso de estándares abiertos de transporte que faciliten la visualización de rutas y tiempos de llegada por parte de los usuarios.
- Desarrollar un panel web administrativo orientado a la visualización y análisis de información operativa del servicio de transporte, incluyendo indicadores relacionados con recorridos, tiempos de operación y comportamiento de las unidades.

A partir de ello, este capítulo reúne los antecedentes necesarios para comprender y justificar los componentes principales de la propuesta. Para ello, se revisan la transformación digital y la movilidad inteligente como marco general del problema, los sistemas inteligentes de transporte como entorno funcional de aplicación, y la geolocalización, el tiempo estimado de llegada y los estándares GTFS y GTFS-Realtime como base para el procesamiento e interoperabilidad de los datos. También se consideran las plataformas digitales de movilidad, la integración con Google Maps y las arquitecturas de software necesarias para soportar el procesamiento y la visualización de la información. Con esta estructura, el capítulo construye la base teórica y técnica que

orienta el desarrollo posterior de la propuesta, evitando repetir en cada apartado el contexto particular de la cooperativa.

El documento se organiza en tres capítulos. El primero desarrolla el marco teórico y el estado del arte sobre movilidad inteligente, sistemas de geolocalización, estándares de transporte y tecnologías relacionadas con el desarrollo del sistema. El segundo presenta la metodología de investigación, el análisis de requerimientos y el diseño de la arquitectura de software. El tercero expone la implementación del prototipo, los resultados obtenidos y las pruebas realizadas para validar la solución propuesta.

CAPÍTULO I. ESTADO DEL ARTE

En este capítulo se revisan los antecedentes teóricos, técnicos y metodológicos que sustentan el diseño de la plataforma propuesta. Para ello, se analizan estudios relacionados con la transformación digital del transporte, los Sistemas Inteligentes de Transporte, la geolocalización vehicular, el tiempo estimado de llegada, los estándares GTFS y GTFS-Realtime, la integración con Google Maps, la gestión operativa y las arquitecturas de software orientadas al procesamiento de datos en tiempo real. Más que reunir definiciones aisladas, este estado del arte busca identificar enfoques, componentes y decisiones funcionales que aportan al problema abordado en esta investigación, especialmente en contextos de transporte urbano donde ya existen datos de geolocalización, pero todavía no se integran dentro de una solución interoperable, analítica y orientada tanto a la gestión como al usuario.

1.1 Transformación digital y movilidad inteligente

La transformación digital del transporte urbano no debe entenderse únicamente como la incorporación de herramientas tecnológicas, sino como un proceso más amplio de reorganización del servicio. En el contexto latinoamericano, este enfoque implica integrar tecnologías digitales a la cadena de valor del sector y generar cambios operacionales, organizativos y en la forma de prestar el servicio (Calatayud et al., 2022). Desde esta perspectiva, el problema no consiste solo en comprobar si las unidades cuentan con GPS institucionales, sino en determinar si la información generada por esos dispositivos realmente se incorpora a un esquema capaz de producir valor operativo, informativo y de gestión.

En ese marco, la movilidad inteligente puede asumirse como una expresión concreta de la transformación digital aplicada al transporte urbano. Su alcance no depende solo de disponer de tecnología, sino de articular procesos de captura, análisis y uso de información que permitan coordinar mejor la operación y responder con mayor precisión a la dinámica de la red (Calatayud et al., 2022). Así, la información deja de ser un simple registro y pasa a convertirse en un recurso útil tanto para la operación del servicio como para la experiencia del usuario.

La revisión del BID también permite diferenciar una digitalización superficial de una transformación más profunda del servicio. Esta distinción es relevante porque evita asumir que la sola presencia de infraestructura tecnológica constituye una prueba suficiente de modernización. El punto crítico se encuentra, más bien, en la brecha entre

la disponibilidad del dato y su aprovechamiento sistémico. A partir de ello, resulta pertinente examinar de qué manera la información de ubicación puede transformarse en insumos para el análisis operativo, la estimación de tiempos de llegada y la generación de salidas interoperables.

Por otro lado, la CAF aborda la movilidad urbana como un problema vinculado con el acceso, la calidad y la equidad. Su análisis muestra que en América Latina y el Caribe persisten tiempos de viaje elevados, mayores tiempos de espera y bajos niveles de confiabilidad del servicio, factores que afectan directamente la experiencia del usuario (Corporación Andina de Fomento [CAF], 2023). Este aporte permite reconocer variables importantes para el análisis del transporte urbano, entre ellas el tiempo de espera, la confiabilidad y la disponibilidad de información para el pasajero.

En conjunto, el BID aporta un marco para comprender la transformación digital del transporte, mientras que la CAF incorpora criterios asociados con la calidad del servicio y la experiencia del usuario en el contexto latinoamericano. A partir de estas bases, los Sistemas Inteligentes de Transporte pueden analizarse como el entorno técnico y funcional en el que dichas capacidades se concretan.

1.2 Sistemas Inteligentes de Transporte (ITS)

Los Sistemas Inteligentes de Transporte constituyen el entorno funcional en el que la transformación digital del transporte se convierte en servicios concretos de monitoreo, coordinación e información. Integran tecnologías de información, comunicación y análisis de datos con el fin de mejorar la operación de la red y la calidad del servicio. En este sentido, Elassy et al. (2024) los describen como sistemas orientados a fortalecer la efectividad, la seguridad y la sostenibilidad de las redes de transporte. Esta definición permite situar la discusión en una lógica de servicio inteligente, donde la infraestructura tecnológica no se incorpora como un fin en sí mismo, sino como una base para sostener funciones operativas identificables.

A partir de esa definición, los ITS abarcan funciones relacionadas con la gestión del tráfico, la operación vehicular y el seguimiento del transporte público. Elassy et al. (2024) incluyen dentro de su alcance la señalización inteligente, la planificación dinámica de rutas y el rastreo de unidades, lo que permite entender que la inteligencia del sistema depende de incorporar información actualizada a la operación cotidiana. Desde esta perspectiva, el interés no recae en la infraestructura por sí sola, sino en su capacidad para

sostener funciones como el seguimiento de unidades, la estructuración de recorridos y el soporte para estimaciones de llegada.

La literatura muestra, sin embargo, que la implementación de ITS no ocurre de la misma manera en todos los contextos. Ramírez-Guerrero et al. (2022) identifican, para ciudades intermedias de Colombia, tres factores decisivos en la integración de servicios tecnológicos al transporte: el costo de las tecnologías, los requisitos funcionales y las condiciones económicas y administrativas de implementación. Esta referencia resulta especialmente útil porque ofrece un marco cercano al de ciudades que no cuentan con los recursos ni con la escala de las grandes metrópolis, y permite analizar los ITS desde una lógica más gradual y ajustada a necesidades concretas.

En esa misma línea, Ramírez-Guerrero et al. (2022) muestran que en estos contextos se priorizan tecnologías basadas en estándares abiertos para proporcionar información a los usuarios. También señalan que los mayores esfuerzos de integración tecnológica suelen concentrarse en la gestión y control de flota, el recaudo y, en algunos casos, el control del tráfico. Este hallazgo sugiere que, en ciudades intermedias, la pertinencia de un ITS no depende de la complejidad de su arquitectura, sino de su capacidad para resolver necesidades reales de seguimiento, organización operativa e información al pasajero.

El análisis de Nunes (2022) complementa esta discusión al enfatizar que los ITS también deben entenderse desde la experiencia del usuario. Su estudio sobre sistemas de información al pasajero muestra que el uso del transporte público depende cada vez más de la disponibilidad de información en tiempo real, la planificación de rutas y la actualización del servicio. Al mismo tiempo, permite anticipar que esa utilidad exige estructuras de datos interoperables capaces de representar tanto la dimensión planificada del servicio como sus cambios dinámicos. De este modo, la inteligencia del sistema no se limita al control interno de la operación, sino que también se expresa en salidas comprensibles y reutilizables para consulta, monitoreo y apoyo a la movilidad.

En conjunto, los ITS no aparecen aquí como una categoría general de innovación, sino como un entorno técnico y funcional que orienta decisiones concretas sobre integración de procesos, priorización de funciones y articulación entre operación e información al usuario. En este marco, la geolocalización en el transporte urbano constituye el componente técnico que alimenta directamente esas capacidades.

1.3 Geolocalización en el transporte urbano

1.3.1 Sistema de Posicionamiento Global (GPS)

La geolocalización en el transporte urbano parte de la capacidad de registrar de forma continua la posición de cada unidad mientras circula por la red vial. En este contexto, los dispositivos GPS o GNSS constituyen la capa inicial de adquisición de datos, ya que permiten capturar coordenadas, velocidad, rumbo y marcas temporales que luego pueden ser procesadas por el sistema. Wong y Tsang (2025) muestran que, dentro de una arquitectura de seguimiento vehicular, la captura GNSS adquiere valor cuando se integra con mecanismos posteriores de transmisión y procesamiento. De esta manera, el GPS aparece como la fuente primaria de captura dentro de un flujo técnico más amplio.

La utilidad del GPS aumenta cuando su registro se combina con mecanismos que mejoran la confiabilidad de la localización. Mansurova et al. (2025) documentan sistemas apoyados en tabletas Android, antenas instaladas en las unidades y servicios de localización acompañados de procesos de validación y filtrado, lo que pone en evidencia que la calidad del dato depende tanto del hardware como de las condiciones de captura. A esto se suma que la precisión del posicionamiento también se ve influida por decisiones de infraestructura, como la ubicación de las antenas, la disponibilidad de señal y la continuidad del registro.

Otro aspecto importante es la frecuencia con la que se capturan las observaciones. Mansurova et al. (2025) trabajan con una resolución temporal de 5 segundos, mientras que Ratneswaran et al. (2024) advierten que la granularidad del muestreo afecta directamente la utilidad de los datos en los procesos posteriores. Una frecuencia baja reduce la precisión en eventos como los tiempos de parada, mientras que una frecuencia alta incrementa las exigencias de almacenamiento y procesamiento. Esta relación resulta especialmente relevante cuando los datos se utilizarán para construir trayectorias, tiempos de recorrido y eventos operativos.

En este sentido, el GPS constituye una capa de adquisición espaciotemporal sobre la cual se apoyan funciones posteriores de modelado y análisis. Su aporte no se limita a indicar dónde se encuentra una unidad, sino que proporciona la base para construir trayectorias, segmentar viajes y derivar eventos operativos. A partir de ello, corresponde analizar el tratamiento y la circulación del dato en tiempo real.

1.3.2 Datos de geolocalización en tiempo real

La geolocalización adquiere un valor mayor cuando las posiciones capturadas no quedan restringidas a consultas posteriores, sino que pueden utilizarse durante la operación del servicio. En ese escenario, los datos de geolocalización en tiempo real permiten conocer el estado actual de la flota y alimentar funciones de seguimiento y análisis. Godfrid et al. (2022) recuerdan que GTFS-Realtime se utiliza para comunicar posiciones vivas de los vehículos, mientras que Wong y Tsang (2025) señalan que GTFS-RT contiene datos sensibles al tiempo, como la posición vehicular y el progreso de viaje. En conjunto, estos trabajos muestran que el tiempo real no consiste solo en recibir coordenadas, sino en integrarlas dentro de un flujo estructurado para su uso inmediato.

Para que ese flujo sea confiable, la literatura destaca la necesidad de validar los datos antes de convertirlos en información operativa. Mansurova et al. (2025) advierten problemas como pérdida de señal, duplicados, desplazamientos implausibles, jitter a baja velocidad e inestabilidad direccional, e incorporan filtros por velocidad, validación espacial, comprobación física del desplazamiento, suavizado exponencial y ventanas móviles. Este aporte muestra que el procesamiento en tiempo real requiere control de calidad desde etapas tempranas, evitando que errores de señal o trayectorias inconsistentes afecten cálculos posteriores.

La transmisión de esa información también exige mecanismos adecuados de comunicación entre la unidad y el sistema central. En el caso documentado por Mansurova et al. (2025), las coordenadas validadas se transmiten mediante MQTT hacia una infraestructura con almacenamiento en PostgreSQL y monitoreo en tiempo real. Este flujo ofrece una referencia clara sobre cómo estructurar el paso desde la adquisición hasta el registro: capturar, validar, transmitir y almacenar.

Además de transmitir datos, el sistema debe ser capaz de estructurarlos para su uso posterior. Ratneswaran et al. (2024) plantean que el procesamiento de GPS crudo requiere, al menos, dos operaciones principales: la extracción de trayectorias por viaje y su segmentación para emparejarlas con la información de paradas. A partir de ese procedimiento, es posible derivar tiempos de recorrido y tiempos de permanencia necesarios para estructurar información compatible con GTFS. La revisión de Godfrid et al. (2022) complementa este punto al mostrar que la disponibilidad de feeds en tiempo real no garantiza por sí sola precisión operativa, pues la calidad y la frecuencia de

actualización afectan la fidelidad con la que el sistema representa el comportamiento real de los vehículos.

En conjunto, los datos de geolocalización en tiempo real constituyen la base que permite pasar de una localización aislada a información operativa estructurada. Sobre esa base, resulta posible examinar cómo la información de múltiples unidades puede integrarse para supervisar el comportamiento global del sistema.

1.3.3 Monitoreo de flotas

El monitoreo de flotas representa la etapa en la que la geolocalización deja de ser un dato puntual de cada vehículo y pasa a convertirse en una herramienta de supervisión operativa del servicio en su conjunto. En la literatura revisada, esta función aparece estrechamente vinculada con el Automatic Vehicle Location (AVL). Elijah et al. (2023) muestran que el AVL proporciona localización geográfica en tiempo real para rastrear flotas y se relaciona con funciones de mayor valor operativo, como el seguimiento del servicio y la estimación de llegada. Esta relación sitúa al monitoreo de flotas como un componente intermedio entre la localización individual y funciones superiores de análisis.

Desde una perspectiva tecnológica, el monitoreo de flotas puede implementarse mediante distintos enfoques. Elijah et al. (2023) señalan que los sistemas basados en GPS han sido ampliamente utilizados para la gestión de flotas, aunque también exploran una arquitectura IoT con BLE como alternativa de menor costo. Esta comparación permite entender que el monitoreo depende de decisiones de arquitectura ajustadas al contexto. En el caso analizado aquí, el GPS institucional disponible se configura como la base más natural para el seguimiento de las unidades.

El monitoreo de flotas también requiere contextualizar las posiciones de los vehículos dentro de la lógica del servicio. Wong y Tsang (2025) muestran este proceso al generar mensajes GTFS-RT VehiclePosition a partir de lecturas GPS, identificadores vehiculares y marcas temporales, cruzando dicha información con relaciones vehículo-ruta y con la programación temporal del viaje. Este aporte demuestra que el seguimiento de flota no consiste solo en registrar ubicaciones, sino en situarlas dentro del contexto operativo correspondiente.

La contribución de Godfrid et al. (2022) amplía esta perspectiva al mostrar que la combinación entre GTFS estático y GTFS-Realtime permite comparar recorridos planificados y recorridos reales, además de analizar velocidades, demoras y diferencias

entre la operación teórica y la observada. De forma complementaria, Mansurova et al. (2025) y Ratneswaran et al. (2024) muestran que la agregación de trayectorias de flota permite construir registros de viajes, eventos de parada, tiempos entre segmentos y tiempos de permanencia en estaciones. En ambos casos, el monitoreo deja de ser una simple observación de vehículos en movimiento y pasa a convertirse en una base de datos operativa capaz de sostener modelado, análisis y generación de formatos estandarizados. Sobre esta base técnica, el siguiente tema es el tiempo estimado de llegada, particularmente las variables que intervienen en su cálculo y la importancia que este indicador tiene para la operación del servicio y la experiencia del usuario.

1.4 Tiempo estimado de llegada (ETA)

1.4.1 Variables que intervienen en el ETA

El tiempo estimado de llegada (ETA) constituye una función central dentro de los sistemas de transporte apoyados en geolocalización, ya que transforma el estado actual del vehículo en una predicción operativa comprensible para usuarios y operadores. A diferencia de la captura de posición, que únicamente indica dónde se encuentra una unidad en un momento determinado, el ETA exige proyectar su comportamiento hacia el siguiente punto de control, parada o destino. En este sentido, Sharma et al. (2023) muestran que esta estimación debe realizarse incluso en contextos con actualizaciones de posición dispersas y bajo dependencias espaciales y temporales del tráfico. Esto permite entender el ETA como un problema de modelado y no como una simple lectura de la ubicación actual del bus.

La literatura revisada coincide en que el ETA no puede calcularse a partir de una sola variable. Rashvand et al. (2023) identifican entre los factores más relevantes el origen, el destino, las coordenadas del vehículo, la densidad del tráfico, la distancia entre paradas y el tipo de día. Esta formulación permite organizar el problema en tres dimensiones principales: variables espaciales, temporales y operativas. Dentro de ese conjunto, la posición actual de la unidad y su relación con la siguiente parada ocupan un lugar especialmente importante. Rashvand et al. (2023) describen atributos como coordenadas de localización, tiempo de llegada programado, distancia a la siguiente parada, referencia del vehículo y sentido de operación, y señalan que la distancia al próximo punto de parada es una de las variables con mayor peso en el modelo.

Sin embargo, la distancia por sí sola no basta para representar el comportamiento del viaje. Rashvand et al. (2023) incorporan también variables como el tipo de día y la condición de hora punta, mientras que Traverso et al. (2025) trabajan con registros históricos de llegadas y salidas junto con parámetros operativos para estimar el comportamiento de los buses del Metropolitano de Lima. A ello se suma el planteamiento de Sharma et al. (2023), quienes calculan el ETA a partir de la suma de los tiempos esperados en los segmentos pendientes del trayecto. En conjunto, estos aportes muestran que la predicción de llegada depende de patrones temporales recurrentes, de la dinámica del recorrido restante y de las condiciones del tráfico. Por ello, el ETA debe entenderse como una función del viaje en curso y no solo como una extrapolación local basada en la posición actual.

La revisión también muestra que la precisión del ETA puede verse afectada por irregularidades operativas como salidas anticipadas, retrasos, detenciones prolongadas o cambios contextuales del servicio. Elijah et al. (2023) advierten que las salidas tempranas y los retrasos impactan directamente la estimación de llegada, mientras que Traverso et al. (2025) señalan como línea de mejora la incorporación de datos de tráfico y clima en tiempo real. En consecuencia, el ETA no puede concebirse como una función rígida, sino como un proceso sensible a la variabilidad real de la operación.

En términos generales, las variables que intervienen en el ETA pueden agruparse en cuatro conjuntos complementarios: localización actual de la unidad, distancia y relación con la siguiente parada o con el resto del trayecto, condiciones temporales de operación y efectos del tráfico o de la variabilidad del servicio. Esta organización ayuda a comprender cómo intervienen en la predicción de llegada y prepara el análisis de su valor estratégico dentro del transporte urbano.

1.4.2 Importancia del ETA en el transporte urbano

La importancia del ETA en el transporte urbano no se limita a su valor técnico como salida de un modelo predictivo. Su principal aporte consiste en convertir datos operativos complejos en información útil para la toma de decisiones del pasajero y para la gestión del servicio. Vera et al. (2023) muestran que las herramientas de consulta de tiempos de espera contribuyen a reducir la ansiedad, disminuir la percepción del tiempo de viaje y planificar mejor la salida desde el origen. En una línea semejante, Rashvand et al. (2023) identifican como un problema persistente la discrepancia entre los tiempos reales de arribo y los tiempos programados, mientras que Traverso et al. (2025) plantean que sus

predicciones en tiempo real buscan reducir la incertidumbre y los tiempos de espera. En conjunto, estas fuentes permiten situar el ETA como un componente con impacto directo en la experiencia del usuario y en la confiabilidad percibida del servicio.

Desde el punto de vista de la gestión, el ETA también aporta valor porque convierte la operación en un insumo útil para la supervisión y la toma de decisiones. Traverso et al. (2025) indican que su propuesta proporciona información basada en datos para optimizar la planificación de rutas y la asignación de buses, y muestran además que las predicciones pueden mejorar la respuesta del sistema ante variaciones del servicio. De esta manera, la estimación de llegada sirve no solo para informar al pasajero, sino también para detectar desajustes, identificar patrones y apoyar decisiones sobre la operación de rutas y unidades.

La importancia del ETA se aprecia con mayor claridad cuando se consideran las consecuencias de ignorar la incertidumbre del tiempo de viaje. Javanmard et al. (2025) muestran que trabajar únicamente con información programada puede distorsionar la evaluación del transporte y reportan una sobreestimación aproximada del 10.97 % en medidas de accesibilidad a servicios de salud cuando no se incorpora la variabilidad real del viaje. Aunque su estudio se centra en accesibilidad, el aporte también resulta pertinente para el ETA, ya que ambos problemas comparten un mismo núcleo: la diferencia entre el tiempo planificado y el tiempo realmente experimentado por el usuario.

En contextos urbanos latinoamericanos, donde la variabilidad del tráfico, las demoras y la heterogeneidad operativa afectan el cumplimiento de itinerarios, el ETA deja de ser una comodidad digital y pasa a convertirse en una herramienta más precisa para representar el estado real del transporte. La evidencia revisada sugiere que, cuando el sistema integra datos históricos, geolocalización en tiempo real y variables operativas, la estimación de llegada mejora la planificación del viaje y, al mismo tiempo, aporta insumos valiosos para la supervisión del servicio.

Una vez revisadas las variables que intervienen en el cálculo del ETA y su importancia dentro del transporte urbano, el análisis puede desplazarse hacia los estándares de información que permiten estructurar, compartir e integrar estos datos en plataformas interoperables. Esto conduce directamente a GTFS y GTFS-Realtime, fundamentales para representar horarios, recorridos, actualizaciones de viaje y posiciones vehiculares dentro de la plataforma propuesta.

1.5 Estándares de información para transporte público

1.5.1 GTFS

A medida que la operación del transporte público depende más de datos digitales, la estandarización de esa información se vuelve necesaria para asegurar interoperabilidad, reutilización y escalabilidad. En este contexto, el GTFS se ha consolidado como uno de los formatos de referencia para describir rutas, paradas, viajes y horarios de forma estructurada. Nunes (2022) recuerda que antes de 2005 no existía un estándar común para compartir información de transporte público y que el GTFS surgió precisamente para resolver esa dispersión, hasta convertirse en el formato más extendido para representar servicios de ruta fija. Este antecedente permite comprender por qué la información del sistema debe organizarse en una estructura normalizada y reutilizable.

La utilidad del GTFS no se explica únicamente por su difusión, sino también por la forma en que representa operativamente el servicio. Liu et al. (2025) muestran que cada feed se compone de múltiples tablas relacionadas entre sí, bajo una lógica similar a la de una base de datos relacional, y sitúan a la tabla trips como eje de articulación entre rutas, servicios y tiempos de parada. En la misma línea, Hassan et al. (2025) describen al GTFS como una representación legible por máquina de los servicios de tránsito, capaz de detallar alineaciones de rutas, paradas, horarios y calendarios de operación. Esto evidencia que la generación de GTFS exige mantener relaciones consistentes entre rutas, viajes, paradas y secuencias temporales.

Desde una perspectiva aplicada, el GTFS ha permitido que la información del transporte no permanezca limitada al uso interno de las operadoras, sino que pueda emplearse en planificación de viajes, análisis de red, visualización y estudios de accesibilidad. Nunes (2022) señala que muchas agencias comparten sus feeds públicamente o mediante APIs, mientras que Liu et al. (2025) explican que la comprensión de esta estructura facilita el desarrollo de aplicaciones de route planning, system visualization y transit analysis. La literatura también presenta al GTFS como una mejora metodológica frente a fuentes no estandarizadas, al ofrecer ventajas en organización, integración espacial, resolución temporal, reproducibilidad y compatibilidad con herramientas analíticas (Hassan et al., 2025).

Sin embargo, el GTFS estático presenta límites cuando se busca representar el comportamiento real del sistema. Nunes (2022) observa que, aunque incorpora atributos

útiles, su capacidad para describir información dinámica sigue siendo restringida frente a otros estándares más completos. De forma similar, Liu et al. (2025) señalan que, para aplicaciones avanzadas, no basta con la estructura estática y suele ser necesario construir redes espaciotemporales que incorporen variaciones a lo largo del día. Estas observaciones muestran que el GTFS resuelve la capa planificada del transporte, pero no alcanza por sí solo para modelar posiciones vehiculares, retrasos o cambios operativos. Esa limitación explica la necesidad de incorporar GTFS-Realtime como complemento.

1.5.2 GTFS-Realtime

GTFS-Realtime surge para representar la dimensión operativa y cambiante del transporte público. Hassan et al. (2025) explican que esta extensión incorpora tres feeds principales —trip updates, vehicle positions y service alerts— que permiten informar retrasos, localización de vehículos y interrupciones del servicio. De forma coincidente, Colque et al. (2021) señalan que este estándar incorpora posiciones de vehículos, actualizaciones de viaje y alertas de servicio. A partir de estos aportes, puede entenderse que GTFS-Realtime no sustituye al GTFS estático, sino que añade la capa dinámica necesaria para el seguimiento, la predicción y la comunicación operativa.

Además de su definición técnica, GTFS-Realtime ha demostrado utilidad en implementaciones reales. En el caso de Arica, Colque et al. (2021) muestran que este estándar permite servir información dinámica mediante HTTP y Protocol Buffers, articulando posiciones de vehículos, actualizaciones de viaje y alertas dentro de una misma lógica de interoperabilidad. Este caso ofrece una referencia útil sobre cómo conectar el procesamiento interno de datos con servicios de consulta orientados a otros sistemas o usuarios.

La contribución de Ratneswaran et al. (2024) refuerza esta línea al subrayar que GTFS-RT incluye ubicaciones de vehículos, predicciones de arribo y alertas de servicio, y que su generación resulta relevante para aplicaciones de seguimiento, programación y predicción de llegada. Esta formulación conecta directamente el estándar con funciones centrales del prototipo, como el monitoreo de unidades, la estructuración temporal del servicio y el soporte al cálculo de ETA.

En conjunto, GTFS y GTFS-Realtime conforman una base complementaria para estructurar la información del transporte público. El primero organiza la dimensión planificada del sistema mediante rutas, viajes, paradas y horarios; el segundo incorpora

la dimensión dinámica de posiciones, actualizaciones y alertas operativas. Con esta base, resulta posible examinar las plataformas digitales de movilidad que consumen estos estándares para ofrecer consulta, visualización y planificación de viajes, especialmente Google Maps y su integración con información de transporte público.

1.6 Plataformas digitales de movilidad

Las plataformas digitales de movilidad funcionan como uno de los puntos de contacto más visibles entre la información del transporte y el pasajero. A través de ellas, los datos sobre rutas, tiempos de viaje, ubicaciones y alternativas de desplazamiento se convierten en información consultable para planificar un recorrido. Por esta razón, su utilidad depende en gran medida de la calidad, actualización y estructura de los datos que reciben. Rashvand et al. (2023) observan que los usuarios recurren con frecuencia a aplicaciones externas como Google Maps y Waze para organizar sus desplazamientos, aunque advierten que estas herramientas no siempre ofrecen la precisión requerida ni se articulan directamente con los operadores locales del transporte. Esto permite entender que las plataformas digitales no sustituyen al sistema de información de transporte, sino que funcionan como interfaces de consumo apoyadas en datos previamente procesados.

Desde el punto de vista técnico, estas plataformas requieren información estructurada para representar recorridos, posiciones, tiempos y alertas de manera comprensible. Colque et al. (2021) muestran que, cuando existe una infraestructura de datos adecuada, es posible desplegar posiciones vehiculares, predicciones de llegada y alertas de servicio en interfaces orientadas al usuario. De forma complementaria, Godfrid et al. (2022) explican que plataformas ampliamente utilizadas, entre ellas Google Maps, procesan información de transporte para calcular tiempos de viaje y rutas, aunque también destacan que muchas de estas soluciones son propietarias y presentan limitaciones para el análisis o la extensión por parte de terceros. En este sentido, las plataformas digitales de movilidad dependen de una arquitectura previa capaz de capturar, depurar y organizar los datos antes de exponerlos al usuario.

1.6.1 Google Maps y la integración de transporte público

Dentro de este ecosistema, Google Maps ocupa un lugar importante por su alcance masivo y por su uso cotidiano en la consulta de rutas y tiempos de viaje. Godfrid et al. (2022) señalan que sus APIs permiten calcular tiempos y distancias de viaje, así como determinar por qué vías circula un vehículo. Sin embargo, los mismos autores aclaran que estas

funcionalidades son propietarias, no cuentan con un lenguaje de consulta flexible y no pueden extenderse libremente por la comunidad SIG o por desarrolladores externos. Esta condición permite reconocer que Google Maps representa una interfaz valiosa para el usuario final, pero no reemplaza la necesidad de contar con una capa propia de procesamiento y control sobre los datos operativos del sistema.

La integración del transporte público con Google Maps depende, en buena medida, de que la información se encuentre estructurada en formatos aceptados por la plataforma. Colque et al. (2021) explican que GTFS-Realtime fue desarrollado por Google como una extensión del GTFS y señalan que esta especificación se utiliza en Google Maps para gestionar información dinámica del transporte público. Por ello, la generación de GTFS y GTFS-Realtime no debe entenderse como un elemento adicional del sistema, sino como una condición técnica para proyectar datos procesados hacia una plataforma externa de consulta masiva.

La evidencia revisada también muestra que la integración con Google Maps puede ir más allá de la representación de recorridos estáticos. Cuando la infraestructura local es capaz de producir información dinámica, es posible incorporar estimaciones de llegada y actualizaciones operativas. En el caso de Arica, Colque et al. (2021) comparan el funcionamiento de su sistema con Google Maps y muestran que, mientras el GTFS estático ofrece horarios programados, GTFS-Realtime permite estimaciones más ajustadas al incorporar atrasos reales de los vehículos. Además, destacan que este tipo de información mejora la gestión del tiempo por parte del usuario y puede alimentar planificaciones de viaje con estimaciones en tiempo real.

Aun así, es necesario matizar el alcance de esta integración. Rashvand et al. (2023) advierten que muchas aplicaciones utilizadas por los usuarios, entre ellas Google Maps, dependen de información crowdsourced y con frecuencia no alcanzan la precisión necesaria para planificar el viaje con suficiente confianza. Su análisis sugiere que el valor de estas plataformas mejora cuando se articulan con modelos predictivos y con una infraestructura en la nube capaz de actualizar con mayor precisión la llegada de los buses. En ese sentido, Google Maps puede funcionar como una salida valiosa para la visualización y consulta del servicio, siempre que exista un sistema local capaz de capturar, depurar, estimar y estructurar la información de manera consistente.

En conjunto, Google Maps puede ampliar el alcance de la información del servicio hacia los usuarios, pero su integración efectiva depende de estándares adecuados y de una base

operativa correctamente procesada. A partir de esta relación entre operación, publicación y consumo de información, resulta pertinente examinar la gestión operativa del transporte, especialmente en lo relacionado con indicadores, toma de decisiones basada en datos y visualización mediante paneles de control.

1.7 Gestión operativa del transporte

1.7.1 Indicadores operativos

La gestión operativa del transporte público necesita convertir el comportamiento diario del servicio en variables observables que permitan monitorear su desempeño y detectar posibles desviaciones. En esta línea, Albán Reyes (2022) plantea que los procesos de monitoreo y evaluación son necesarios para orientar la toma de decisiones y medir avances a partir de una línea base. Desde esta perspectiva, los indicadores operativos no deben entenderse como datos aislados, sino como instrumentos de seguimiento continuo del servicio.

Estos indicadores deben construirse a partir de dimensiones concretas de la prestación. Albán Reyes (2022) organiza las medidas de calidad en categorías vinculadas con disponibilidad, confort y conveniencia, e incluye variables como frecuencia, horas de servicio, cobertura, carga de pasajeros, confiabilidad y tiempo de viaje. Así, los datos sobre rutas, tiempos y comportamiento de las unidades pueden transformarse en indicadores comparables para evaluar la regularidad, cobertura y desempeño del sistema.

Vera et al. (2023) complementan esta visión al mostrar que los indicadores también pueden estimarse a partir de flujos digitales generados por herramientas de uso cotidiano. Su estudio utiliza información proveniente de una aplicación móvil para calcular tiempo de espera, estado de paraderos y evaluación del servicio, con el propósito de producir reportes útiles para operadores y reguladores. A esta lógica se suman los trabajos basados en GTFS, donde la operación se traduce en métricas comparables y visualizables. Para et al. (2024) muestran que G2Viz fue diseñado para ofrecer métricas como *service rate*, *speed* y *headway*, representadas a lo largo del día mediante recursos gráficos. Estas variables resultan coherentes con el uso de datos GPS, rutas y horarios para evaluar la regularidad y eficiencia de la operación.

1.7.2 Toma de decisiones basada en datos

La utilidad de los indicadores se vuelve más evidente cuando estos se integran a procesos de decisión. La toma de decisiones basada en datos implica pasar de una gestión

principalmente reactiva a una gestión capaz de interpretar patrones, comparar desempeños y priorizar acciones correctivas con evidencia. Albán Reyes (2022) señala que los estándares de servicio y su control mediante indicadores deben incorporarse a los mecanismos de gestión del sistema, lo que convierte al monitoreo en un soporte directo para la acción.

Los estudios revisados muestran, además, que una arquitectura adecuada de datos permite generar análisis más amplios que el simple reporte instantáneo. Liu et al. (2025) sostienen que la transformación de GTFS en redes espaciotemporales habilita análisis detallados de accesibilidad y permite desarrollar evaluaciones personalizadas sobre el sistema. En una línea semejante, Tóth et al. (2025) muestran que el cruce entre GTFS y GTFS-RT permite reconstruir viajes observados, estudiar retrasos, interrupciones y comportamiento histórico del sistema. Estos aportes refuerzan una idea central: el valor analítico no surge automáticamente del dato bruto, sino del tratamiento técnico que permite organizarlo, cruzarlo e interpretarlo.

Vera et al. (2023) aportan una dimensión institucional complementaria al señalar que los datos digitales permiten generar reportes periódicos, e incluso en tiempo real, que ayudan a mantener un registro histórico del desempeño del servicio y respaldan decisiones sobre el estado actual de la red. En ese sentido, la toma de decisiones basada en datos puede entenderse como la capacidad de convertir registros operativos en evidencia útil para operadores y administradores. Para que esa evidencia pueda utilizarse adecuadamente, la visualización se vuelve un componente necesario, ya que facilita la interpretación clara de la información.

1.7.3 Visualización de datos y paneles de control

La visualización de datos permite que los resultados del análisis operativo sean comprensibles para los actores encargados de supervisar el sistema. En este punto, los paneles de control cumplen un papel importante, porque traducen rutas, posiciones, horarios e indicadores en representaciones que facilitan la supervisión, la interpretación y el seguimiento del servicio. Para et al. (2024) muestran que G2Viz responde precisamente a esa necesidad al integrar capacidades de visualización y analítica para apoyar procesos de planificación y operación.

El valor de estos paneles aumenta cuando la visualización permite explorar la información desde distintos niveles. En G2Viz, los datos se presentan mediante mapas, controles y

gráficos asociados a indicadores de desempeño. De manera complementaria, Tóth et al. (2025) describen una visualización interactiva basada en capas de formas, paradas y posiciones vehiculares para analizar comportamiento histórico, retrasos e interrupciones. Ambos trabajos muestran que un panel efectivo no se limita a resumir datos, sino que ayuda a relacionar desempeño, espacio y tiempo dentro de una misma interfaz.

Desde el punto de vista de implementación, esta capacidad depende de que la información haya sido previamente procesada y organizada por el sistema. Por ello, el panel de control no debe verse solo como un recurso gráfico, sino como la interfaz mediante la cual los resultados del procesamiento operativo se vuelven utilizables para supervisión y análisis.

En conjunto, la gestión operativa del transporte se fortalece cuando la información del servicio se transforma en indicadores, esos indicadores se incorporan a procesos de decisión y sus resultados pueden visualizarse con claridad. A partir de esta relación, el siguiente paso consiste en examinar las arquitecturas de software que permiten estructurar y exponer estas funciones de manera ordenada.

1.8 Arquitecturas de software para sistemas de transporte

1.8.1 API REST

En los sistemas de transporte basados en datos, la arquitectura de software cumple un papel decisivo porque define cómo circula la información entre las distintas partes de la solución. En este tipo de plataformas, no basta con recibir datos de ubicación; también es necesario organizar servicios que permitan procesarlos, consultarlos y reutilizarlos desde diferentes módulos. Colque et al. (2021) muestran esta lógica en una arquitectura compuesta por un sistema de localización vehicular automática, un servidor GTFS-Realtime y un sistema web destinado a mostrar alertas, posiciones y actualizaciones de viaje. Este antecedente permite observar la importancia de distribuir responsabilidades entre componentes con funciones claramente diferenciadas.

Un elemento clave dentro de esa organización es la presencia de interfaces que desacoplan la fuente de datos del resto del sistema. Colque et al. (2021) explican que su servidor de aplicación dispone de APIs que obtienen información desde la base de datos y la entregan en formato JSON para la generación posterior de feeds GTFS-Realtime. Este punto muestra el papel de la capa de servicios como mediadora entre almacenamiento, procesamiento y consumo externo. En el caso de una API REST, esta capa permite

exponer información estructurada tanto al panel administrativo como a los módulos encargados de transformación e integración.

Esta separación también responde al principio de modularidad. Velepucha y Flores (2023) señalan que las arquitecturas distribuidas buscan reducir dependencias, facilitar el mantenimiento y mejorar la comprensión del sistema mediante la descomposición de funciones en componentes más manejables. Aunque en este caso no se plantea necesariamente una arquitectura completa de microservicios, el aporte resulta útil como criterio de diseño: evitar que toda la lógica quede concentrada en un único bloque. En un prototipo de transporte, esto implica separar con claridad la recepción de datos, el procesamiento operativo, la generación de GTFS y GTFS-Realtime, y la visualización administrativa.

En esa misma línea, Velepucha y Flores (2023) destacan que patrones como Backend for Frontend y los mecanismos de comunicación entre servicios permiten desacoplar los clientes del núcleo del sistema y ordenar el crecimiento de la solución. Dentro de este enfoque, una API REST funciona como punto de integración entre los distintos componentes: el panel web administrativo, los procesos de transformación de datos y futuras integraciones pueden trabajar sobre una misma capa de servicios reutilizables. Esto reduce la duplicación de lógica y favorece la mantenibilidad del sistema.

1.8.2 Procesamiento de datos en tiempo real

La segunda dimensión crítica de la arquitectura corresponde al procesamiento de datos en tiempo real. En este caso, el desafío no consiste únicamente en recibir información GPS, sino en organizar un sistema capaz de contextualizarla, transformarla y entregarla con baja latencia. Wong y Tsang (2025) describen un sistema orientado a GTFS-RT Vehicle Positions, cuyo flujo integra hardware sensor, firmware, red y software de servidor para producir un feed en tiempo real altamente disponible. Esta aproximación permite entender que el tiempo real no se limita a actualizar posiciones en una interfaz, sino que exige una organización técnica completa del sistema.

En la arquitectura propuesta por Wong y Tsang (2025), el servidor distribuye el trabajo en varios hilos dedicados a la ingestión, el análisis y el servicio HTTP/HTTPS. Además, el análisis no se limita a recibir coordenadas, sino que las cruza con estructuras previamente cargadas, como relaciones vehículo-ruta y árboles de viajes indexados por tiempo, para construir mensajes GTFS-RT válidos. Esto evidencia que el valor del

procesamiento en tiempo real depende de la capacidad de ubicar cada lectura GPS dentro de una estructura operativa más amplia, compuesta por vehículo, ruta, viaje y secuencia temporal.

Mansurova et al. (2025) complementan esta perspectiva al mostrar que una arquitectura orientada al tiempo real requiere mecanismos que aseguren continuidad entre captura, transmisión y almacenamiento. Su trabajo documenta una infraestructura en la que los datos GPS, una vez tratados, pueden sostener monitoreo continuo y análisis posterior dentro de una base estructurada. Este planteamiento refuerza una exigencia arquitectónica importante: el sistema debe mantener coherencia entre entrada, procesamiento y registro para que la información pueda reutilizarse de forma consistente en otros módulos.

Asimismo, Mansurova et al. (2025) muestran que, cuando esa organización se apoya en una infraestructura adecuada, los datos pueden alimentar capas de monitoreo y visualización en tiempo real. Aunque no sea necesario replicar exactamente la misma pila tecnológica, sí se puede recuperar un criterio útil para este proyecto: una arquitectura de transporte orientada al tiempo real necesita un canal de entrada confiable, almacenamiento estructurado y servicios capaces de exponer resultados procesados a otros componentes del sistema. En este contexto, el backend no actúa solo como repositorio, sino como núcleo de procesamiento y distribución de información operativa.

En conjunto, una arquitectura de software para sistemas de transporte puede entenderse a partir de dos pilares complementarios: una capa de servicios, como la API REST, que desacople componentes y exponga funciones reutilizables; y un procesamiento en tiempo real capaz de contextualizar y transformar los datos GPS dentro de una estructura operativa consistente. Con esta base, una plataforma apoyada en GPS institucional, recepción de datos desde el proveedor, backend, PostgreSQL, generación de GTFS/GTFS-Realtime y panel web administrativo puede organizar el flujo de datos del transporte y ponerlo al servicio del análisis operativo y de la información al usuario.

CAPÍTULO II. MATERIALES Y MÉTODOS

En este capítulo se describe el enfoque metodológico adoptado para el desarrollo de la presente investigación, orientada al diseño y construcción de una plataforma digital para el procesamiento y análisis de datos de geolocalización del transporte urbano, con generación de información operativa útil e integración con Google Maps mediante estándares de transporte público. La metodología se estructura para abordar el problema identificado en la Cooperativa 28 de Septiembre y organizar de forma sistemática el proceso de análisis, diseño, desarrollo, demostración, evaluación y comunicación del prototipo propuesto. En este sentido, el capítulo expone las generalidades de la investigación, el marco metodológico del estudio, la incorporación de Scrum en el desarrollo del prototipo y las fases aplicadas para la construcción de la solución tecnológica.

2.1 Generalidades de la investigación

Esta investigación se desarrolló bajo un enfoque aplicado y tecnológico, ya que buscó responder a una necesidad identificada en el contexto operativo de la Cooperativa 28 de Septiembre. El problema se relacionó con la existencia de datos GPS institucionales provenientes de las unidades de transporte, los cuales no contaban con una plataforma que permitiera procesarlos y convertirlos en información útil para la operación del servicio. A partir de esta situación, el estudio se orientó al diseño y construcción de un prototipo como respuesta técnica a una necesidad específica del transporte urbano.

El propósito del trabajo fue diseñar y desarrollar un prototipo capaz de transformar datos de geolocalización del transporte urbano en información operativa para el monitoreo, el análisis de rutas, la estimación de tiempos de llegada y la generación de estructuras interoperables como GTFS y GTFS-Realtime, con miras a su integración con Google Maps. Con ello, se buscó aprovechar de mejor manera los datos GPS institucionales disponibles en la Cooperativa 28 de Septiembre y establecer una base técnica que pudiera servir para futuras ampliaciones del sistema.

El alcance del estudio se delimitó al diseño, desarrollo y validación funcional de un prototipo en un entorno controlado. La propuesta incluyó la recepción y procesamiento de datos GPS, su almacenamiento en una base de datos relacional, la aplicación de reglas de validación y depuración, la reconstrucción de trayectorias, el cálculo de tiempos estimados de llegada y la visualización administrativa mediante un panel web. La investigación no tuvo como finalidad implementar una solución definitiva para toda la

flota, sino demostrar la viabilidad técnica y funcional del prototipo dentro del contexto definido.

2.2 Metodología general del estudio

Para el desarrollo del estudio se adoptó como marco metodológico principal la Design Science Research Methodology (DSRM), debido a que la investigación estuvo orientada a la creación de un artefacto tecnológico destinado a responder una necesidad del transporte urbano. En este caso, la problemática identificada en la Cooperativa 28 de Septiembre no requería únicamente describir la situación existente, sino diseñar, construir y validar una solución capaz de transformar datos GPS en información operativa útil y en estructuras interoperables. Por ello, la DSRM resultó adecuada para organizar el proceso investigativo desde la identificación del problema hasta la demostración, evaluación y comunicación del prototipo propuesto.

De manera complementaria, para la construcción del prototipo se empleó Scrum como metodología de desarrollo. Su aplicación permitió organizar el trabajo técnico de forma iterativa y orientar la implementación progresiva de los componentes del sistema. Así, la DSRM proporcionó la estructura general de la investigación, mientras que Scrum guio la ejecución del artefacto dentro de la fase de diseño y desarrollo.

2.2.1 Fases de la DSRM aplicadas en la investigación

En la presente investigación, la DSRM se aplicó como marco general para organizar el proceso de creación y validación del prototipo. Su uso permitió estructurar el estudio en una secuencia metodológica orientada a la resolución de un problema real mediante el diseño de un artefacto tecnológico.

La fase de identificación del problema y motivación se centró en reconocer la necesidad existente en la Cooperativa 28 de Septiembre respecto al aprovechamiento de los datos GPS institucionales. La fase de planteamiento de la solución permitió definir la propuesta tecnológica y delimitar el alcance funcional del prototipo. La fase de diseño y desarrollo del artefacto comprendió la planificación del trabajo, el diseño técnico de la solución y la construcción progresiva del sistema.

Posteriormente, las fases de demostración y evaluación permitieron comprobar el funcionamiento general del prototipo y valorar sus resultados dentro del entorno de prueba definido para la investigación. Finalmente, la fase de comunicación se materializó en la presente tesis, en la que se documentaron de manera estructurada el problema

abordado, la metodología empleada, el diseño del artefacto, su construcción, su demostración y los resultados obtenidos.

2.3 Desarrollo del prototipo con Scrum

Para el desarrollo del prototipo se empleó Scrum como metodología de trabajo, debido a que permitió organizar la implementación del sistema de forma iterativa y progresiva. Su aplicación facilitó la priorización de funcionalidades, la planificación técnica de las actividades y el seguimiento de los avances obtenidos durante cada sprint.

Aunque Scrum fue concebido originalmente para equipos de trabajo, en esta investigación se aplicó de manera adaptada al contexto de una tesis individual, conservando sus principios de planificación iterativa, priorización de tareas, revisión de avances e incremento funcional del producto.

Su uso se reflejó en la definición de roles, la utilización de artefactos de trabajo y la aplicación de eventos orientados a la planificación, revisión y mejora continua del proceso. Así, el desarrollo del sistema se organizó como una construcción incremental, en la que cada iteración aportó avances verificables al prototipo.

2.3.1 Roles de Scrum asumidos en la investigación

Dado que el presente trabajo se desarrolló en el contexto de una tesis individual, los roles de Scrum fueron asumidos por el mismo investigador, en correspondencia con el carácter del estudio. El rol de Product Owner se vinculó con la definición y priorización de los requerimientos del prototipo a partir de las necesidades identificadas en la Cooperativa 28 de Septiembre. El rol de Scrum Master se relacionó con la organización del trabajo, la planificación de los sprint y el seguimiento del desarrollo. Finalmente, el rol de equipo de desarrollo correspondió a la implementación técnica del sistema, incluyendo la base de datos, el backend, los módulos de procesamiento, la generación de GTFS y GTFS-Realtime, y el panel web administrativo. Esta diferenciación permitió organizar metodológicamente el desarrollo del prototipo dentro del contexto individual de la investigación.

2.3.2 Artefactos de Scrum aplicados al prototipo

La aplicación de Scrum en el prototipo se apoyó en tres artefactos principales: Product Backlog, Sprint Backlog e Incremento. Estos elementos permitieron organizar el trabajo,

traducir los requerimientos en unidades concretas de implementación y evidenciar avances funcionales a lo largo del proceso.

El Product Backlog estuvo conformado por los requisitos funcionales y no funcionales, las historias de usuario y las tareas derivadas definidas para el prototipo. Este artefacto permitió ordenar las funcionalidades de acuerdo con su importancia y dependencia técnica. El Sprint Backlog se reflejó en la distribución de tareas por sprint de dos semanas, seleccionadas según prioridad funcional, tiempo disponible y viabilidad de implementación. Por su parte, el Incremento correspondió a los avances funcionales obtenidos al finalizar cada iteración, materializados en componentes operativos del prototipo.

2.3.3 Eventos de Scrum implementados en la investigación

Los eventos de Scrum se implementaron de forma compatible con el carácter individual del proyecto, manteniendo su finalidad principal de planificación, seguimiento y revisión del trabajo. En este sentido, el desarrollo se organizó en sprint de duración fija, con actividades de planificación al inicio de cada iteración, revisión de resultados al cierre y retrospectiva para identificar ajustes de mejora.

La planificación del sprint permitió definir las tareas que serían desarrolladas en cada iteración, de acuerdo con la prioridad funcional del prototipo y las dependencias técnicas entre módulos. La revisión del sprint se utilizó para contrastar las funcionalidades implementadas con las tareas previstas en cada bloque de trabajo. Por su parte, la retrospectiva del sprint permitió identificar ajustes necesarios para mejorar la continuidad del desarrollo, reorganizar prioridades y corregir dificultades detectadas durante la implementación.

2.4 Fase 1 – Identificación del problema y motivación

La fase de identificación del problema y motivación se enfocó en reconocer la situación existente en la Cooperativa 28 de Septiembre respecto al uso de los datos GPS institucionales. Aunque la cooperativa disponía de información de geolocalización generada por sus unidades, estos datos no eran transformados en información útil para apoyar el seguimiento operativo del servicio. Esta condición evidenció una brecha entre la disponibilidad de registros de ubicación y su aprovechamiento práctico dentro de la operación diaria.

Desde el punto de vista metodológico, esta fase permitió justificar la pertinencia del artefacto propuesto. La problemática no estuvo relacionada con la falta de datos, sino con la ausencia de una solución capaz de estructurarlos, procesarlos y presentarlos de manera funcional para la supervisión del servicio. Por ello, la motivación del estudio surgió de la necesidad de cerrar esa brecha mediante el desarrollo de un prototipo orientado al tratamiento de la información GPS institucional.

Asimismo, la identificación del problema permitió delimitar el sentido de la investigación dentro del contexto específico de la cooperativa. No se buscó proponer una solución genérica para cualquier sistema de transporte, sino responder a una situación concreta vinculada con el uso operativo de la información de geolocalización. Esta fase estableció la base que justificó el desarrollo posterior de la plataforma y orientó las decisiones metodológicas y técnicas adoptadas en las etapas siguientes.

2.5 Fase 2 – Planteamiento de la solución

A partir del problema identificado, se planteó una solución orientada al diseño y desarrollo de un prototipo capaz de recibir, almacenar, procesar y transformar datos GPS en información operativa útil. La propuesta se concibió como una respuesta tecnológica relacionada directamente con la situación detectada en la Cooperativa 28 de Septiembre, con el propósito de dar valor funcional a los registros de geolocalización disponibles.

La solución se estructuró alrededor de una base de datos relacional, una capa de procesamiento backend, módulos de tratamiento operativo de la información, generación de salidas interoperables y un panel web administrativo para consulta. Esta organización permitió definir los componentes esenciales del prototipo y la forma en que debían articularse para responder al problema planteado.

Esta fase también permitió precisar el alcance funcional del artefacto dentro del entorno de investigación. La propuesta no se orientó a implementar un sistema definitivo para toda la flota, sino a desarrollar un prototipo funcional en un entorno controlado, con capacidad para demostrar la viabilidad técnica de la solución. De esta manera, el planteamiento estableció qué debía resolver el prototipo y cómo se proyectaría su desarrollo en las fases posteriores.

2.6 Fase 3 – Diseño y desarrollo del artefacto

En esta fase se integraron la planificación del desarrollo, el diseño técnico de la solución y la construcción progresiva del prototipo. Su propósito fue traducir la solución planteada

en una estructura funcional organizada, capaz de responder al problema identificado mediante una implementación técnica coherente con los requerimientos definidos.

2.6.1 Product Backlog y priorización de funcionalidades

La planificación del desarrollo se apoyó en un Product Backlog conformado por los requisitos funcionales y no funcionales, las historias de usuario y las tareas derivadas del prototipo. Este backlog permitió organizar el trabajo de acuerdo con la prioridad funcional del sistema y con las dependencias técnicas entre sus componentes.

A partir de estos elementos se definió una secuencia de desarrollo progresiva, iniciando con la recepción y almacenamiento de datos GPS, continuando con su validación y procesamiento, y finalizando con la visualización administrativa y la generación de salidas interoperables. Esta organización permitió mantener coherencia entre los requerimientos del prototipo y las actividades previstas para su implementación.

Los requisitos funcionales definidos en esta fase se orientaron principalmente al núcleo operativo del prototipo, es decir, a la recepción y procesamiento de telemetría GPS, la reconstrucción de trayectorias, el cálculo de tiempos estimados de llegada, la generación de estructuras interoperables y la visualización del servicio. De forma complementaria, durante la construcción del sistema se incorporaron funcionalidades de soporte e interacción, como la autenticación de usuarios, el marcado de rutas favoritas y el registro de incidencias ciudadanas. Estas funciones fortalecieron la usabilidad y la supervisión operativa de la plataforma, sin modificar el alcance funcional central definido para el artefacto.

Requisitos funcionales y no funcionales

Tabla 1

Requisitos funcionales

Código	Requisito
RF-1	El sistema debe permitir la recepción de datos de geolocalización provenientes de la fuente disponible del proveedor GPS.
RF-2	El sistema debe almacenar los datos GPS en una base de datos estructurada para su consulta y procesamiento posterior.
RF-3	El sistema debe validar y depurar los datos GPS para tratar registros duplicados, coordenadas inválidas e inconsistencias temporales o espaciales.
RF-4	El sistema debe reconstruir trayectorias de las unidades a partir de los datos GPS procesados.
RF-5	El sistema debe asociar las trayectorias de las unidades con rutas y paradas definidas dentro del prototipo.
RF-6	El sistema debe calcular tiempos estimados de llegada a partir de la información procesada de geolocalización.
RF-7	El sistema debe generar archivos GTFS con la información planificada del servicio de transporte.
RF-8	El sistema debe generar estructuras GTFS-Realtime para posiciones vehiculares y actualizaciones de viaje, conforme al alcance del prototipo.
RF-9	El sistema debe permitir la visualización administrativa de las unidades sobre un mapa.
RF-10	El sistema debe permitir la visualización de indicadores operativos en un panel web administrativo.
RF-11	El sistema debe preparar y estructurar la información del servicio en formatos GTFS y GTFS-Realtime compatibles con Google Maps.
RF-12	El sistema debe permitir la planificación de itinerarios entre un origen y un destino, combinando tramos peatonales y transporte público, considerando restricciones de caminata y tiempos estimados de espera.

Requisitos no funcionales

Tabla 2

Requisitos no funcionales

Categoría	Requisito
Usabilidad	El panel web administrativo debe presentar una interfaz clara, comprensible y organizada para facilitar la consulta de información operativa.
Rendimiento	El sistema debe procesar, consultar y mostrar la información operativa en tiempos adecuados para el entorno de prueba del prototipo.
Integridad de datos	La plataforma debe preservar la consistencia de los datos almacenados, evitando duplicidades e inconsistencias entre captura, validación y registro.
Interoperabilidad	La información generada por la plataforma debe estructurarse en formatos compatibles con GTFS y GTFS-Realtime para facilitar su reutilización externa.
Mantenibilidad	La solución debe organizar sus componentes de forma modular, permitiendo separar recepción, procesamiento, almacenamiento y visualización.
Confiabilidad	El sistema debe mantener continuidad entre la entrada de datos, su procesamiento y su almacenamiento, evitando pérdidas de información durante el flujo principal.

Una vez definidos los requisitos, se elaboraron las historias de usuario con el fin de traducir esas necesidades en objetivos concretos de desarrollo. Estas historias permitieron expresar de forma más clara qué se esperaba del prototipo, cuál era la prioridad de cada funcionalidad y qué condiciones debían cumplirse para considerarla aceptada. La Tabla 3 recoge las historias de usuario planteadas para la plataforma propuesta.

Tabla 3*Historias de usuario de la plataforma propuesta*

ID	Nombre breve	Estimación de esfuerzo	Importancia	Descripción de historia de usuario	Dependencias	Criterios de aceptación
H U-1	Recepción de datos GPS	12 h	900	Como administrador, quiero que el sistema reciba datos de geolocalización desde la fuente disponible del proveedor GPS para disponer de información actualizada de las unidades.	Acceso a la fuente de datos del proveedor GPS	El sistema recibe registros con los campos mínimos definidos, los registra sin errores críticos y conserva su marca temporal.
H U-2	Almacenamiento estructurado de datos	10 h	850	Como administrador, quiero que los datos GPS se almacenen de forma estructurada para consultarlos y utilizarlos en procesos posteriores de análisis y transformación.	HU-1	Los datos quedan almacenados en PostgreSQL con estructura consistente y pueden recuperarse correctamente mediante consulta.
H U-3	Validación y depuración de datos GPS	14 h	950	Como administrador, quiero validar y depurar los datos GPS para evitar que registros erróneos afecten el análisis operativo y las	HU-2	El sistema identifica y trata duplicados, coordenadas inválidas, saltos imposibles e inconsistencias

				estimaciones de llegada.		temporales, generando un conjunto de datos limpio.
H U- 4	Reconstrucción de trayectorias	14 h	800	Como administrador, quiero reconstruir trayectorias de las unidades a partir de los datos GPS para analizar recorridos y comportamiento operativo.	HU-3	El sistema organiza los puntos GPS por unidad y tiempo, reconstruye trayectorias y permite su consulta para análisis posterior.
H U- 5	Asociación de unidades con rutas y paradas	12 h	850	Como administrador, quiero relacionar las trayectorias de las unidades con rutas y paradas para contextualizar operativamente la información de geolocalización.	HU-4	El sistema vincula las trayectorias observadas con las rutas y paradas definidas en el prototipo sin inconsistencias relevantes.
H U- 6	Estimación de tiempos de llegada	16 h	1000	Como administrador, quiero generar y consultar tiempos estimados de llegada para disponer de información operativa útil y habilitar su posterior publicación.	HU-3, HU-5	El sistema calcula tiempos estimados de llegada a partir de datos procesados y muestra resultados coherentes con el

						estado del servicio.
H U- 7	Generación de GTFS	14 h	750	Como administrador, quiero generar archivos GTFS para estructurar la información planificada del servicio en un formato interoperable.	HU-5	El sistema genera archivos GTFS válidos para las rutas, paradas, viajes y horarios contemplados en el prototipo.
H U- 8	Generación de GTFS-Realtime	14 h	900	Como administrador, quiero generar información en formato GTFS-Realtime para representar posiciones vehiculares y actualizaciones de viaje del servicio.	HU-3, HU-5, HU-6	El sistema genera estructuras GTFS-Realtime válidas para posiciones vehiculares y actualizaciones de viaje, sin errores estructurales críticos.
H U- 9	Visualización administrativa en mapa	12 h	700	Como administrador, quiero visualizar la ubicación de las unidades en un mapa para supervisar la operación en tiempo real.	HU-3, HU-5	El panel muestra la ubicación de las unidades en el mapa con datos procesados y sin errores críticos de representación.
H U- 10	Visualización de	12 h	650	Como administrador, quiero visualizar	HU-4, HU-5, HU-6	El panel presenta indicadores

	indicadores operativos			indicadores operativos en un panel web para analizar el desempeño de las rutas y unidades.		definidos para el prototipo y permite su consulta sin errores de carga.
H U-11	Preparación de datos para Google Maps	10 h	600	Como administrador, quiero preparar datos compatibles con Google Maps mediante GTFS y GTFS-Realtime para facilitar su integración técnica con plataformas externas de consulta.	HU-7, HU-8	Los datos generados cumplen la estructura requerida por GTFS y GTFS-Realtime y quedan listos para validación técnica.

A partir de las historias de usuario, se identificaron las tareas específicas necesarias para llevar cada funcionalidad a la práctica. Esta descomposición permitió organizar de manera más precisa el trabajo técnico, estimar el esfuerzo requerido y establecer condiciones claras para verificar la finalización de cada actividad. La Tabla 4 muestra las tareas derivadas de las historias de usuario, organizadas de acuerdo con su prioridad.

Tabla 4*Tareas derivadas de las historias de usuario, organizadas por prioridad*

Prioridad	N.º HU	ID de tarea	Tarea	Horas	Condición de DONE
1000	HU-6	T6-1	Definir variables de entrada y reglas base para el cálculo del ETA	3 h	Variables, supuestos y lógica base del ETA definidos por escrito y listos para implementación.
		T6-2	Implementar cálculo de distancia entre unidad y siguiente parada	4 h	El sistema calcula correctamente la distancia operativa entre la unidad y la parada objetivo.
		T6-3	Implementar lógica de estimación de tiempo de llegada	5 h	El sistema genera ETA a partir de datos procesados sin errores críticos de ejecución.
		T6-4	Validar y ajustar el cálculo del ETA con casos de prueba	4 h	Resultados contrastados con datos de prueba y ajustes realizados cuando existen desviaciones relevantes.
950	HU-3	T3-1	Definir reglas de validación y depuración de registros GPS	3 h	Reglas para duplicados, coordenadas inválidas e inconsistencias temporales o espaciales documentadas.
		T3-2	Implementar tratamiento de duplicados y coordenadas inválidas	5 h	El sistema detecta y corrige o excluye registros duplicados y coordenadas no válidas.
		T3-3	Implementar control de inconsistencias temporales y desplazamientos imposibles	3 h	El sistema identifica saltos imposibles o secuencias temporales incoherentes en los registros.
		T3-4	Probar el módulo de depuración con datos reales o simulados	3 h	Módulo probado y salida limpia verificada sin errores críticos.
900	HU-1	T1-1	Analizar la estructura de datos entregada por la fuente GPS	3 h	Campos mínimos requeridos identificados

					y documentados para el flujo del prototipo.
		T1-2	Implementar el módulo de recepción de datos GPS	5 h	El sistema recibe registros desde la fuente disponible y los procesa sin fallos críticos.
		T1-3	Verificar recepción continua y registro de fecha y hora	4 h	Flujo de recepción comprobado con marcas temporales correctas en los registros almacenados.
900	HU-8	T8-1	Definir la estructura de salida para GTFS-Realtime	3 h	Estructura de feeds definida para posiciones vehiculares y actualizaciones de viaje según el alcance del prototipo.
		T8-2	Implementar generación de VehiclePosition	5 h	El sistema genera posiciones vehiculares en formato GTFS-Realtime sin errores estructurales.
		T8-3	Implementar generación de TripUpdate	3 h	El sistema genera actualizaciones de viaje con estructura válida y datos consistentes.
		T8-4	Validar la estructura del GTFS-Realtime generado	3 h	Feed revisado y validado técnicamente sin errores formales críticos.
850	HU-2	T2-1	Diseñar el esquema de base de datos para almacenamiento GPS	3 h	Tablas, relaciones y campos definidos de forma consistente con el flujo del prototipo.
		T2-2	Implementar almacenamiento de registros GPS en PostgreSQL	4 h	Registros almacenados correctamente y disponibles para consulta posterior.
		T2-3	Validar integridad y recuperación de los datos almacenados	3 h	Consultas ejecutadas con resultados correctos y sin inconsistencias de almacenamiento.

850	HU-5	T5-1	Definir correspondencia entre unidades, rutas y paradas del prototipo	3 h	Relaciones operativas documentadas para las unidades, rutas y paradas definidas.
		T5-2	Implementar asociación de trayectorias con rutas	4 h	El sistema vincula trayectorias observadas con la ruta correspondiente sin errores críticos.
		T5-3	Implementar vinculación de trayectorias con paradas	3 h	El sistema identifica la relación entre trayectoria y paradas definidas dentro del prototipo.
		T5-4	Validar asociaciones con casos de prueba	2 h	Asociación unidad-ruta-parada verificada con resultados coherentes.
800	HU-4	T4-1	Ordenar registros GPS por unidad y secuencia temporal	3 h	Registros organizados correctamente por unidad y tiempo para su procesamiento posterior.
		T4-2	Implementar reconstrucción de trayectorias a partir de datos GPS	5 h	Trayectorias generadas correctamente para las unidades incluidas en el prototipo.
		T4-3	Validar continuidad y consistencia de trayectorias reconstruidas	3 h	Trayectorias comprobadas sin discontinuidades relevantes para los casos de prueba.
		T4-4	Preparar salida de trayectorias para análisis o visualización	3 h	Trayectorias disponibles para consumo posterior por módulos de análisis o panel.
750	HU-7	T7-1	Definir archivos y campos GTFS requeridos para el prototipo	3 h	Estructura mínima de GTFS documentada conforme al alcance definido.
		T7-2	Implementar generación de archivos base de GTFS	4 h	Archivos generados con información de rutas, paradas y viajes del prototipo.

		T7-3	Implementar generación de tiempos y recorridos asociados	4 h	Horarios y recorridos estructurados correctamente dentro del conjunto GTFS generado.
		T7-4	Validar la estructura y consistencia del GTFS generado	3 h	Archivos revisados y validados sin errores estructurales críticos.
700	HU-9	T9-1	Diseñar la vista de mapa del panel administrativo	3 h	Vista de mapa definida e implementada con estructura funcional.
		T9-2	Integrar posiciones de las unidades sobre el mapa	4 h	El mapa muestra correctamente la ubicación de las unidades con datos procesados.
		T9-3	Incorporar información básica de contexto por unidad	2 h	La vista permite identificar unidad y relación operativa básica en el mapa.
		T9-4	Probar actualización y representación en mapa	3 h	Visualización validada con datos reales o simulados sin errores críticos de carga.
650	HU-10	T10-1	Definir indicadores operativos a mostrar en el panel	2 h	Indicadores seleccionados y documentados conforme al alcance del prototipo.
		T10-2	Implementar consultas o cálculos para indicadores operativos	4 h	Indicadores calculados correctamente a partir de los datos procesados.
		T10-3	Implementar componentes visuales para indicadores	3 h	Panel muestra indicadores con estructura legible y funcional.
		T10-4	Validar consistencia y legibilidad de la visualización	3 h	Resultados probados y presentados sin errores de interpretación o carga.
600	HU-11	T11-1	Revisar los criterios de compatibilidad de GTFS y GTFS-	3 h	Requisitos técnicos identificados y contrastados con las

			Realtime con Google Maps		salidas generadas por el prototipo.
		T11-2	Ajustar la estructura de los archivos y feeds generados para cumplir con los formatos requeridos	4 h	Archivos o feeds preparados conforme a la estructura requerida para validación técnica.
		T11-3	Validar técnicamente la consistencia de los datos preparados para su integración con Google Maps	3 h	Datos revisados y verificados sin errores formales críticos.

2.6.2 Planificación de sprint e incremento del prototipo

Con las tareas ya definidas, el siguiente paso consistió en distribuir las en sprint de duración aproximada, siguiendo la organización iterativa del trabajo establecida con Scrum. Esta planificación se realizó considerando el orden técnico de implementación de los componentes del sistema, desde la recepción y tratamiento de los datos GPS hasta la visualización administrativa y la generación de estructuras interoperables.

La distribución por sprint permitió organizar el desarrollo en bloques de trabajo consecutivos, cada uno con un conjunto específico de tareas priorizadas. Esto facilitó la construcción progresiva del prototipo y el seguimiento de avances parciales durante el proceso de implementación.

Tabla 5

Planificación de tareas distribuidas por sprint

Sprint	Duración	Tareas asignadas
Sprint 1	2 semanas	T1-1, T1-2, T1-3, T2-1, T2-2
Sprint 2	2 semanas	T2-3, T3-1, T3-2, T3-3, T3-4
Sprint 3	2 semanas	T4-1, T4-2, T4-3, T4-4, T5-1
Sprint 4	2 semanas	T5-2, T5-3, T5-4, T6-1, T6-2
Sprint 5	2 semanas	T6-3, T6-4, T7-1, T7-2, T7-3
Sprint 6	2 semanas	T7-4, T8-1, T8-2, T8-3, T8-4
Sprint 7	2 semanas	T9-1, T9-2, T9-3, T9-4, T10-1, T10-2
Sprint 8	1 semana	T10-3, T10-4, T11-1, T11-2, T11-3

Al finalizar cada sprint se obtuvo un incremento funcional del prototipo, entendido como un avance verificable dentro del sistema. Estos incrementos correspondieron a la

incorporación progresiva de módulos y componentes necesarios para consolidar la solución propuesta.

2.6.3 Diseño de la solución

El diseño de la solución permitió definir la estructura técnica del prototipo antes de su implementación. En esta etapa se establecieron los componentes que intervendrían en el funcionamiento del sistema, sus relaciones principales, la lógica de procesamiento y la organización visual de la plataforma. De este modo, el diseño constituyó la base técnica que orientó la construcción posterior del prototipo.

2.6.3.1 Arquitectura general del sistema

La arquitectura general del sistema se definió bajo un enfoque distribuido y orientado a servicios, con el propósito de organizar los componentes principales del prototipo y mantener un flujo de información claro entre ellos. A partir de este criterio, el ciclo de vida de los datos de transporte se estructuró en cuatro capas principales: ingestión de datos, persistencia, procesamiento para interoperabilidad y visualización.

En la capa de ingestión y backend se concentraron la lógica de negocio y el procesamiento principal de la información. Para ello, se desarrolló un backend con NestJS, sobre el entorno de ejecución de Node.js y el lenguaje TypeScript. Este componente expuso una API REST para la comunicación con el cliente y, de forma complementaria, incorporó servicios encargados de consumir los flujos de geolocalización provenientes de los proveedores GPS de las unidades. Dentro de esta misma capa se integraron las reglas de validación de entrada, la depuración de ruido geográfico, el cálculo del rumbo o azimut de los vehículos y la estimación del tiempo de llegada (ETA). Estos procesos se realizaron a partir de la asociación entre las coordenadas reportadas, la geometría de las rutas y la estructura operativa definida en el sistema.

La capa de persistencia se implementó con PostgreSQL como motor de base de datos relacional, debido a su capacidad para almacenar de manera estructurada la información relacionada con rutas, paradas, itinerarios, unidades y registros históricos de geolocalización. La comunicación entre el backend y la base de datos se gestionó mediante Prisma ORM, lo que permitió conservar la integridad del esquema de datos, simplificar la administración del modelo relacional y facilitar las consultas requeridas para el tratamiento espacio-temporal de la información del prototipo.

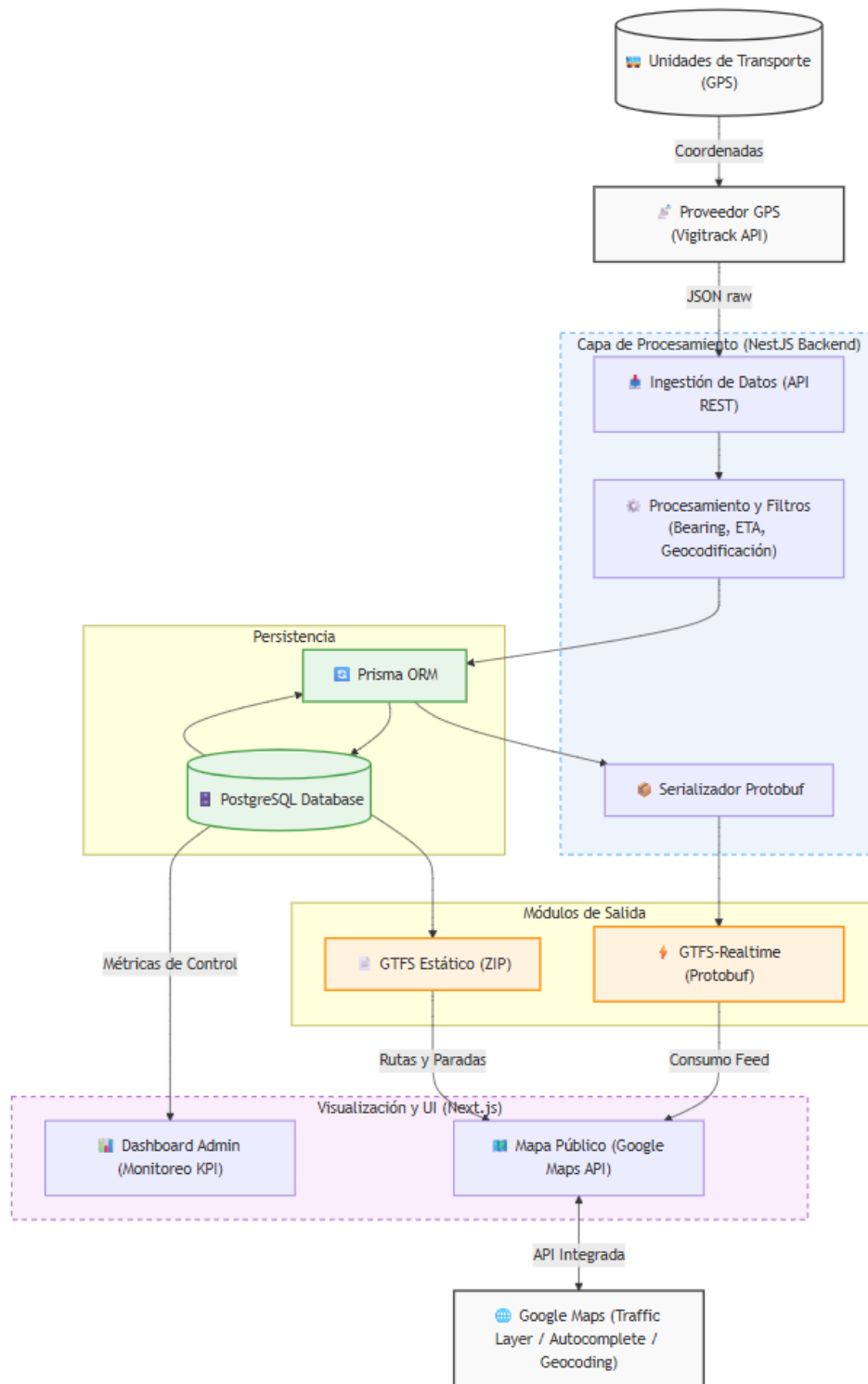
Sobre esta base se definió el módulo de interoperabilidad, encargado de estructurar la información conforme al estándar internacional de transporte público GTFS y a su extensión GTFS-Realtime. En su dimensión estática, el sistema generó archivos de texto estructurado como stops.txt, trips.txt y shapes.txt, entre otros requeridos por el modelo de datos del servicio. En su dimensión dinámica, el módulo serializó actualizaciones de viaje (TripUpdates), posiciones de vehículos (VehiclePositions) y alertas de servicio (Alerts) mediante Protocol Buffers (Protobuf). Con ello, se buscó cumplir la especificación de GTFS-Realtime y permitir que agentes externos pudieran consumir información interoperable en tiempo real.

La capa de visualización y experiencia de usuario se construyó sobre Next.js, framework utilizado para desarrollar la interfaz web del prototipo. Esta capa se organizó en dos entornos principales. El primero correspondió al panel administrativo, orientado al monitoreo de indicadores clave de rendimiento, como puntualidad, velocidad promedio y cobertura del servicio, con el fin de apoyar la supervisión operativa. El segundo correspondió al mapa público de usuario, concebido para ofrecer una interacción visual directa con la operación del transporte. Este módulo integró la API de Google Maps mediante la librería @react-google-maps/api, lo que permitió visualizar la ubicación en tiempo real de los autobuses mediante marcadores SVG dinámicos con rotación automática según el rumbo reportado por el GPS. De forma complementaria, se incorporó la capa de tráfico en tiempo real (TrafficLayer), la geocodificación inversa para autocompletar puntos de origen y destino a partir de eventos sobre el mapa, y la búsqueda predictiva de direcciones mediante Places Autocomplete.

Finalmente, para facilitar el despliegue, la disponibilidad y la escalabilidad del prototipo, la arquitectura completa se orquestó en la nube mediante Railway. En esta plataforma se desplegaron la base de datos relacional, la API desarrollada en NestJS y el frontend construido en Next.js. En conjunto, la arquitectura permitió integrar en una misma solución la captura de datos GPS, su persistencia estructurada, el procesamiento operativo e interoperable, y la visualización administrativa y pública del sistema.

Figura 1

Arquitectura general del sistema



Como se observa en la Figura 1, la arquitectura general del sistema organiza el flujo de información desde la captura de la geolocalización de las unidades hasta su procesamiento, almacenamiento y visualización. A partir de esta estructura, el backend gestiona la validación y el tratamiento de los datos, PostgreSQL conserva la información

operativa y los módulos de interoperabilidad generan salidas bajo el estándar GTFS. Sobre esta base se apoyan tanto el panel administrativo como el mapa público integrado con Google Maps.

2.6.3.2 Diagrama de casos de uso

El diagrama de casos de uso permitió delimitar el alcance funcional del prototipo mediante la representación de las interacciones entre los distintos actores del sistema y las funciones principales que ofrece la plataforma. Para este diseño, los actores se organizaron en dos grupos: actores primarios, correspondientes a los usuarios que interactúan directamente con la solución, y actores secundarios, vinculados con servicios externos que proveen o consumen información dentro del flujo operativo del prototipo.

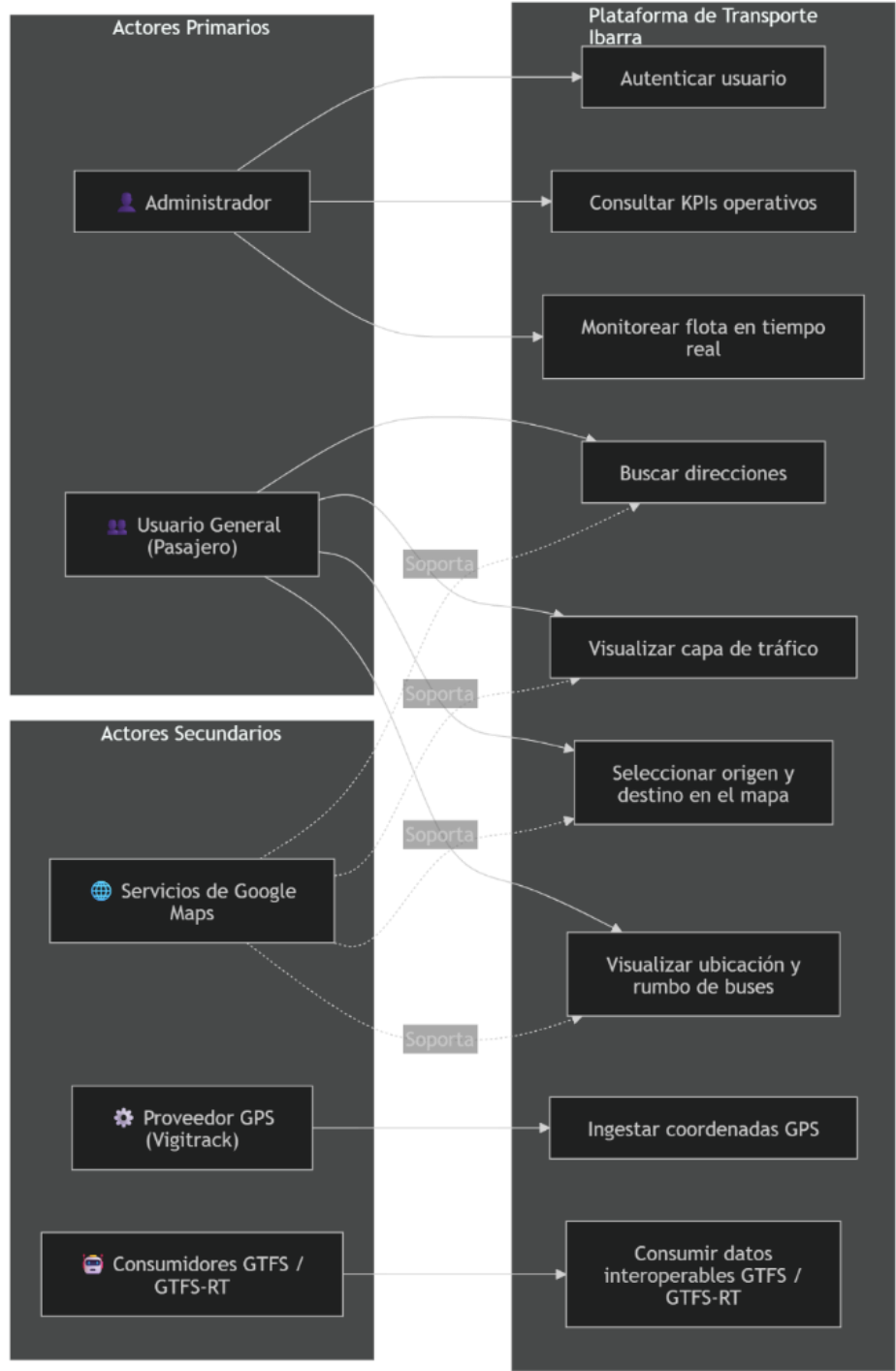
Entre los actores primarios se identificó, en primer lugar, al administrador, responsable de la gestión interna y de la supervisión del sistema. Sus principales casos de uso incluyen la autenticación en la plataforma, la consulta de indicadores clave de rendimiento y el monitoreo de la flota en tiempo real. En segundo lugar, se definió al usuario general, entendido como pasajero o ciudadano que accede a la plataforma pública de forma abierta. Sus casos de uso comprenden la búsqueda de direcciones, la visualización de la capa de tráfico, la selección de origen y destino en el mapa y la consulta en tiempo real de la ubicación y el rumbo de los buses.

Dentro de los actores secundarios se identificó al proveedor GPS, representado por la API externa de Vigitrack, cuya función consiste en suministrar de forma continua las coordenadas geográficas de las unidades hacia el backend. De igual manera, se consideraron los servicios de Google Maps como componentes externos que proporcionan el soporte cartográfico, la capa de tráfico y las funcionalidades de búsqueda asociadas a la interacción con el mapa. Finalmente, se incluyó al consumidor de datos interoperables, correspondiente a aplicaciones de terceros o servicios externos que utilizan la información generada por el prototipo en formatos GTFS y GTFS-Realtime.

Las funciones principales representadas en el diagrama se organizaron en tres grupos: administración del sistema, interacción del usuario con el mapa y automatización del flujo de datos. En el ámbito administrativo se incluyeron la autenticación, la consulta de indicadores operativos y el monitoreo de la flota. En la interacción ciudadana se contemplaron la búsqueda de direcciones, la visualización de la capa de tráfico, la selección de origen y destino sobre el mapa y la consulta de la ubicación y rumbo de los

buses. Por su parte, en la automatización del flujo de datos se representaron la ingestión de coordenadas GPS y el consumo de salidas interoperables. De esta manera, el diagrama de casos de uso permitió establecer con claridad las fronteras funcionales del prototipo y la relación entre actores internos, servicios externos y funciones principales del sistema.

Figura 2
Diagrama de casos de uso de la plataforma propuesta



2.6.3.3 Diseño de la base de datos

Como parte del diseño técnico del prototipo, se implementó una base de datos relacional en PostgreSQL, cuya interacción y mapeo objeto-relacional se gestionó mediante Prisma ORM. Esta estructura se definió para almacenar, de manera organizada, tanto los datos GPS recibidos desde las unidades de transporte como la información necesaria para la visualización del sistema, el análisis operativo y la generación de salidas interoperables en formatos GTFS y GTFS-Realtime.

A partir del modelo entidad-relación definido para el prototipo, las tablas del sistema se organizaron en cuatro módulos funcionales. El primero correspondió al módulo de infraestructura y operación de transporte, encargado de almacenar la información estática del servicio. En este grupo se incluyeron las tablas unidades, destinada a registrar el identificador interno, la placa, el código de unidad y el estado operativo de cada autobús; rutas, orientada a almacenar la información general de las líneas de transporte, incluyendo origen, destino, nombre de la ruta, coordenadas de salida y llegada, además de parámetros de control relacionados con radios y tiempos mínimos y máximos de recorrido; y paradas, donde se registró la ubicación georreferenciada y el nombre de cada punto autorizado de ascenso y descenso. A este mismo módulo pertenecieron ruta_paradas, que definió la secuencia ordenada y el sentido del recorrido de las paradas dentro de cada ruta, y ruta_shapes, que almacenó la secuencia de coordenadas utilizada para representar el trazado geométrico de las rutas sobre el mapa.

El segundo grupo correspondió al módulo de telemetría y monitoreo en tiempo real, orientado al tratamiento de la información dinámica proveniente del sistema GPS. En esta categoría se incluyó la tabla registros_gps, diseñada para conservar el historial de transmisiones de cada unidad, almacenando fecha, hora, coordenadas, velocidad, rumbo y otros atributos asociados al comportamiento operativo del vehículo. De forma complementaria, la tabla trayectorias permitió registrar el inicio y el fin de los recorridos ejecutados por cada unidad, así como su estado dentro de la operación del sistema.

El tercer grupo se definió como módulo de estimaciones y tiempos de recorrido, destinado al almacenamiento de resultados derivados del procesamiento operativo. Dentro de este conjunto se incluyó la tabla estimaciones_eta, encargada de registrar las proyecciones de tiempo estimado de llegada hacia las paradas junto con su estado de confiabilidad; tiempos_tramo, utilizada para conservar la duración histórica de desplazamiento entre paradas consecutivas y apoyar el refinamiento de los cálculos de ETA; y

pasos_parada_actual, diseñada para identificar la última parada registrada dentro del trayecto activo de una unidad.

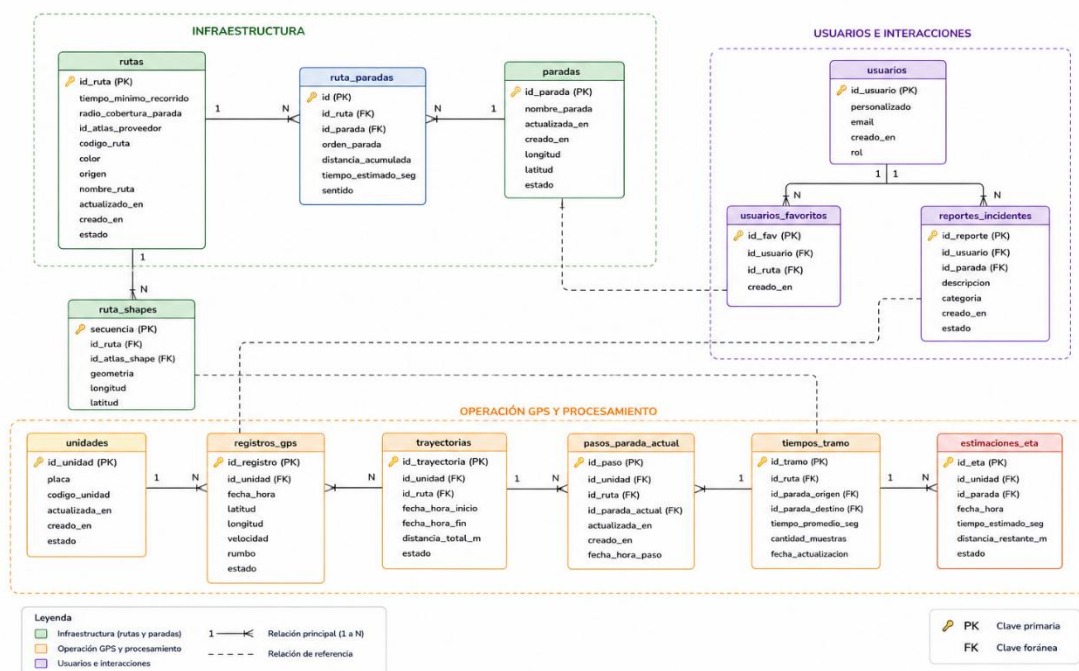
El cuarto grupo correspondió al módulo de usuarios e interacción social, incorporado para gestionar la relación entre la plataforma y sus usuarios. En este módulo se incluyeron las tablas usuarios, usuarios_favoritos y reportes_incidentes, destinadas a administrar el acceso, la personalización de rutas y el registro de eventos reportados por los usuarios.

Además, el diseño de la base de datos contempló criterios de optimización y consistencia relacional. Debido a la frecuencia de transmisión de los datos GPS, se definieron índices sobre tablas críticas como registros_gps, trayectorias, estimaciones_eta y tiempos_tramo, priorizando campos relacionados con unidad, ruta y marcas temporales. Esto permitió agilizar las consultas requeridas por el monitoreo y la visualización en tiempo real. De igual forma, se establecieron restricciones de unicidad en relaciones clave para evitar duplicidades y preservar la integridad del modelo de datos.

En conjunto, esta estructura permitió articular la información estática del servicio, los datos dinámicos de geolocalización, los resultados del procesamiento operativo y la interacción con los usuarios dentro de un mismo modelo relacional. Con ello, la base de datos proporcionó un soporte coherente para el funcionamiento integral del prototipo.

Figura 3

Modelo entidad-relación de la base de datos



Como se aprecia en la Figura 3, la base de datos se estructuró como un modelo relacional orientado a integrar, dentro de una misma arquitectura, tanto la información estática del sistema de transporte como los datos dinámicos de geolocalización, las estimaciones operativas y los elementos de interacción con el usuario. Esta organización permitió sostener de forma coherente el almacenamiento histórico de registros GPS, el cálculo de resultados operativos como trayectorias y tiempos estimados de llegada, y la preparación de información interoperable para los módulos GTFS y GTFS-Realtime dentro del prototipo.

2.6.3.4 Componentes de la arquitectura del sistema

El diagrama de componentes permitió representar la organización interna de los módulos lógicos que conforman la plataforma digital, precisando la distribución de responsabilidades de software, sus dependencias y las interfaces mediante las cuales se comunican. Este diseño ayudó a estructurar el prototipo de forma desacoplada, evitando concentrar toda la lógica en un único bloque y favoreciendo la mantenibilidad, la escalabilidad y la evolución independiente de sus componentes.

La arquitectura del prototipo se organizó en cuatro grupos principales. El primero correspondió a los componentes de ingestión e integración externa, encargados de capturar y validar inicialmente la información proveniente de los dispositivos GPS. En este grupo se ubicó el servicio responsable de programar y ejecutar consultas automáticas hacia la API externa del proveedor GPS en intervalos regulares, junto con el controlador de ingestión, encargado de recibir, validar y filtrar las tramas de datos crudas antes de incorporarlas al flujo operativo del sistema.

El segundo grupo estuvo conformado por los componentes de procesamiento central y lógica espacio-temporal, responsables del tratamiento operativo de la información. En esta categoría se incluyó el módulo de reconstrucción de trayectorias y asociación con rutas, cuya función fue agrupar registros GPS individuales para representar recorridos continuos y verificar su correspondencia con la estructura física del servicio. También se integró el componente de predicción de tiempos estimados de llegada, encargado de asociar la última posición reportada de cada unidad con los puntos de control definidos en las paradas y calcular el ETA con apoyo en registros históricos de desplazamiento entre tramos. Como apoyo a estos procesos, se incorporó un componente utilitario para operaciones geométricas, utilizado en el cálculo de rumbos, distancias y relaciones espaciales entre coordenadas.

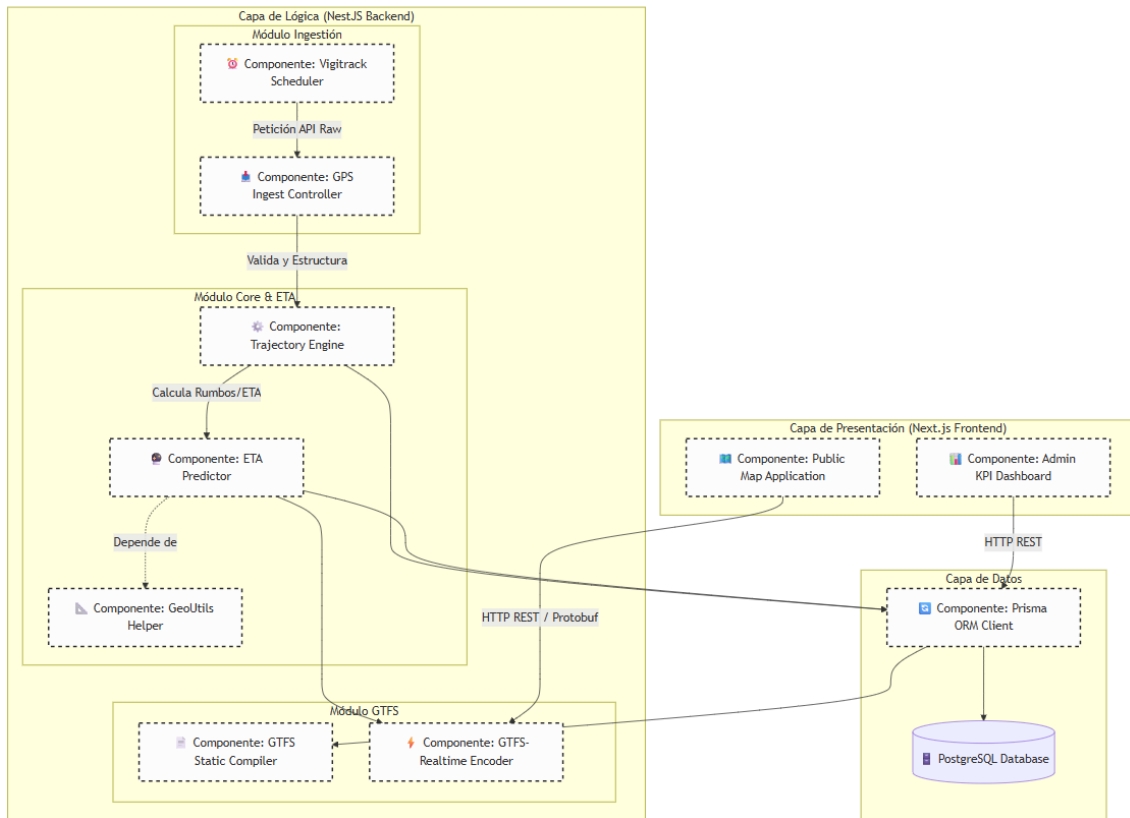
El tercer grupo correspondió a los componentes de persistencia e interoperabilidad. En esta parte de la arquitectura se ubicó el cliente de Prisma como capa de abstracción para las operaciones de lectura y escritura sobre PostgreSQL, facilitando el acceso tipado al modelo relacional. Junto a ello, se integraron los módulos encargados de generar salidas interoperables. Por un lado, el componente de generación de GTFS estático estructuró la información histórica y operativa en archivos comprimidos conforme al estándar. Por otro lado, el codificador de GTFS-Realtime tradujo el estado dinámico de los buses y las estimaciones ETA al formato Protocol Buffers (Protobuf), exponiendo canales de salida como TripUpdates y VehiclePositions.

El cuarto grupo estuvo integrado por los componentes de visualización, desarrollados sobre el frontend en Next.js. En este ámbito se distinguieron dos interfaces principales: el panel administrativo, orientado a consultar y representar indicadores clave de rendimiento relacionados con la puntualidad, la cobertura y el comportamiento del servicio; y la aplicación pública de mapa, destinada al usuario general para visualizar buses en movimiento, consultar la capa de tráfico urbano y buscar direcciones mediante servicios de geocodificación y autocompletado integrados con Google Maps.

En conjunto, esta organización permitió definir canales claros de comunicación mediante protocolos y formatos estándar, como REST, JSON y Protocol Buffers, además de interfaces seguras hacia la base de datos. De esta manera, cada componente asumió una responsabilidad específica dentro del sistema, sin afectar la coherencia funcional del prototipo.

Figura 4

Diagrama de componentes de la plataforma digital



2.6.3.5 Algoritmos principales del prototipo

Dentro del diseño técnico del prototipo se definieron los procedimientos lógicos encargados de transformar los registros de geolocalización crudos en información operativa estructurada y en funciones orientadas a la planificación de recorridos. Estos algoritmos conformaron el núcleo de procesamiento del backend, ya que permitieron convertir las tramas provenientes del proveedor GPS en variables útiles para el monitoreo de flota, la estimación de tiempos de llegada, la planificación de viajes y la generación de salidas interoperables. Para su implementación, la lógica del sistema se organizó en siete etapas secuenciales e interrelacionadas.

El primer procedimiento correspondió a la ingestión, validación y depuración de datos GPS, concebido como un filtro inicial de calidad antes de persistir la información. Cada trama recibida desde el proveedor incluyó, al menos, el identificador de la unidad, coordenadas geográficas, velocidad reportada y marca temporal. Sobre esta entrada, el sistema aplicó verificaciones de integridad estructural para asegurar la existencia de campos obligatorios, controles de consistencia geográfica para excluir coordenadas fuera

de la envolvente espacial definida para el área metropolitana de Ibarra y un filtro de remoción de duplicados mediante el rechazo de tramas idénticas ya registradas en la base de datos. Además, para mitigar saltos temporales incoherentes o físicamente improbables, el control de velocidad y duración se trasladó a las capas lógicas superiores de procesamiento de tramos y estimación, evitando que el ruido del sensor afectara directamente el cálculo operativo.

El segundo procedimiento fue la reconstrucción dinámica de trayectorias. Su propósito consistió en agrupar los registros GPS de una misma unidad de transporte y ordenarlos cronológicamente para representar recorridos observables dentro del sistema. A partir de esta secuencia, el algoritmo aplicó una lógica basada en estados operativos. Una trayectoria fue clasificada como EN_CURSO cuando la unidad transmitía regularmente dentro de una ruta asignada; como COMPLETADA o INCOMPLETA cuando se detectaba un cierre del recorrido por inactividad prolongada o por cruce de los terminales oficiales de inicio y fin; y como DESVÍO_PROLONGADO cuando la distancia respecto a la geometría oficial de la ruta superaba un umbral de tolerancia. Esta etapa permitió representar el comportamiento real del vehículo a partir de la continuidad temporal y de su coherencia con la infraestructura del servicio.

El tercer procedimiento correspondió a la asociación con rutas y secuencia de paradas, implementado mediante una lógica de *map matching* entre las trayectorias reconstruidas, el trazado geométrico almacenado en RutaShape y la secuencia ordenada de paradas registrada en RutaParada. Su objetivo fue traducir coordenadas geográficas en variables propias del transporte público. Para ello, el algoritmo utilizó dos límites de influencia espacial diferenciados: un radio de control de 125 m en los puntos iniciales y finales para la apertura y cierre de trayectorias, y un radio de influencia de 80 m para la detección geoespacial del paso por paradas intermedias. Cuando el vehículo se alejaba de forma sistemática de los límites de tolerancia, el sistema evitaba asociaciones artificiales y asignaba el estado FUERA_DE_RUTA. Esta etapa permitió identificar con mayor precisión la correspondencia entre la posición del vehículo y la secuencia de paradas relevante para el cálculo posterior del ETA.

El cuarto procedimiento fue el algoritmo híbrido de estimación del tiempo de llegada (ETA), definido para calcular de forma acumulativa el tiempo estimado de arribo hacia cada parada objetivo. La lógica inició con la determinación de la velocidad utilizada en el cálculo, denotada como V_u . Para evitar fluctuaciones extremas y errores de predicción

causados por detenciones momentáneas, congestión o aceleraciones bruscas, la velocidad se acotó mediante la siguiente regla de limitación:

$$V_u = \begin{cases} \frac{18 \text{ km}}{h}, & \text{si } V_{gps} < \frac{5 \text{ km}}{h} \\ \min\left(V_{gps}, \frac{30 \text{ km}}{h}\right), & \text{si } V_{gps} \geq \frac{5 \text{ km}}{h} \end{cases}$$

Una vez estabilizada la velocidad, el algoritmo determinó dinámicamente el siguiente punto de control, denotado como P_{sig} . Debido a que el servicio puede operar sobre rutas circulares o con giros en U, el sistema evaluó continuamente las dos paradas siguientes dentro de la secuencia ordenada, $(i+1)$ e $(i+2)$, seleccionando aquella con la distancia euclidiana más corta respecto a la posición actual de la unidad. Esta lógica evitó bloqueos en el cálculo cuando un autobús omitía una parada de la secuencia oficial o presentaba una transición espacial no estrictamente lineal.

Con el siguiente punto de control definido, el tiempo estimado de llegada hacia una parada posterior $Destino_j$ se calculó como la suma del tiempo requerido para alcanzar la primera parada objetivo más la acumulación de los tiempos de los tramos posteriores, conforme a la siguiente expresión:

$$T_j = t_1 + \sum_{k=sig+1}^j t_k$$

donde T_j representa el tiempo estimado de llegada hacia la parada destino j , t_1 corresponde al tiempo del primer tramo y $\sum t_k$ representa la acumulación de los tiempos de los tramos siguientes. El primer tramo se obtuvo en tiempo real a partir de la distancia física entre la posición GPS actual del bus y la primera parada objetivo, dividida para la velocidad utilizada en el cálculo:

$$t_1 = \frac{D_{real}}{V_u}$$

Para los tramos subsiguientes, el sistema siguió una lógica de prioridad en cascada. En primer lugar, utilizó el promedio histórico de tiempo de viaje almacenado en la tabla `tiempos_tramo` para cada transición específica entre paradas consecutivas. Cuando no existieron suficientes observaciones históricas, se aplicó un respaldo geográfico basado en la distancia del trayecto y una velocidad urbana media constante de 18 km/h, equivalente a 5 m/s, marcando la predicción con el estado de confiabilidad `SIN_HISTORICO`. A nivel de tramo, también se estableció un control de desplazamientos improbables, descartando transiciones que superaran una duración lógica de 20 minutos, es decir, 1200 s, ya que ello indicaría detenciones prolongadas o desvíos significativos de la unidad. Si la antigüedad del último dato GPS superó los 2 minutos, el algoritmo interrumpió el proceso y etiquetó la estimación como `DATOS_DESACTUALIZADOS`. Esta estrategia híbrida permitió combinar observación en tiempo real, información histórica y mecanismos de respaldo para producir estimaciones más robustas.

El quinto procedimiento correspondió al algoritmo de planificación de itinerarios multimodales, diseñado para evaluar las mejores opciones de movilidad entre un origen y un destino georreferenciados, priorizando el uso del transporte público frente al desplazamiento completamente a pie. Como primera regla, la opción de caminata directa entre origen y destino quedó restringida a una distancia máxima de 800 m. Si la distancia física superaba dicho umbral, el planificador descartaba esta posibilidad para evitar sugerencias poco viables para el usuario.

Posteriormente, el sistema ejecutó una detección de paradas cercanas, ordenando todas las paradas según su distancia respecto al origen y al destino. Sobre esta base, se seleccionaron las cinco paradas más próximas dentro de un radio máximo de 1,2 km. Si no existía una ruta directa entre los puntos seleccionados, el algoritmo activaba la planificación de itinerarios con transbordo, buscando conexiones de una sola transferencia entre diferentes líneas de autobús. En esta etapa, la distancia de caminata entre paradas de transferencia se restringió a un máximo de 250 m, garantizando transbordos físicamente realizables para el ciudadano.

Como elemento diferenciador frente a planificadores estáticos, el sistema integró el ETA en tiempo real para calcular el tiempo de espera en parada. Para ello, el algoritmo consultó el componente de estimación de llegadas con el fin de identificar la cercanía del siguiente

bus disponible. Cuando no existían vehículos operando activamente en la ruta, se asignó un valor de respaldo de 5 minutos como tiempo estimado de espera.

Finalmente, los itinerarios resultantes fueron ordenados mediante una función de costo percibido, destinada a reflejar que el usuario percibe el tiempo de caminata como más costoso que el tiempo a bordo del vehículo. La función utilizada fue la siguiente:

$$C_p = T_{total} + 4 \left(\frac{D_{cam}}{V_{cam}} \right)$$

donde C_p representa el costo percibido del itinerario, T_{total} es la duración total del viaje en segundos, la cual ya incorpora implícitamente el tiempo real de caminata, D_{cam} corresponde a la distancia total recorrida a pie y V_{cam} representa la velocidad media de caminata humana, establecida en 1,39 m/s, equivalente a 5 km/h. Al sumarse nuevamente el término de caminata ponderado por un coeficiente de 4, el efecto total del esfuerzo peatonal dentro de la función equivale a una penalización aproximada de 5 veces frente al tiempo de tránsito. Con ello, el sistema priorizó itinerarios con menor desplazamiento a pie.

El sexto procedimiento se orientó a la generación de feeds interoperables GTFS estáticos. Su función consistió en compilar de forma periódica las tablas relacionales de infraestructura y planificación del servicio, tales como rutas, paradas, secuencia de paradas y geometrías de ruta. A partir de estas tablas, el sistema generó un conjunto estandarizado de archivos planos delimitados por comas, posteriormente empaquetados en un archivo comprimido .zip, cumpliendo con la especificación general requerida para el intercambio interoperable de información estática.

Finalmente, el séptimo procedimiento correspondió a la generación de feeds GTFS-Realtime. En esta etapa, el sistema transformó las variables dinámicas almacenadas en la base de datos, como últimas posiciones, rumbos, estados de trayectoria y estimaciones ETA, en la estructura binaria jerárquica de Protocol Buffers (Protobuf) utilizada por GTFS-Realtime. Este proceso generó dos entidades principales: VehiclePositions, encargada de transmitir el identificador de la unidad, latitud, longitud, rumbo, velocidad y marca temporal; y TripUpdates, responsables de estructurar las actualizaciones de tiempos estimados de llegada y salida por parada para los viajes en curso.

En conjunto, estos procedimientos definieron la lógica central del prototipo y organizaron, de manera secuencial, el paso desde la recepción de datos GPS hasta la obtención de resultados operativos, navegacionales e interoperables.

2.6.3.6 Diagramas de secuencia

Los diagramas de secuencia se definieron para representar la interacción temporal y el intercambio de mensajes entre los componentes del sistema durante la ejecución de los procesos principales del prototipo. Su diseño permitió observar la dinámica interna del software, detallando los flujos de control y las dependencias funcionales entre los actores externos, las interfaces de usuario, los controladores del backend desarrollados en NestJS, los servicios de procesamiento, el cliente Prisma ORM y la base de datos PostgreSQL.

Para el análisis dinámico de la plataforma se modelaron tres secuencias principales. La primera correspondió a la ingestión, validación y almacenamiento de telemetría GPS, proceso ejecutado automáticamente en segundo plano mediante un programador de tareas del backend. En esta secuencia, el servicio encargado de planificar las consultas periódicas realiza peticiones hacia la API del proveedor GPS para obtener las tramas crudas de geolocalización de la flota. Una vez recibida la respuesta en formato JSON, la información se transfiere a los componentes responsables de la validación y depuración, donde se aplican controles de consistencia espacial, filtrado de coordenadas inválidas y verificación de condiciones operativas. Posteriormente, los registros validados se persisten mediante Prisma ORM en las tablas correspondientes de PostgreSQL, principalmente aquellas relacionadas con registros GPS y trayectorias. Esta secuencia permitió representar el flujo automático de captura, limpieza y almacenamiento de datos de telemetría.

La segunda secuencia se orientó a la planificación de viajes y consulta dinámica de rutas, representando la interacción del usuario general con el mapa público para obtener un recorrido óptimo entre un origen y un destino. El proceso inicia cuando el pasajero ingresa direcciones en la interfaz de búsqueda o selecciona puntos directamente sobre el mapa. A partir de esta acción, el frontend desarrollado en Next.js envía una solicitud HTTP al backend, donde el controlador correspondiente delega la operación al servicio de planificación. Este servicio consulta la base de datos para recuperar rutas, paradas y geometrías activas dentro del entorno de búsqueda y, además, solicita al componente de ETA la información de tiempos estimados de arribo de las unidades en circulación. Con estos insumos, el algoritmo aplica la lógica de planificación multimodal, ordena los

itinerarios mediante la función de costo percibido y retorna al frontend las mejores alternativas de viaje. Finalmente, la interfaz pública representa gráficamente la ruta seleccionada, los tramos peatonales y la información dinámica asociada al recorrido.

La tercera secuencia correspondió a la generación y publicación de feeds GTFS-Realtime. En este proceso, la información dinámica persistida en la base de datos se transforma en una salida interoperable compatible con integradores externos. La secuencia se activa cuando un cliente externo, como un planificador de transporte o un servicio de integración, realiza una solicitud HTTP a los endpoints destinados a publicar posiciones vehiculares o actualizaciones de viaje. El controlador del backend canaliza la petición hacia el servicio correspondiente, el cual consulta en PostgreSQL las últimas posiciones operativas y las estimaciones ETA de las unidades activas. Una vez obtenidos estos datos, el codificador de GTFS-Realtime organiza la información bajo la estructura oficial de Google Transit y la serializa en formato binario mediante Protocol Buffers. Como resultado, el backend responde con un archivo binario listo para ser consumido por sistemas externos de consulta e integración.

En conjunto, estos diagramas complementaron la descripción de la arquitectura, la base de datos y los algoritmos, ofreciendo una visión más precisa del comportamiento secuencial del sistema.

Figura 5

Diagrama de secuencia de recepción, validación y almacenamiento de datos GPS

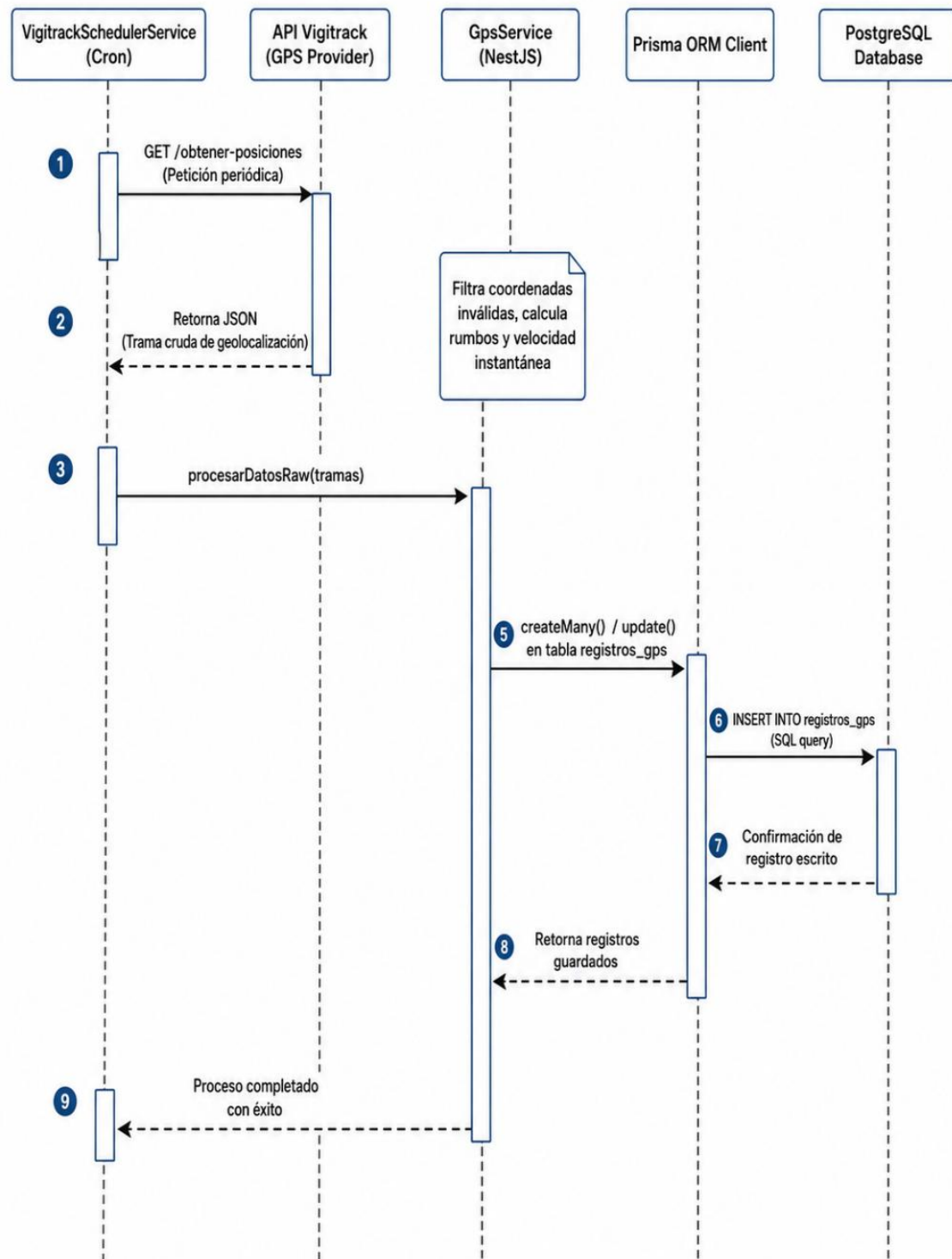


Figura 6

Diagrama de secuencia de procesamiento operativo y cálculo de ETA

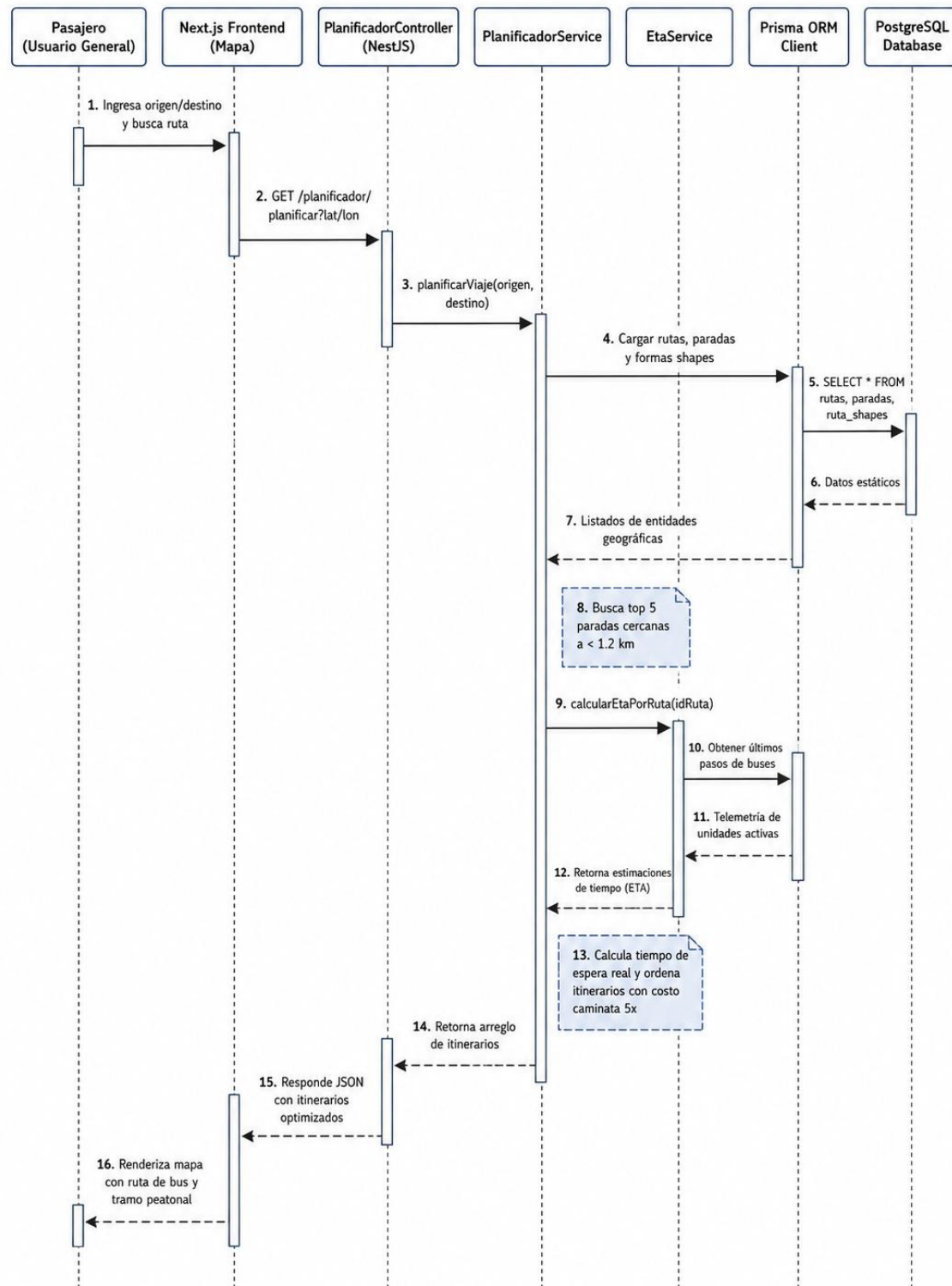
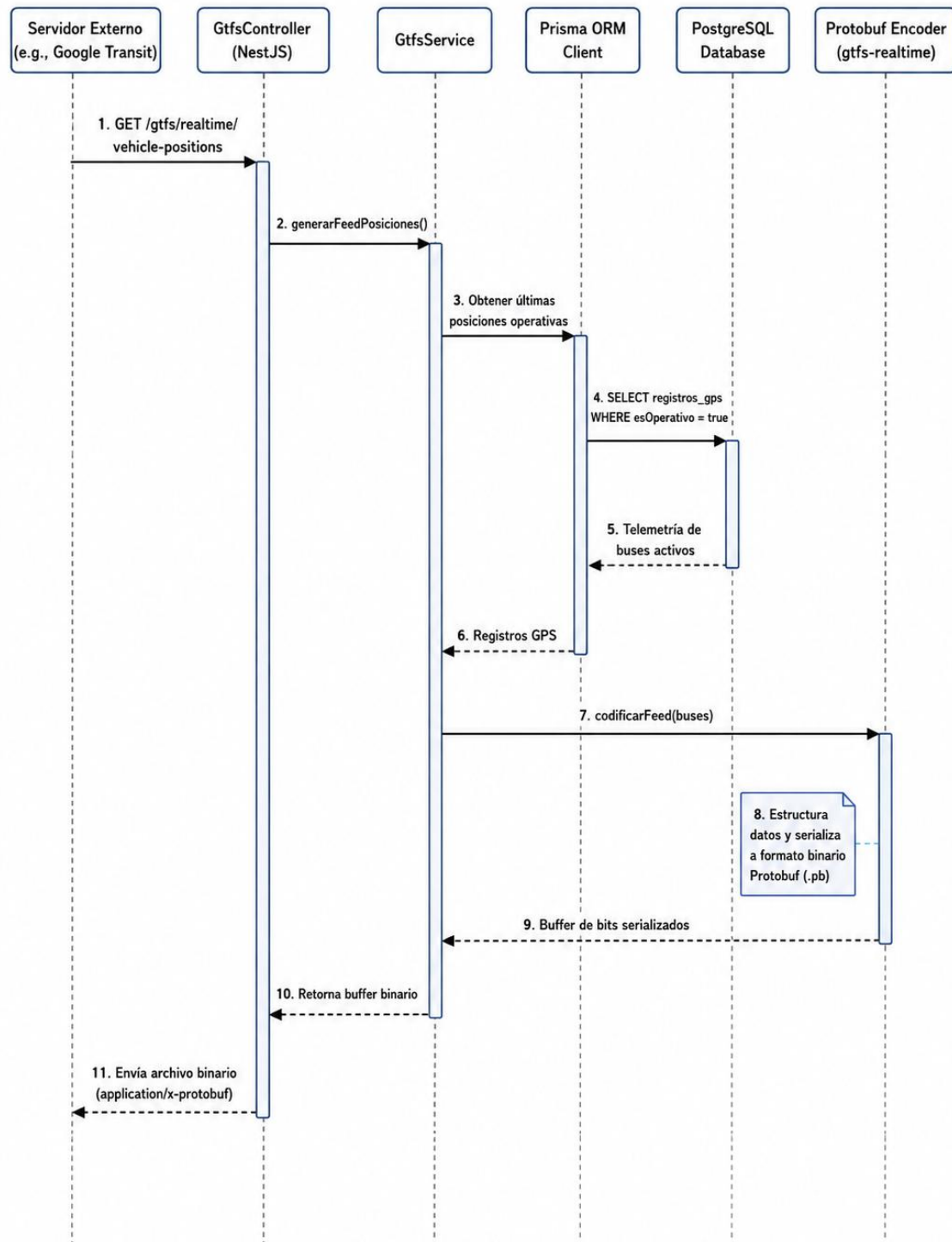


Figura 7

Diagrama de secuencia de generación de GTFS-Realtime



2.6.3.7 Diseño de interfaz

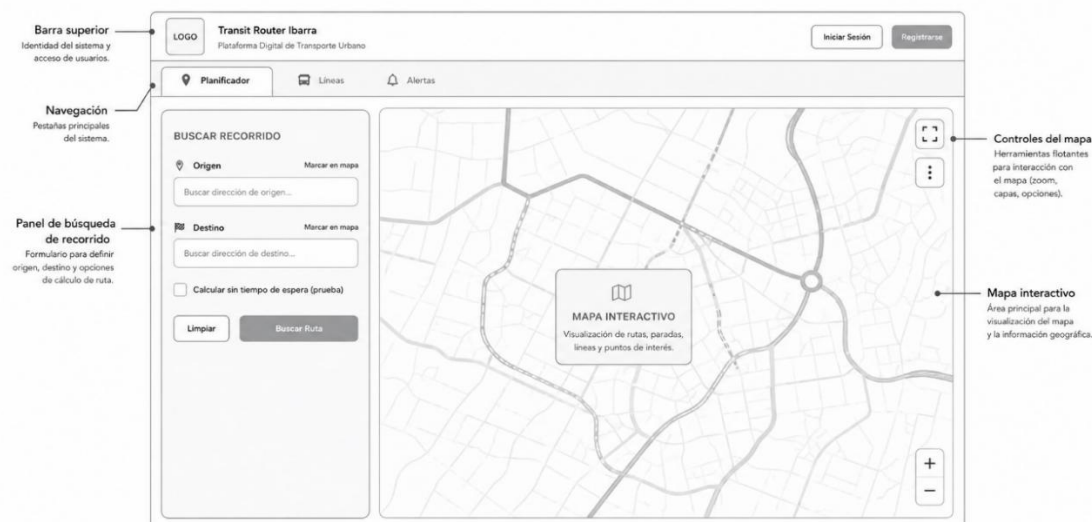
El diseño de la interfaz de usuario se planteó para organizar de forma clara y accesible la información del transporte urbano, considerando dos perfiles principales: el ciudadano que consulta el servicio y el administrador que supervisa su operación. Para ello, la interfaz se desarrolló sobre Next.js, bajo criterios de diseño responsivo, navegación fluida y distribución ordenada de los componentes visuales. Además, se incorporaron recursos como transparencias, desenfoques de fondo, jerarquía tipográfica y microinteracciones orientadas a mejorar la experiencia de uso.

La plataforma se estructuró en dos entornos web principales. El primero correspondió a la interfaz ciudadana o mapa público del pasajero, concebida como un espacio de acceso libre para consultar rutas, visualizar unidades en movimiento y planificar recorridos. Esta interfaz se diseñó en un formato de pantalla dividida, con un panel lateral de opciones y un mapa interactivo como elemento central. Dentro de este entorno se definieron tres vistas funcionales. La primera fue el buscador de itinerarios, orientado a la planificación de viajes a partir del ingreso de puntos de origen y destino mediante autocompletado de direcciones. Esta vista permitió consultar cómo llegar a un destino utilizando el transporte público, mostrar alternativas ordenadas según el criterio de menor caminata y desglosar los tiempos de espera en parada y el tiempo estimado de viaje.

La segunda vista fue el monitoreo de líneas y ETA, destinada a mostrar las rutas activas, sus trazados y paradas, así como la ubicación en tiempo real de los autobuses mediante marcadores dinámicos con rotación según el rumbo reportado por el GPS. La tercera correspondió al módulo de reportes viales ciudadanos, concebido como un mecanismo de participación colaborativa mediante el cual los usuarios autenticados pueden registrar incidentes de tránsito, como congestión, retrasos o buses llenos, directamente sobre el mapa a partir de coordenadas obtenidas por geocodificación inversa.

Figura 8

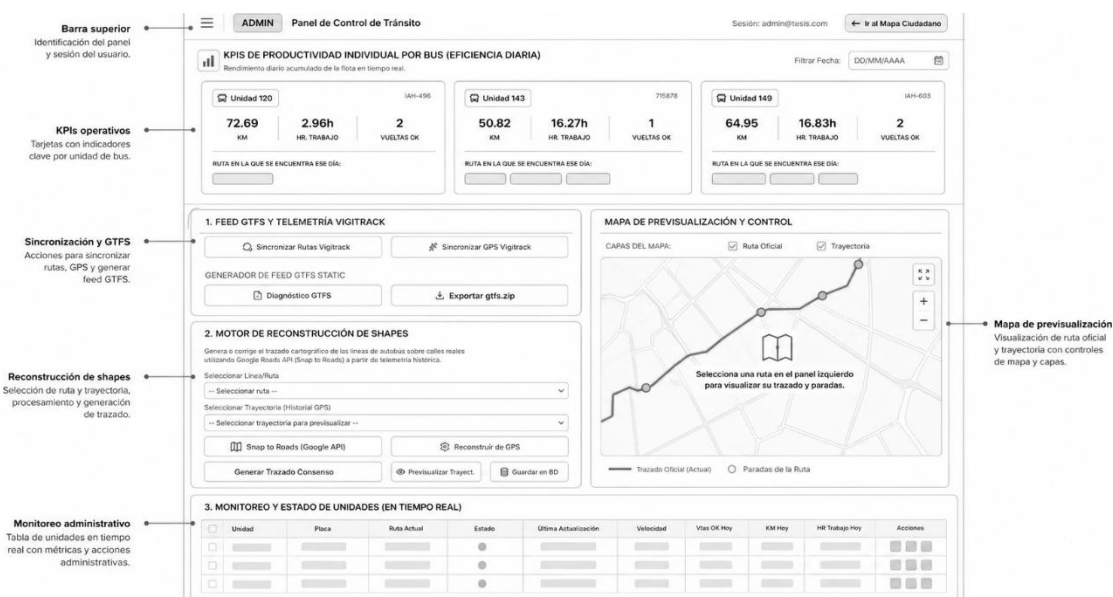
Diseño del panel principal del sistema



El segundo entorno correspondió a la interfaz administrativa o dashboard de control, orientada exclusivamente a la supervisión operativa y a la revisión técnica del sistema por parte de los operadores. Esta interfaz se organizó en tres módulos funcionales. El primero fue el módulo de indicadores analíticos, compuesto por tarjetas de resumen y gráficos comparativos destinados a mostrar métricas operativas como kilometraje acumulado, horas de servicio, número de vueltas completadas y tiempos de recorrido por ruta. El segundo fue el mapa de previsualización y auditoría, diseñado para contrastar visualmente la trayectoria oficial de una ruta con la trayectoria realmente ejecutada por una unidad en un recorrido histórico, lo que permite identificar desvíos, interrupciones o paradas omitidas. El tercer módulo correspondió al área de moderación y control GTFS, orientada a la descarga del feed interoperable, la revisión del estado de consistencia de la base de datos y la administración de reportes ciudadanos visibles en la plataforma pública.

Figura 9

Diseño del panel administrativo



En conjunto, ambos entornos permitieron organizar de manera coherente el acceso a la información del sistema, integrando monitoreo geoespacial, planificación de viajes, participación ciudadana y supervisión operativa dentro de una misma solución web. Esta organización favoreció la separación funcional entre las necesidades del usuario final y los requerimientos de control administrativo, manteniendo una experiencia de uso consistente y escalable. En el presente diseño, los entornos ciudadano y administrativo se describen a nivel funcional general. Durante la implementación, cada uno de estos bloques se descompuso en submódulos específicos de interacción, monitoreo, auditoría y control, cuya materialización se presenta con mayor detalle en el Capítulo III.

2.6.4 Herramientas de desarrollo

Para la implementación e integración del prototipo se seleccionó un conjunto de herramientas, librerías y entornos de desarrollo que permitieron sostener la interoperabilidad, la escalabilidad y la robustez de la solución. Estas herramientas se organizaron de acuerdo con su función dentro de la arquitectura del sistema: persistencia de datos, backend, frontend, infraestructura de despliegue y pruebas.

En la capa de datos y persistencia se utilizó PostgreSQL como sistema gestor de base de datos relacional, debido a su robustez, capacidad de procesamiento concurrente y eficiencia para almacenar tanto la información estructurada del servicio como el historial de telemetría de las unidades. La interacción entre el backend y la base de datos se

gestionó mediante Prisma ORM, lo que permitió definir el esquema de datos, administrar migraciones de forma segura y ejecutar consultas tipadas dentro del entorno de desarrollo en TypeScript.

En el backend se utilizó NestJS, framework basado en Node.js, como base para estructurar la lógica modular del sistema. Esta herramienta permitió desarrollar los endpoints de la API REST y organizar los servicios responsables del procesamiento de trayectorias, el cálculo de ETA, la generación de datos interoperables y la ejecución de tareas programadas para la ingesta automática de datos GPS. Como lenguaje principal de desarrollo se empleó TypeScript, incorporando tipado estático para reducir errores en tiempo de ejecución y mejorar la mantenibilidad del código. Para la serialización binaria de los feeds en tiempo real se utilizaron las librerías `protobufjs` y `gtfs-realtime-bindings`, necesarias para codificar la información conforme al estándar Protocol Buffers empleado por GTFS-Realtime.

En el frontend se empleó Next.js como framework principal para el desarrollo del panel administrativo y del mapa público orientado al usuario general. Su adopción permitió trabajar sobre una arquitectura reactiva y optimizada para la carga de componentes, además de facilitar la interacción con la API del backend. Para el diseño visual y responsivo de la interfaz se utilizó Tailwind CSS, lo que facilitó la implementación de estilos consistentes, diseño adaptable y elementos visuales como transparencias, desenfoces y microinteracciones. En la capa cartográfica se integró la Google Maps JavaScript API mediante la librería `@react-google-maps/api`, permitiendo incorporar renderizado de mapas, trazado de rutas, capas de tráfico, autocompletado predictivo de direcciones y geocodificación inversa dentro de la experiencia del usuario.

En el ámbito de infraestructura, despliegue y pruebas, se utilizó Docker como herramienta de contenedorización para aislar y homogeneizar los entornos locales de desarrollo. Para la publicación y orquestación del prototipo en la nube se empleó Railway, plataforma que permitió alojar la base de datos PostgreSQL, ejecutar el backend desarrollado en NestJS y desplegar el frontend construido en Next.js dentro de un mismo entorno de servicio. Durante la fase de desarrollo, la verificación funcional de los endpoints de la API REST se realizó con Postman, mientras que el control de versiones y el resguardo del código fuente se gestionaron mediante Git y GitHub.

Finalmente, para la elaboración de diagramas técnicos se utilizó draw.io, herramienta empleada en la construcción de los modelos de arquitectura, casos de uso, componentes y secuencias incorporados en el documento.

2.6.5 Construcción del prototipo

La construcción del prototipo se realizó de manera progresiva y modular, siguiendo una secuencia organizada en cinco fases consecutivas. Esta estructura permitió respetar las dependencias técnicas entre los componentes y consolidar gradualmente la base de datos, la lógica de negocio, los módulos de interoperabilidad, la interfaz web y el despliegue final en la nube.

La primera fase correspondió a la implementación de la base de datos y del mapeo relacional. En esta etapa se aprovisionó PostgreSQL y se definió la estructura de tablas, relaciones y restricciones de integridad a partir del modelo descrito en el archivo schema.prisma. La creación del esquema no se realizó mediante sentencias SQL directas, sino a través del sistema de migraciones automatizadas proporcionado por Prisma ORM. Con este mecanismo se establecieron las entidades necesarias para almacenar la infraestructura estática del servicio y los flujos dinámicos asociados con registros GPS, trayectorias y estimaciones de arribo.

La segunda fase se orientó al desarrollo del backend y de la lógica de negocio. En este punto se estructuró el servidor bajo una arquitectura modular y desacoplada, organizada en componentes responsables de la ingesta, el procesamiento y la exposición de servicios. Dentro de este bloque se implementó la ingestión asíncrona de telemetría mediante un servicio de tareas programadas que consume periódicamente la API de Vigitrack y canaliza las tramas hacia el módulo de validación GPS. También se desarrollaron los servicios de reconstrucción de trayectorias, asociación espacial entre unidades y rutas, cálculo de tiempos históricos entre paradas, estimación híbrida de arribo y planificación de itinerarios multimodales.

La tercera fase correspondió al desarrollo de los módulos de interoperabilidad GTFS y GTFS-Realtime. Una vez consolidada la lógica operativa del sistema, se implementaron los componentes encargados de transformar la información relacional y dinámica en salidas compatibles con el estándar internacional de transporte. En el caso del módulo estático, los datos del servicio se estructuraron y empaquetaron en archivos planos comprimidos en formato .zip. En el módulo en tiempo real, la telemetría dinámica y las

estimaciones ETA fueron serializadas en formato binario mediante Protocol Buffers, exponiendo endpoints HTTP destinados a consumidores externos de información interoperable.

La cuarta fase se enfocó en la construcción del frontend y del mapa interactivo. En esta etapa se desarrollaron dos entornos diferenciados. El primero correspondió a la aplicación ciudadana, en la que se integraron funcionalidades de autocompletado predictivo de direcciones, geocodificación inversa, visualización de rutas, trazado de recorridos peatonales y representación dinámica de autobuses en movimiento sobre el mapa. El segundo correspondió al dashboard administrativo, concebido como un entorno de supervisión y control, donde se incorporaron módulos de indicadores operativos, auditoría visual de trayectorias y herramientas de moderación de alertas registradas por los usuarios.

La quinta fase correspondió a la integración y despliegue continuo en la nube. Para ello, los distintos componentes del prototipo fueron orquestados en Railway, plataforma en la que se configuró el entorno de producción para la base de datos PostgreSQL, el backend desarrollado en NestJS y el frontend construido en Next.js. Este despliegue se vinculó con el repositorio del proyecto en GitHub, lo que permitió automatizar la compilación, construcción y publicación de cambios mediante un flujo continuo de integración y despliegue.

En conjunto, la construcción del prototipo siguió una secuencia coherente con el ciclo de desarrollo del software. Cada componente se implementó sobre una base técnica previamente consolidada, favoreciendo la integración progresiva de la persistencia de datos, la lógica de negocio, la interoperabilidad GTFS y la visualización web.

2.7 Fase 4 – Demostración del prototipo

En esta fase se realizó la demostración funcional del prototipo dentro de un entorno controlado. El propósito fue comprobar que los componentes implementados respondieran de forma coherente al flujo previsto del sistema, desde la recepción de telemetría GPS hasta la visualización web de la información operativa y la generación de estructuras interoperables. La demostración se aplicó únicamente sobre los módulos efectivamente construidos y sobre el flujo funcional definido para el alcance del prototipo.

La primera etapa correspondió a la recepción e ingesta de telemetría cruda. En este punto se verificó la conexión activa entre el backend desarrollado en NestJS y la API REST del

proveedor tecnológico Vigitrack, mediante llamadas programadas ejecutadas de forma periódica. También se comprobó que el sistema capturara correctamente los paquetes de datos recibidos y asignara sus atributos a las variables de negocio definidas para el prototipo, preservando marcas de tiempo, coordenadas geográficas y parámetros operativos de cada unidad.

Posteriormente, se revisó el funcionamiento del proceso de validación, depuración y persistencia. En esta etapa se constató que los registros duplicados en coordenadas y tiempo fueran filtrados antes de su almacenamiento en PostgreSQL, y que las coordenadas corruptas, inconsistentes o no asociadas a rutas activas fueran excluidas del flujo principal o marcadas con un estado de inoperatividad. Con ello, se evidenció que la información persistida mantenía un nivel de consistencia suficiente para el procesamiento posterior.

La siguiente etapa se orientó al procesamiento operativo y a la reconstrucción de trayectorias. En ella se verificó la capacidad del backend para agrupar registros por unidad y tiempo, reconstruyendo recorridos individuales a partir de la telemetría histórica. También se comprobó la segmentación automática de trayectorias, las cuales podían iniciar al aproximarse la unidad al terminal de salida y cerrarse al arribar al terminal de llegada, actualizando su estado conforme a la lógica definida en el sistema. Además, se evidenció la asociación de tiempos de tránsito entre paradas a partir de la proximidad espacial entre las coordenadas recibidas y la red física de paradas registrada en la base de datos.

Una cuarta etapa se concentró en el cálculo de estimaciones de arribo y en la planificación de itinerarios. En este punto se comprobó la generación en tiempo real de tiempos estimados de llegada a paradas futuras a partir de las coordenadas activas de la flota. Del mismo modo, se verificó que, ante escenarios de inactividad o pérdida de señal, el motor de ETA aplicara la lógica de contingencia definida por el sistema para mantener continuidad informativa. Sobre esta base, también se demostró que el planificador de itinerarios ordenara dinámicamente las rutas recomendadas utilizando el criterio de costo percibido, priorizando trayectos con menor carga de caminata y respetando las restricciones máximas establecidas para accesos peatonales y transbordos.

La quinta etapa correspondió a la generación de estructuras interoperables. En esta fase se comprobó la compilación de la información del servicio en archivos planos correspondientes al estándar GTFS estático, incluyendo los conjuntos mínimos

requeridos para representar agencias, rutas, paradas, recorridos, viajes y secuencias temporales. Estos archivos fueron empaquetados automáticamente en un archivo comprimido para su distribución. De manera complementaria, se demostró la serialización dinámica de feeds GTFS-Realtime, particularmente posiciones de vehículos y actualizaciones de viaje, codificados en formato Protocol Buffers.

Finalmente, se verificó la visualización en el entorno web, tanto en la interfaz ciudadana como en el panel administrativo. En la primera se comprobó la búsqueda de origen y destino, la consulta de rutas y la representación del recorrido sobre el mapa. En la segunda se verificó la visualización dinámica de unidades, el despliegue de indicadores operativos y la consulta de información de apoyo para la supervisión del servicio. De este modo, la fase de demostración permitió comprobar que el prototipo operaba de manera integrada sobre el conjunto de módulos que conforman la solución propuesta.

2.8 Fase 5 – Evaluación y validación

En esta fase se definió el proceso de evaluación y validación del prototipo, con el fin de establecer los criterios para analizar el cumplimiento de las funcionalidades previstas y la consistencia técnica de los resultados generados durante su ejecución. La finalidad fue determinar, dentro del alcance del estudio, si la solución desarrollada respondía adecuadamente a la necesidad identificada en la Cooperativa de Transportes “28 de Septiembre”. Para ello, se consideró tanto el funcionamiento técnico del sistema como la coherencia entre los datos de entrada, su procesamiento interno y las salidas producidas.

La evaluación se orientó, en primer lugar, a verificar el cumplimiento de los requisitos funcionales establecidos para el prototipo. En este sentido, se definió una validación sistemática de procesos como la recepción de datos GPS, su almacenamiento estructurado, la validación y depuración de registros, la reconstrucción de trayectorias, la asociación con rutas y paradas, el cálculo de tiempos estimados de llegada, la planificación de itinerarios y la generación de estructuras GTFS y GTFS-Realtime conforme al alcance previsto.

Además del cumplimiento funcional, la validación consideró la consistencia técnica de los resultados producidos por el prototipo. En esta etapa se planteó comprobar si los registros procesados mantenían coherencia temporal y espacial, si las trayectorias reconstruidas guardaban relación con los recorridos definidos en la red de transporte y si las estimaciones de llegada respondían de manera lógica a la posición de las unidades y

a la información histórica disponible. Del mismo modo, se contempló la verificación de la estructura y conformidad de las salidas interoperables generadas por el sistema.

Otro aspecto considerado fue la correspondencia entre la lógica de procesamiento interno y la información presentada en la interfaz web. Para ello, se propuso revisar que la ubicación de las unidades, la representación de trayectorias, la planificación de itinerarios y los indicadores analíticos reflejaran de forma consistente los resultados obtenidos por el backend.

En conjunto, esta fase permitió establecer la base metodológica para la comprobación funcional del prototipo y para la valoración posterior de su desempeño dentro del entorno de prueba. La aplicación de los casos de prueba definidos y los resultados concretos obtenidos se presentan en el Capítulo III.

2.8.1 Casos de prueba

Los casos de prueba se definieron para verificar el comportamiento funcional de los módulos principales del prototipo dentro del entorno de validación. Su planteamiento permitió relacionar las entradas de prueba, los resultados esperados y los criterios de verificación aplicables a procesos clave del sistema, como la recepción de datos GPS, el almacenamiento estructurado, la validación y depuración de registros, la reconstrucción de trayectorias, la asociación con rutas y paradas, el cálculo de ETA, la generación de archivos GTFS y estructuras GTFS-Realtime, y la visualización de información en el panel web administrativo.

Estos casos se derivaron de los requisitos funcionales, de los módulos implementados y de la lógica operativa definida para el prototipo. Para su construcción se consideraron entradas de prueba, resultados esperados y criterios de verificación que permitieran identificar tanto el cumplimiento de las funcionalidades previstas como posibles desviaciones en el comportamiento del sistema. De esta manera, los casos de prueba constituyeron una base metodológica para la validación funcional del prototipo y para el análisis de su comportamiento integral.

Tabla 6*Casos de prueba aplicados al prototipo*

ID	Módulo / proceso	Entrada de prueba	Resultado esperado	Criterio de verificación
CP-01	Recepción de datos GPS	Registros enviados desde la fuente disponible del proveedor GPS con campos completos de unidad, fecha, hora, latitud y longitud	El sistema recibe correctamente los registros y los pone a disposición del flujo interno de procesamiento	Confirmar que los registros ingresan al backend sin error y conservan los campos mínimos requeridos
CP-02	Recepción de datos incompletos	Registros GPS con ausencia de uno o más campos obligatorios	El sistema identifica la inconsistencia y evita el procesamiento normal del registro	Verificar que los registros incompletos sean rechazados o marcados como inválidos
CP-03	Recepción de datos con formato inválido	Registros con estructura no compatible con el formato esperado por el sistema	El sistema detecta el formato incorrecto y evita su incorporación al flujo principal	Comprobar que los registros inválidos no sean aceptados para validación y almacenamiento
CP-04	Recepción continua de múltiples registros	Secuencia consecutiva de registros GPS de distintas unidades	El sistema recibe y procesa múltiples registros sin pérdida de información	Confirmar que todos los registros enviados sean capturados correctamente por el backend
CP-05	Detección de duplicados	Registros repetidos de una misma unidad con igual marca temporal y coordenadas	El sistema detecta y excluye los duplicados del flujo principal	Comprobar que no se almacenen o procesen registros repetidos

CP-06	Detección de coordenadas inválidas	Registros con latitud o longitud fuera de rangos válidos	El sistema identifica las coordenadas no válidas y las excluye del procesamiento	Verificar que esos registros no pasen a etapas posteriores
CP-07	Detección de inconsistencias temporales	Registros con secuencias horarias incorrectas o no cronológicas	El sistema detecta inconsistencias temporales y aplica las reglas definidas	Comprobar que la secuencia final sea cronológicamente consistente
CP-08	Detección de desplazamientos imposibles	Registros consecutivos con cambios espaciales incoherentes en tiempos muy cortos	El sistema identifica desplazamientos no plausibles y los trata según las reglas de depuración	Verificar que esos registros no afecten la trayectoria reconstruida
CP-09	Conservación de registros válidos tras depuración	Conjunto mixto de registros válidos e inválidos	El sistema conserva únicamente los registros válidos para el procesamiento posterior	Verificar que los datos correctos permanezcan disponibles tras el proceso de limpieza
CP-10	Almacenamiento en base de datos	Registros GPS previamente validados	Los datos se almacenan correctamente en PostgreSQL y quedan disponibles para consulta posterior	Confirmar persistencia correcta, integridad de campos y recuperación mediante consulta
CP-11	Recuperación de registros almacenados	Consulta sobre datos GPS previamente persistidos	El sistema recupera correctamente la información solicitada desde la base de datos	Verificar correspondencia entre lo almacenado y lo recuperado
CP-12	Integridad entre entrada y persistencia	Registros válidos recibidos	La información almacenada mantiene	Comprobar consistencia de campos clave entre

		y posteriormente almacenados	correspondencia con los datos procesados por el sistema	entrada, validación y almacenamiento
CP-13	Ordenamiento por unidad y tiempo	Registros válidos de varias unidades en diferente orden de llegada	El sistema organiza los registros por unidad y secuencia temporal	Verificar que la salida quede ordenada correctamente antes del procesamiento
CP-14	Reconstrucción de trayectorias	Secuencia de registros GPS válidos de una misma unidad ordenados por tiempo	El sistema agrupa los puntos y reconstruye una trayectoria coherente para la unidad evaluada	Verificar continuidad temporal y correspondencia entre los puntos almacenados y la trayectoria generada
CP-15	Asociación con rutas	Trayectorias reconstruidas y estructura operativa del prototipo con rutas definidas	El sistema vincula la trayectoria con la ruta correspondiente	Confirmar que la trayectoria se relacione con la estructura operativa definida sin inconsistencias relevantes
CP-16	Asociación con paradas	Trayectorias reconstruidas y paradas definidas en la base de datos	El sistema relaciona la trayectoria con las paradas correspondientes	Verificar correspondencia entre desplazamiento observado y puntos de parada definidos
CP-17	Cálculo de ETA con datos válidos	Posición actual de la unidad, ruta asociada y parada objetivo	El sistema genera una estimación de llegada coherente con la información procesada	Verificar que el ETA se calcule y se presente como salida operativa válida
CP-18	Cálculo de ETA con datos insuficientes	Registros incompletos o sin relación operativa	El sistema controla adecuadamente la situación y evita	Comprobar que el sistema no produzca ETA

		suficiente para estimar llegada	generar resultados inconsistentes	inválidos ante datos insuficientes
CP-19	Presentación del ETA	Resultado de cálculo de tiempo estimado de llegada	El sistema utiliza el ETA como salida operativa disponible para consulta	Verificar que el tiempo estimado pueda visualizarse correctamente en el sistema
CP-20	Generación de GTFS	Datos planificados de rutas, paradas y relaciones operativas	El sistema genera archivos GTFS con estructura compatible con el estándar esperado	Confirmar que los archivos contengan la información requerida y mantengan consistencia estructural
CP-21	Generación de GTFS-Realtime	Datos operativos procesados de posiciones y estado de viaje	El sistema genera estructuras GTFS- Realtime acordes con la información disponible	Verificar que la salida mantenga formato y contenido compatibles con la lógica interoperable del prototipo
CP-22	Flujo integral del prototipo	Datos GPS válidos desde la entrada hasta la salida administrativa e interoperable	El sistema mantiene continuidad entre recepción, procesamiento, almacenamiento, visualización y generación de salidas	Verificar que el flujo completo opere sin inconsistencias críticas

Los casos de prueba definidos en esta sección constituyeron la base de la validación funcional del prototipo. Su ejecución y los resultados obtenidos se presentan posteriormente en el Capítulo III, dentro del apartado correspondiente a las pruebas de caja negra.

2.8.2 Resultados de validación

Los resultados de validación permitieron definir los criterios con los que se comprobaría el funcionamiento del prototipo frente a los requerimientos y condiciones técnicas establecidos durante la investigación. A partir de las pruebas funcionales y de la revisión de consistencia técnica, se planteó verificar si los módulos implementados respondían adecuadamente a las funciones previstas dentro del alcance del estudio.

Esta validación no se limitó a comprobar cada componente de forma aislada, sino que también consideró la continuidad del flujo general del prototipo. Por ello, la evaluación abarcó procesos como la recepción y depuración de telemetría GPS, el almacenamiento en base de datos, la reconstrucción de trayectorias, la asociación con rutas y paradas, el cálculo del tiempo estimado de llegada, la generación de estructuras GTFS y GTFS-Realtime, y la visualización de la información en la interfaz.

De esta manera, la validación se planteó como un mecanismo para comprobar la viabilidad técnica del prototipo y su utilidad frente al problema identificado en la Cooperativa de Transportes “28 de Septiembre”. La ejecución de los casos de prueba definidos y los resultados concretos obtenidos se presentan en el Capítulo III, dentro del apartado correspondiente a las pruebas de caja negra.

2.9 Fase 6 – Comunicación de la investigación

La fase de comunicación se concretó en la presente tesis, donde se documentaron de manera estructurada el problema identificado, la metodología empleada, el diseño técnico del artefacto, la construcción del prototipo, su demostración funcional y los resultados de la evaluación realizada. El propósito de esta etapa fue organizar el conocimiento generado durante el desarrollo del proyecto y ponerlo a disposición tanto del ámbito académico como del contexto profesional relacionado con la ingeniería de software y los sistemas de transporte.

A través de esta documentación se explicaron las decisiones metodológicas y técnicas que sustentaron la solución propuesta, incluyendo la arquitectura general del sistema, el modelo de datos, los algoritmos de procesamiento de telemetría, la estimación de tiempos de llegada, la planificación de itinerarios y los mecanismos de interoperabilidad basados en los estándares GTFS y GTFS-Realtime. Asimismo, la tesis permitió dejar constancia de cómo estos elementos fueron articulados dentro del contexto operativo de la Cooperativa de Transportes “28 de Septiembre”.

De esta manera, la fase de comunicación permitió cerrar el ciclo metodológico de la investigación aplicada, integrando en un solo documento el desarrollo de la solución, su validación y los principales hallazgos obtenidos durante el proceso. Con ello, se consolidó una referencia técnica y metodológica útil para futuras iniciativas orientadas a la modernización de sistemas de información en organizaciones de transporte público con características similares.

CAPÍTULO III. RESULTADOS Y DISCUSIÓN

En este capítulo se presentan los resultados obtenidos durante el desarrollo e implementación del prototipo. La finalidad es mostrar, de forma clara y verificable, cómo se cumplieron los objetivos planteados en la investigación, considerando el funcionamiento de los módulos construidos, las pruebas realizadas, el despliegue del sistema y las condiciones técnicas bajo las cuales opera la solución. Además, se incluye la discusión de los hallazgos más relevantes para interpretar el alcance del prototipo dentro del contexto de la Cooperativa de Transportes “28 de Septiembre”.

3.1 Resultados de la implementación del prototipo

En esta sección se describen los resultados alcanzados durante la construcción del sistema, tomando como referencia las principales interfaces y módulos implementados. El propósito es evidenciar que la solución fue desarrollada conforme a los requerimientos funcionales definidos y que dispone de una estructura operativa adecuada para el monitoreo, la planificación de itinerarios y la interoperabilidad del servicio de transporte. También se consideran funcionalidades de soporte e interacción incorporadas durante el desarrollo, como la autenticación de usuarios, la gestión básica de perfiles, el marcado de rutas favoritas y el registro de incidencias ciudadanas, debido a que complementan el funcionamiento integral de la plataforma.

3.1.1 Interfaces de acceso y autenticación al sistema

El sistema incorpora un flujo de control de acceso organizado en dos pantallas principales. El formulario de inicio de sesión de la plataforma valida el ingreso mediante credenciales de usuario, compuestas por correo electrónico y contraseña. Una vez autenticado, el sistema identifica el rol correspondiente y habilita el acceso al panel administrativo o al entorno ciudadano. De forma complementaria, la interfaz de registro permite la creación de nuevas cuentas y la asignación del perfil correspondiente dentro de la plataforma, como funcionalidad de soporte para la gestión de acceso al sistema.

Figura 10

Interfaz de inicio de sesión

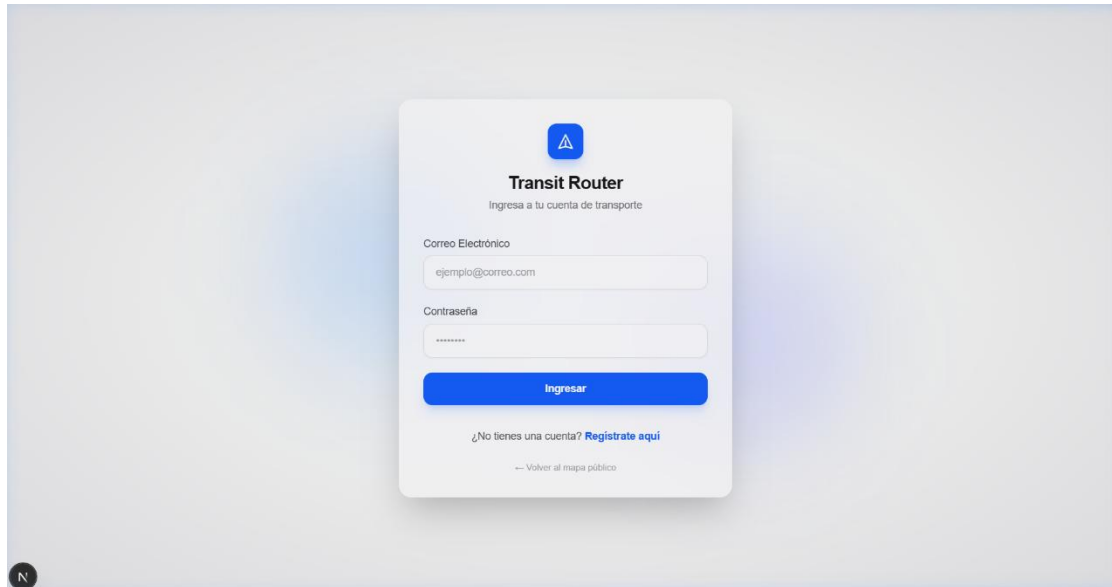
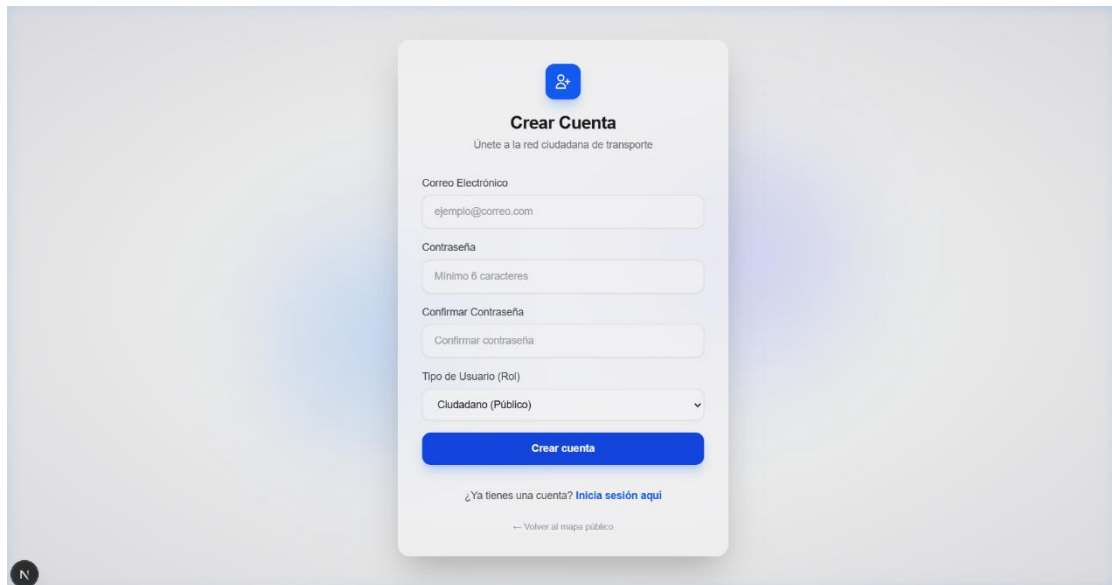


Figura 11

Interfaz de registro de usuarios y asignación de perfiles



3.1.2 Interfaz principal del mapa público

La interfaz principal del mapa público corresponde al entorno ciudadano del sistema y se organiza en pestañas funcionales para facilitar la navegación del usuario. En ella se integra el mapa interactivo de la API de Google Maps, donde se visualiza la ubicación en tiempo real de los autobuses activos y se habilitan los componentes de búsqueda espacial requeridos para la consulta de recorridos. Dentro de este entorno se destaca la pestaña del

módulo de consulta de líneas, concebida para la exploración general de la red de transporte. Esta sección permite al usuario listar las líneas operativas de la cooperativa, seleccionar un recorrido específico y proyectar de forma exclusiva sobre el mapa su trazado geométrico oficial, sus paradas asociadas y las unidades que circulan en esa ruta en ese momento. De forma complementaria, la interfaz incorpora opciones de apoyo a la navegación, como el marcado de líneas favoritas para consultas posteriores.

Figura 12

Entorno ciudadano interactivo del mapa público

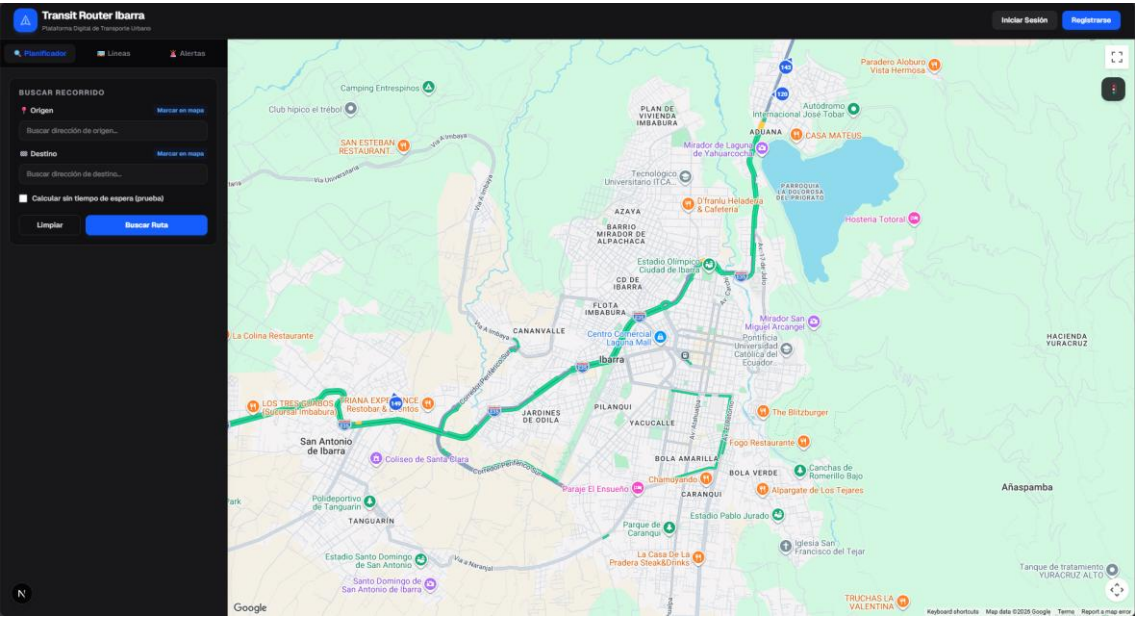
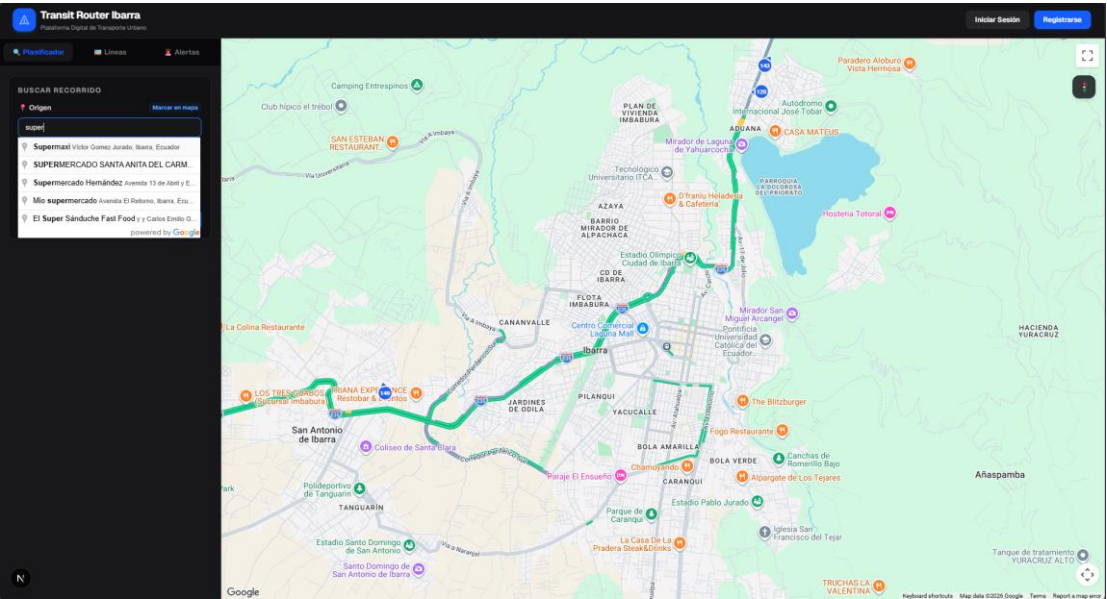


Figura 13

Componentes de búsqueda geoespacial con autocompletado



3.1.3 Módulo de planificación de itinerarios

Este módulo permite que el usuario ingrese un punto de origen y un destino para obtener alternativas de viaje que combinan tramos peatonales y transporte público. El planificador evalúa las rutas operativas disponibles y ordena las opciones mediante el criterio de costo percibido, aplicando una penalización de cinco veces al esfuerzo de caminata. Con ello, el sistema evita recomendar recorridos con desplazamientos excesivos a pie y prioriza alternativas más viables para el usuario.

Como apoyo a la validación funcional, la interfaz del planificador incorpora el control “Calcular sin tiempo de espera”. Al activar esta opción, el frontend envía un parámetro al backend para anular temporalmente los tiempos de espera dinámicos calculados en parada, asignándolos a cero segundos. Esta función permite comparar el itinerario recomendado en una condición ideal, sin espera, frente al cálculo real basado en datos operativos en tiempo real. De esta manera, se puede observar con mayor claridad el impacto de la telemetría en la selección de itinerarios.

Figura 14

Visualización del itinerario recomendado y desglose de tramos

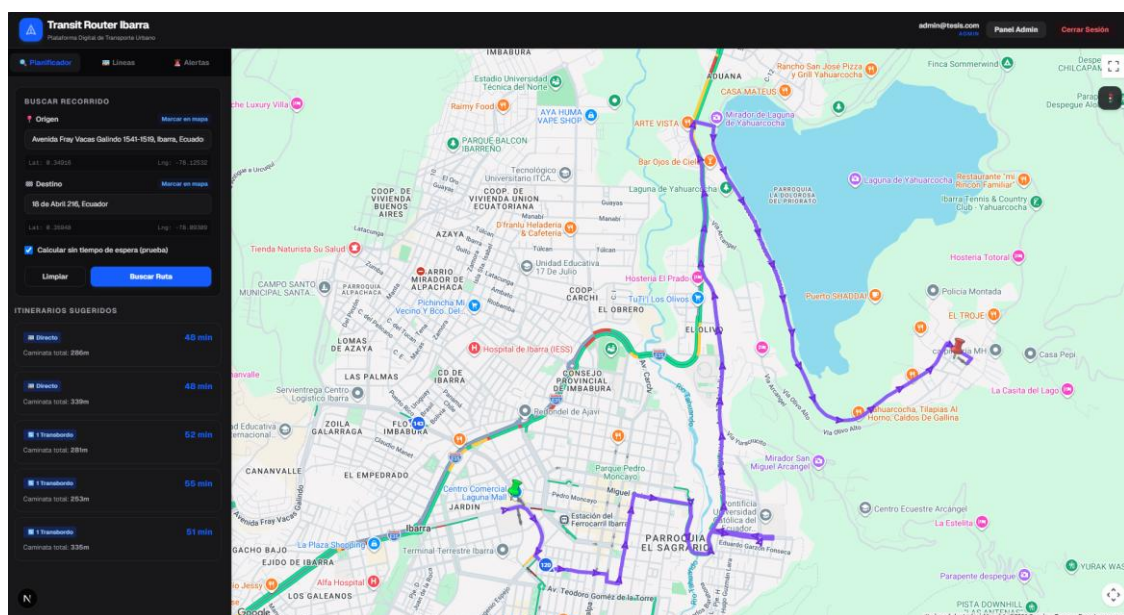
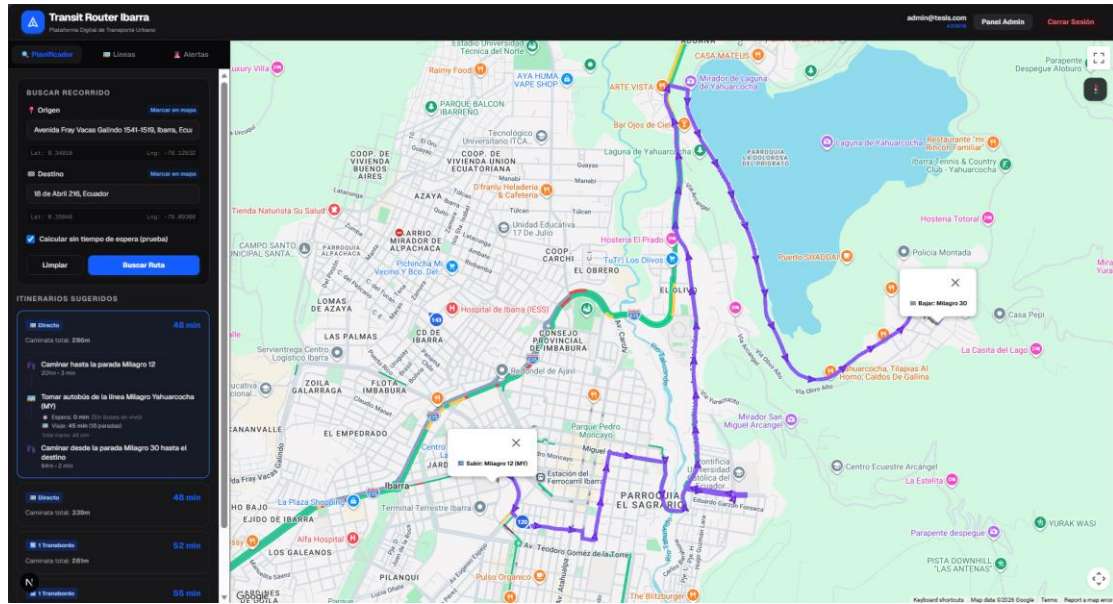


Figura 15

Visualización de tramos detallados y conexiones peatonales

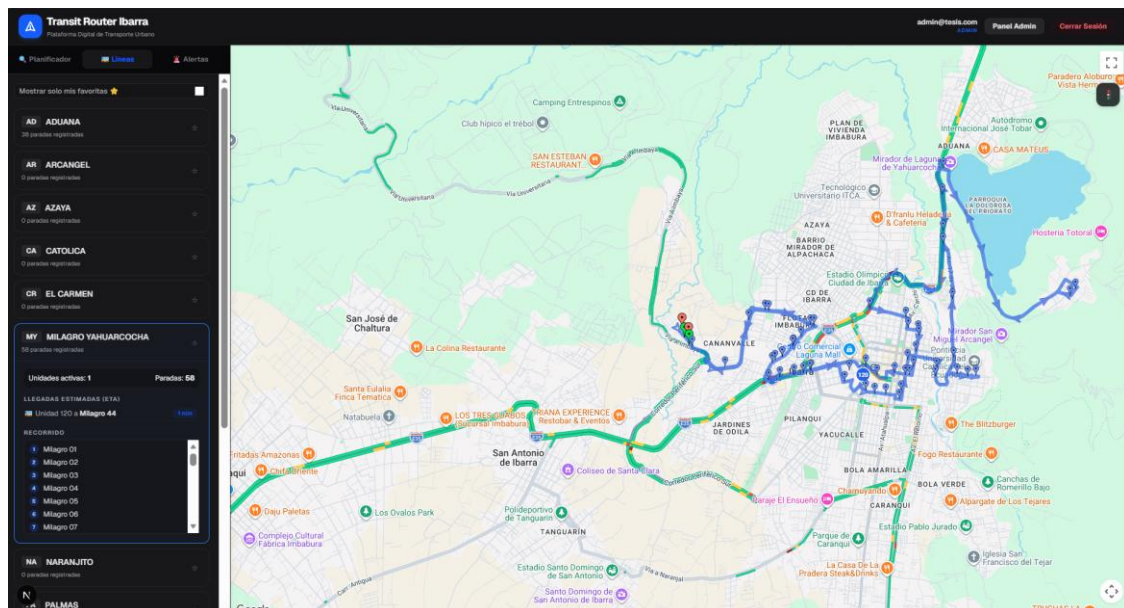


3.1.4 Módulo de monitoreo de rutas y ETA

Este módulo permite consultar de forma específica una línea de transporte, proyectando sobre el mapa interactivo el trazado oficial de la ruta y la ubicación en tiempo real de las unidades en circulación. Adicionalmente, el panel lateral muestra de forma dinámica la estimación del tiempo de llegada de los autobuses hacia las paradas objetivo. La estimación realiza una transición automática hacia la siguiente parada cuando el autobús se encuentra a menos de 20 segundos del arribo previsto.

Figura 16

Visualización del trazado de ruta y tiempos de arribo por parada



3.1.5 Módulo de reportes ciudadanos o incidencias

Este módulo permite registrar alertas relacionadas con eventos que afectan la calidad o la fluidez del servicio, como congestión vehicular, accidentes, retrasos operativos o unidades con alta ocupación. Su finalidad es complementar la supervisión del sistema con información generada directamente por los usuarios.

Para facilitar el registro y mejorar la precisión espacial de la información, el prototipo implementa dos mecanismos de ingreso de reportes. El primero corresponde al reporte georreferenciado sobre el mapa, orientado a personas que se encuentran en paradas o en puntos cercanos a la vía. En este caso, el usuario selecciona directamente sobre la interfaz cartográfica el lugar donde ocurre la incidencia y el sistema captura la coordenada del evento. El segundo mecanismo corresponde al reporte vinculado a una unidad activa, orientado a usuarios que se encuentran a bordo de un autobús en circulación. En este caso, el sistema permite seleccionar la unidad en tránsito y vincula automáticamente la ubicación del reporte con la coordenada GPS de telemetría asociada a dicha unidad, relacionándola además con la ruta correspondiente.

En ambos casos, el sistema almacena la información vinculando el usuario emisor, el tipo de incidencia, una descripción opcional y la coordenada geográfica asociada. Posteriormente, estos datos pueden visualizarse de forma temporal sobre el mapa público y alimentar el módulo administrativo de supervisión operativa. Aunque estas funciones no forman parte del núcleo operativo principal del prototipo, su incorporación permitió complementar la supervisión del servicio con información generada directamente por los usuarios.

Figura 17

Registro de incidencias mediante selección directa sobre el mapa

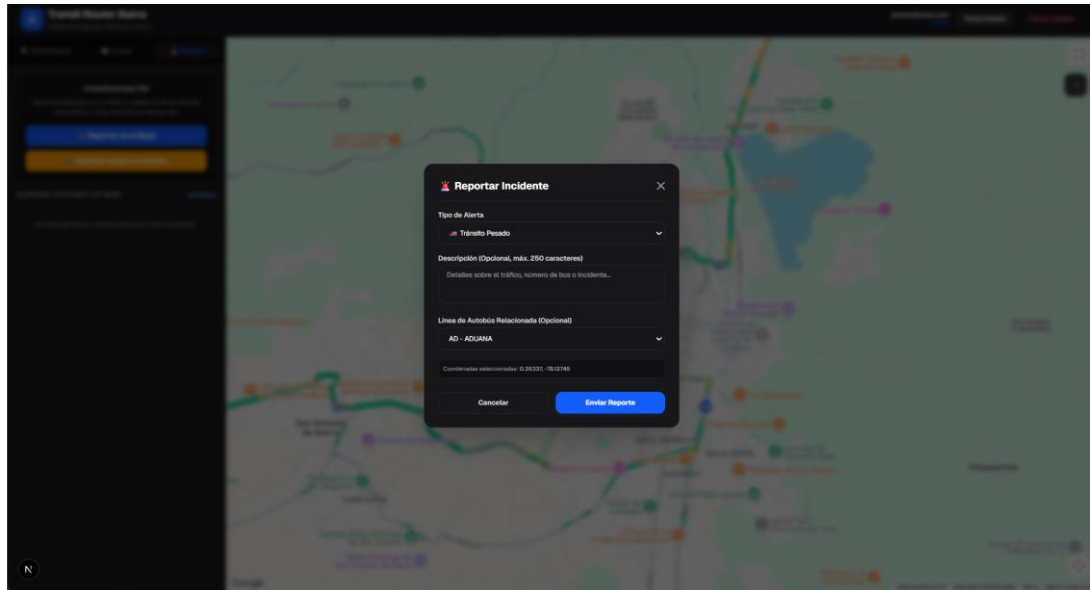
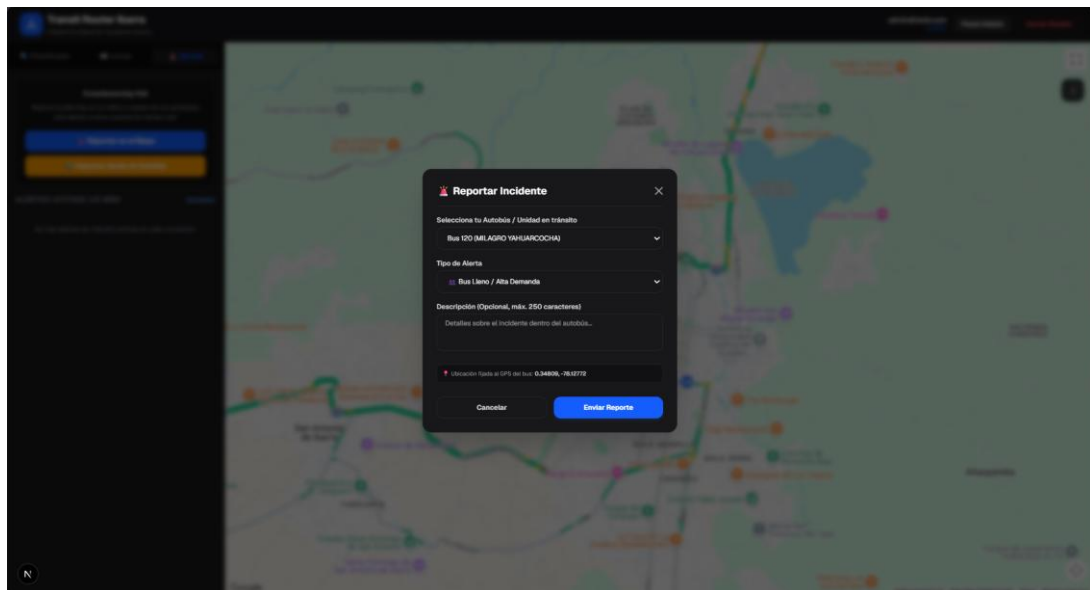


Figura 18

Registro de incidencias asociado a una unidad activa en circulación



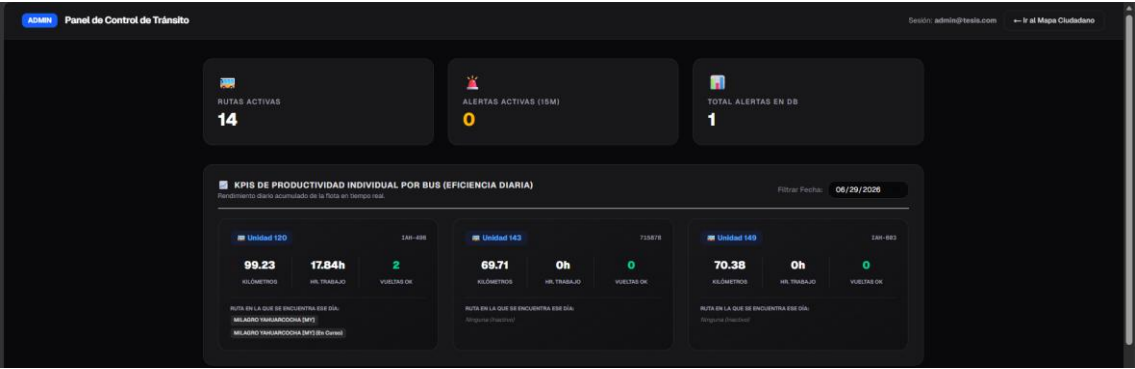
3.1.6 Dashboard administrativo

El dashboard administrativo constituye el entorno principal de supervisión interna del sistema. Para asegurar una gestión integral de la flota, este módulo organiza los datos operativos y los controles de integración en tres paneles diferenciados.

Panel A: KPIs de productividad individual por bus. Este panel despliega tarjetas analíticas por cada autobús, filtrables por fecha en tiempo real. El backend calcula

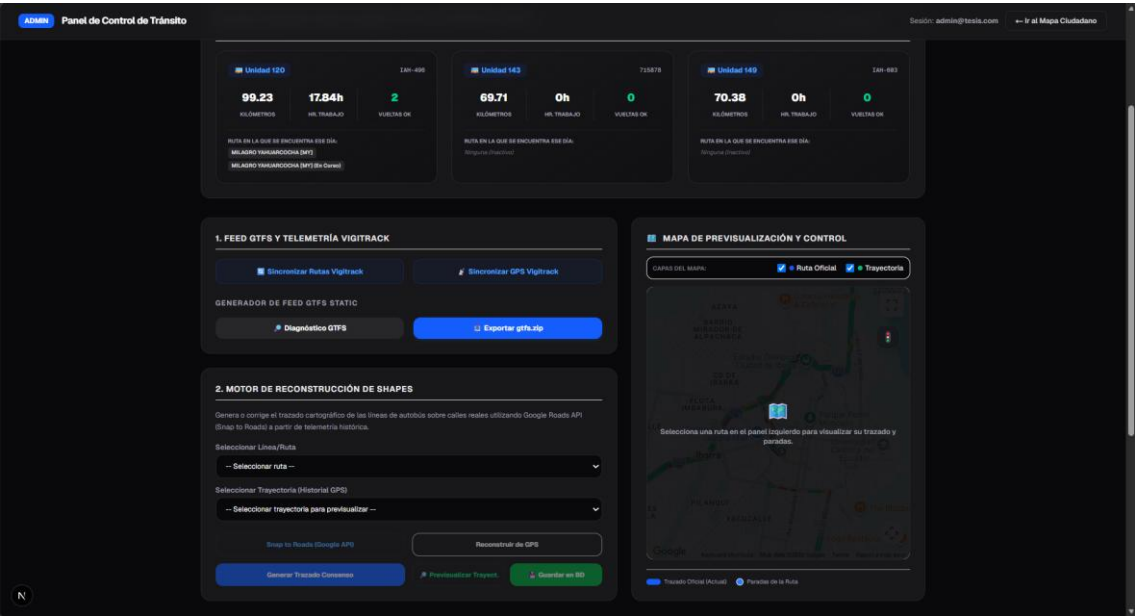
automáticamente los kilómetros recorridos acumulados, las horas operativas y el número de vueltas completadas con éxito durante la jornada a partir de los datos de telemetría.

Figura 19
KPIs de eficiencia individual y productividad de la flota



Panel B: Sincronización de feed GTFS y telemetría Vigitrack. Este panel reúne los controles necesarios para sincronizar, cuando se requiere, las rutas configuradas y las últimas posiciones satelitales directamente desde la API del proveedor tecnológico Vigitrack.

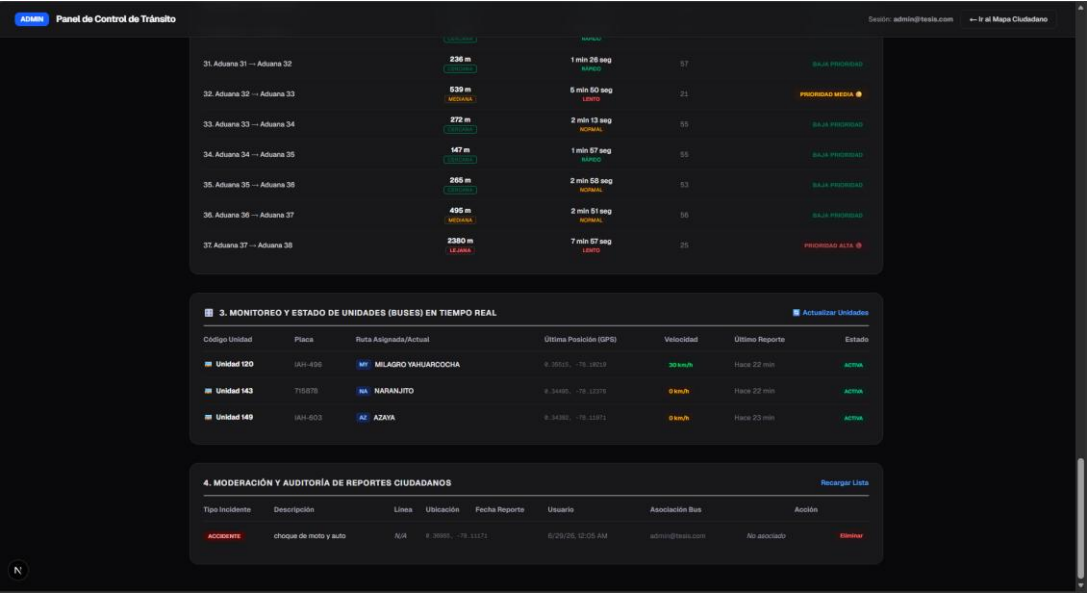
Figura 20
Interfaz de sincronización externa de telemetría



Panel C: Monitoreo y estado de unidades en tiempo real. Este panel presenta una tabla de seguimiento en vivo con el estado actual de cada autobús, su placa, la ruta asignada, la velocidad operativa calculada y el tiempo transcurrido desde su última transmisión GPS.

Figura 21

Tabla de monitoreo y estado de transmisión de la flota



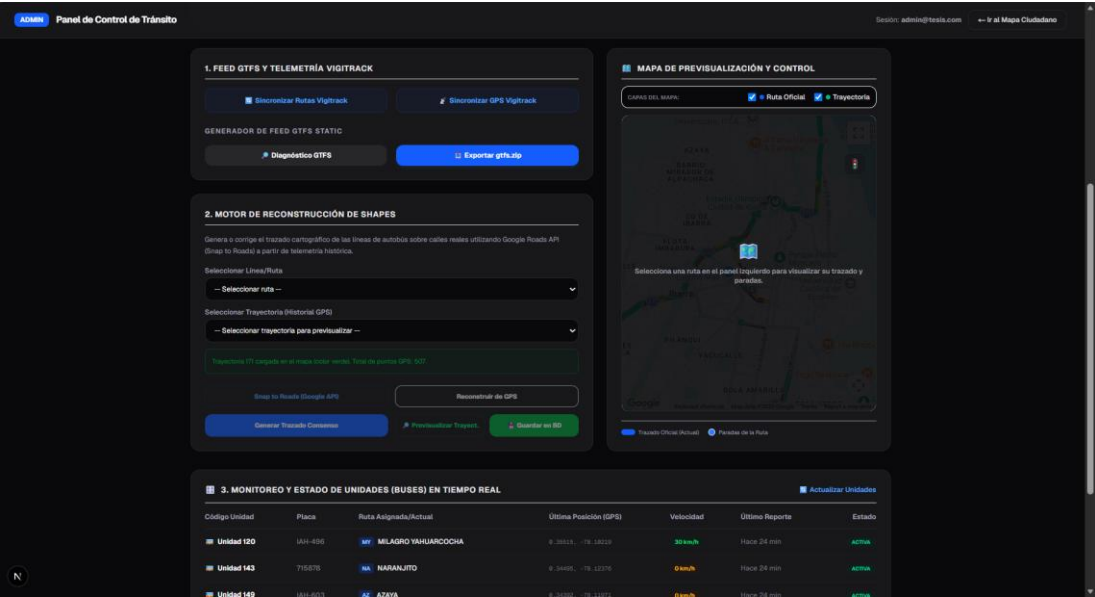
3.1.7 Módulo de auditoría de trayectorias y desvíos

Este componente permite al administrador examinar y evaluar viajes específicos, históricos o en curso, comparándolos con el trazado oficial y los tiempos esperados mediante cuatro paneles especializados.

Panel D: Reconstrucción de shapes. Este panel alinea coordenadas GPS dispersas y genera trayectorias corregidas utilizando Google Roads API, lo que permite mejorar la representación geométrica de los recorridos de la cooperativa.

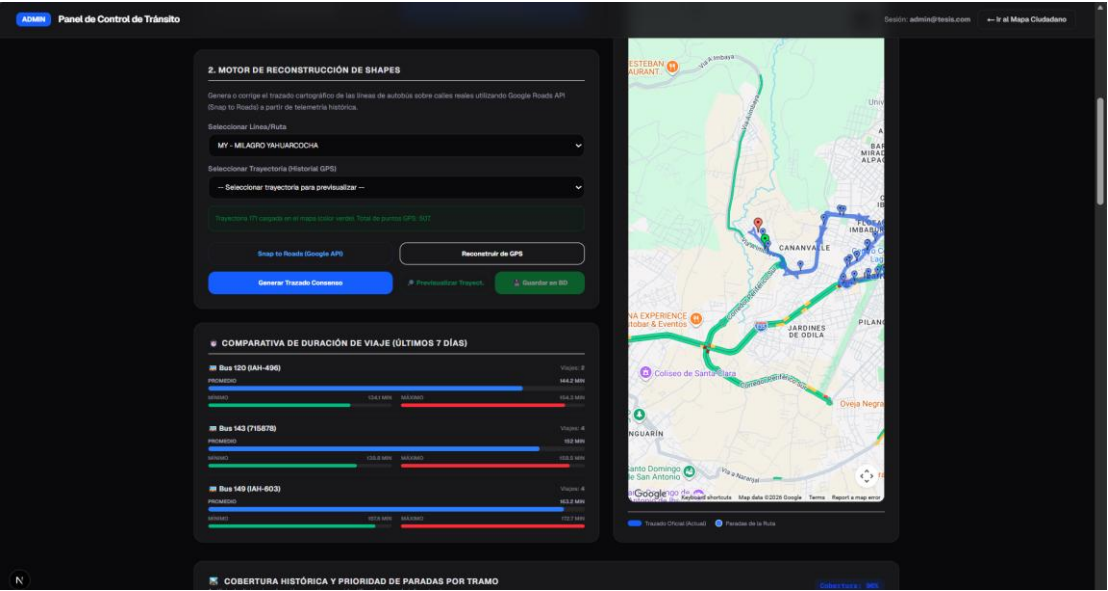
Figura 22

Interfaz de reconstrucción cartográfica y alineación de trazados



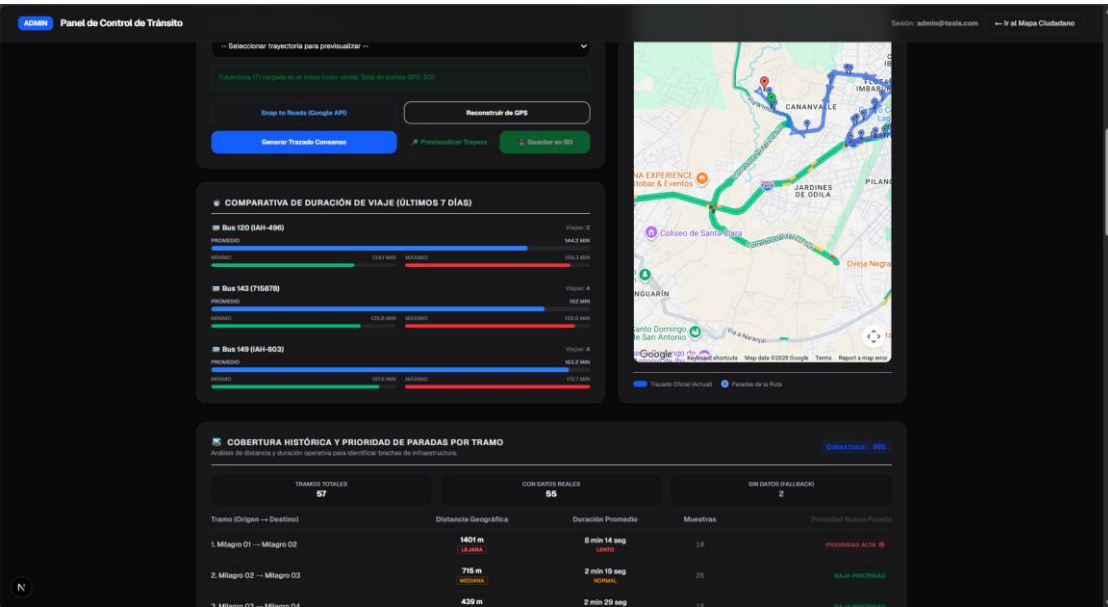
Panel E: Comparativa de duración de viaje. Este panel presenta diagramas comparativos con el tiempo promedio, mínimo y máximo registrado en los últimos siete días frente a la duración programada teóricamente.

Figura 23
Comparativa estadística de tiempos de tránsito por línea



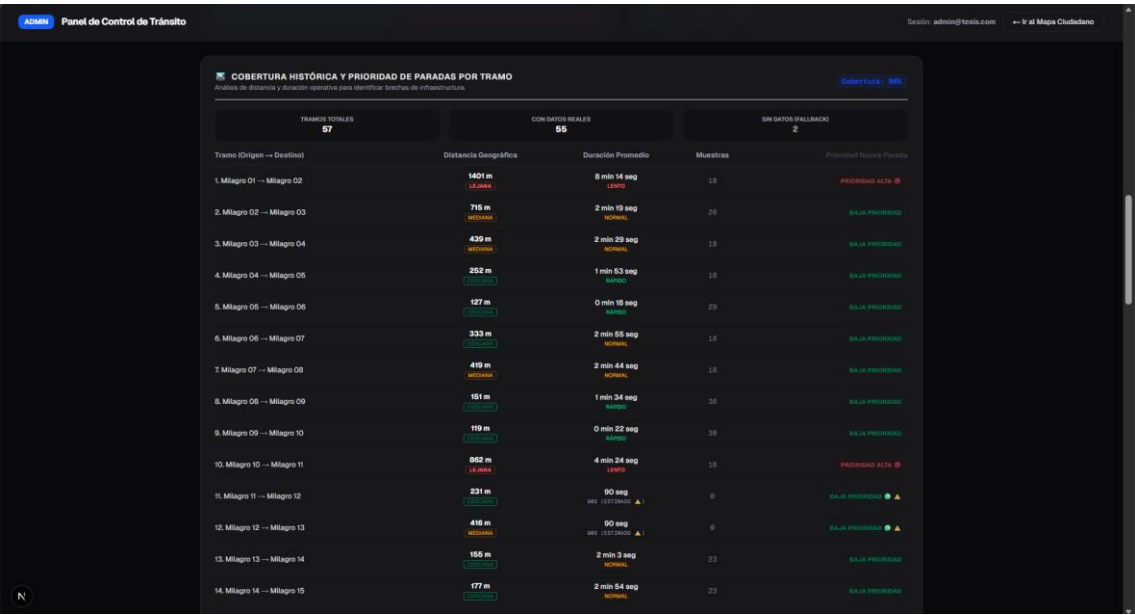
Panel F: Auditoría de desvíos y paradas omitidas por trayectoria. Este panel evalúa el nivel de cumplimiento general del viaje. El backend detecta cuántos puntos se desviaron del umbral aceptable, superior a 150 metros, e identifica el estado de cada parada como visitada u omitida.

Figura 24
Detalle de auditoría de paradas omitidas y desvíos espaciales



Panel G: Cobertura histórica y prioridad de paradas por tramo. Este panel analiza la velocidad media y el tiempo promedio de tránsito registrado entre cada par de paradas consecutivas, con el fin de clasificar y priorizar operativamente los tramos y paradas con mayores brechas de infraestructura.

Figura 25
Análisis de velocidad por tramos y prioridad de infraestructura

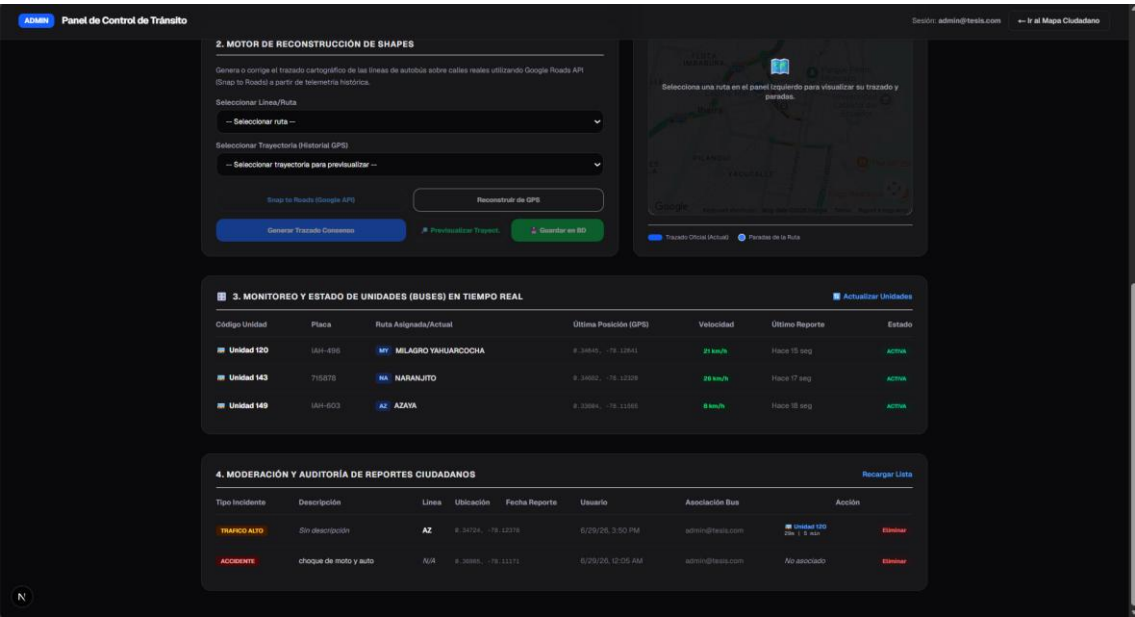


3.1.8 Módulo de moderación y auditoría de reportes ciudadanos

Como extensión de las funcionalidades de interacción ciudadana incorporadas al prototipo, esta sección gestiona la moderación y supervisión interna de los reportes enviados por los usuarios. Para ello, el backend ejecuta un algoritmo de correlación espacio-temporal que procesa cada incidente dentro de una ventana de tiempo de ± 10 minutos. Si una unidad de transporte en circulación se encontraba dentro de un radio menor o igual a 500 metros respecto al lugar del incidente, el sistema asocia automáticamente dicho autobús al reporte generado. Este módulo administrativo permite auditar la validez de los reportes, visualizar la distancia y la diferencia temporal con la que coincidieron el vehículo y la alerta, y aplicar acciones de moderación o eliminación de incidentes según corresponda.

Figura 26

Consola de moderación e incidentes ciudadanos correlacionados

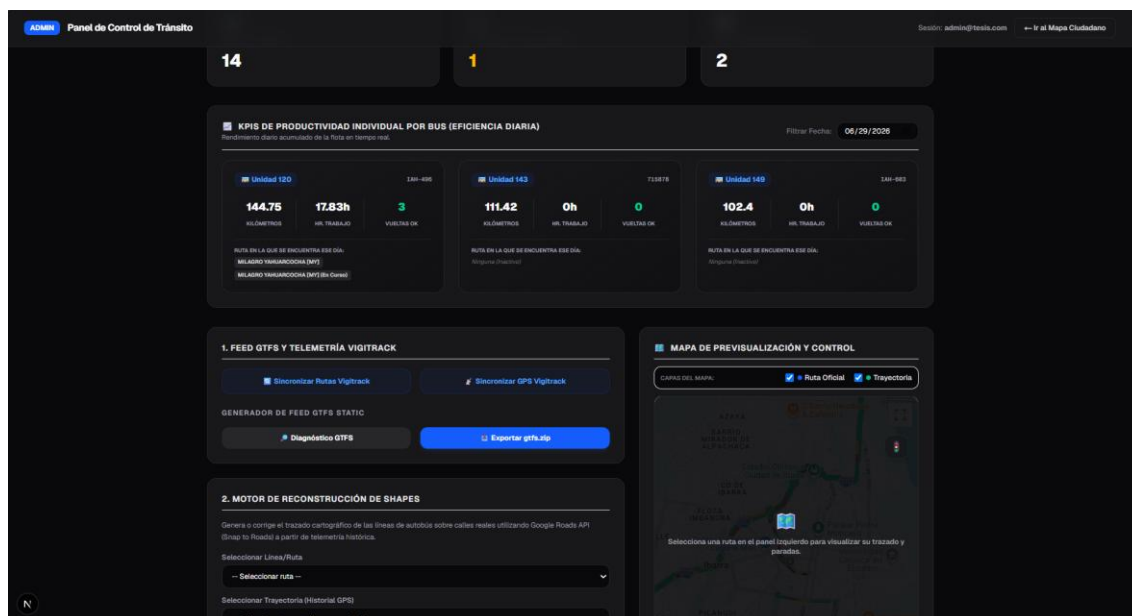


3.1.9 Módulo de diagnóstico y descarga de GTFS

Esta interfaz está destinada al control de interoperabilidad y al intercambio de información del sistema de transporte con plataformas externas de planificación de viajes, como Google Maps, mediante el estándar internacional GTFS (General Transit Feed Specification). El panel expone herramientas de diagnóstico estructural para verificar en el backend la consistencia lógica de rutas, paradas huérfanas, trazados geométricos y secuencias válidas de horarios antes de empaquetar y generar la descarga del archivo comprimido .zip, el cual contiene las tablas normalizadas del estándar, tales como agency.txt, routes.txt, stops.txt, shapes.txt, stop_times.txt y calendar.txt.

Figura 27

Herramienta de auditoría GTFS y exportación de archivos comprimidos



3.2 Cumplimiento de los requisitos funcionales

Con el fin de verificar el alcance funcional de la solución desarrollada, se evaluó el cumplimiento de los requisitos definidos durante la fase de análisis. Esta comprobación se centró en los requerimientos funcionales prioritarios que conforman el núcleo operativo del prototipo, relacionados con la recepción y procesamiento de telemetría GPS, la reconstrucción de trayectorias, el cálculo de ETA, la planificación de itinerarios, la generación de estructuras interoperables y la visualización de la información operativa. La verificación se realizó contrastando cada requerimiento con el componente implementado, su manifestación observable en las interfaces y los resultados obtenidos durante las pruebas funcionales.

Tabla 7*Matriz de cumplimiento de requisitos funcionales*

Código	Requisito funcional	Componente o módulo	Evidencia de cumplimiento	Estado
RF-1	Recepción de datos de geolocalización desde la fuente del proveedor GPS.	Módulo de integración de telemetría	Sincronización periódica con la API de monitoreo del proveedor.	Cumplido
RF-2	Almacenar datos GPS en una base de datos estructurada para consulta posterior.	Persistencia de datos	Estructura relacional con almacenamiento organizado de los registros GPS.	Cumplido
RF-3	Validar y depurar datos GPS ante registros duplicados e inconsistencias.	Módulo de saneamiento de datos	Exclusión automática de coordenadas duplicadas e inválidas antes del almacenamiento.	Cumplido
RF-4	Reconstruir trayectorias de las unidades a partir de los datos GPS.	Módulo de trayectorias	Segmentación de recorridos con marcas de tiempo de inicio, fin y estado del viaje.	Cumplido
RF-5	Asociar trayectorias con rutas y paradas definidas.	Módulo de asociación espacial	Detección geoespacial de arribo a paradas y construcción de tiempos por tramo.	Cumplido
RF-6	Calcular tiempos estimados de llegada de unidades.	Motor de ETA	Predicción basada en posición actual, tiempos históricos y reglas de respaldo.	Cumplido
RF-7	Generar archivos GTFS estáticos de la planificación del servicio.	Exportador GTFS	Generación y empaquetado del archivo comprimido con la estructura estándar.	Cumplido
RF-8	Generar feeds GTFS-Realtime para posiciones y actualizaciones de viaje.	Módulo GTFS-Realtime	Exposición de salidas dinámicas en formato Protocol Buffers.	Cumplido

RF-9	Visualización administrativa de unidades sobre un mapa interactivo.	Interfaz de monitoreo	Visualización dinámica de marcadores sobre el mapa administrativo.	Cumplido
RF-10	Visualización de indicadores operativos en panel administrativo.	Dashboard de métricas	Representación gráfica de vueltas, kilómetros y tiempos promedio.	Cumplido
RF-11	Estructurar datos GTFS y GTFS-Realtime compatibles con Google Maps.	Módulo de interoperabilidad	Generación de estructuras estáticas y dinámicas conforme al estándar.	Cumplido
RF-12	Permitir la planificación de itinerarios entre un origen y un destino, combinando tramos peatonales y transporte público.	Módulo de planificación de itinerarios	Generación de alternativas de viaje con criterio de costo percibido y restricción de caminata.	Cumplido

3.3 Cumplimiento de requisitos no funcionales

Además de las funcionalidades principales del sistema, se revisó el cumplimiento de criterios no funcionales relacionados con seguridad, despliegue, persistencia de datos y escalabilidad técnica. Esta evaluación permitió comprobar que el prototipo no solo responde a los requerimientos operativos definidos, sino que también mantiene condiciones técnicas adecuadas para su funcionamiento dentro del alcance del proyecto.

3.3.1 Seguridad

La seguridad del entorno administrativo se implementó mediante autenticación con tokens JWT. Las contraseñas de los usuarios se almacenan en la base de datos PostgreSQL protegidas mediante el algoritmo de hash bcryptjs. De forma adicional, el frontend valida el rol del perfil autenticado y restringe el acceso al panel administrativo, redirigiendo al formulario de inicio de sesión cuando un usuario no autorizado intenta acceder a rutas protegidas.

3.3.2 Alojamiento y despliegue

El sistema fue publicado en la plataforma Railway, estructurándose a través de servicios conectados para la base de datos, el backend y el frontend. La base de datos PostgreSQL aloja la estructura relacional del sistema; el backend desarrollado en NestJS opera con variables de entorno configuradas para la conexión y persistencia; y el frontend construido en Next.js fue ajustado para su carga dinámica en el cliente, reduciendo problemas de prerenderizado asociados a los componentes cartográficos.

3.3.3 Persistencia de datos

La persistencia de la información histórica de telemetría se apoya en PostgreSQL, garantizando la consistencia de la estructura relacional. Para evitar escrituras innecesarias en la base de datos, se aplican filtros de duplicidad y validaciones de tipos de datos a nivel de servidor. Esto permite conservar la información operativa de forma organizada y útil para consultas históricas y generación de reportes. Asimismo, la infraestructura utilizada permite incorporar mecanismos de respaldo y recuperación de datos de acuerdo con las necesidades de despliegue del sistema.

3.3.4 Escalabilidad técnica

El prototipo fue diseñado e implementado siguiendo principios de arquitectura modular. Los servicios del backend y los componentes visuales del frontend se encuentran desacoplados mediante el uso de API REST, lo que facilita futuras extensiones funcionales, la incorporación de nuevos proveedores de datos GPS o la integración de nuevas interfaces de visualización sin comprometer la base de datos ni el núcleo de procesamiento.

3.4 Validación consolidada del prototipo

A partir de la verificación de los requisitos funcionales, los criterios no funcionales y los casos de prueba definidos en la metodología, se elaboró una matriz consolidada de validación del prototipo. Esta matriz resume las principales dimensiones evaluadas, la evidencia revisada y el resultado obtenido durante la comprobación funcional del sistema. Su propósito es presentar una visión general de la validación antes de detallar los resultados específicos de las pruebas de caja negra.

Tabla 8

Matriz consolidada de resultados de validación del prototipo

Dimensión validada	Evidencia revisada	Resultado obtenido	Estado
Recepción de datos GPS	Sincronización con la API del proveedor GPS	Los registros de geolocalización ingresaron correctamente al flujo interno del sistema	Validado
Almacenamiento y persistencia	Registro de datos en PostgreSQL	La información se conservó de forma estructurada y disponible para consulta posterior	Validado
Validación y depuración de datos	Filtros de duplicidad, coordenadas inválidas e inconsistencias	Los registros inconsistentes fueron excluidos o tratados antes del procesamiento operativo	Validado
Reconstrucción de trayectorias	Agrupación de registros por unidad y secuencia temporal	Las trayectorias se generaron de forma coherente a partir de los datos GPS procesados	Validado
Asociación con rutas y paradas	Relación entre trayectorias, rutas y puntos de parada	El sistema vinculó las posiciones procesadas con la estructura operativa definida en el prototipo	Validado
Cálculo de ETA	Estimaciones generadas con datos activos, históricos y reglas de respaldo	Los tiempos estimados de llegada se calcularon de acuerdo con la lógica definida para el prototipo	Validado
Planificación de itinerarios	Consulta de rutas entre origen y destino	El sistema generó alternativas de viaje combinando tramos peatonales y transporte público	Validado
Generación de GTFS	Exportación de archivos estáticos del servicio	El archivo comprimido GTFS fue generado correctamente con la información requerida	Validado
Generación de GTFS-Realtime	Validación de feeds dinámicos de posiciones y actualizaciones	Los feeds dinámicos fueron interpretados sin errores críticos durante la validación	Validado

Visualización web	Mapa público y dashboard administrativo	La información operativa se visualizó correctamente en las interfaces implementadas	Validado
Seguridad de acceso	Autenticación, roles y rutas protegidas	El sistema restringió el acceso administrativo a usuarios autenticados y autorizados	Validado

3.5 Resultados de las pruebas de caja negra

La validación funcional del prototipo se complementó con pruebas de caja negra orientadas a verificar el comportamiento observable del sistema ante entradas simuladas y reales, contrastando los resultados obtenidos con las especificaciones definidas para el prototipo. A través de estas pruebas se evaluaron los módulos de recepción, validación, persistencia, reconstrucción de trayectorias, cálculo de ETA, exportación GTFS y planificación de rutas.

Tabla 9

Matriz de resultados de pruebas de caja negra

Código	Módulo o proceso	Entrada de prueba	Resultado esperado	Resultado obtenido	Estado
CP-01	Recepción GPS	Tramas completas desde el proveedor.	El sistema recibe los registros y los procesa para el flujo interno.	Registro recibido y almacenado correctamente.	Exitoso
CP-02	Recepción GPS	Tramas sin latitud, longitud o código de unidad.	El sistema rechaza el registro por falta de campos mínimos.	Registro descartado con error de validación.	Exitoso
CP-03	Recepción GPS	Tramas con formato o tipo de dato inválido.	El sistema detecta el error y evita el procesamiento.	La trama fue descartada con respuesta de error.	Exitoso
CP-04	Recepción GPS	Secuencia simultánea de tramas de varias unidades.	El sistema procesa de forma concurrente las tramas.	Ingesta concurrente procesada sin pérdida de registros.	Exitoso
CP-05	Validación	Trama idéntica a	El sistema identifica	Registro omitido por duplicidad.	Exitoso

		una ya persistida.	duplicidad y evita guardarla.		
CP-06	Validación	Coordenadas no numéricas u omitidas.	El sistema detecta la anomalía y excluye el registro.	Registro descartado por inconsistencia geográfica.	Exitoso
CP-07	Validación	Tramas con secuencia temporal desordenada.	El sistema ordena cronológicamente los puntos.	Registros ordenados para la reconstrucción posterior.	Exitoso
CP-08	Validación	Tramas con duración implícita incoherente.	El sistema detecta la anomalía y omite el tramo.	Tramo descartado por inconsistencia temporal.	Exitoso
CP-09	Validación	Mezcla de registros válidos e inválidos.	El sistema conserva únicamente los válidos.	Registros inválidos descartados y registros válidos almacenados.	Exitoso
CP-10	Base de datos	Escritura de registros GPS validados.	Almacenamiento exitoso en PostgreSQL.	Datos almacenados correctamente.	Exitoso
CP-11	Base de datos	Consulta de tramos históricos.	Recuperación rápida y ordenada.	Registros devueltos sin alteraciones.	Exitoso
CP-12	Base de datos	Contraste entre datos ingresados y persistidos.	Coincidencia de campos obligatorios.	Los valores almacenados coinciden con los enviados.	Exitoso
CP-13	Base de datos	Ingesta de tramas de múltiples flotas.	Agrupación correcta por unidad.	Datos agrupados eficientemente por vehículo y tiempo.	Exitoso
CP-14	Trayectorias	Coordenadas de autobús cruzando el punto de salida.	Apertura de trayectoria en curso.	Trayectoria creada correctamente.	Exitoso

CP-15	Trayectorias	Asociación de trayectorias con la geometría oficial.	Vinculación correcta con la ruta.	Asociación realizada de forma exitosa.	Exitoso
CP-16	Trayectorias	Autobús dentro del radio de una parada.	Registro de paso por parada.	Arribo registrado correctamente.	Exitoso
CP-17	ETA	Unidad activa con históricos disponibles.	Estimación dinámica de arribo.	ETA calculado correctamente.	Exitoso
CP-18	ETA	Unidad sin señal activa o detenida.	Aplicación del valor de respaldo.	ETA asignado automáticamente al valor por defecto.	Exitoso
CP-19	ETA	Consulta del ETA desde el módulo de monitoreo.	Visualización correcta en la interfaz.	ETA mostrado correctamente en el panel.	Exitoso
CP-20	GTFS	Exportación del catálogo estático.	Generación de archivo comprimido GTFS.	Exportación realizada sin errores.	Exitoso
CP-21	GTFS-Realtime	Solicitud a los feeds dinámicos.	Generación de respuestas en Protocol Buffers.	Feed dinámico respondido correctamente.	Exitoso
CP-22	Ruteador	Solicitud de ruta entre origen y destino con distancias variables.	Itinerarios calculados con penalización de 5 veces al esfuerzo peatonal.	Resultados ordenados priorizando menor caminata.	Exitoso

3.5.1 Pruebas del acceso al sistema

Se evaluó el comportamiento de la pantalla de inicio de sesión ingresando credenciales correctas, erróneas y campos vacíos. El sistema denegó el acceso ante datos inválidos y bloqueó la navegación hacia el dashboard cuando no existía autenticación activa. Solo

ante credenciales válidas fue posible recuperar el perfil del usuario y redirigir correctamente a la vista autorizada.

3.5.2 Pruebas del mapa público y monitoreo

Se verificó la carga de las interfaces geográficas y la actualización dinámica de la ubicación de las unidades dentro del mapa público. Durante la prueba, la integración del mapa en el navegador se mantuvo estable y permitió representar rutas, autobuses activos e incidencias asociadas sin afectar la interacción general del sistema.

3.5.3 Pruebas del planificador de rutas

Se ingresaron puntos de origen y destino dentro del mapa público para evaluar el comportamiento del planificador de rutas. Cuando la distancia de caminata directa fue menor a 800 metros, el sistema sugirió un desplazamiento peatonal directo. En cambio, ante distancias mayores, el algoritmo recomendó alternativas en autobús, identificó correctamente opciones con transbordos a pie inferiores a 250 metros y ordenó los itinerarios conforme al criterio de costo percibido.

3.5.4 Pruebas de interoperabilidad y validación de feeds GTFS y GTFS-Realtime

La consistencia y conformidad técnica de los datos de salida se validaron tanto para las estructuras estáticas GTFS como para las transmisiones en tiempo real GTFS-Realtime. Para ello, se utilizó una herramienta especializada capaz de consumir el archivo GTFS estático y los feeds dinámicos generados por el backend. Esta revisión permitió comprobar la sintaxis, la estructura y el cumplimiento de los campos obligatorios requeridos para la interoperabilidad del sistema.

En la primera etapa se configuró el validador con la ruta del archivo GTFS estático y con los feeds dinámicos correspondientes a VehiclePositions y TripUpdates. Esta configuración permitió que la herramienta actuara como consumidor externo de la información generada por el prototipo, reproduciendo así un escenario real de interoperabilidad.

Figura 28

Configuración del validador GTFS-Realtime con el feed GTFS estático y los feeds dinámicos del sistema

Welcome to GTFS-realtime Validator

What is the GTFS-Realtime Validator?

This tool takes GTFS and GTFS-realtime feeds as inputs, and examines the GTFS-realtime feed to determine if it properly meets GTFS-realtime best practices, both those explicitly listed in the spec and those based on the needs of consuming applications.

The rules that this tool uses to validate feeds are listed [here](#)

Getting Started

1. Add a link to your GTFS feed zip file (e.g., <http://mygtfs.org/gtfs.zip>)
2. Add a link to your GTFS-realtime feed (e.g., <http://mygtfs.org/trip-updates>)
3. Press " + Add Feed " if more feed needs to be monitored
4. Press the Start button to begin validation
5. Reset button can be used to reset all fields

Feed Details

Step 1: Add the URL for your GTFS zip file

GTFS feed

☒ Run GTFS validation

☒ Use shapes.txt for GTFS-realtime rules

Step 2: Add the URL for your GTFS-realtime feed

GTFS-Realtime Feed 1

GTFS-Realtime Feed 2

Interval (seconds):

Posteriormente, el proceso de validación permitió comprobar que los feeds en tiempo real podían ser interpretados correctamente, sin registrar errores ni advertencias críticas durante la auditoría. De esta manera, se verificó que la serialización en Protocol Buffers y la estructura general de los datos resultaban compatibles con el consumo externo esperado para este tipo de integración.

Figura 29

Resultados de la validación de los feeds GTFS-Realtime sin errores críticos

Feed - <http://host.docker.internal:3001/api/gtfs/realtime/vehicle-positions>

Summary

Http requests: 41

Unique responses: 41

[Most recent response](#)

ID	Title	Severity	Last iteration	Last time	Count	Show in log
----	-------	----------	----------------	-----------	-------	-------------

Log

Iteration	ID	Title	Severity	Timestamp ?
-----------	----	-------	----------	-------------

Feed - <http://host.docker.internal:3001/api/gtfs/realtime/trip-updates>

Summary

Http requests: 42

Unique responses: 42

[Most recent response](#)

ID	Title	Severity	Last iteration	Last time	Count	Show in log
----	-------	----------	----------------	-----------	-------	-------------

Log

Iteration	ID	Title	Severity	Timestamp ?
-----------	----	-------	----------	-------------

Por su parte, el feed GTFS estático también fue auditado mediante el módulo de diagnósticos integrado en el entorno administrativo del prototipo, el cual verifica de forma preventiva la consistencia de las entidades necesarias antes de exportar el archivo comprimido.

3.6 Discusión de resultados

Los resultados obtenidos muestran que el prototipo logró integrar de manera funcional los componentes principales previstos en la investigación. Entre ellos se incluyen la captura de telemetría GPS, la validación de datos, la reconstrucción de trayectorias, la estimación de tiempos de llegada, la planificación de itinerarios multimodales, la visualización web y la generación de información interoperable mediante GTFS y GTFS-Realtime. Desde el punto de vista funcional, la solución incorporó procesamiento dinámico y mecanismos de consulta ajustados a las condiciones operativas del entorno de prueba.

En cuanto al desempeño del sistema, la combinación de datos GPS en tiempo real, promedios históricos de tránsito y reglas lógicas de control permitió generar estimaciones de llegada más estables dentro de las limitaciones propias de la telemetría disponible. Entre estas reglas se incluyeron el acotamiento de velocidad y los valores de respaldo para el cálculo de ETA. De igual manera, la organización de la interfaz en dos entornos diferenciados permitió atender necesidades distintas dentro de la plataforma. El entorno ciudadano se orientó principalmente a la planificación de viajes y a la consulta de rutas, mientras que el entorno administrativo concentró herramientas de monitoreo en mapa, auditoría de cumplimiento y consulta de indicadores diarios de rendimiento por unidad.

No obstante, los resultados también evidencian que el alcance del prototipo depende de ciertas condiciones operativas. La regularidad de transmisión del proveedor GPS y la exactitud con la que las unidades siguen los recorridos oficiales influyen directamente en la precisión de las trayectorias reconstruidas y en las alertas de desvío generadas por el sistema. Por esta razón, la utilidad de la solución podría fortalecerse en futuras fases mediante la incorporación de un mayor volumen de datos históricos reales, pruebas de campo más prolongadas y la ampliación del número de rutas y unidades operativas.

En conjunto, los resultados permiten afirmar que el prototipo respondió de manera satisfactoria a los objetivos específicos planteados en la investigación. Se atendió el análisis de las necesidades de información del servicio, el diseño de la arquitectura

tecnológica, el procesamiento de datos de geolocalización para la generación de rutas y tiempos estimados de llegada, la estructuración interoperable de información compatible con Google Maps y el desarrollo de un panel administrativo orientado a la supervisión operativa.

En síntesis, los hallazgos demuestran la viabilidad técnica de construir una plataforma integrada para el procesamiento, análisis y visualización de datos de geolocalización del transporte público urbano. La solución desarrollada ofrece una base funcional útil y adaptable al contexto operativo de la Cooperativa de Transportes “28 de Septiembre”.

CONCLUSIONES

1. Se identificó que la Cooperativa de Transportes “28 de Septiembre” disponía de datos de geolocalización de sus unidades, pero carecía de una plataforma que permitiera procesarlos y transformarlos en información útil para la gestión operativa y la consulta ciudadana, justificando el desarrollo de la solución propuesta.
2. La arquitectura tecnológica diseñada permitió integrar de forma modular la recepción de datos GPS, el almacenamiento, el procesamiento de información, la interoperabilidad mediante estándares GTFS y GTFS-Realtime, y la visualización web, favoreciendo la escalabilidad y el mantenimiento del sistema.
3. La implementación del prototipo demostró que los datos de geolocalización pueden utilizarse para reconstruir trayectorias, estimar tiempos de llegada y planificar rutas, convirtiendo la telemetría GPS en una herramienta de apoyo para la operación del transporte urbano.
4. La generación de información en formatos GTFS y GTFS-Realtime permitió preparar el sistema para su integración con Google Maps y otras plataformas compatibles, estableciendo una base para la interoperabilidad del servicio de transporte.
5. El desarrollo del mapa público y del panel administrativo permitió ofrecer herramientas para la consulta de rutas e información del servicio, así como funcionalidades de monitoreo y gestión dirigidas a la cooperativa.
6. En conjunto, el prototipo cumplió el objetivo general de desarrollar una plataforma para el procesamiento y análisis de datos de geolocalización del transporte urbano con integración en Google Maps, demostrando su viabilidad técnica y su potencial para fortalecer la gestión operativa y mejorar el acceso a la información para los usuarios.

RECOMENDACIONES

1. Ampliar el prototipo mediante la incorporación de un mayor número de rutas, paradas y unidades, con el propósito de evaluar su comportamiento bajo condiciones de operación más cercanas al entorno real.
2. Incrementar la disponibilidad de datos históricos de geolocalización para mejorar la precisión de las estimaciones de llegada, fortalecer la reconstrucción de trayectorias y optimizar el cálculo de tiempos por tramo.
3. Mantener un proceso continuo de validación y actualización de los archivos GTFS y GTFS-Realtime, garantizando la consistencia de la información publicada conforme evolucionen las rutas, paradas, horarios y unidades del sistema.
4. Fortalecer el panel administrativo y la interfaz ciudadana mediante la incorporación de nuevas funcionalidades, indicadores operativos y mejoras de usabilidad que incrementen el aprovechamiento de la información generada.
5. Realizar pruebas de campo con usuarios y personal de la cooperativa para validar el desempeño del prototipo en un entorno operativo real e identificar oportunidades de mejora antes de su implementación definitiva.

REFERENCIAS BIBLIOGRÁFICAS

- Albán Reyes, M. L. (2022). *Propuesta de un diseño de rutas y frecuencias del transporte público urbano para mejorar la movilidad en el cantón Morona, periodo 2019* [Trabajo de titulación de maestría, Escuela Superior Politécnica de Chimborazo]. <https://dspace.espoch.edu.ec/handle/123456789/14998>
- Calatayud, A., Katz, R., & Riobó, P. (2022). *Impulsando la transformación digital del transporte en América Latina y el Caribe*. Banco Interamericano de Desarrollo. <https://publications.iadb.org/publications/spanish/document/Impulsando-la-transformacion-digital-del-transporte-en-America-Latina-y-el-Caribe.pdf>
- Colque, A., Valdivia, R., Navarrete, M., & Aracena, S. (2021). Un sistema de información geográfico para el transporte público basado en el estándar GTFS realtime. *Ingeniare. Revista Chilena de Ingeniería*, 29(1), 51–62. <https://doi.org/10.4067/S0718-33052021000100051>
- Corporación Andina de Fomento. (2023). *Movilidad urbana para más y mejores oportunidades*. CAF. <https://www.caf.com/media/4666452/impacto-caf-movilidad-urbana-informe-completo.pdf>
- Elassy, M., Al-Hattab, M., Takruri, M., & Badawi, S. (2024). Intelligent transportation systems for sustainable smart cities. *Transportation Engineering*, 16, 100252. <https://doi.org/10.1016/j.treng.2024.100252>
- Elijah, O., Keoh, S. L., bin Abdul Rahim, S. K., Seow, C. K., Cao, Q., bin Sarijari, M. A., Ibrahim, N. F., & Basuki, A. (2023). Transforming urban mobility with internet of things: Public bus fleet tracking using proximity-based bluetooth beacons. *Frontiers in The Internet of Things*, 2, 1255995. <https://doi.org/10.3389/friot.2023.1255995>
- Godfrid, J., Radnic, P., Vaisman, A., & Zimányi, E. (2022). Analyzing public transport in the city of Buenos Aires with MobilityDB. *Public Transport*, 14, 287–321. <https://doi.org/10.1007/s12469-022-00290-8>
- Hassan, M., Mahin, H. D., Ahmed, F., Hassan, M. M., Rahaman, A., & Abdullah, M. (2025). Assessing public transit network efficiency and accessibility in Johor Bahru and Penang, Malaysia: A data-driven approach. *Results in Engineering*, 27, 106126. <https://doi.org/10.1016/j.rineng.2025.106126>
- Javanmard, R., Liu, L., Long, J. A., & Lee, J. (2025). Using Realtime GTFS to generate easy-to-use transit accessibility measures under travel time uncertainty. *Travel Behaviour and Society*, 41, 101054. <https://doi.org/10.1016/j.tbs.2025.101054>

- Liu, D., Guo, J., Gu, Y., King, M., Han, L. D., & Brakewood, C. (2025). Analyzing transit systems using General Transit Feed Specification (GTFS) by generating spatiotemporal transit networks. *Information*, 16(1), 24.
<https://doi.org/10.3390/info16010024>
- Mansurova, A., Mussina, A., Aubakirov, S., Nugumanova, A., & Yedilkhan, D. (2025). From raw GPS to GTFS: A real-world open dataset for bus travel time prediction. *Data*, 10(8), 119. <https://doi.org/10.3390/data10080119>
- Nunes, C. A. do E. S. (2022). *Public transport management systems with real-time passenger information* [Dissertação de mestrado, Universidade de Évora].
https://dspace.uevora.pt/rdpc/bitstream/10174/32110/1/Mestrado-Engenharia_Informatica_E_Learning-Catarina_Alexandra_do_Espirito_Santos_Nunes%20.pdf
- Ramirez-Guerrero, T., Toro, M., Tabares, M. S., Salazar-Cabrera, R., & Pachón de la Cruz, Á. (2022). Key aspects for IT-services integration in urban transit service of medium-sized cities: A qualitative exploratory study in Colombia. *Sustainability*, 14(5), 2478. <https://doi.org/10.3390/su14052478>
- Rashvand, N., Hosseini, S. S., Azarbayjani, M., & Tabkhi, H. (2023). Real-time bus arrival prediction: A deep learning approach for enhanced urban mobility [Preprint]. *arXiv*. <https://doi.org/10.48550/arXiv.2303.15495>
- Ratneswaran, S., Thayasivam, U., & Thillaiambalam, S. (2024). GPS-2-GTFS: A Python package to process and transform raw GPS data of public transit to GTFS format [Preprint]. *arXiv*. <https://doi.org/10.48550/arXiv.2412.15221>
- Sharma, S., Mawane, N., Gowda, D. M., Taware, M., Kumar, C., Dixit, Y. C., & Ramesh, R. (2023). Estimating time of arrival of vehicle fleets with GCN based traffic prediction [Preprint]. *arXiv*. <https://doi.org/10.48550/arXiv.2311.12758>
- Tóth, R., Ispány, M., & Zichar, M. (2025). Interactive visualization for the GTFS and GTFS-RT data of Budapest. *ISPRS International Journal of Geo-Information*, 14(7), 245. <https://doi.org/10.3390/ijgi14070245>
- Traverso, D., Pacheco, G., & Castañeda, P. (2025). T-RAPPI: A machine learning model for the Corredor Metropolitano. En *Proceedings of the 11th International Conference on Vehicle Technology and Intelligent Transport Systems (VEHITS 2025)* (pp. 374–381). SCITEPRESS. <https://doi.org/10.5220/0013220700003941>

Velepucha, V., & Flores, P. (2023). A survey on microservices architecture: Principles, patterns and migration challenges. *IEEE Access*, *11*, 88339–88358.

<https://doi.org/10.1109/ACCESS.2023.3305687>

Vera, F., Arriagada, J., Hernández, F., & Munizaga, M. (2023). Indicadores de calidad de servicio de transporte público a partir de datos de la aplicación móvil de transporte TranSapp. *Estudios de Transporte*, *24*(2), 46–60.

<https://www.estudiosdetransporte.org/sochitran/article/view/296>

Para, S., Wirotasithon, T., Jundee, T., Demissie, M. G., Sekimoto, Y., Biljecki, F., & Phithakkitnukoon, S. (2024). G2Viz: An online tool for visualizing and analyzing a public transit system from GTFS data. *Public Transport*, *16*(3), 893–928.

<https://doi.org/10.1007/s12469-024-00362-x>

Wong, J., & Tsang, K. (2025). A highly available GTFS-RT positions system [Preprint]. arXiv. <https://doi.org/10.48550/arXiv.2508.01121>

ANEXOS

Carta de recepción de la propuesta tecnológica por parte del cliente



**COOPERATIVA DE TRANSPORTE
"28 DE SEPTIEMBRE"**

Oficio N°095-CTU28S-2026
Ibarra, 23 de Julio del 2026

Yo, Ing. Diana Estévez Granja, portadora de la cédula de ciudadanía N° 1002429304, en mi calidad de Presidenta de la Cooperativa de Transporte "28 de Septiembre", por medio de la presente,

CERTIFICO

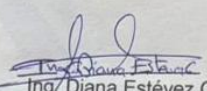
Que en representación de la Cooperativa de Transportes "28 de Septiembre", dejamos constancia de que hemos recibido y revisado el trabajo de titulación denominado "Plataforma digital para el procesamiento y análisis de datos de geolocalización del transporte urbano con integración en Google Maps", desarrollado por el estudiante Jean Pierre Ramírez Cevallos, portador de la cédula de ciudadanía N.° 1003688791.

Una vez evaluado el proyecto, manifestamos nuestra conformidad con los resultados obtenidos, ya que la propuesta aporta una herramienta tecnológica para gestionar y analizar los datos de geolocalización de las unidades, visualizar información de rutas y generar estimaciones útiles para la operación del servicio.

Asimismo, certificamos que el estudiante cumplió con los objetivos establecidos y desarrolló sus actividades con responsabilidad y profesionalismo.

Por lo expuesto, la Cooperativa acepta satisfactoriamente el trabajo realizado y emite la presente carta para los fines académicos correspondientes.

Atentamente,


Ing. Diana Estévez G.
PRESIDENTA DE LA COOPERATIVA
Correo: cooperativa28deseptiembre@hotmail.com
Celular: 099 3237187
Dirección: Av. Cristóbal de Troya 2-151 y Obispo Jesús Yerovi.

Dir.: El Camal, Av. Cristóbal de Troya 2-151 y Jesús Yerovi ☎ 2547 246 / 2547 185
Email: cooperativa28deseptiembre@hotmail.com / gerencia_cooperativa28deseptiembre@hotmail.com