# UNIVERSIDAD ESTATAL DE BOLÍVAR

## FACULTAD DE CIENCIAS ADMINISTRATIVAS, GESTIÓN EMPRESARIAL E INFORMÁTICA

CARRERA DE SOFTWARE

**TRABAJO DE INTEGRACIÓN CURRICULAR**
**PREVIO A LA OBTENCIÓN DEL TÍTULO DE**
**INGENIERO(A)(S) EN SOFTWARE**

**FORMA:** PROYECTO TECNOLÓGICO

**TEMA:**

SISTEMA WEB PARA EL SEGUIMIENTO DE ENCOMIENDAS DE LA COOPERATIVA DE TRANSPORTE FLOTA BOLÍVAR


**AUTOR(A):**

Leydi Marianela Galeas Armijo
Ronny Johel Cabezas Lechón

**DIRECTORA:**

Ing. Mónica Elizabeth Bonilla Manobanda


**GUARANDA – ECUADOR**

**2025**

**TEMA DEL PROYECTO TECNOLÓGICO**

SISTEMA WEB PARA EL SEGUIMIENTO DE ENCOMIENDAS DE LA COOPERATIVA DE TRANSPORTE FLOTA BOLIVAR

**DEDICATORIA**

Con todo mi amor y gratitud, dedico o este trabajo: a mis amados padres, quienes, con su esfuerzo, su amor incondicional y su ejemplo de vida, han sido mi más grande inspiración y el motor de mi formación. Este logro es el fruto tangible de todo el apoyo y sacrificio que me han brindado a lo largo de nuestras vidas.

A mis hermanos y familiares, por su incondicional aliento y por compartir la alegría de este camino. Y a mis familiares a quienes ya no están físicamente con nosotros, cuya memoria y espíritu me acompañaron en cada jornada de esfuerzo. Se que su ejemplo sigue vivo y que celebren conmigo la culminación de este sueño. A todos ustedes, dedico este éxito.

**Ronny J. Cabezas.**

Con profundo amor y sincera gratitud, dedico este trabajo a las personas que han sido el sostén fundamental en mi vida.

A mis padres, por sembrar en mí valores, principios y la fortaleza necesaria para no rendirme ante las dificultades. Su ejemplo constante de esfuerzo y honestidad ha guiado cada uno de mis pasos y ha hecho posible la culminación de esta meta.

A mi esposo, por su apoyo incondicional, por su paciencia en los momentos de mayor dedicación y por motivarme a seguir adelante aun cuando el camino parecía difícil. Su compañía ha sido esencial en este proceso.

A mis hijos, quienes representan mi mayor inspiración y la razón de mi superación diaria. Todo sacrificio realizado tiene sentido al pensar en el futuro que deseo construir junto a ustedes.

Este logro no es solo mío, sino también de todos ustedes, que con amor y confianza hicieron posible que este sueño se convierta en realidad.

**Leydi M. Galeas**

**AGRADECIMIENTO**

Al culminar esta etapa fundamental que representa nuestro trabajo de grado, deseamos expresar nuestro más sincero agradecimiento a quienes hicieron posible este logro. En primer lugar, a Dios, por ser la guía y la fortaleza que nos permitió culminar la carrera universitaria y alcanzar esta meta profesional. De igual forma, extendemos nuestro agradecimiento a nuestras respectivas familias, por ser la base y el soporte esencial que permitió la culminación de este proyecto. Sin su aliento y comprensión, nada de esto hubiese sido posible.

Agradecemos de manera especial a la ingeniera Mónica Bonilla, nuestra directora de trabajo de titulación, por su dirección experta, su paciencia y su invaluable conocimiento que nos guio paso a paso en el desarrollo e implementación del sistema web para el seguimiento de encomiendas de la Cooperativa de Transporte Flota Bolívar. También queremos hacer un reconocimiento especial a los ingenieros Maricela Espín y Darwin Carrión, cuya mentoría y enseñanzas fueron fundamentales no solo para el desarrollo metodológico de esta investigación, sino también para nuestra formación integral a lo largo de toda la carrera. A nuestros profesores y compañeros, con quienes hemos compartido estos años de formación, muchas gracias por formar parte de este proceso, por las valiosas lecciones y por el apoyo mutuo que nos permitió convertirnos en profesionales. A todos, por su fe, su ayuda y su inspiración: MUCHAS GRACIAS.

# CERTIFICADO DE VALIDACIÓN

UNIVERSIDAD ESTATAL DE BOLIVAR

FACULTAD DE CIENCIAS ADMINISTRATIVAS, GESTIÓN EMPRESARIAL E INFORMÁTICA

## CERTIFICADO DE VALIDACIÓN

Ing. Mónica Bonilla, Ing. Maricela Espín e Ing. Darwin Carrión, en su orden directora y Pares Académicos del Trabajo de Integración Curricular "SISTEMA WEB PARA EL SEGUIMIENTO DE ENCOMIENDAS DE LA COOPERATIVA DE TRANSPORTE FLOTA BOLÍVAR" desarrollado por el señor Ronny Johel Cabezas Lechón y la señorita Leydi Marianela Galeas Armijo.

## CERTIFICAN

Que, luego de revisado el Trabajo de Integración Curricular en su totalidad, cumple con las exigencias académicas de la carrera SOFTWARE, por lo tanto, autorizamos su presentación y defensa.

Guaranda, 23 de febrero del 2026




Ing. Mónica Bonilla
**Directora**




Ing. Maricela Espín
**Par Académico**




Ing. Darwin Carrión
**Par Académico**

Dirección: Av. Ernesto Che Guevara y Gabriel Secaira
Guaranda-Ecuador
Teléfono: (593) 3220 6059
**www.ueb.edu.ec**

## DERECHOS DE AUTOR

Nosotros **CABEZAS LECHÓN RONNY JOHEL** y **GALEAS ARMIJO LEYDI MARIANELA** portadores de la Cédula de Identidad No. **1727598888** y **0202321642** en calidad de autores y titulares de los derechos morales y patrimoniales del Trabajo de Titulación: **SISTEMA WEB PARA EL SEGUIMIENTO DE ENCOMIENDAS DE LA COOPERATIVA DE TRANSPORTE FLOTA BOLÍVAR**. Modalidad **Proyecto Tecnológico**, de conformidad con el Art. 114 del CÓDIGO ORGÁNICO DE LA ECONOMÍA SOCIAL DE LOS CONOCIMIENTOS, CREATIVIDAD E INNOVACIÓN, concedemos a favor de la Universidad Estatal de Bolívar, una licencia gratuita, intransferible y no exclusiva para el uso no comercial de la obra, con fines estrictamente académicos. Conservamos a mi/nuestro favor todos los derechos de autor sobre la obra, establecidos en la normativa citada.

Así mismo, autorizamos a la Universidad Estatal de Bolívar, para que realice la digitalización y publicación de este trabajo de titulación en el Repositorio Digital, de conformidad a lo dispuesto en el Art. 144 de la Ley Orgánica de Educación Superior.

El (los) autor (es) declara (n) que la obra objeto de la presente autorización es original en su forma de expresión y no infringe el derecho de autor de terceros, asumiendo la responsabilidad por cualquier reclamación que pudiera presentarse por esta causa y liberando a la Universidad de toda responsabilidad.

**Ronny Johel Cabezas Lechón**
**C.I.: 1727598888**

**Leydi Marianela Galeas Armijo**
**C.I.: 0202321642**

# ÍNDICE DE CONTENIDOS

TEMA DEL PROYECTO TECNOLÓGICO ..... i
DEDICATORIA ..... ii
AGRADECIMIENTO ..... iii
CERTIFICADO DE VALIDACIÓN ..... iv
ÍNDICE DE CONTENIDOS ..... vi
ÍNDICE DE TABLAS ..... ix
ÍNDICE DE FIGURAS ..... xi
INTRODUCCIÓN ..... 1
RESUMEN ..... 2
ABSTRACT ..... 3
KICHWA ..... 4
CAPÍTULO I ..... 5
FORMULACIÓN GENERAL DEL PROYECTO ..... 5
1.1 Tema ..... 5
1.2 Descripción del Problema ..... 5
1.3 Justificación ..... 5
1.4 Objetivos: General y Específicos ..... 6
1.4.1 Objetivo General ..... 6
1.4.2 Objetivos Específicos ..... 6
CAPÍTULO II ..... 7
MARCO TEÓRICO ..... 7
2.1 Antecedentes ..... 7
2.2 Científico ..... 11
2.3 Conceptual ..... 16
2.4 Legal ..... 21

2.5 Georreferencial ..... 28
CAPÍTULO III ..... 29
METODOLOGÍA ..... 29
3.1 Metodología de Desarrollo de Software ..... 29
3.2 Técnicas e Instrumentos de Recopilación de Datos ..... 33
CAPÍTULO IV ..... 34
INGENIERÍA DEL PROYECTO ..... 34
4.1 Planificación ..... 34
4.1.1 Gestión del trabajo mediante Kanban. ..... 43
4.2 Diseño ..... 44
4.2.1 Arquitectura del software ..... 44
4.2.2 Vistas arquitectónicas 4 + 1 ..... 45
4.3 Codificación ..... 56
4.4 Pruebas ..... 69
4.5 Entrega ..... 77
CONCLUSIONES ..... 78
RECOMENDACIONES ..... 79
BIBLIOGRAFÍA ..... 80
ANEXOS ..... 83
Objeto del documento ..... 24
Elaborado ..... 24
Objetivo ..... 24
Requerimientos ..... 24
Manual de usuario ..... 25
1. Pantalla Inicial ..... 25
2. Iniciar sesión ..... 25
Registro de usuario ..... 27

3. Usuario con Rol de Gerente ..... 28
Presentación de la pantalla principal “Panel de Control” ..... 28
Gestión de usuarios ..... 29
Gestión de Oficinas ..... 31
Gestión de Rutas ..... 33
Reportes ..... 35
4. Usuario con Rol de Oficinista ..... 37
Presentación de la pantalla principal “Panel de Control” ..... 37
Gestión de Encomiendas ..... 38
Filtros ..... 40
Rastreo de Encomiendas ..... 43
Introducción ..... 52
Objetivo ..... 52
Manual técnico ..... 52
1. Especificaciones técnicas ..... 52
2. Instalación de dependencias ..... 53
3. Ejecución en modo de desarrollo ..... 54
4. Alojamiento del sistema en Render ..... 56
Alojamiento del backend Laravel en Web Service ..... 56
Alojamiento del frontend React en Static site. ..... 58
Crear la base de datos en Postgres. ..... 59
5. Conexión y configuración al dominio www.ctflotabolivar.com ..... 60
6. Estructura del sistema ..... 61
Frontend React ..... 61
Backend Laravel ..... 61

# ÍNDICE DE TABLAS

Tabla 1. *Características principales de las Metodologías Ágiles.* ..... 13
Tabla 2. *Sistemas Gestores de Base de Datos* ..... 17
Tabla 3. *Lenguajes de programación utilizados para el desarrollo de aplicativos webs.* ..... 18
Tabla 4. *Historia de usuario 1* ..... 34
Tabla 5. *Historia de usuario 2* ..... 34
Tabla 6. *Historia de usuario 3* ..... 35
Tabla 7. *Historia de usuario 4* ..... 35
Tabla 8. *Historia de usuario 5* ..... 36
Tabla 9. *Historia de usuario 6* ..... 36
Tabla 10. *Historia de usuario 7* ..... 37
Tabla 11. *Historia de usuario 8* ..... 37
Tabla 12. *Historia de usuario 9* ..... 38
Tabla 13. *Historia de usuario 10* ..... 38
Tabla 14. *Historia de usuario 11* ..... 39
Tabla 15. *Historia de usuario 12* ..... 40
Tabla 16. *Historia de usuario 13* ..... 40
Tabla 17. *Historia de usuario 14* ..... 41
Tabla 18. *Historia de usuario 15* ..... 42
Tabla 19. *Registro de riesgos* ..... 42
Tabla 20. *Stakeholders* ..... 70
Tabla 21. *Gestor de pruebas* ..... 70
Tabla 22. *Pruebas de registro de oficinistas.* ..... 71
Tabla 23. *Gestión de oficinas* ..... 71
Tabla 24. *Gestión de rutas* ..... 72
Tabla 25. *Reportes de encomiendas* ..... 72

Tabla 26. *Gestión de encomiendas* ..... 72

Tabla 27. *Pruebas de funcionalidad* ..... 73

Tabla 28. *Pruebas de disponibilidad* ..... 74

Tabla 29. *Pruebas de rendimiento* ..... 74

Tabla 30. *Pruebas de seguridad* ..... 75

Tabla 31. *Pruebas de usabilidad* ..... 75

Tabla 32. *Pruebas de integración* ..... 76

# ÍNDICE DE FIGURAS

Figura 1. *Mapa Geográfico de la ubicación de la Cooperativa de Transporte "Flota Bolívar" del cantón Guaranda.* ..... 28

Figura 2. *Implementación de metodología Kanban.* ..... 43

Figura 3. *Diagrama de Entidad – Relación.* ..... 44

Figura 4. *Diagrama de patrón arquitectónico de N-capas* ..... 45

Figura 5. *Diagrama de caso de uso (escenarios)* ..... 46

Figura 6. *Diagrama de clases (Vista lógica)* ..... 47

..... 47

Figura 7. *Diagrama de secuencia del proceso de registro y seguimiento de encomienda.* ..... 49

Figura 8. *Diagrama de secuencia del proceso de entrega de encomienda al chofer y recepción en destino.* ..... 50

Figura 9. *Diagrama de secuencia del proceso de entrega final de encomienda al destinatario.* ..... 50

Figura 10. *Diagrama de paquetes (Vista de despliegue)* ..... 52

Figura 11. *Diagrama de actividad, proceso de registro de encomienda.* ..... 53

Figura 12. *Diagrama de actividad, entrega de la encomienda al destinatario* ..... 54

Figura 13. *Diagrama de despliegue.* ..... 55

Figura 14. *Logo de la Cooperativa de Transportes Flota Bolívar* ..... 61

Figura 15. *Login para ingreso al interfaz gerente y oficinista* ..... 61

Figura 16. *Interfaz de seguimiento de encomienda* ..... 62

Figura 17. *Interfaz principal* ..... 62

Figura 18. *Panel de control.* ..... 63

Figura 19. *Gestión de encomiendas* ..... 63

Figura 20. *Formulario de registro de encomienda.* ..... 64

Figura 21. *Gestión de usuarios.* ..... 64

Figura 22. *Gestión de choferes.* ..... 65

Figura 23. *Gestión de oficinas.* ..... 65

Figura 24. *Gestión de rutas.* ..... 66

Figura 25. *Gestión de buses.* ..... 66

Figura 26. *Reportes de encomiendas.* ..... 67

# INTRODUCCIÓN

En la actualidad, la transformación digital se ha convertido en un factor determinante para la competitividad y eficiencia de las organizaciones, especialmente en sectores que requieren control logístico y gestión de información en tiempo real. El sector del transporte terrestre de encomiendas no es ajeno a esta realidad, ya que enfrenta desafíos relacionados con la trazabilidad de envíos, el control administrativo y la atención oportuna al usuario. En muchas cooperativas de transporte, los procesos de registro y seguimiento de encomiendas aún se realizan de manera manual o mediante sistemas poco integrados, lo que genera retrasos, pérdida de información y dificultades en la supervisión operativa.

Los procesos de encomiendas de la Cooperativa de Transporte Bolívar, presenta la necesidad de modernizar sus procesos mediante la implementación de herramientas tecnológicas que permitan optimizar la gestión de encomiendas. La ausencia de un sistema digital de seguimiento limita la capacidad de ofrecer información precisa y en tiempo real a los usuarios, afectando la eficiencia del servicio y la confianza del cliente.

En este contexto, el presente proyecto propone el diseño y desarrollo de un sistema web orientado al registro, control y seguimiento de encomiendas, con el propósito de mejorar la trazabilidad, fortalecer la gestión administrativa y contribuir a la modernización tecnológica de la cooperativa. El sistema se fundamenta en principios de arquitectura de software de tres capas, garantizando una adecuada separación de responsabilidades entre la interfaz de usuario, la lógica de negocio y la base de datos. Asimismo, se adopta una metodología de desarrollo ágil que permite una implementación flexible y adaptable a los requerimientos institucionales.

Desde el punto de vista académico, esta investigación integra fundamentos teóricos relacionados con sistemas de información, trazabilidad logística y metodologías ágiles de desarrollo de software, articulándolos con un enfoque aplicado orientado a la solución de un problema real. Además, el proyecto considera el cumplimiento de la normativa ecuatoriana vigente en materia de protección de datos personales y propiedad intelectual, asegurando el uso responsable de la información.

**RESUMEN**

El presente proyecto tecnológico tiene como objetivo el diseño y desarrollo de un sistema web para el seguimiento y control de encomiendas en la Cooperativa de Transporte Bolívar, con el propósito de optimizar los procesos administrativos y mejorar la trazabilidad de los envíos. Actualmente, la gestión de encomiendas en muchas cooperativas de transporte se realiza mediante procedimientos manuales o sistemas poco integrados, lo que genera retrasos, pérdida de información y dificultades en el control operativo.

La investigación se fundamenta en teorías relacionadas con sistemas de información, trazabilidad logística, arquitectura de software y metodologías ágiles de desarrollo. Se adopta un enfoque metodológico aplicado con un diseño orientado a la solución de un problema real dentro del contexto organizacional. Para el desarrollo del sistema se emplea una arquitectura de tres capas (presentación, lógica de negocio y datos), garantizando escalabilidad, seguridad y mantenibilidad.

El sistema propuesto permitirá el registro digital de encomiendas, la consulta del estado de los envíos en tiempo real y la generación de reportes administrativos, fortaleciendo la transparencia y la eficiencia del servicio. Asimismo, se consideran los lineamientos establecidos en la normativa ecuatoriana vigente respecto a la protección de datos personales y propiedad intelectual.

Como resultado, el proyecto contribuye a la transformación digital del sector transporte local, promoviendo una gestión moderna, segura y orientada a la satisfacción del usuario.

**Palabras clave:** sistema web, trazabilidad, encomiendas, transporte terrestre, transformación digital.

**ABSTRACT**

This research project aims to design and develop a web-based system for tracking and controlling parcels in the Bolívar Transportation Cooperative, in order to optimize administrative processes and improve shipment traceability. Currently, parcel management in many transportation cooperatives is carried out through manual procedures or poorly integrated systems, which leads to delays, information loss, and operational control difficulties.

The study is based on theoretical foundations related to information systems, logistics traceability, software architecture, and agile development methodologies. An applied methodological approach was adopted, focused on solving a real organizational problem. The system was developed using a three-tier architecture (presentation, business logic, and data layer), ensuring scalability, security, and maintainability.

The proposed system enables digital parcel registration, real-time shipment tracking, and administrative report generation, strengthening service transparency and operational efficiency. Additionally, the project complies with Ecuadorian regulations regarding personal data protection and intellectual property.

As a result, this project contributes to the digital transformation of the local transportation sector, promoting modern, secure, and user-oriented management practices.

**Keywords:** web system, traceability, parcels, land transportation, digital transformation.

**KICHWA**

Kay yachay ruraypa hatun yuyayka shina kan: Bolívar Transporte Cooperativapi encomiendas katina, rikuna, kamachina sistema web ruray, shuk alli kamachiyta churanapak, llankaykunata sumakyachinapak, chaymantapash apamushka kargakunapa ñanpi katinata alli rikurinapak. Kunankaman ashka transporte cooperativakunapi encomiendas kamachiyka makipi rurashka o mana alli tinkishka sistemakunawan rurakun, chaymantami ñawpakman purishpa llakikuna rikurin: shuyanakuy, willay chinkay, kamachiy mana alli rikuy.

gKay yachayka sistemas de información, trazabilidad logística, arquitectura de software, metodologías ágiles nishkakunapi yachaykunata hapishpa rurashka kan. Kay proyecto ruraypika kimsa patapi arquitectura (rikuchiy, negocio yuyay, datos wakichiy) churashka kan, shinallatak wiñarinapak, alli wañuchiy y mana sasachinakuyta rikuchin.

Rurashka sistemaka encomiendas killkakatina digitalpi, katina tiempo realpi, shinallatak reportekunata rurana ushanka, chaymantami servicio transparencia shinallatak llankay alli puriyta sumakyachin. Kay proyecto Ecuador llaktapi datos personales wañuchiy kamachikunata shinallatak propiedad intelectual kamachikunata respetashpa rurashka kan.

Tukuy yachay rurayka transporte sectorpi tecnología mushukyachiyta yanapan, alli, seguro, usuario-man rikurishka kamachiyta wiñachinapak.

**Shimikuna hatun yuyaykuna:** sistema web, trazabilidad, encomiendas, transporte terrestre, tecnología mushukyachiy.

# CAPÍTULO I

# FORMULACIÓN GENERAL DEL PROYECTO

## 1.1 Tema

Sistema web para el seguimiento de encomiendas de la Cooperativa de Transporte Flota Bolívar

## 1.2 Descripción del Problema

Actualmente, la Cooperativa de Transporte Flota Bolívar realiza el proceso de envío y recepción de paquetes de forma manual o semi automatizada con el uso de hojas de cálculo, lo que genera incertidumbre en los clientes respecto al estado y ubicación de sus envíos. Esta falta de trazabilidad provoca retrasos en la entrega de información, dificultades para atender reclamos y una percepción negativa del servicio. El personal operativo enfrenta desafíos en la organización y control de las encomiendas, lo que puede derivar en pérdidas o entregas incorrectas.

El control logístico es un pilar fundamental para la cooperativa, dada la gran cantidad de rutas y el volumen de encomiendas que manejan. Conectan Guaranda con las principales localidades de la provincia de Bolívar, además de destinos estratégicos como Riobamba, Chillanes, Caluma, San Luis, Santo Domingo, Guayaquil, Quito y Babahoyo; además, cuenta con servicios de encomiendas en la ciudad de Ambato. El manejo de un volumen aproximado de 250 a 300 encomiendas semanales que actualmente se gestiona de forma manual, ha generado la necesidad de una solución más eficiente y confiable. El registro y la entrega de información a los clientes son particularmente propensos a errores con el método actual.

Por ello, la automatización del proceso es la clave para asegurar un servicio logístico eficiente, transparente y sin fallas, que cumpla con las expectativas de los clientes y optimice las operaciones de la cooperativa.

## 1.3 Justificación

La creciente demanda de servicios de transporte de encomiendas requiere soluciones tecnológicas que brinden transparencia, rapidez y confiabilidad a sus clientes, lo que permitirá a la empresa ser competitiva y mantener un estándar de calidad en su servicio. En el caso de Flota Bolívar, la ausencia de un sistema automatizado para el seguimiento de encomiendas, limita la capacidad de respuesta ante consultas de los usuarios y genera inconvenientes en la gestión interna de los envíos.

Con la implementación de un sistema de tracking, la cooperativa de transporte podrá mejorar la experiencia del cliente al brindarle funcionalidades de consulta y mensajería

por WhatsApp incrementando su confianza y fidelidad, reducir errores operativos en el registro de encomiendas, actualización del estado de envió (entrega digitalizada, en ruta, recibido en oficina, entregado al cliente), registro de entrega de encomiendas y la optimización de la trazabilidad de los paquetes en la gestión de envíos. Esta iniciativa no solo representa un aporte tecnológico para la cooperativa de transporte, sino también una oportunidad de modernización que puede escalar a otras áreas del servicio.

Cabe destacar que este proyecto se encuadra dentro de la línea de Ingeniería de Software, Redes y Telecomunicaciones, específicamente en la sub- línea de Diseño e Implementación de Sistemas de Información.

## 1.4 Objetivos: General y Específicos

### 1.4.1 Objetivo General

Implementar un Sistema web para el seguimiento de encomiendas de la Cooperativa de Transporte Flota Bolívar de la ciudad de Guaranda, provincia Bolívar

### 1.4.2 Objetivos Específicos

- Analizar el proceso actual de gestión de encomiendas en la Cooperativa de Transporte Flota Bolívar para identificar sus requerimientos
- Diseñar una solución tecnológica que permita el seguimiento de las encomiendas, adaptada a las necesidades de la cooperativa.
- Codificar los módulos frontend y backend del sistema, garantizando la integración de la mensajería por WhatsApp.
- Realizar las pruebas necesarias al software que garantice su funcionamiento.

# CAPÍTULO II

# MARCO TEÓRICO

## 2.1 Antecedentes

Vásquez Vega, S. P. (2023). Aplicación web de trazabilidad para la gestión logística de encomiendas de la empresa “ServiQuito”. El actual proyecto presenta un sistema web desarrollado para la empresa “SERVIQUITO S.A.”, empresa dedicada a brindar el servicio de paquetería, la cual cuenta con varias sucursales y con diversas alianzas estratégicas que le permiten tener cobertura a nivel nacional. La finalidad de este sistema es mejorar la gestión de encomiendas dentro de la empresa, brindar a los clientes un mejor servicio y permitir a los directivos poder llevar un mejor control sobre las encomiendas; además, este sistema logró aportar una ventaja competitiva a la empresa debido a que con su uso proporciona al cliente un servicio de mayor calidad. A su vez, con la aplicación de distintas técnicas de recolección de datos como la observación directa y entrevistas aplicadas en la empresa, se pudo identificar oportunidades de mejora y desarrollar módulos que funcionen de una manera mucho más eficiente y ágil. El sistema fue desarrollado bajo la arquitectura MVC (Modelo Vista Controlador) utilizando el framework CodeIgniter basado en PHP. Al utilizar este framework se obtuvo como resultado un sistema muy completo, distribuido ordenadamente en distintos módulos, además cuenta con un esquema de seguridad robusto y brinda un gran beneficio debido a que en un futuro este sistema se podrá escalar o modificar sin mayor problema. Para poder organizar de mejor manera el desarrollo del proyecto, se utilizó la metodología ágil SCRUM la cual permitió dividir el proyecto en 4 fases llamadas sprints las cuales tuvieron una duración de 4 semanas, una vez finalizada cada uno de los sprints se presentaron avances, logrando así mantener totalmente involucrados a los directivos de la empresa.

León, M. A. G. (2018). Servicio de tracking online para empresas de encomiendas que no tienen portal de información. El proyecto presentado, tiene por objeto aportar eficiencia al modelo de negocio de las empresas de encomienda, generando una actitud positiva y un espíritu progresista, elevando el nivel de competitividad y reputación que actualmente tienen en el mercado, en esta industria. Mediante la investigación realizada, los dueños de estas empresas tienen una visión incompleta del modelo de negocio actual para cualquier empresa, esto quiere decir, tener una comunicación constante y online con el cliente, a través de plataformas o sitios web. Los dueños entrevistados se analizaron, mayormente son personas sin estudios académicos, por ende, la experiencia del trabajo

se enfoca más a las percepciones, que a los resultados que puede darnos el papel al hacer un análisis más detallado. La perspectiva ha sido mayormente la mejor arma, para defenderse ante la incertidumbre que todo negocio tiene que lidiar. Es precisamente ahí, donde el proyecto presentado quiere entrar, pues el prototipo del "Portal de Recepciones" busca entregar información relevante de manera sencilla a los ojos de los usuarios. Este proyecto viene a solucionar el déficit de información con el que cuentan los clientes al momento de esperar la recepción de su encomienda. No existe información fidedigna del despacho, cuando estos se atrasan por algún problema (Murphy). En la práctica, nadie sabe que pasó y el cliente de forma personal averigua porque su pedido se atrasó. La plataforma, busca acercar a todas las empresas de encomienda, con sus clientes que no tienen información del estado del envío de sus bultos. Esta cercanía, se basa en un portal que, de forma online, informe el estado de los envíos de sus clientes. Pudiendo revisar detalladamente fechas de envío, recepción, foto del Boucher y valor de encomienda. Toda esta información que pareciera ser básica y corriente es la oportunidad que este proyecto modeló.

Pizzol, P. G. D. (2022). Desarrolló un sistema de rastreamiento de equipamientos médico-hospitalarios. El presente proyecto desarrolla un sistema de información web para la gestión técnica y el mantenimiento preventivo y correctivo de equipos biomédicos, con el objetivo de optimizar su uso y reducir los tiempos de inactividad. Se identificaron los requisitos del sistema mediante encuestas y entrevistas al personal técnico, lo que permitió definir funcionalidades clave como gestión de usuarios y roles, planificación de mantenimientos, notificaciones automáticas y generación de reportes gráficos. El desarrollo se realizó bajo la metodología ágil Scrum, priorizando las funcionalidades esenciales mediante el método MoSCoW. Se utilizaron tecnologías modernas como PHP con CodeIgniter, MySQL, Bootstrap y JavaScript, garantizando escalabilidad y seguridad en la plataforma. Las pruebas del sistema incluyeron técnicas de caja blanca y caja negra, verificando su correcto funcionamiento y la integridad de los datos. Los resultados evidencian que el sistema mejora la organización y trazabilidad de los mantenimientos, proporcionando una solución eficiente y automatizada para la gestión de equipos biomédicos en entornos hospitalarios.

Gómez Valdelomar, D. (2012). Sistema de seguimiento de envíos, “Desarrollo de una aplicación para Windows Phone 7 que trabaje en modo online/offline, y envíe

información a un servidor". En la actualidad existen muchas empresas de mensajería con una gran carga de trabajo debido al gran aumento de las ventas por internet de todo tipo de artículos como ropa, calzados, aparatos electrónicos, videojuegos, libros, materiales de estudios de universidades online, etc.… Estas empresas necesitan saber en qué estado se encuentran sus envíos en cada momento, para que en caso de que exista alguna incidencia puedan solucionarlo al instante. De esta manera obtendrán una muy buena imagen de cara a sus clientes. Para conseguir que las empresas de mensajería puedan consultar rápidamente el estado en que se encuentran sus envíos se generará una solución que les permitirá realizar dicha consulta de una forma ágil y eficaz. Hoy en día ya existen dispositivos llamados Smartphone, a precios cada vez más asequibles, que nos permiten recibir y enviar información a través de las redes de comunicación, por tanto, con ellos podremos actualizar en tiempo real el estado de los envíos en el servidor de datos central de la empresa. De este modo, desde la oficina central de la empresa se podrá conocer inmediatamente el estado actual de cada uno de los envíos y solucionar cualquier problema que surgiese, lo cual contribuirá positivamente en la percepción que se hagan los clientes sobre el servicio ofrecido por la empresa. Por ello, obtener actualizados los estados de los envíos es el principal objetivo de la solución que se detalla en este documento.

Vivas Mena, J. B. (2019). Propuesta de implementación del sistema Web de venta de boletos de viaje y gestión de encomiendas para la empresa Transportes Montero SAC Piura; 2018. La presente tesis fue desarrollada bajo la línea de investigación en Tecnologías de la Información y Comunicación de la Escuela Profesional de Ingeniería de Sistemas de la Universidad Católica Los Ángeles de Chimbote, para la mejora continua de la calidad de los procesos en las organizaciones del Perú; cuyo objetivo general consistió en proponer la implementación de un sistema web de venta de boletos de viaje y gestión de encomiendas para contribuir al mejoramiento de procesos de la empresa TRANSPORTES MONTERO S.A.C. en el año 2018; teniendo un diseño de tipo cuantitativo, descriptivo, usando el diseño de investigación no experimental, de corte transversal; trabajando con una muestra delimitada por 14 trabajadores, obteniendo como resultados: En lo que corresponde a la dimensión: calidad de procesos y servicios, el 63% de los trabajadores encuestados afirma que la empresa Transportes Montero S.A.C. SÍ brinda calidad de procesos y servicios a los clientes. En lo que se refiere a la dimensión los sistemas web para la agilización de procesos, el 84% de los trabajadores encuestados

afirmó que los sistemas web SÍ agilizan los procesos. Y finalmente en la dimensión eficiencia, interacción y seguridad, se observó que el 81% de los trabajadores encuestados afirma que los sistemas web SÍ son eficientes. Estos resultados tienen similitud con lo planteado en la hipótesis, por lo que se concluye que la hipótesis queda aceptada.

Al revisar los aportes de Vásquez Vega (2023), León (2018), Pizzol (2022), Gómez Valdelomar (2012) y Vivas Mena (2019), se identifican coincidencias importantes: todos los proyectos buscan mejorar la trazabilidad de procesos logísticos, la transparencia hacia el cliente y la eficiencia en la gestión empresarial. Además, la mayoría recurre a metodologías ágiles como Scrum, destacando la importancia de presentar avances iterativos y de involucrar a los usuarios finales en el desarrollo.

En cuanto a diferencias, se observa que algunos trabajos utilizaron CodeIgniter (Vásquez Vega, 2023; Pizzol, 2022), otros enfocaron sus soluciones en plataformas móviles (Gómez Valdelomar, 2012) o en sistemas web especializados en ventas y encomiendas (Vivas Mena, 2019), mientras que León (2018) trabajó más en el aspecto de comunicación online entre empresa–cliente. Estas variaciones muestran que no existe un único modelo de implementación, sino múltiples enfoques que dependen del contexto y necesidades de cada organización.

A partir de esta revisión se identifican vacíos relevantes:

- Varios sistemas no fueron diseñados con un enfoque totalmente responsivo, lo que limita su uso en smartphones.
- En varios casos, la seguridad y escalabilidad no fueron descritas como un eje central, lo que podría limitar su crecimiento futuro.

Con base en ello, el proyecto para la Cooperativa de Transporte Flota Bolívar aporta un enfoque diferencial al:

- Desarrollar un sistema bajo Laravel, framework moderno y robusto que facilita la escalabilidad y la seguridad y frontend React, para para garantizar una interfaz de usuario altamente reactiva, fluida y optimizada para cualquier dispositivo.
- Integrar módulos de registro y seguimiento de encomiendas en un solo sistema, logrando mayor eficiencia organizacional.

Luego del análisis de los trabajos relacionados al tema se establece que es adecuado implementar un sistema web para el seguimiento de encomiendas de la Cooperativa de Transporte Flota Bolívar de la ciudad de Guaranda, provincia Bolívar, con el fin de agilizar los procesos de registro de encomienda y el estado de la misma, usando un diseño responsivo en el framework Laravel.

## 2.2 Científico

### Metodología de Desarrollo de Software

Maida y Paziela. (2015) indican que: "Una metodología de desarrollo de software es un marco de trabajo que se usa para estructurar, planificar y controlar el proceso de desarrollo de sistemas de información. Una gran variedad de estos marcos de trabajo ha evolucionado durante los años, cada uno con sus propias fortalezas y debilidades. Una metodología de desarrollo de sistemas no tiene que ser necesariamente adecuada para usarla en todos los proyectos. Cada una de las metodologías disponibles es más adecuada para tipos específicos de proyectos, basados en consideraciones técnicas, organizacionales, de proyecto y de equipo".

### Metodologías Ágiles

García, I. M. (2015) indican que: "La metodología ágil propone una forma de trabajo flexible cuya planificación se actualiza continuamente. Esto contrasta con las metodologías tradicionales, las cuales siguen una planificación precisa desde el principio. Las metodologías ágiles es una evolución de las metodologías iterativas, ya que ambas se basan en objetivos a corto plazo. Su objetivo es tener el software funcionando lo antes posible, ya que así el cliente tendrá primeras versiones donde podrá comprobar y aportar su idea de negocio."

### Beneficios de la metodología ágil

- Mayor flexibilidad: uno de los principales principios es la flexibilidad ante las solicitudes de cambio.
- Eficacia operativa: el aumento de la eficacia operativa, la reducción del tiempo y el aumento del control permiten que los proyectos se realicen con un riesgo minimizado y una mayor eficacia.
- Satisfacción de los usuarios finales: las empresas informan una mayor satisfacción de los usuarios finales lo que refleja el mayor valor final generado gracias a este enfoque (Digital, s.f.).

**Scrum**

Según Navarro et al. (2013, como se citó en Merchán-Narváez, N. J., Palma-Peralta, E. E., & Poma-Japón, D. X., 2024) es un marco de trabajo ágil para conseguir una cooperación efectiva en el equipo en diferentes proyectos, además utiliza reglas, artefactos y permite definir roles que crean la estructura indispensable para obtener la funcionalidad correcta. Scrum adopta un enfoque incremental basado en la teoría de gestión de procesos. Dicha teoría se basa en la transparencia, verificación y adaptación; la transparencia asegura la claridad durante el proceso de cosas que pueden influir en el resultado; la verificación permite descubrir desviaciones no deseadas del proceso; y la adaptación, efectúa arreglos apropiados para minimizar su impacto (p. 5057).

**Kanban**

Según Martins (2024, como se citó en Merchán-Narváez, N. J., Palma-Peralta, E. E., & Poma-Japón, D. X., 2024) es un método visual para la gestión de proyectos que se ejecuta mediante la utilización de tablero Kanban. Estos tableros permiten a los equipos observar de forma gráfica su flujo y carga de trabajo, además se encuentran organizados en columnas que indican distintas fases que conlleva el proceso laboral, tales como: "pendiente", "en progreso" y "completado". Las tareas específicas se representan mediante tarjetas que progresan a lo largo de las columnas hasta su finalización. Este enfoque se fundamenta en la transparencia y representación visual del proceso de trabajo, permitiendo a los equipos detectar cuellos de botella y mejorar continuamente el proceso (p. 5058).

**Xtreme Programing (XP)**

Según Penadés y Letelier (2006, como se citó en Merchán-Narváez, N. J., Palma-Peralta, E. E., & Poma-Japón, D. X., 2024) pone énfasis en la relevancia de perfeccionar las interacciones entre personas como un elemento fundamental para lograr el éxito a la hora de desarrollar software. XP promueve la colaboración, enseñanza y un entorno laboral positivo. Fundamenta su enfoque en la retroalimentación entre desarrolladores y cliente, la comunicación flexible entre los involucrados, la implementación de soluciones de manera sencilla y la disposición para realizar cambios. Es particularmente apropiado para proyectos con requerimientos indefinidos, altamente variables y con un riesgo técnico considerable (p. 5059).

**LEAN DEVELOPMENT**

Según Maida et al. (2015) el desarrollo Lean es una adaptación a los entornos de desarrollo de software, del método de producción que desarrolló Toyota, para equipos

pequeños de programadores. Se fundamenta principalmente en constituir un equipo fuerte y altamente preparado capaz de llevar a cabo cualquier tarea en poco tiempo, legando todo a la eficacia y la cohesión de los componentes del equipo y obviando los procesos y la burocracia que conlleva normalmente el tener un sistema de producción preestablecido (p. 98).

**Tabla 1.**

*Características principales de las Metodologías Ágiles.*

| Característica | Scrum | Kanban | Extreme Programming (XP) | Lean |
|---|---|---|---|---|
| **Enfoque Principal** | Marco de trabajo para la gestión de proyectos, basado en iteraciones (sprints) y roles definidos. | Método visual para gestionar el flujo de trabajo, centrado en la mejora continua y la eficiencia. | Metodología de ingeniería de software que promueve prácticas técnicas y la calidad del código. | Principio de gestión que se enfoca en eliminar el "desperdicio" (todo lo que no agrega valor) para maximizar la eficiencia. |
| **Ciclos** | Utiliza "Sprints" de duración fija (generalmente de 1 a 4 semanas). Al final de cada sprint se entrega un incremento de producto funcional. | No tiene ciclos de duración fija. El trabajo fluye de manera continua. | Ciclos de desarrollo cortos (típicamente de 1 a 3 semanas). | No tiene ciclos fijos, se centra en un flujo de valor continuo. |

| | | | | |
|---|---|---|---|---|
| **Roles** | **Product Owner:** Prioriza el trabajo.<br><br>**Scrum Master:** Facilita el proceso y elimina obstáculos.<br><br>**Equipo de Desarrollo:** Autogestionado y multidisciplinar. | No define roles específicos, pero sí promueve la colaboración y el liderazgo a todos los niveles. | No tiene roles fijos como Scrum, pero promueve una estrecha colaboración entre el cliente y el equipo. | No tiene roles prescritos, pero fomenta la mejora continua por parte de todo el equipo. |
| **Eventos** | **Daily Scrum:** Reunión diaria de 15 minutos. **Planificación de Sprint:** Al inicio. **Revisión de Sprint:** Al final. **Retrospectiva de Sprint:** Al final. | No tiene eventos formales. Las reuniones son "justo a tiempo" para resolver problemas. | **Daily Stand-up:** Reunión diaria. **Planificación de Iteración:** Al inicio.<br><br>**Revisión y Retrospectiva:** Al final de cada iteración. | No tiene eventos formales, pero fomenta una cultura de mejora continua y aprendizaje. |
| **Gestión del Trabajo** | Se utiliza un "Product Backlog" priorizado y un "Sprint Backlog" para las tareas del sprint actual. | Se utiliza un tablero visual (Kanban) con columnas para "Pendiente", "En Progreso" y "Terminado". Se limita el trabajo en curso (WIP). | Se priorizan las historias de usuario y se divide el trabajo en iteraciones cortas. | Se basa en el mapeo del flujo de valor para identificar y eliminar el desperdicio. |

| | | | | |
|---|---|---|---|---|
| **Ventajas** | ·Gran estructura y organización.<br><br>·Fomenta la colaboración del equipo.<br><br>·Permite una inspección y adaptación constante.<br><br>·Adecuado para proyectos con requisitos cambiantes. | ·Gran flexibilidad y adaptabilidad.<br><br>·Visualización clara del flujo de trabajo.<br><br>·Reduce el tiempo de entrega y aumenta la eficiencia.<br><br>·Ideal para equipos de mantenimiento o soporte. | ·Alta calidad del código gracias a sus prácticas técnicas.<br><br>·Fomenta la comunicación y la retroalimentación constante.<br><br>·Permite la adaptación continua a los cambios. | ·Maximiza la eficiencia y la entrega de valor.<br><br>·Reduce costos y tiempos de producción.<br><br>·Se puede aplicar a cualquier proceso, no solo software. |
| **¿Cuándo usarla?** | ·Proyectos complejos e inciertos.<br><br>·Cuando se necesita un marco de trabajo claro para equipos autogestionados.<br><br>·Proyectos donde la adaptación y la retroalimentación son cruciales. | · Ideal cuando el trabajo es imprevisto y requiere cambios de prioridad en tiempo real.<br><br>. Es ideal si ya se tiene un flujo de trabajo y solo quieres optimizarlo gradualmente mediante la visualización. | ·Proyectos de desarrollo de software donde la calidad y la ingeniería son prioritarias.<br><br>·Equipos pequeños y altamente colaborativos.<br><br>·Cuando el cliente está dispuesto a participar activamente en el proceso. | ·Proyectos o procesos donde se busca maximizar la eficiencia y eliminar pasos innecesarios.<br><br>·Proyectos con un flujo de trabajo bien definido, pero con oportunidades de mejora. |

*Nota.* Esta tabla muestra las características principales de las metodologías de desarrollo ágiles de software más utilizadas.

La elección de Extreme Programming (XP) y Kanban como metodologías ágiles para el desarrollo del sistema web de seguimiento de encomiendas de la Cooperativa de Transporte Flota Bolívar se justifica por la naturaleza del proyecto y las necesidades tanto de la organización como de los futuros usuarios.

En primer lugar, XP (Extreme Programming) es adecuado porque los requerimientos del sistema pueden cambiar en el transcurso del desarrollo, a medida que se ajusten los procesos internos de la cooperativa. El enfoque de XP en la comunicación constante con el cliente y en la retroalimentación continua asegura que el software se adapte de manera flexible a estos cambios. Además, las prácticas técnicas propias de XP como las pruebas unitarias, la integración continua y la simplicidad en el diseño garantizan que el sistema sea robusto, confiable y de calidad, lo cual es fundamental para una aplicación crítica como la gestión y el seguimiento de encomiendas.

Por otra parte, Kanban aporta valor al proyecto mediante una gestión visual del flujo de trabajo. La representación de las tareas en un tablero Kanban con estados como, "entrega digitalizada, en ruta, recibido en oficina, entregado al cliente" permitirá al equipo de desarrollo y a la cooperativa observar claramente el avance del proyecto y detectar posibles cuellos de botella en tiempo real. Además, al no requerir ciclos rígidos, Kanban se adapta perfectamente a un contexto en el que surgen tareas de manera continua, como ajustes de funcionalidad, mejoras o solicitudes de soporte que puedan presentarse durante y después del desarrollo. La combinación de XP y Kanban permite equilibrar la calidad técnica del software con la eficiencia en la gestión del trabajo. Mientras XP asegura que el sistema web se desarrolle con prácticas sólidas de ingeniería y centrado en las necesidades del cliente, Kanban facilita la organización, transparencia y mejora continua del proceso. En conjunto, estas metodologías brindan un marco flexible y colaborativo, ideal para un proyecto que requiere adaptarse a los cambios, garantizar la trazabilidad de las encomiendas y ofrecer un servicio confiable a los usuarios de la cooperativa.

## 2.3 Conceptual

### Hardware

El hardware es la parte física y tangible del sistema, como los componentes electrónicos que cumplen funciones como la alimentación eléctrica del sistema (Pérez, 2021).

### Software

El software es la parte intangible y digital del sistema, que se ocupa del área conceptual o representacional, dentro de un entorno simulado que ayuda a la interacción con el usuario (Pérez, 2021).

**Tracking**

El tracking hace referencia a un sistema que gracias a las nuevas tecnologías es posible la localización de los productos en todo momento o por etapas (Gaete, 2018, p.22).

**Sistema tecnológico**

Un sistema tecnológico es aquel que ayuda a eliminar los cuellos de botella, actividades críticas, retrasos en las entregas o la falta de documentos de soporte, debido que estas actividades son el reclamo frecuente en el seguimiento de encomiendas (Gaete, 2018, p.37).

**Sistemas operativos**

Los sistemas operativos son un conjunto de programas que actúa como intermediarios que nos permiten manejar recursos del hardware como la memoria, discos, medios de almacenamientos de información y diversos periféricos de la computadora (Pérez, 2021).

**Líneas de comando**

Las líneas de comando es una interfaz de usuario de computadora, que permite a los usuarios dar instrucciones a algún programa informático por medio de líneas de texto en base a su sintaxis (Pérez, 2021).

**Gestor de base de datos**

El gestor de base de datos es una interfaz que interactúa con el usuario y que permite almacenar, organizar, recuperar, comunicar y administrar información de un modo especial en la computadora (Beekman, 2005).

**Tabla 2.**

*Sistemas Gestores de Base de Datos*

| Sistema | Descripción |
|---|---|
| **MongoDB** | Es un SGBD multiplataforma no relacional y usa BSON con esquema dinámico. Es un sistema NoSQL orientado a documentos y es de código abierto escrito en lenguaje C + + con licencia GNU AGPL 3.0. Este sistema gestor permite realizar todo tipo de consultas y operaciones de indexación muy eficaces, ofrecen un buen soporte a la replicación y el balanceo de carga y, además, tiene una alta eficiencia en la exploración de los recursos (Postigo, 2023, p. 9). |

| | |
|---|---|
| **MySQL y MariaDB** | Es un SGBD relacional, multihilo, multiusuario y multiplataforma. Está diseñado para su exploración en entornos web, y también funciona bien en base de datos de tamaño medio (Postigo, 2023, p. 9). |
| **SQL Server de Microsoft** | Es de arquitectura principalmente cliente/servidor, relacional y es el mayor competidor de Oracle. La instalación típica incluye herramientas relacionadas con la inteligencia artificial empresarial. Ofrece mecanismos muy cómodos de respaldo y recuperación, comprensión, replicación de alta disponibilidad y recuperación ante desastres (Postigo, 2023, p. 9 - 10). |
| **PostgreSQL** | Es un SGBD multiplataforma que permite la gestión de objetos, por lo que se convierte en el mejor sistema relacional-objetos. Es el SGBD libre más avanzado del mundo (Postigo, 2023, p. 10). |

*Nota.* Principales gestores de base de datos, y sus conceptualizaciones.

**Lenguaje SQL**

El lenguaje SQL (Structured Query Language) es una herramienta para organizar, gestionar y recuperar datos almacenados en una base de datos relacional, por tanto, permite la comunicación con el sistema de gestión de la base de datos. (Martín, M. J. R., Rodríguez, F. M., & Martin, A. R., 2006).

**Lenguajes de programación**

Los lenguajes de programación son conjunto de herramientas que permite darle instrucciones a la computadora, actuando como intermediario entre el desarrollador y la máquina, a la vez nos permiten desarrollar desde aplicaciones prácticas hasta sistemas complejos en diversas áreas tecnológicas (Beekman, 2005).

**Tabla 3.**

*Lenguajes de programación utilizados para el desarrollo de aplicativos webs.*

| Lenguaje de programación | Descripción |
|---|---|
| **HTML** | Es un lenguaje de marcado de los contenidos de un sitio web, es decir, para designar la función de cada elemento dentro de la página: titulares, párrafos, listas, tablas, etc. Es el esqueleto de la web y la base de todo (Mestres, 2018, p. 6). |

| | |
|---|---|
| **CSS** | Es un lenguaje de hojas de estilo creado para controlar la presentación de páginas definiendo colores, tamaños, tipos de letra, posiciones, espaciados, etc (Mestres, 2018, p. 6). |
| **PHP** | Este lenguaje gratuito sirve de gran ayuda para aquellos programadores que empiezan a manejarse en la creación de páginas web y aplicaciones web 12 dinámicas, gracias a su facilidad de comprensión y compatibilidad con el lenguaje estático HTML. Además, es un lenguaje que está muy vinculado a la creación y mantenimiento de bases de datos y, por supuesto, sirve de soporte a WordPress, el CMS más usado en la actualidad por los desarrolladores web para crear sus proyectos o gestionar las webs de sus clientes. (Rodríguez, 2020, como se citó en Macario, E., & Macario, N. I. E., 2023). |
| **Javascript** | En esta lista, JavaScript es el único lenguaje que sirve tanto al lado del servidor como al cliente. Al igual que HTML y CSS, JavaScript desempeña un papel fundamental en la programación de front-end interactivo y fácil de usar de la mayoría de los sitios web, independientemente de las escalas y la complejidad. Por lo tanto, en muchas encuestas de Stack Overflow, GitHub y Hired, siempre sigue siendo su posición de liderazgo en los principales lenguajes de codificación de uso común. (Designveloper, 2020, como se citó en Macario, E., & Macario, N. I. E., 2023). |

*Nota.* Lista de lenguajes de programación, que se implantara en el proyecto tecnológico.

**Trazabilidad**

La trazabilidad son los procedimientos definidos que nos permite conocer el historial, la ubicación y la trayectoria de un producto o lote de productos a lo largo de una cadena de suministros en un momento dado, con el apoyo de herramientas predefinidas (Cepeda, 2010).

**Backend**

Backend es aquello que se encuentra del lado del servidor y se encarga de interactuar con bases de datos, verificar maniobras de sesiones de usuarios, montar la página en un servidor y servir todas las vistas creadas por el desarrollador frontend. En este caso el número de tecnologías es mucho menos limitado, puesto que la programación backend puede alcanzar lenguajes como PHP, Python, .NET, Java, etc., y las bases de datos sobre las que se trabaja pueden ser SQL, MongoDB, MySQL, entre otras (Mestres, 2018, p. 5).

**Laravel**

Morales. (2023) como se citó en Treviño, C. E., Moreno, M. M., Gaffare, M. P. G., Ramírez, R. E. M., & Jiménez, J. R. (2023), indica que Laravel:

> Es un framework de desarrollo web de código abierto escrito en PHP que se utiliza para crear aplicaciones web de alta calidad y escalables, siendo uno de los frameworks de PHP más populares. Se basa en el patrón de diseño MVC (Model-View-Controller) y utiliza una sintaxis intuitiva que facilita la lectura y el mantenimiento del código. Además de contar con una arquitectura modular que permite a los desarrolladores reutilizar componentes de código y crear aplicaciones escalables y fáciles de mantener (p. 3220).

**Patrón Observer**

Según Debrauwer (2013), menciona que "el patrón Observer tiene como objetivo construir una dependencia entre un sujeto y los observadores de modo que cada modificación del sujeto sea notificada a los observadores para que puedan actualizar su estado" (p. 144).

**Frontend**

El frontend son aquellas tecnologías de desarrollo web del lado del cliente, es decir, las que corren en el navegador del usuario y que son básicamente tres: HTML, CSS y JavaScript. El frontend se enfoca en el usuario, en todo con lo que puede interactuar y lo que ve mientras navega. Una buena experiencia de usuario, inmersión y usabilidad son algunos de los objetivos que busca un buen desarrollador frontend, y hoy en día hay una gran variedad de frameworks, preprocesadores y librerías que ayudan en esta tarea (Mestres, 2018, p. 5).

**React**

React, también conocida como React.js o ReactJS es una biblioteca JavaScript de código abierto diseñada para crear interfaces de usuario con el objetivo de facilitar el desarrollo de aplicaciones en una sola página. Es mantenido por Facebook y la comunidad de software libre. En el proyecto hay más de mil desarrolladores libres (Medrano, 2022, p. 16).

**Relación entre patrón de diseño gof, framework y marco de trabajo.**

En el desarrollo del Sistema web de seguimiento de encomiendas para la Cooperativa de Transporte Flota Bolívar, se utiliza el framework Laravel para el backend y React para el frontend, estableciendo una Arquitectura N-Capas sobre una comunicación API REST. Esta estructura facilita la separación estricta de responsabilidades, un principio fundamental para la programación basada en XP (Extreme Programming).

A diferencia de una arquitectura monolítica tradicional, el uso de N-Capas permite dividir la lógica del sistema en: Capa de presentación Frontend React, encargada de la interfaz intuitiva y modular. Capa de Servicios Backend Laravel: Donde reside la lógica de negocio y se orquesta el comportamiento del sistema. Capa de Datos: Encargada de la persistencia de la información.

En este contexto, el Patrón Observer se implementa estratégicamente dentro de la Capa de Servicios de Laravel para garantizar la trazabilidad y las notificaciones automáticas. Cuando se ejecuta la lógica de cambio de estado de una encomienda (Sujeto), el servicio dispara eventos que notifican a los suscriptores (Observadores), como el módulo de Bot de WhatsApp, sin contaminar la lógica transaccional.

El uso de Kanban con su tablero visual complementa esta arquitectura asegurando un flujo de trabajo constante, mientras que las prácticas de XP garantizan que esta separación en capas se traduzca en un código escalable, testeable y de alta calidad.

## 2.4 Legal

### Ley de Propiedad Intelectual

Según la ley orgánica de propiedad Intelectual en su Sección II refiere, Objetivo del derecho de autor:

> La protección del derecho de autor recae sobre todas las obras del ingenio, en el ámbito literario o artístico, cualquiera que sea su género, forma de expresión, mérito o finalidad. Los derechos reconocidos por el presente Título son independientes de la propiedad del objeto material en el cual está incorporada la obra y su goce o ejercicio no están supeditados al requisito del registro o al cumplimiento de cualquier otra formalidad (Asamblea Nacional del Ecuador, 2006, Art. 8).

Entre las diferentes obras protegidas por este artículo se encuentran los programas de ordenador.

Sección V.

De los programas de ordenador

> Los programas de ordenador se consideran obras literarias y se protegen como tales. Dicha protección se otorga independientemente de que hayan sido incorporados en un ordenador y cualquiera sea la forma en que estén expresados, ya sea en forma legible por el hombre (código fuente) o en forma legible por máquina (código objeto), ya sean programas operativos y programas aplicativos, incluyendo diagramas de flujo, planos, manuales de uso, y en general, aquellos

elementos que conformen la estructura, secuencia y organización del programa (Asamblea Nacional del Ecuador, 2006, Art. 28).

El artículo 29 de la Ley de Propiedad intelectual manifiesta que:

> Es titular de un programa de ordenador, el productor, esto es la persona natural o jurídica que toma la iniciativa y responsabilidad de la realización de la obra. Se considerará titular, salvo prueba en contrario, a la persona cuyo nombre conste en la obra o sus copias de la forma usual. Dicho titular está además legitimado para ejercer en nombre propio los derechos morales sobre la obra, incluyendo la facultad para decidir sobre su divulgación. El productor tendrá el derecho exclusivo de realizar, autorizar o prohibir la realización de modificaciones o versiones sucesivas del programa, y de programas derivados del mismo. Las disposiciones del presente artículo podrán ser modificadas mediante acuerdo entre los autores y el productor (Asamblea Nacional del Ecuador, 2006, Art. 29).

**Ley Orgánica de Protección de Datos Personales**

El Artículo 42 de la Ley Orgánica de Protección de Datos Personales establece la obligatoriedad de una evaluación de impacto bajo ciertas condiciones:

> Evaluación del impacto del tratamiento de datos personales. -El responsable realizará una evaluación de impacto del tratamiento de datos personales cuando se haya identificado la probabilidad de que dicho tratamiento, por su naturaleza, contexto o fines, conlleve un alto riesgo para los derechos y libertades del titular o cuando la Autoridad de Protección de Datos Personales lo requiera.
>
> La evaluación de impacto relativa a la protección de los datos será de carácter obligatoria en caso de:
>
> a) Evaluación sistemática y exhaustiva de aspectos personales de personas físicas que se base en un tratamiento automatizado, como la elaboración de perfiles, y sobre cuya base se tomen decisiones que produzcan efectos jurídicos para las personas naturales;
>
> b) Tratamiento a gran escala de las categorías especiales de datos, o de los datos personales relativos a condenas e infracciones penales, o
>
> c) Observación sistemática a gran escala de una zona de acceso público.
>
> La Autoridad de Protección de Datos Personales establecerá otros tipos de operaciones de tratamiento que requieran una evaluación de impacto relativa a la protección de datos.

La evaluación de impacto deberá efectuarse previo al inicio del tratamiento de datos personales (Asamblea Nacional del Ecuador, 2021, Art. 42).

El Artículo 43 de la Ley Orgánica de Protección de Datos Personales establece los plazos para la notificación de vulneraciones de seguridad:

> Art. 43.-Notificación de vulneración de seguridad.-El responsable del tratamiento deberá notificar la vulneración de la seguridad de datos personales a la Autoridad de Protección de Datos Personales y la Agencia de Regulación y Control de las Telecomunicaciones, tan pronto sea posible, y a más tardar en el término de cinco (5) días después de que haya tenido constancia de ella, a menos que sea improbable que dicha violación de la seguridad constituya un riesgo para los derechos y las libertades de las personas físicas. Si la notificación a la Autoridad de Protección de Datos no tiene lugar en el término de cinco (5) días, deberá ir acompañada de indicación de los motivos de la dilación.
>
> El encargado del tratamiento deberá notificar al responsable cualquier vulneración de la seguridad de datos personales tan pronto sea posible, y a más tardar dentro del término de dos (2) días contados a partir de la fecha en la que tenga conocimiento de ella (Asamblea Nacional del Ecuador, 2021, Art. 43).

**Código INGENIOS**

> **Artículo 1.-** Objeto. - El presente Código tiene por objeto normar el Sistema Nacional de Ciencia, Tecnología, Innovación y Saberes Ancestrales previsto en la Constitución de la República del Ecuador y su articulación principalmente con el Sistema Nacional de Educación, el Sistema de Educación Superior y el Sistema Nacional de Cultura, con la finalidad de establecer un marco legal en el que se estructure la economía social de los conocimientos, la creatividad y la innovación. (Asamblea Nacional del Ecuador, 2016, Art. 1)
>
> **Artículo 2.-** Ámbito. - Se rigen por el presente Código todas las personas naturales, jurídicas y demás formas asociativas que desarrollen actividades relacionadas a la economía social de los conocimientos, la creatividad y la innovación.
>
> Las actividades relacionadas a la economía social de los conocimientos, la creatividad y la innovación son aquellas enfocadas a la creación de valor a partir del uso intensivo de la generación, transmisión, gestión y aprovechamiento del bien de interés público conocimiento, que incluye los conocimientos tradicionales; promoviendo en todos los sectores sociales y productivos la

colaboración y potenciación de las capacidades individuales y sociales, la democratización, distribución equitativa, y aprovechamiento eficiente de los recursos en armonía con la naturaleza, dirigido a la obtención del buen vivir (Asamblea Nacional del Ecuador, 2016, Art. 2).

**Artículo 3.-** Fines. - El presente Código tiene, como principales, los siguientes fines:

1. Generar instrumentos para promover un modelo económico que democratice la producción, transmisión y apropiación del conocimiento como bien de interés público, garantizando así la acumulación y redistribución de la riqueza de modo justo, sostenible y en armonía con la naturaleza;
2. Promover el desarrollo de la ciencia, la tecnología, la innovación y la creatividad para satisfacer necesidades y efectivizar el ejercicio de derechos de las personas, de los pueblos y de la naturaleza;
3. Incentivar la producción del conocimiento de una manera democrática colaborativa y solidaria;
4. Incentivar la circulación y transferencia nacional y regional de los conocimientos y tecnologías disponibles, a través de la conformación de redes de innovación social, de investigación, académicas y en general, para acrecentarlos desde la práctica de la complementariedad y solidaridad;
5. Generar una visión pluralista e inclusiva en el aprovechamiento de los conocimientos, dándole supremacía al valor de uso sobre el valor de cambio;
6. Desarrollar las formas de propiedad de los conocimientos compatibles con el buen vivir, siendo estas: pública, privada, comunitaria, estatal, asociativa y mixta;
7. Incentivar la desagregación y transferencia tecnológica a través de mecanismos que permitan la generación de investigación, desarrollo de tecnología e innovación con un alto grado de componente nacional;
8. Promover la distribución justa y equitativa de los beneficios derivados de las actividades vinculadas a la generación, transmisión, gestión, uso y aprovechamiento de los conocimientos, la tecnología, la innovación y los conocimientos tradicionales, así

como el uso eficiente de los factores sociales de la producción para incrementar el acervo de conocimiento e innovación;

9. Establecer las fuentes de financiamiento y los incentivos para el desarrollo de las actividades de la economía social de los conocimientos, la creatividad y la innovación;

10. Fomentar el desarrollo de la sociedad del conocimiento y de la información como principio fundamental para el aumento de productividad en los factores de producción y actividades laborales intensivas en conocimiento; y,

11. Fomentar la protección de la biodiversidad como patrimonio del Estado, a través de las reglas que garanticen su aprovechamiento soberano y sustentable, proteger y precautelar los derechos de las comunidades, pueblos y nacionalidades sobre sus conocimientos tradicionales y saberes ancestrales relacionados a la biodiversidad; y evitar la apropiación indebida de la biodiversidad y los conocimientos tradicionales asociados a esta (Asamblea Nacional del Ecuador, 2016, Art. 3).

**Código interno Flota Bolívar**

**EL PRESIDENTE DE LA REPÚBLICA**

**CONSIDERANDO:**

Que se ha enviado al Ministerio de Previsión Social la documentación correspondiente al Estatuto de la Cooperativa de Transportes FLOTA BOLÍVAR, domiciliada en la ciudad de Guaranda, Provincia de Bolívar, aprobada mediante Acuerdo Ministerial N°2522 de 14 de Octubre de 1964 e inscrita en el Registro General de Cooperativas con el n° orden 1221 de 16 de Octubre del mismo año, documentación que contiene todos los datos actualizados de acuerdo con las disposiciones constantes en la Ley de Cooperativas y su Reglamento General.

Que de conformidad con el Art. 2° del Acuerdo Ministerial N° 8460 de 10 de marzo de 1967 las cooperativas tienen la obligación de reencuadrar su estatuto y reglamentos internos a la Ley de Cooperativas y a su Reglamento General, para que puedan seguir haciendo uso de todos sus derechos.

**Acuerda:**

**Art. 1°** Dar por fenecida la existencia legal de la Cooperativa introduciendo las siguientes reformas a su estatuto:

1°. - En el Art. 31, cámbiese: "Tres vocales" por "cinco vocales". En el mismo artículo, tercer inciso, cámbiese "Tres vocales suplentes" por "cinco vocales suplentes".

2°. - En el Art. 55, suprímase lo siguiente: "y la bonificación a los empleados de la Cooperativa, de acuerdo al Código del Trabajo".

3°. - En el Art. 56, en su parte final añádase lo siguiente: "El 15% por concepto de bonificación a los empleados de la Cooperativa de acuerdo al Código del Trabajo".

4°. - En el Art. 59, suprímase "de las dos terceras partes de los asistentes" y en su lugar póngase: "de las dos terceras partes de la totalidad de los socios de la Cooperativa".

**Art. 2°** La Dirección Nacional de Cooperativas de conformidad a sus facultades legales califique a las personas que a continuación se mencionan en calidad de socios activos: [Sigue una lista de nombres ilegibles].

**Art. 3°** Los socios que posteriormente ingresen a la Cooperativa para hacer uso de sus derechos y obligaciones como tales, previamente deberán ser calificados por la Dirección Nacional de Cooperativas, caso contrario las personas que no cumplan con esta disposición serán excluidas de la Cooperativa.

**EL PRESIDENTE DE LA JUNTA MILITAR DE GOBIERNO**

**CONSIDERANDO:**

Que se han enviado al Ministerio de Previsión Social los Estatutos de la Cooperativa de Transportes "FLOTA BOLÍVAR", domiciliada en la Ciudad de Guaranda, Provincia de Bolívar.

En uso de las atribuciones que le concede el Art. 5° de la Ley de Cooperativas,

**ACUERDA:**

**Art. 1°** Aprobar los Estatutos de la Cooperativa de Transportes "FLOTA BOLÍVAR", con las siguientes modificaciones:

**1a.** En el literal c) del Art. 16°, hágase constar solamente "Por fallecimiento" y suprímase el resto del artículo.

**2a.** Al Art. 20° añádase lo siguiente: "además de los excedentes sociales que le correspondan hasta el momento de su retiro".

**3a.** Al Art. 20° añádase el siguiente inciso: "La liquidación de los haberes del socio que se retire por cualquier concepto se efectuará al momento de realizarse los balances semestrales".

**4a.** Luego del Art. 21° hágase constar los siguientes artículos: "Art... Para la expedición de una resolución que disponga la expulsión de un socio

deberá levantarse previamente una información sumaria en la cual podrá el acusado ejercer su derecho de defensa.".

**5a.** Al Art. 23° añádase lo siguiente: "descompuesto en 84 certificados de aportación de un valor de $100 cada uno y el capital suscrito es el de $25.500".

**6a.** En el Art. 29° en lugar de: "un mil quinientos sucres" dígase "mil sucres".

**7a.** El literal d) del Art. 31° dirá: El saldo se repartirá entre los socios de acuerdo al monto de operaciones y al trabajo realizado con la Cooperativa.

**8a.** En el Art. 47° en lugar de "Presidente" dígase: "El Gerente" y en lugar de "la mitad más uno de los socios" dígase "la tercera parte de los socios".

**Art. 4°** La Cooperativa no podrá apartarse de las actividades específicas para las cuales se constituye ni operar en otra clase que no sea la de transportes, pudiendo en caso de contravenir a esta disposición caer en las sanciones legales previstas en la Ley y Reglamento General de Cooperativas en que está contemplado inclusive la disolución de la Cooperativa.

**Art. 5°** Inscríbase definitivamente en el Libro respectivo a la antedicha Cooperativa.

El análisis del marco legal nos ayuda a establecer límites y derechos del proyecto y los artículos que lo encaminan son los siguientes:

- **La Ley de Propiedad Intelectual:** La ley de Propiedad Intelectual garantiza que el código fuente y la estructura del Sistema web para el seguimiento de encomiendas serán protegidos confiriendo el derecho exclusivo de realizar, autorizar o prohibir modificaciones futuras.
- **La Ley Orgánica de Protección de Datos Personales:** La Ley Orgánica de Protección de Datos Personales es fundamental, dado que el sistema gestionará información sensible de los clientes (nombres, direcciones, números de contacto, etc.). Esta ley exige que el sistema se diseñe con principios de privacidad por defecto y por diseño.
- **El Código INGENIOS:** El Código INGENIOS enmarca este proyecto dentro de la economía social de los conocimientos y la creatividad. El desarrollo de una solución tecnológica específica para optimizar los procesos logísticos en la provincia de Bolívar contribuye directamente a los fines de este código,

promoviendo la innovación, la gestión eficiente del conocimiento y la modernización de un servicio público esencial en la región.

Después de haber realizado un análisis del marco legal, de acuerdo con el reglamento interno de la Cooperativa de Transporte Flota Bolívar no existe ningún impedimento legal para la implementación de un sistema de seguimiento de encomiendas, por lo que se ha determinado que es factible su ejecución. El marco legal no representa un obstáculo, sino una guía indispensable para construir un sistema que no solo sea funcional, sino también ético, seguro, legalmente protegido y alineado con la visión de desarrollo tecnológico del Ecuador.

**2.5 Georreferencial**

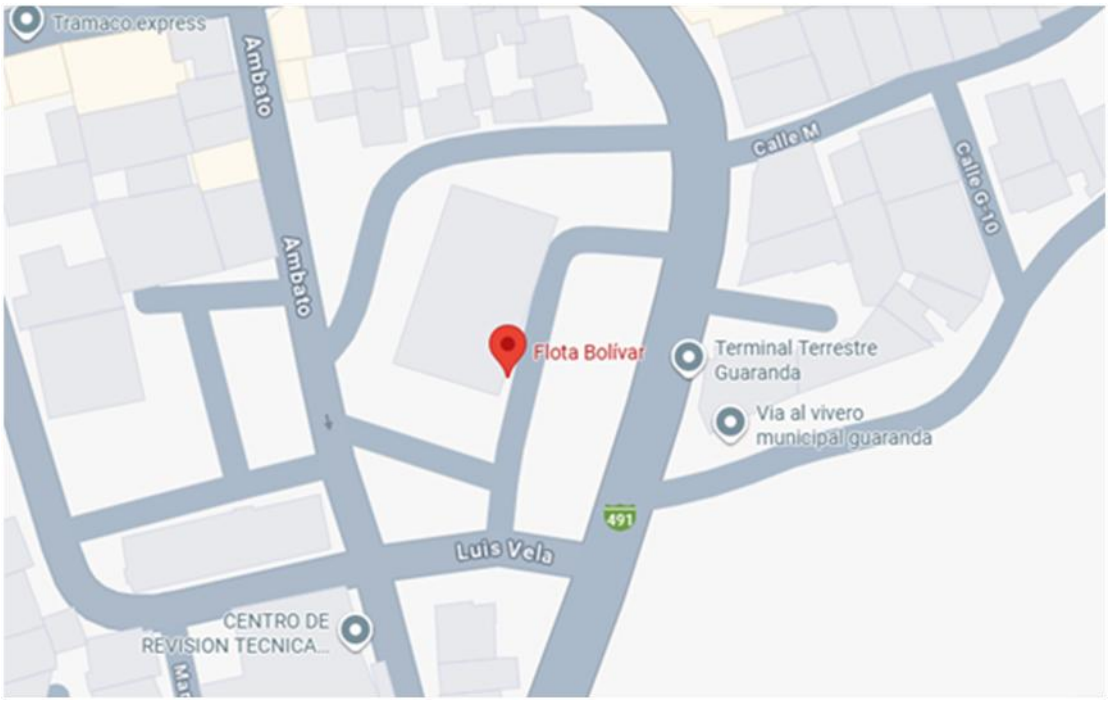
**Cantón:** Guaranda

**Provincia:** Bolívar

**Dirección:** Av. Elisa Mariño de Carvajal frente al Terminal Terrestre, Guaranda, Ecuador.

**Figura 1.**

*Mapa Geográfico de la ubicación de la Cooperativa de Transporte "Flota Bolívar" del cantón Guaranda.*




| | |
|---|---|
| **Latitud:** 1º35'29.094"S | **Longitud:** 78º59'37.489"W |

# CAPÍTULO III

# METODOLOGÍA

## 3.1 Metodología de Desarrollo de Software

Para desarrollar el sistema web de seguimiento de encomiendas de la Cooperativa de Transporte Flota Bolívar, se decidió trabajar con la metodología XP (Extreme Programming), ya que se adapta muy bien a proyectos donde los requerimientos pueden cambiar con frecuencia y se busca mantener una alta calidad en el código. Además, se complementará con la metodología Kanban, que facilita la gestión visual del trabajo, ayudando a organizar tareas, detectar bloqueos y mantener un flujo de desarrollo constante. En este proyecto no se realizarán encuestas, por lo que la obtención de información se basará en una entrevista, la revisión de documentación existente y la observación del proceso operativo dentro de la Cooperativa. A partir de ello, se identificarán las principales necesidades, como el registro de encomiendas, la generación de guías, la consulta de estados, el control de entregas y el seguimiento de paquetes. El uso combinado de XP y Kanban permitirá desarrollar el sistema de forma ágil, colaborativa y con entregas frecuentes, garantizando que el producto final sea funcional, escalable y fácil de mantener. Fases a utilizar en el desarrollo:

- **Planificación:**

En este punto se define el alcance del proyecto. Como equipo de desarrollo conjunto con el cliente definiremos las historias de los usuarios, que representan los requerimientos funcionales desde la perspectiva del cliente de la Cooperativa de Transporte Flota Bolívar. Durante el proceso de planeación se elabora un reléase de plan, que organiza las funcionalidades a desarrollar a interacciones cortas, con la toma de decisiones en qué funcionalidades implementar en cada una. Teniendo en cuenta de que el sistema es vital para la competitividad de la Cooperativa, se priorizan funciones críticas como el registro y consultas de encomiendas. Dejando las mejoras visuales o complementarias para fases posteriores.

El equipo de desarrollo conjunto con la estrecha colaboración del personal de la Cooperativa de Transporte Flota Bolívar, se determinará y asegura la alineación del proyecto con la metodología, evaluaremos puntos importantes como:

- Historias del usuario: se realizan las historias de usuario para asegurar que son simples o directas. Lo más importante es confirmar la disponibilidad del personal de la cooperativa como cliente activo, quien participará en la definición, refinamiento y priorización de estas historias.

- Valor comercial y prioridad: en esta etapa, se priorizan las funcionalidades que aportan mayor valor, como el registro y la consulta de encomiendas, dejando las mejoras visuales o funcionales secundarias para futuras interacciones.

La creación de cronogramas de lanzamiento es una guía flexible para la entrega continua y de valor. El objetivo es lanzar tan pronto como sea posible para obtener el feedback del cliente.

- División en Interacciones y Planes: Se divide el proyecto en iteraciones cortas y gestionables, de una a dos semanas de duración. Se crea un plan específico para cada una, definiendo las funcionalidades a desarrollar en cada ciclo.

- **Diseño:**

Durante la fase de diseño, se busca definir la estructura técnica y funcional del sistema web de seguimiento de encomiendas, asegurando simplicidad, escalabilidad y facilidad de mantenimiento, principios fundamentales de la metodología XP. En esta etapa, el diseño evoluciona conforme avanza el desarrollo, permitiendo incorporar cambios y mejoras basadas en la retroalimentación del cliente.

**1.- Patrón Arquitectónico** Para el desarrollo del sistema se adoptará el patrón arquitectónico en N-Capas, el cual separa las responsabilidades en diferentes niveles para mantener una estructura modular y organizada. Este patrón contempla tres capas principales:

- Capa de presentación, que gestiona la interacción entre el usuario y el sistema mediante una interfaz web intuitiva y accesible.
- Capa de lógica de negocio, donde se implementan los procesos y reglas que controlan el registro, consulta y seguimiento de las encomiendas.
- Capa de datos, encargada de la conexión con la base de datos centralizada que almacena la información de los usuarios, encomiendas, entregas y movimientos. Esta separación permite un mantenimiento más sencillo, una mayor reutilización del código y una mejor escalabilidad del sistema, alineándose con las buenas prácticas del desarrollo ágil.

**2.- Vista Arquitectónica** Para representar la estructura del sistema desde diferentes perspectivas, se utilizará el modelo de vistas arquitectónicas 4+1, el cual facilita la comprensión global del sistema y su interacción con los actores involucrados. Este modelo incluye:

- Vista Lógica: muestra la funcionalidad del sistema desde la perspectiva del usuario final. En esta vista se emplearán diagramas de clases y secuencia que describen la interacción entre los objetos y las operaciones principales del sistema.
- Vista de Procesos: se enfoca en el comportamiento dinámico y la concurrencia del sistema. A través de diagramas de actividad se representarán los flujos de trabajo relacionados con el seguimiento de encomiendas, priorizando el rendimiento y la escalabilidad.
- Vista Física: representa la topología del sistema en términos de infraestructura. Se utilizará un diagrama de despliegue para mostrar la distribución del sistema entre los servidores, la base de datos y los dispositivos de acceso del usuario.
- Vista de Despliegue: describe cómo los componentes del software se gestionan e interconectan dentro del entorno de desarrollo. Aquí se aplicarán diagramas de paquetes para definir la organización del código y sus dependencias.
- Vista de Escenarios (+1): integra las vistas anteriores a través de diagramas de casos de uso, mostrando cómo los actores como gerente, oficinistas y clientes interactúan con el sistema para realizar las funciones principales, como registrar, rastrear y entregar encomiendas.

- **Codificación:**

Durante la fase de codificación, se implementan los requerimientos definidos en las historias de usuario, asegurando calidad, colaboración y adaptación continua. Esta etapa se desarrollará mediante iteraciones, siguiendo los principios de Extreme Programming (XP) en conjunto con la metodología Kanban, permitiendo mantener un flujo de trabajo visual, constante y flexible.

Las principales prácticas XP que se aplicarán son:

- Programación en parejas: Los desarrolladores colaborarán directamente en el mismo entorno de trabajo, revisando y mejorando el código en tiempo real. Mientras un integrante codifica, el otro valida la lógica, detecta errores y aporta sugerencias. Por ejemplo, en el módulo de seguimiento de encomiendas, un programador podrá desarrollar la lógica de actualización del estado de la encomienda mientras el otro valida la correcta integración con la base de datos. Esta práctica fomenta la calidad del código y el intercambio de conocimiento dentro del equipo.

- Desarrollo guiado por pruebas: Antes de escribir el código funcional, se crearán pruebas automatizadas que inicialmente fallarán. Luego, se escribirá el código mínimo necesario para que estas pruebas pasen con éxito. Este enfoque garantiza que cada componente del sistema cumpla con los criterios funcionales esperados y evita errores en etapas posteriores.
- Integración continua: Cada vez que se implementen nuevas funcionalidades, estas serán integradas al sistema y se ejecutarán pruebas automáticas para detectar posibles fallos o conflictos entre módulos. Esta práctica asegura que el sistema mantenga su estabilidad y compatibilidad durante todo el desarrollo.
- Refactorización: El código será mejorado constantemente sin alterar su comportamiento externo, permitiendo mantener un diseño limpio, legible y escalable. Esta práctica garantiza que el sistema pueda adaptarse fácilmente a futuras modificaciones solicitadas por la cooperativa.

Cada iteración incluirá estas prácticas, abordando los módulos definidos en la planificación:

Iteración 1: Módulo de gerente, oficinista y cliente.

Iteración 2: Registros de encomiendas y actualización de estados.

Iteración 3: Modulo de generación de reportes de roles.

Iteración 4: Generación de código de seguimiento y consulta.

Iteración 5: Integración de mapa de seguimiento.

- **Pruebas**

Las pruebas no solo van a ser implementadas al final del sistema, sino que, por cada actividad constante desarrolladas, en este sistema web implementaremos estos tipos de pruebas:

- Pruebas unitarias: realizaremos pruebas automáticas que validen funciones específicas del sistema, con el enfoque a verificar su correcto funcionamiento.
- Pruebas de integración: verificaremos que cada módulo del sistema funcione conjuntamente al objetivo deseado.
- Pruebas de aceptación: realizada en conjunto con el cliente, se validará que las funcionalidades cumplen con las expectativas y las funcionalidades impuestos en la fase de implementación en base a los requisitos y lógica del negocio.

- Pruebas de regresión: se ejecutan todas las pruebas de aceptación y unitarias para asegurar que las funcionalidades no han desencadenado las ya existentes. Con el objetivo de que el sistema funcione correctamente.

Al trabajar con Kanban se podrá identificar de manera inmediata si una funcionalidad queda estancada en la fase de pruebas, permitiendo al equipo reaccionar de forma ágil para recibir bloqueos y mantener el flujo de trabajo constante.

- **Entrega**

Una vez finalizadas las pruebas de aceptación, como equipo estaremos listos para la entrega del incremento del software al cliente. Esto significa que el personal de la Cooperativa de Transportes Flota Bolívar podrá utilizar el sistema en una versión preliminar y dar su feedback si es necesario.

En esta fase, la metodología Kanban cobra relevancia, ya que el tablero visual permitirá mostrar de manera clara qué funcionalidades están listas para ser entregadas, cuáles requieren ajustes y cuáles se encuentran planificadas para próximas iteraciones. De esta forma, el cliente tiene una visión transparente del avance, puede dar retroalimentación inmediata sobre lo que ya está disponible y se facilita la toma de decisiones sobre las siguientes prioridades.

### 3.2 Técnicas e Instrumentos de Recopilación de Datos

Se recopiló información mediante entrevistas al personal seleccionado y observación de procesos. Estas técnicas permitieron obtener información organizada y oportuna.

**Entrevista.**

La aplicación de la entrevista a los encargados de la recepción, el registro y la entrega de encomiendas dentro de la Cooperativa de Transporte Flota Bolívar nos permitió recopilar información específica para la obtención de los requerimientos del sistema web. Anexo 4.

**Observación.**

La observación nos permitió conocer cómo se realizan los procesos relacionados con la gestión del envío de las encomiendas, incluyendo los horarios de atención en las distintas oficinas, los protocolos de atención al cliente, los horarios que tienen los buses y conocer las distintas rutas que cubren los buses. Anexo 5.

# CAPÍTULO IV

# INGENIERÍA DEL PROYECTO

## 4.1 Planificación

### Especificación de requerimientos software

A continuación, se redacta las historias de usuario con el propósito de recopilar información para el desarrollo del sistema.

**Tabla 4.**

*Historia de usuario 1*

| **Historia de usuario** | | | |
|---|---|---|---|
| **Numero:** 1 | | **Usuario:** | Gerente |
| **Nombre de historia:** | Registro del oficinista | | |
| **Prioridad de negocio:** | Alta | **Riesgo en desarrollo:** | Alta |
| **Precondiciones:** | | | |
| **Programador responsable:** | Leydi Galeas y Ronny Cabezas | | |
| **Descripción:** | El sistema debe permitir al gerente registrar a un oficinista encargado. Dicho proceso incluirá la captura de sus datos esenciales: nombre, apellido, correo electrónico y la asignación de una contraseña para el acceso al sistema. | | |
| **Criterio de validación:** | El sistema debe ser probado previamente y comprobar su funcionamiento.<br>El gerente debe tener credenciales de acceso válidas para realizar el registro inicial de los oficinistas. | | |

**Tabla 5.**

*Historia de usuario 2*

| **Historia de usuario** | | | |
|---|---|---|---|
| **Numero:** 2 | | **Usuario:** | Gerente y oficinista |
| **Nombre de historia:** | Autenticación | | |
| **Prioridad de negocio:** | Alta | **Riesgo en desarrollo:** | Alta |
| **Precondiciones:** | | | |
| **Programador responsable:** | Leydi Galeas y Ronny Cabezas | | |

| | |
|---|---|
| **Descripción:** | El sistema debe contar con un proceso seguro y eficiente de inicio de sesión que permita al gerente y a los oficinistas acceder a sus respectivas cuentas. |
| **Criterio de validación:** | El sistema debe validar las credenciales ingresadas y permitir el acceso solo a usuarios autorizados.<br>El sistema debe proporcionar mensajes claros de error en caso de credenciales incorrectas o cuentas bloqueadas. |

**Tabla 6.**

*Historia de usuario 3*

| **Historia de usuario** | | | |
|---|---|---|---|
| **Numero:** 3 | | **Usuario:** Gerente | |
| **Nombre de historia:** | Gestión de oficinas | | |
| **Prioridad de negocio:** | Alta | **Riesgo en desarrollo:** | Media |
| **Precondiciones:** | | | |
| **Programador responsable:** | Leydi Galeas y Ronny Cabezas | | |
| **Descripción:** | El sistema debe contar con un módulo que permita el registro, editar y actualizar las oficinas establecidas. | | |
| **Criterio de validación:** | El sistema muestra lista actualizada de oficinas.<br>Solo el gerente puede modificar o eliminar registros.<br>Los cambios se reflejan de inmediato en las consultas de los oficinistas. | | |

**Tabla 7.**

*Historia de usuario 4*

| **Historia de usuario** | | | |
|---|---|---|---|
| **Numero:** 4 | | **Usuario:** Gerente | |
| **Nombre de historia:** | Gestión de choferes | | |
| **Prioridad de negocio:** | Alta | **Riesgo en desarrollo:** | Media |
| **Precondiciones:** | | | |
| **Programador responsable:** | Leydi Galeas y Ronny Cabezas | | |

| | |
|---|---|
| **Descripción:** | El sistema debe contar con un módulo que permita el registro, editar, eliminar y actualizar los choferes. |
| **Criterio de validación:** | El sistema valida que el número de licencia sea único.<br>El sistema impide el registro de choferes que no tengan la licencia.<br>La lista de choferes se actualiza automáticamente tras cada operación. |

**Tabla 8.**

*Historia de usuario 5*

| **Historia de usuario** | | | |
|---|---|---|---|
| **Numero:** 5 | | **Usuario:** Gerente | |
| **Nombre de historia:** | Gestión de rutas | | |
| **Prioridad de negocio:** | Alta | **Riesgo en desarrollo:** | Media |
| **Precondiciones:** | | | |
| **Programador responsable:** | Leydi Galeas y Ronny Cabezas | | |
| **Descripción:** | El sistema debe contar con un módulo que permita el registro, editar, eliminar y actualizar las rutas. | | |
| **Criterio de validación:** | Las rutas se muestran correctamente con sus datos asociados.<br>Solo el gerente puede crear o modificar rutas.<br>Las rutas eliminadas no deben estar asociadas a encomiendas activas. | | |

**Tabla 9.**

*Historia de usuario 6*

| **Historia de usuario** | | | |
|---|---|---|---|
| **Numero:** 6 | | **Usuario:** Gerente | |
| **Nombre de historia:** | Gestión de horarios | | |
| **Prioridad de negocio:** | Alta | **Riesgo en desarrollo:** | Media |
| **Precondiciones:** | | | |
| **Programador responsable:** | Leydi Galeas y Ronny Cabezas | | |

| | |
|---|---|
| **Descripción:** | El sistema debe contar con un módulo que permita el registro, editar, eliminar y actualizar los horarios. |
| **Criterio de validación:** | Los horarios se registran correctamente con validación de fechas y horas.<br>No se permiten solapamientos de horario para el mismo bus o chofer. |

**Tabla 10.**

*Historia de usuario 7*

| **Historia de usuario** | | | |
|---|---|---|---|
| **Numero:** 7 | | **Usuario:** Gerente | |
| **Nombre de historia:** | Gestión de buses | | |
| **Prioridad de negocio:** | Alta | **Riesgo en desarrollo:** | Media |
| **Precondiciones:** | | | |
| **Programador responsable:** | Leydi Galeas y Ronny Cabezas | | |
| **Descripción:** | El sistema debe contar con un módulo que permita el registro, editar, eliminar y actualizar las unidades. | | |
| **Criterio de validación:** | El sistema valida que la placa del bus sea única.<br>El gerente puede modificar el estado (activo, en mantenimiento).<br>No se puede eliminar un bus con rutas activas. | | |

**Tabla 11.**

*Historia de usuario 8*

| **Historia de usuario** | | | |
|---|---|---|---|
| **Numero:** 8 | | **Usuario:** Oficinista | |
| **Nombre de historia:** | Registrar encomienda | | |
| **Prioridad de negocio:** | Alta | **Riesgo en desarrollo:** | Alta |
| **Precondiciones:** | | | |
| **Programador responsable:** | Leydi Galeas y Ronny Cabezas | | |
| **Descripción:** | El sistema debe permitir a los oficinistas encargados registrar nuevas encomiendas, capturando todos los | | |

| | |
|---|---|
| | datos necesarios: remitente, destinatario, peso, dimensiones, costo de envío, origen y destino. |
| **Criterio de validación:** | El sistema genera un número único de guía automáticamente.<br>Se valida que todos los datos obligatorios estén completos.<br>Se confirma visualmente que la encomienda fue registrada con éxito. |

**Tabla 12.**

*Historia de usuario 9*

| **Historia de usuario** | | | |
|---|---|---|---|
| **Numero:** 9 | | **Usuario:** | Gerente |
| **Nombre de historia:** | Visualizar reportes | | |
| **Prioridad de negocio:** | Alta | **Riesgo en desarrollo:** | Alta |
| **Precondiciones:** | | | |
| **Programador responsable:** | Leydi Galeas y Ronny Cabezas | | |
| **Descripción:** | El sistema permitirá al gerente generar reportes detallados de encomiendas, con la capacidad de filtrar por un oficinista específico dentro de una sucursal o emitir un reporte consolidado a nivel general. | | |
| **Criterio de validación:** | El gerente podrá visualizar campos como los nombres del remitente y destinatario, el estado de la encomienda registrada y su ruta correspondiente. | | |

**Tabla 13.**

*Historia de usuario 10*

| **Historia de usuario** | | | |
|---|---|---|---|
| **Numero:** 10 | | **Usuario:** | Oficinista y cliente |
| **Nombre de historia:** | Monitoreo de encomienda | | |
| **Prioridad de negocio:** | Alta | **Riesgo en desarrollo:** | Alta |
| **Precondiciones:** | | | |

| | |
|---|---|
| **Programador responsable:** | Leydi Galeas y Ronny Cabezas |
| **Descripción:** | El sistema permitirá al cliente y oficinista rastrear el estado de su encomienda.<br>El cliente podrá consultar en línea al ingresar el código se seguimiento generado al ser registrado la encomienda. |
| **Criterio de validación:** | El sistema muestra estado actual, oficina y fecha de actualización.<br>Se informa si el código no existe o está inactivo.<br>La información se actualiza automáticamente al cambiar el estado de la encomienda.<br>El sistema debe proporcionar mensaje claro de error en caso de que el código ingresado es incorrecto. |

**Tabla 14.**

*Historia de usuario 11*

| **Historia de usuario** | | | |
|---|---|---|---|
| **Numero:** 11 | | **Usuario:** Oficinista | |
| **Nombre de historia:** | Modificar estado de encomienda | | |
| **Prioridad de negocio:** | Alta | **Riesgo en desarrollo:** | Alta |
| **Precondiciones:** | | | |
| **Programador responsable:** | Leydi Galeas y Ronny Cabezas | | |
| **Descripción:** | El sistema permitirá al oficinista cambiar el estado de una encomienda (Recibido en oficina, En ruta, Recibido en oficina, Entregado al cliente) para mantener actualizado el flujo logístico. Al realizar el cambio, el sistema debe registrar la oficina actual y la fecha/hora del movimiento de forma automática. | | |
| **Criterio de validación:** | El sistema muestra estado actual, oficina y fecha de actualización.<br>El sistema debe presentar una lista desplegable con los estados permitidos según el flujo lógico. | | |

| | |
|---|---|
| | El sistema no debe permitir saltos de estado imposibles<br>El sistema debe validar que el oficinista solo pueda marcar una encomienda como "Recibida en oficina" si su cuenta está asociada a la oficina de destino correspondiente. |

**Tabla 15.**

*Historia de usuario 12*

| **Historia de usuario** | | | |
|---|---|---|---|
| **Numero:** 12 | | **Usuario:** | Oficinista |
| **Nombre de historia:** | Reporte de encomienda | | |
| **Prioridad de negocio:** | Alta | **Riesgo en desarrollo:** | Alta |
| **Precondiciones:** | | | |
| **Programador responsable:** | Leydi Galeas y Ronny Cabezas | | |
| **Descripción:** | El sistema debe generar reportes detallados sobre encomiendas enviadas, entregadas, filtradas por fechas y rutas establecidas. | | |
| **Criterio de validación:** | Los reportes se generan solamente en formatos PDF.<br>Los datos son precisos y actualizados en base al registro realizado de la encomienda.<br>Solo el oficinista puede generar reportes consolidados. | | |

**Tabla 16.**

*Historia de usuario 13*

| **Historia de usuario** | | | |
|---|---|---|---|
| **Numero:** 13 | | **Usuario:** | Gerente y cliente |
| **Nombre de historia:** | Consulta de rutas y horarios | | |
| **Prioridad de negocio:** | Alta | **Riesgo en desarrollo:** | Media |
| **Precondiciones:** | | | |
| **Programador responsable:** | Leydi Galeas y Ronny Cabezas | | |

| | |
|---|---|
| **Descripción:** | El sistema debe permitir tanto al gerente como al cliente consultar la información actualizada de las rutas y horarios disponibles. El cliente podrá visualizar los horarios de salida, llegada, las oficinas de origen y destino, y el chofer asignado. El gerente podrá revisar y verificar la información completa para su control administrativo. |
| **Criterio de validación:** | El sistema debe mostrar los datos de rutas y horarios correctamente organizados. La información consultada debe coincidir con los registros almacenados por el gerente.<br>El cliente podrá filtrar rutas por origen, destino o fecha. Los datos deben actualizarse automáticamente al modificarse una ruta u horario por parte del gerente. |

**Tabla 17.**

*Historia de usuario 14*

| **Historia de usuario** | | | |
|---|---|---|---|
| **Numero:** 14 | | **Usuario:** Cliente | |
| **Nombre de historia:** Consultar oficinas | | | |
| **Prioridad de negocio:** Alta | | **Riesgo en desarrollo:** Media | |
| **Precondiciones:** | | | |
| **Programador responsable:** | Leydi Galeas y Ronny Cabezas | | |
| **Descripción:** | El sistema debe permitir al cliente consultar la información detallada de las oficinas registradas. Se mostrará el nombre de la oficina, dirección, encargado, número de contacto, correo y horario de atención. Esta funcionalidad facilita la ubicación de las oficinas para el envío o recepción de encomiendas. | | |
| **Criterio de validación:** | El sistema muestra correctamente la lista actualizada de oficinas registradas. | | |

| | |
|---|---|
| | La información visualizada debe coincidir con la base de datos.<br>Los usuarios podrán filtrar oficinas por ciudad o tipo de servicio.<br>En caso de no existir registros, el sistema mostrará un mensaje informativo. |

**Tabla 18.**

*Historia de usuario 15*

| **Historia de usuario** | | | |
|---|---|---|---|
| **Numero:** 15 | | **Usuario:** Gerente y oficinista | |
| **Nombre de historia:** Cerrar sesión | | | |
| **Prioridad de negocio:** Alta | | **Riesgo en desarrollo:** Alta | |
| **Precondiciones:** | | | |
| **Programador responsable:** | Leydi Galeas y Ronny Cabezas | | |
| **Descripción:** | Permite a los usuarios cerrar sesión de forma segura, eliminando los datos de acceso de la sesión activa. | | |
| **Criterio de validación:** | Al cerrar sesión se eliminan las credenciales temporales.<br>El sistema redirige al inicio de sesión.<br>No es posible acceder a funciones sin autenticación. | | |

**Tabla 19.**

*Registro de riesgos*

| **Registro de riesgos** | | | | | |
|---|---|---|---|---|---|
| **N°** | **Riesgos** | **Probabilidad (1-5)** | **Impacto (1-5)** | **Severidad** | **Plan de mitigación** |
| 1 | Retraso de implementación del sistema | 3 | 5 | 15 | Implementar herramientas para la gestión del despliegue en GitHub. |

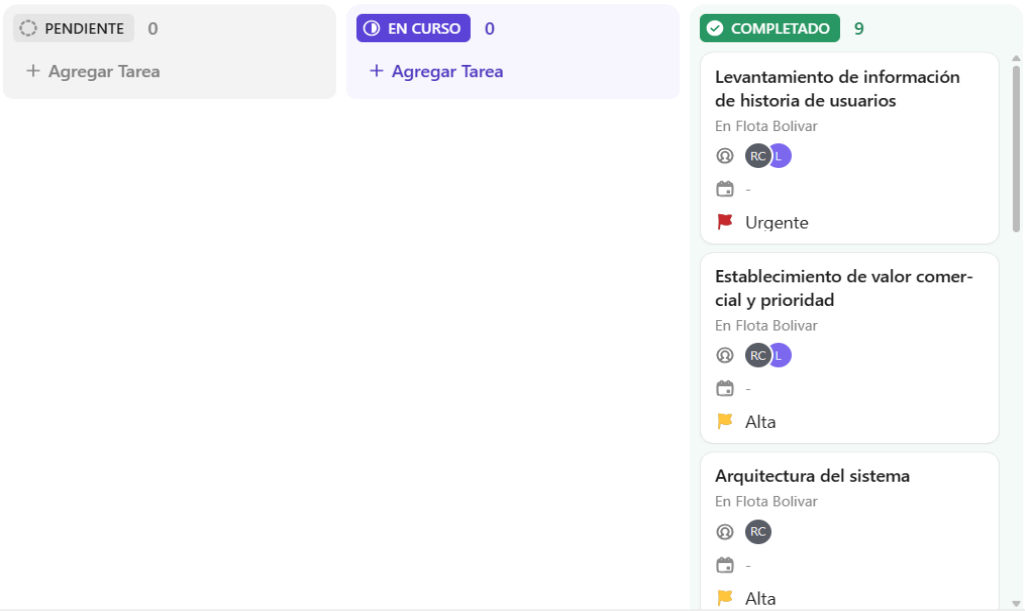
| | | | | | |
|---|---|---|---|---|---|
| 2 | Infraestructura deficiente | 2 | 5 | 10 | Garantizar que el entorno de ejecución sea el adecuado para el correcto funcionamiento. |
| 3 | Comunicación de componentes bajo determinadas circunstancias | 3 | 5 | 15 | Asegurar que todos los componentes sean escalables y en caso de fallos, sea corregible de manera modular. |

### 4.1.1 Gestión del trabajo mediante Kanban.

Para la gestión y seguimiento de las historias de usuario, se implementó un tablero Kanban utilizando la herramienta ClickUp. En este tablero se organizaron las tareas en columnas como “Pendiente”, “En curso” y “Completado”, permitiendo visualizar el avance del proyecto en tiempo real y detectar posibles bloqueos durante el desarrollo.

**Figura 2.**

*Implementación de metodología Kanban.*



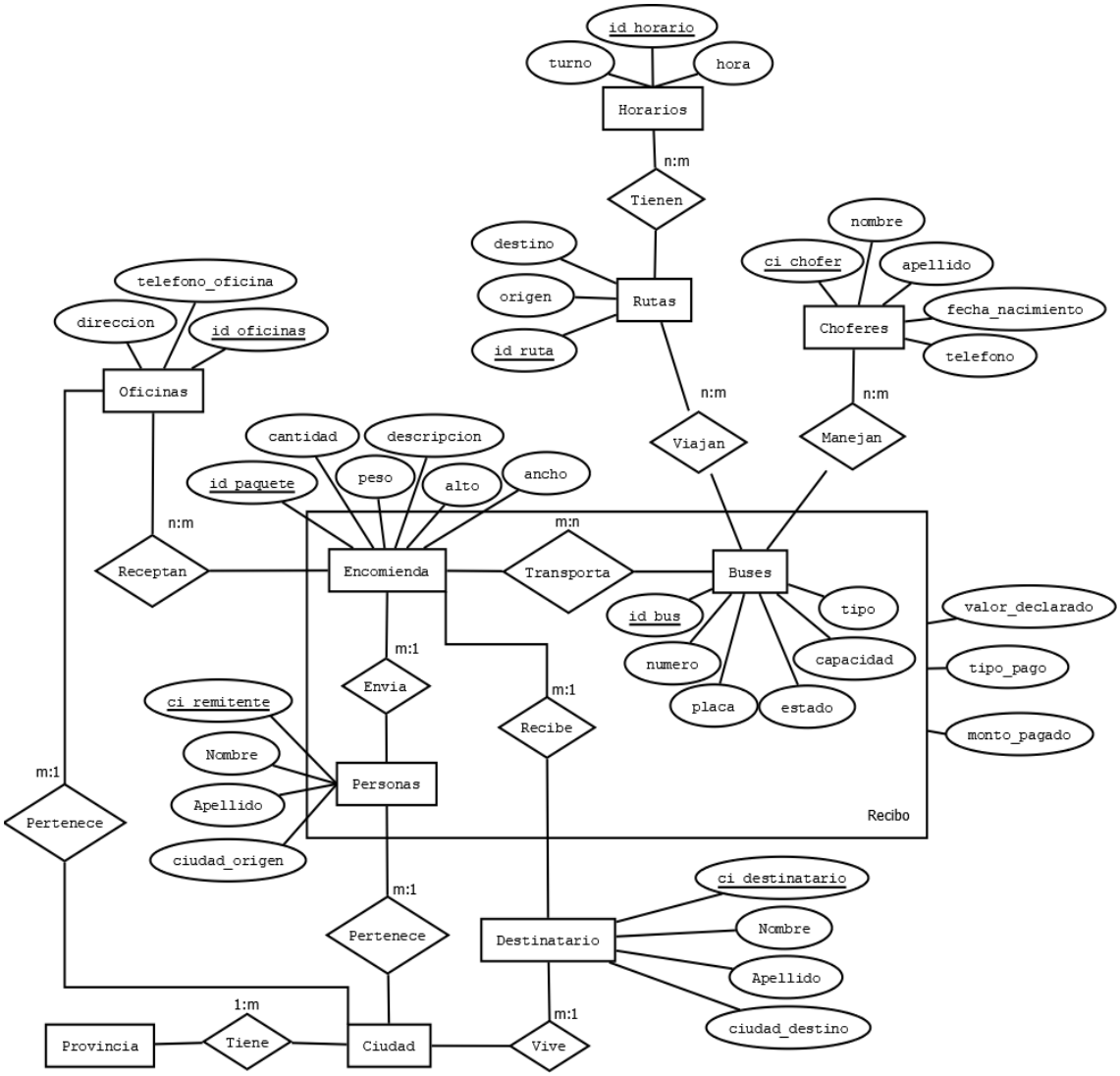

**Fuente:** Elaboración propia.

## 4.2 Diseño

### 4.2.1 Arquitectura del software

**Figura 3.**

*Diagrama de Entidad – Relación.*

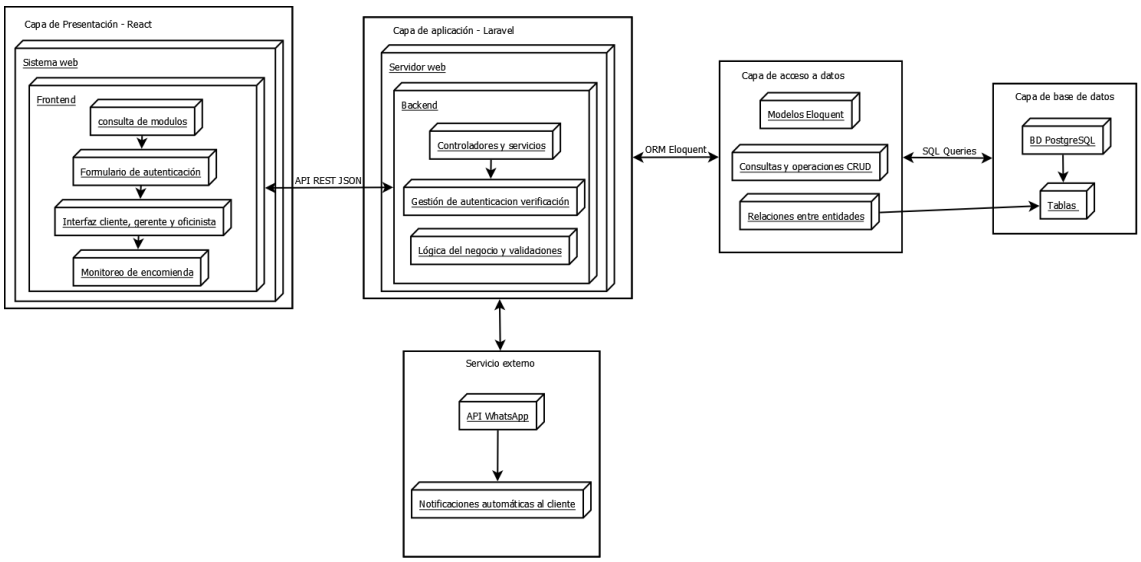



***Nota.*** Representación modular de las entidades y sus relaciones.

#### *4.2.1.1Patrón de diseño arquitectónico N-capas*

**Figura 4.**

*Diagrama de patrón arquitectónico de N-capas*




***Nota.*** Estructura del sistema en el patrón arquitectónico n-capas.

### 4.2.2 Vistas arquitectónicas 4 + 1

- **Vista de escenarios**

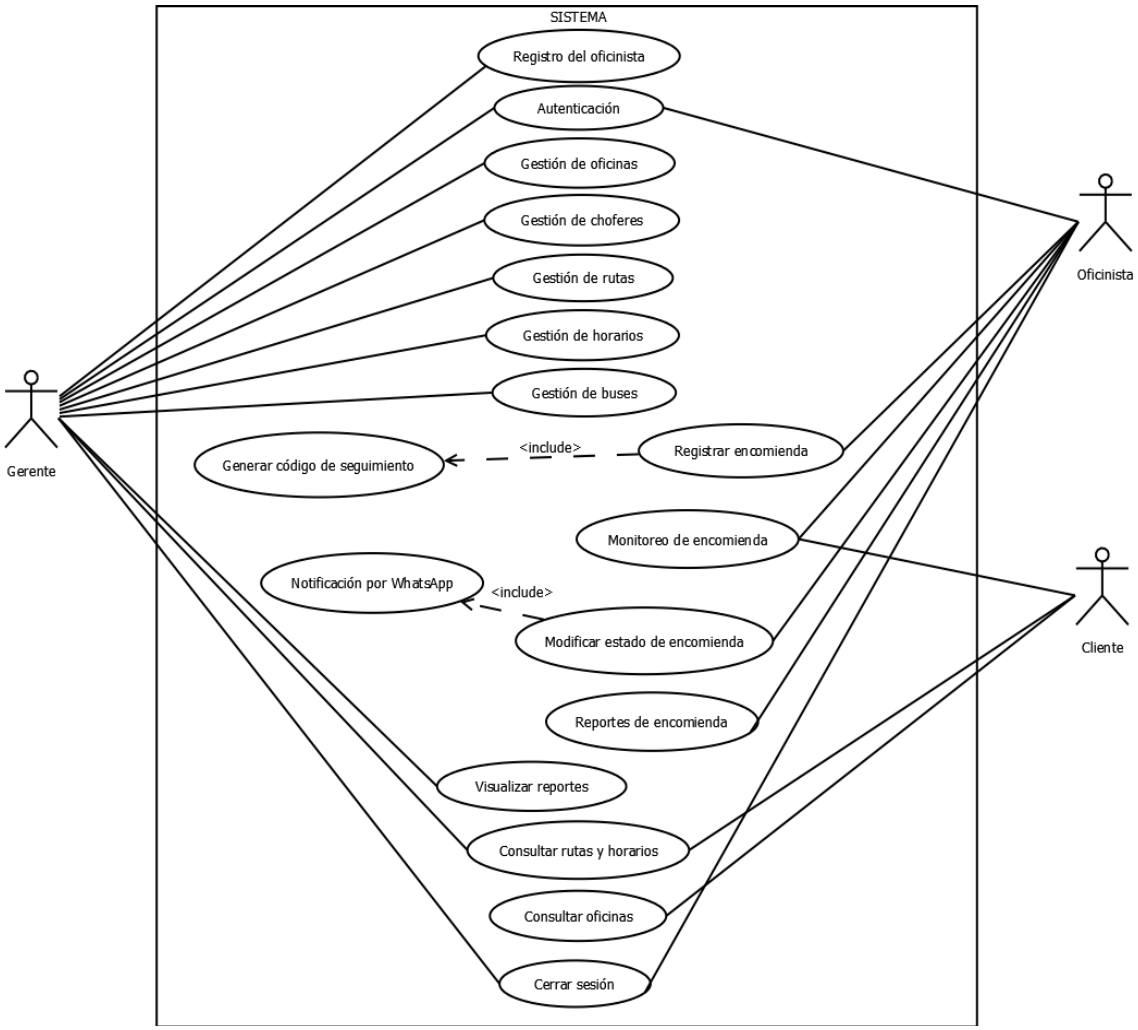
Esta vista presenta los actores clave y sus interacciones con el sistema. Los actores son:

- **Gerente autenticado:** Este usuario es el administrador del sistema, contando con recursos operativos (CRUD) en módulos de buses, oficinas, rutas y horarios. Además, es responsable de la administración de los oficinistas y choferes registrados y por registrar. A nivel estratégico, cuenta con la visualización de reportes de las encomiendas registradas y entregadas para el control y análisis de paquetes enviados.
- **Oficinista autenticado:** Es el autor que realiza las operaciones clave para la gestión de envíos. Teniendo accesibilidad en el registro de las encomiendas, acción que una vez realizada nos generara automáticamente un código de seguimiento. Dentro de sus roles también incluye la capacidad de monitorear la encomienda y la crítica función de modificar el estado de la misma. Es importante destacar que una vez realizado la modificación del estado de la encomienda a "recibido en oficina" en la oficina destino, se enviara automáticamente un mensaje a WhatsApp al cliente destinatario.

- **Usuario cliente:** Es el actor externo principal del sistema, que no requiere de autenticación. Su rol se centra en la obtención de información sobre los servicios como las consultas a rutas y horarios, consulta a oficinas y el seguimiento de los envíos. La funcionalidad más crítica es el mensaje enviado al WhatsApp del cliente destinatario, que es un mecanismo de comunicación pasivo que se activa automáticamente por el sistema cuando el oficinista realiza la modificación de la encomienda a "recibido en oficina". De esta manera, el cliente tendrá la facilidad de ser informado cuando llegue su encomienda al destino registrado.

**Figura 5.**

*Diagrama de caso de uso (escenarios)*




***Nota.*** Autores involucrados en el sistema.

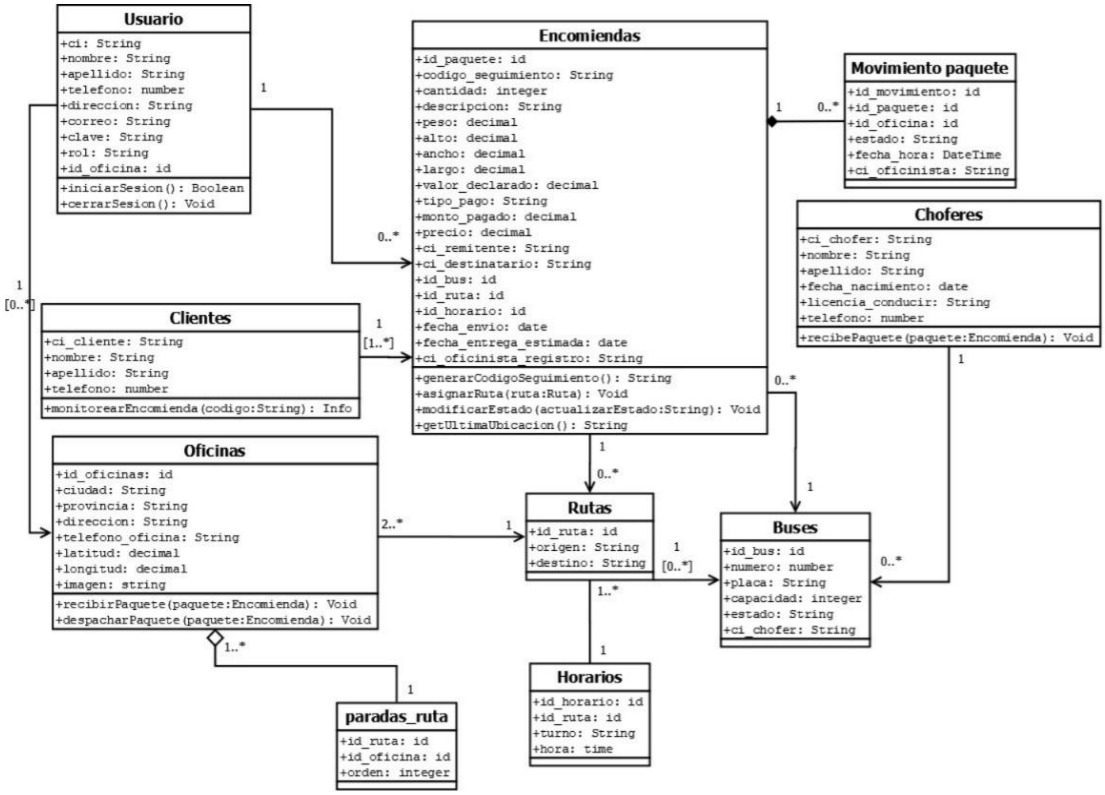
- **Vista de diseño lógico.**

Esta vista se centra en la estructura funcional y las interacciones del sistema. Se definirá el diagrama de clases donde se realizará la estructura estática de los datos, y el diagrama de secuencia que definirá el comportamiento dinámico de los objetos.

El diagrama de clases construido directamente sobre el modelo de la base de datos, definiendo componentes conceptuales del sistema y sus relaciones. Esta estructura organiza el sistema en clases fundamentales como la Gestión de personal y acceso de oficinistas y gerente, la logística y operación de buses, rutas y oficinas, y el seguimiento de encomienda. La clase encomiendas es la entidad central, vinculada a las especificaciones de envío y a la operación, mientras que la clase movimiento_paquete es clave en la trazabilidad, registrando cambios de estado y ubicación. Las relaciones entre estas clases garantizan la integridad y coherencia de los datos, elementos esenciales para el monitoreo de encomienda y la visualización de reportes.

**Figura 6.**

*Diagrama de clases (Vista lógica)*




***Nota.*** Estructura de diagrama de clases de la base de datos del sistema.

El comportamiento dinámico del sistema es representado con el diagrama de secuencia, ilustrándonos un flujo de control para los casos de usos operativos.

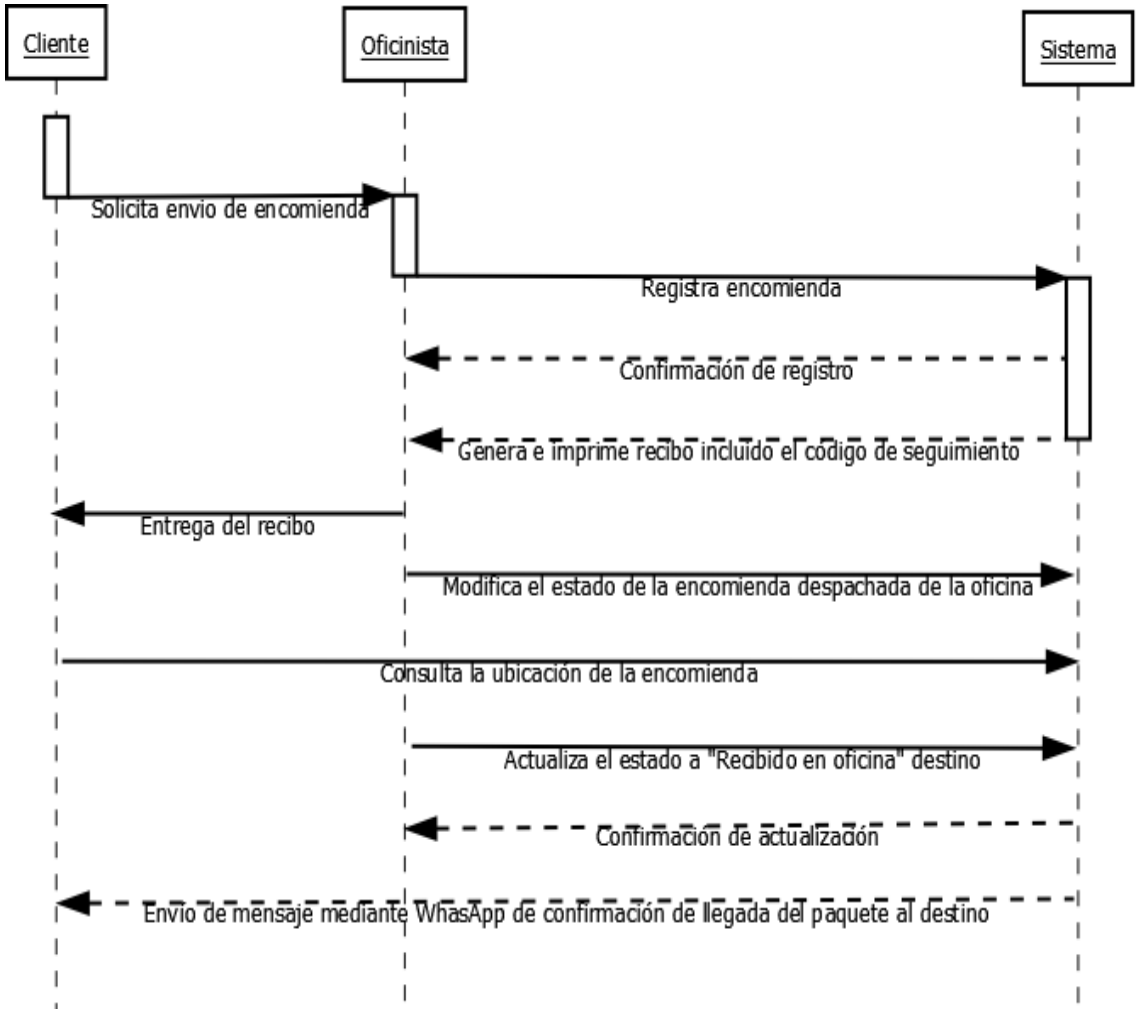
El proceso inicia cuando el cliente remitente realiza una solicitud de envió de encomienda. En respuesta, el oficinista toma la responsabilidad de registrar la solicitud al sistema. Este paso de registro es fundamental e implica el ingreso al sistema de los datos completos del cliente remitente y destinatario, una vez registrado la encomienda el sistema genera un código único que ayudara al monitoreo de la encomienda.

Una vez realizado el registro de la encomienda, el oficinista procede con la preparación logística. Esto implica una generación de reportes de las encomiendas que están listas para ser despachadas y deben ser enviadas a sus respectivos lugares de destino. Realiza la entrega de las encomiendas al chofer del bus para su envío respectivo.

Finalmente, el ciclo concluye con la entrega al destinatario. El cliente destinatario solicita la entrega de su encomienda, conjunto con su identidad que será procesado por el oficinista. Después de verificar la identidad. Se le realizará una toma a la cedula de identidad y el recibo firmado por parte del cliente destinatario, y será subido al sistema como evidencia de entrega, y por último la entrega física del paquete.

**Figura 7.**

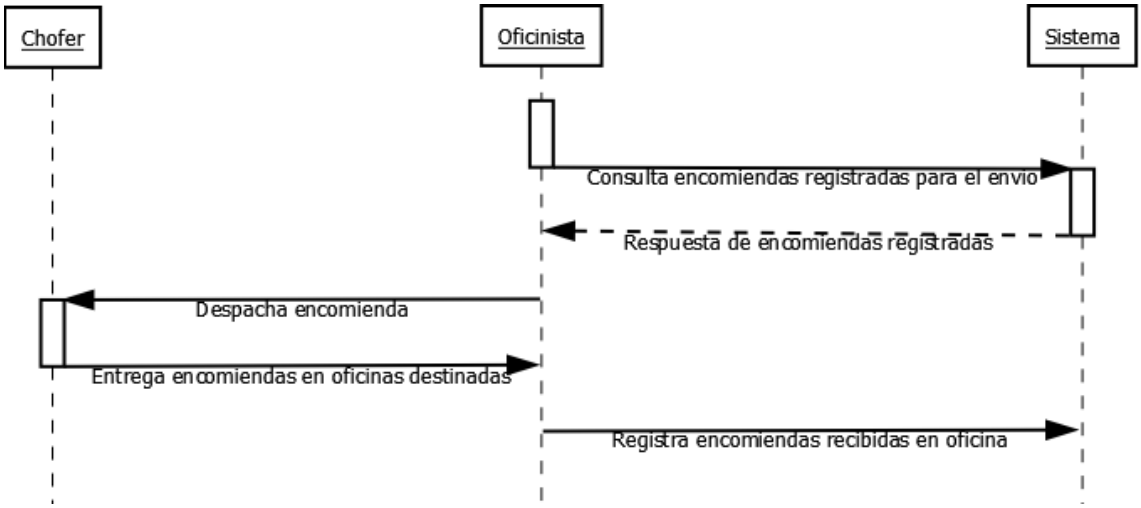
*Diagrama de secuencia del proceso de registro y seguimiento de encomienda.*




***Nota.*** El diagrama describe la interacción entre el cliente, el oficinista y el sistema, detallando desde la solicitud inicial hasta la notificación automática vía WhatsApp.

**Figura 8.**

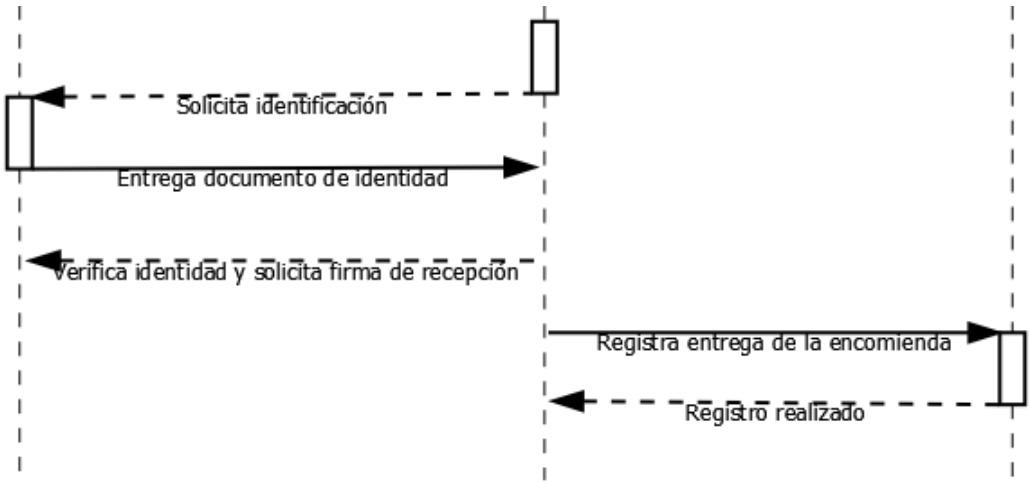
*Diagrama de secuencia del proceso de entrega de encomienda al chofer y recepción en destino.*




***Nota.*** El diagrama ilustra el flujo de trabajo entre el oficinista y chofer abarcando la logística de entrega de paquetes por oficinas y registro de llegada.

**Figura 9.**

*Diagrama de secuencia del proceso de entrega final de encomienda al destinatario.*




***Nota.*** La ilustración refleja la validación de identidad del destinatario, la solicitud de firma de registro de entrega de la encomienda al cliente.

- **Vista de despliegue**

En la vista de despliegue se establece mediante un diagrama de paquetes distribuidos y sigue una arquitectura de tres capas, donde el Frontend React se comunicará exclusivamente con los servicios expuestos en el Backend Laravel, y este último interactúa con la base de datos.

La Vista de Despliegue describe la asignación de los elementos de software del Sistema Web de Seguimiento de Encomiendas de Flota Bolívar a los recursos de hardware o nodos. La arquitectura se basa en un esquema de tres nodos físicos/lógicos principales para asegurar la disponibilidad y el rendimiento.

1. Nodo cliente

Es la representación del dispositivo utilizado por el gerente, oficinista y cliente para acceder al sistema mediante un navegador web.

Los componentes desplegados en la capa de presentación Frontend en React se descargan y se ejecutan en este nodo, incluyendo paquetes de vistas y componentes de interfaz de usuario, y el paquete de servicios encargado de la comunicación mediante peticiones HTTP/HTTPS hacia el servidor web.

2. Nodo de aplicación

El servidor dedicado que aloja la lógica central del negocio. Este nodo debe ser escalable, para manejar las transacciones.

Los componentes desplegados en la capa de aplicación Backend en Laravel se despliegan en este nodo, e incluye paquetes de rutas API, controladores HTTP y Middlewares para la gestión de autenticación. Además, desde aquí se gestiona la comunicación con servicios externos como el API de WhatsApp al cliente destinatario.

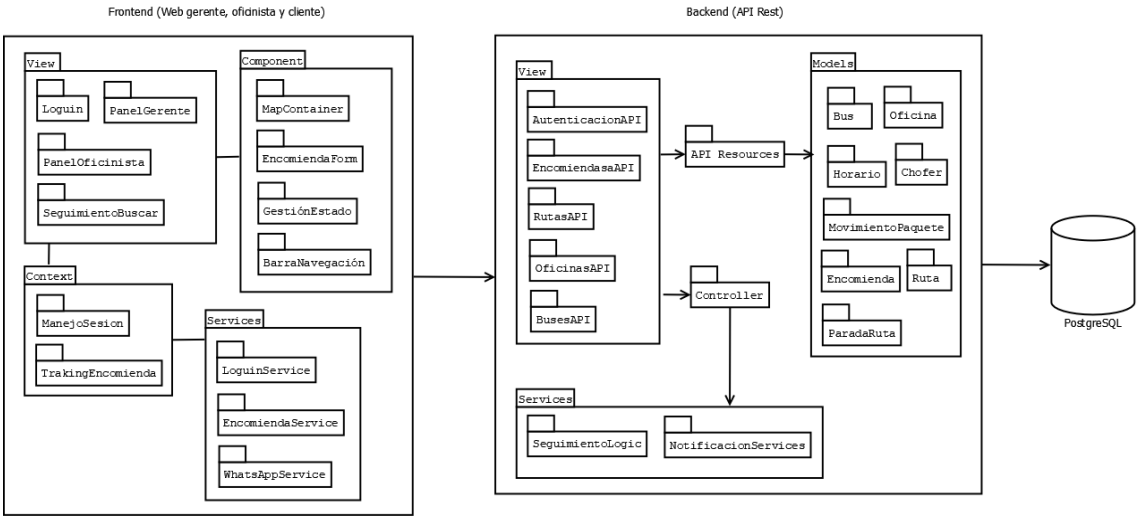
3. Nodo de base de datos

Es el servidor optimizado para el almacenamiento y la recuperación rápida de datos. Por lo general, se mantendrá separado físicamente del servidor web por seguridad y rendimiento. 01

En este nodo residen los componentes de la Capa de Datos, que incluye el esquema físico de PostgreSQL y las Migraciones que definen la estructura de las tablas relacionales. La interacción es gestionada exclusivamente por el servidor web a través de conexiones seguras**.**

**Figura 10**.

*Diagrama de paquetes (Vista de despliegue)*



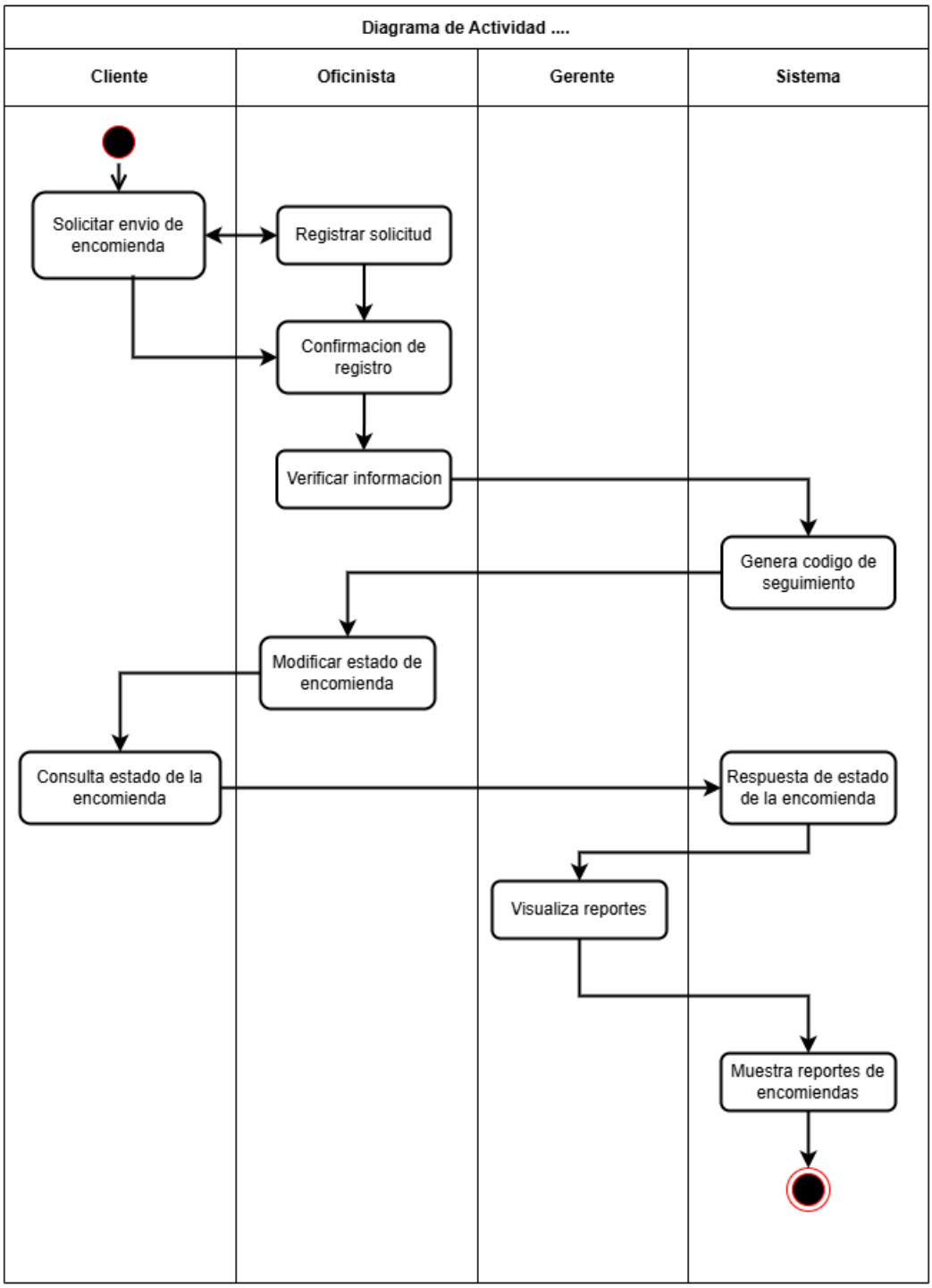

***Nota.*** Estructura de Frontend y Backend del sistema.

- **Vista de procesos**

Para la vista de procesos se representa en un diagrama de actividades, los cuales representa el flujo principal que hay en el sistema para las funcionalidades importantes como: registrar encomienda y la consulta el estado del envió por parte del cliente.

**Figura 11.**

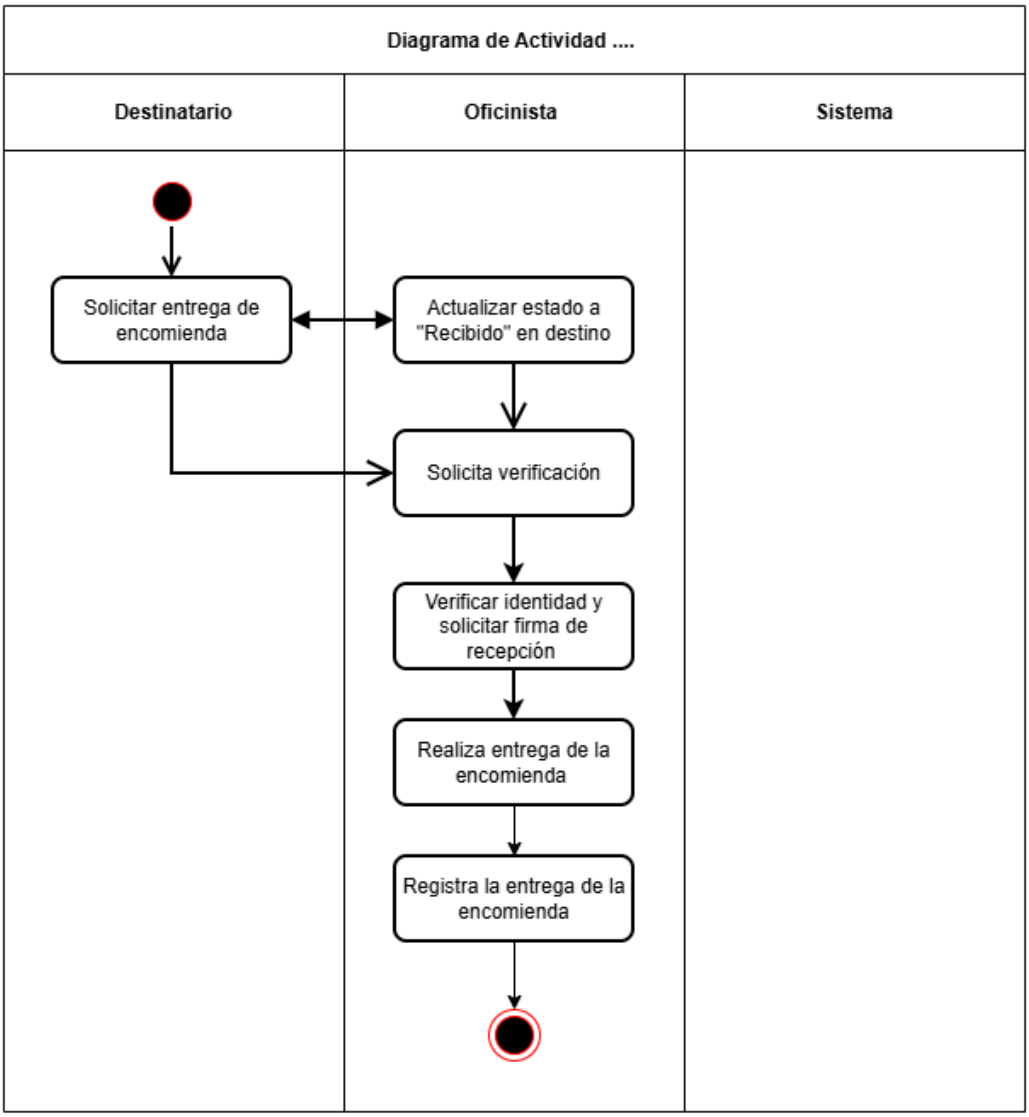
*Diagrama de actividad, proceso de registro de encomienda.*




***Nota.*** Flujo de registro y recepción de la encomienda del cliente remitente.

**Figura 12.**

*Diagrama de actividad, entrega de la encomienda al destinatario*




***Nota.*** Flujo de entrega de la encomienda al cliente destinatario.

- **Vista física**

Para la vista física del sistema, se representa en un diagrama de despliegue que ilustra la topología del hardware y los entornos de ejecución sobre los cuales se distribuyen los componentes del software. A continuación, se describe los nodos que proponen la estructura:

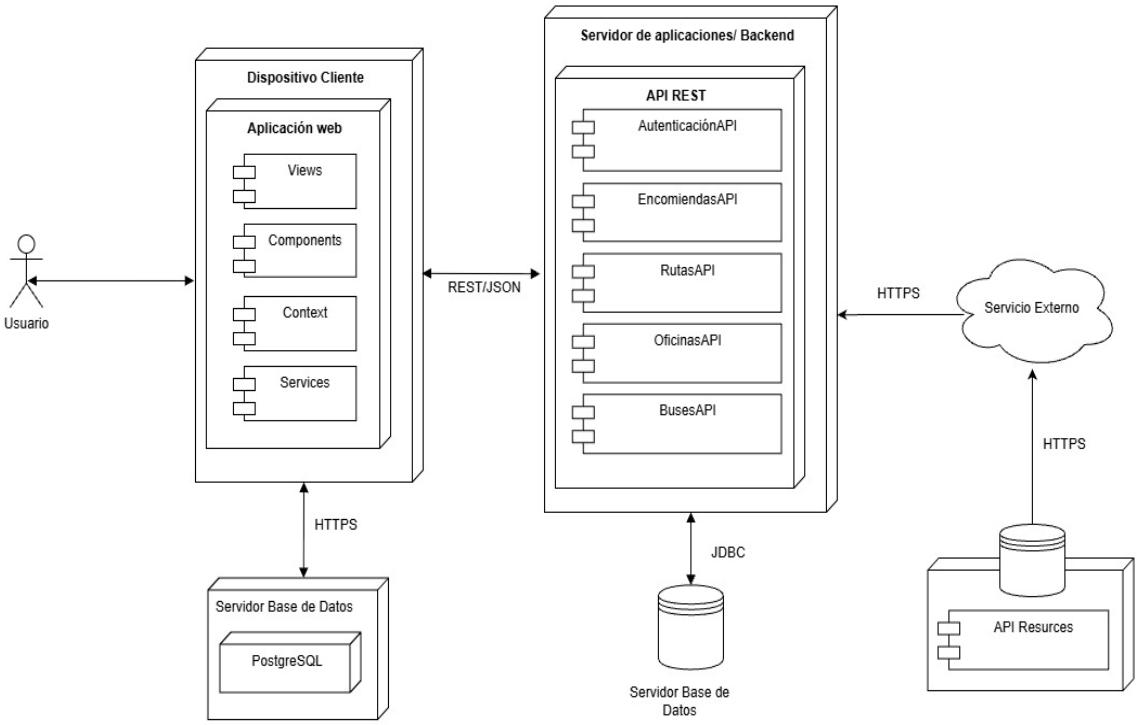
1. Dispositivo cliente: representa el dispositivo portátil o móvil con el que los autores cliente, gerente y oficinista usaran la aplicación. Enfocado en ser compatible con diversos navegadores web como Chrome, Firefox, Egde, etc. El entorno se descarga y ejecuta en el Frontend desarrollado en React, cuenta con los

componentes de interfaz de usuario, formulario de registro, mapas de visualización de ruta y módulo de autenticación que consume los servicios del Backend.

2. Servidor web: estará desplegado en una nube privada, contendrá el API Rest del sistema realizado con Laravel. Este nodo centraliza la lógica de negocio y tiene los componentes de los controladores, la gestión de seguridad, los modelos Eloquent y las rutas que procesan la asignación de encomiendas y choferes.
3. Servidor de base de datos: gestionado en postgres, nodo independiente optimizado para el almacenamiento de datos relacionales. Aquí se ejecuta el motor de PosgreSQL, en donde se almacenará toda la información estructurada que requiere el sistema, garantizando la integridad y persistencia de los datos.
4. API de WhatsApp: un servicio externo de mensajería que se conecta con el servidor web. Este nodo recibe las peticiones del backend para realizar el envío automático de notificaciones a los clientes destinatario una vez que la encomienda llegue a la oficina destino.

**Figura 13.**

*Diagrama de despliegue.*




***Nota.*** Especificación técnica del sistema construido mediante el framework backend Laravel, frontend React y PostgreSQL como gestor de base de datos.

### 4.3 Codificación

En esta fase codificación, el proyecto se dividió en backend en laravel y frontend en React, ambas integradas mediante una API REST y PostgreSQL como base de datos, está organizado en carpetas que cumplen un rol en específico dentro de la lógica del sistema de seguimiento de encomiendas.

- **Backend Laravel**

**/app/Http/Controllers/Api**

Contiene los controladores responsables de gestionar las solicitudes provenientes del frontend React a través de la API REST. Cada controlador corresponde a un módulo operativo del sistema, actuando como intermediarios entre el frontend y la lógica de negocio:

- BusController.php: administra el registro, consulta y actualización de los buses utilizados en las rutas.
- ChoferController.php: gestiona la información de los choferes asociados a los envíos.
- ClienteController.php: administra datos de los clientes, su creación y validaciones.
- DashboardController.php: genera estadísticas operativas y consolidadas para el rol gerente.
- EncomiendaController.php: gestiona el registro de encomiendas, la asignación de guías, el cambio de estado y la lógica principal del ciclo de vida del paquete.
- MovimientoImagenController.php: controla la carga y registro de imágenes asociadas a movimientos del paquete.
- MovimientoPaqueteController.php: administra cada evento asociado al proceso logístico (recibido, despacho, en ruta, en oficina destino, entregado).
- OficinaController.php: gestiona las oficinas de la cooperativa donde se reciben o entregan las encomiendas.
- ReporteEncomiendaController.php: genera reportes históricos para usuarios internos.

**/app/Models**

Aloja los modelos Eloquent que representa las tablas del sistema, contiene archivos de bus, chofer, cliente, encomienda, horarios, movimiento imagen, movimiento paquete, oficina, parada ruta, ruta y user.

Estos modelos encapsulan reglas de negocio esenciales, tales como relaciones entre entidades, validaciones internas, casteo de datos y definición de atributos asignables. Además, permiten interactuar con la base de datos de manera estructurada mediante Eloquent ORM, facilitando operaciones como la inserción, actualización, eliminación y consulta de información sobre las encomiendas y sus movimientos.

**/app/Services**

Contiene servicios especializados que encapsulan lógica de negocio compleja que no debe residir ni en los controladores ni en los modelos.

- WhatsAppService.php: gestiona él envió de mensajes automáticos por WhatsApp hacia los clientes. Se utiliza cada vez que una encomienda cambia de estado, enviando notificaciones como “Paquete registrado”, “En ruta”, “Llego a oficina destino” o “Entregado al cliente”.

  Internamente se comunica con una API externa enviando solicitudes HTTP con los datos del cliente y el mensaje correspondiente.
- DateService: centrada en operaciones relacionadas con el manejo de fechas y formatos, tales como conversión de zona horaria, formateo de fechas para reportes, calculo en diferencias de tiempo o estandarización de formatos.
- PdfService.php: genera los formatos PDF, tales como el comprobante de envio, reportes de encomiendas o respaldos para el cliente. A través de una librería de creación de PDFs, convierte la información del sistema en archivos descargables.

**/app/Providers**

Este reúne clases fundamentales que permiten configurar de forma centralizada multiples servicios del framework laravel. Actúan como puntos de inicialización del sistema, ya que se encargan automáticamente durante el arranque de la aplicación.

- AppServiceProvider.php: se encarga de configurar servicios globales para el sistema. En el contexto del sistema, incluye instrucciones como el formateo predeterminado de fechas, la configuración general de paginadores, la carga de utilidades globales o la vinculación de interfaces con sus implementaciones concretas.

- AuthServiceProvider.php: define las políticas de autorización del sistema, dependiendo de su rol de registro, esto garantiza que cada ruta de la API sea accedida únicamente por el rol adecuado.
- EventServiceProvider.php: conecta eventos con sus respectivas acciones automáticas. Dado que realiza acciones como el envío de notificaciones vía WhatsApp una vez que la encomienda haya sido registrada en el sistema como "recibido en oficina".
- RouteServiceProvider.php: organiza las rutas REST dentro de routes/api.php definiendo como se agrupan y que middlewares las protegen. Esto permite que el sistema mantenga un flujo bien organizado entre el backend y frontend, facilitando que React consuma los endpoints sin inconsistencias.

**/app/Traits**

Reúne fragmentos de código reutilizables diseñados para ser incluidos en múltiples modelos, servicios y controladores, permitiendo compartir funcionalidades comunes sin duplicarlas.

- HandlesDates.php: contiene métodos especializados en el manejo de fechas dentro. Usado para sellos temporales dentro de la base de datos, usando registros como el estado de la encomienda, conjunto con los controladores y modelos pueden manejar fechas de forma uniforme, reduciendo errores y garantizando que la información temporal sea consistente en todo el flujo del proceso lógico.
- ManejaClientes.php: agrupa funciones relacionadas en la gestión de clientes, como el registro de una nueva encomienda, la vinculación de clientes a movimientos o la verificación de contactos previos para él envió de notificaciones por WhatsApp.

**/badabase/migration**

Contiene la estructura completa de la base de datos del sistema. Aquí se encuentran las definiciones de las tablas como encomiendas, movimiento_paquetes, movimiento_imagenes, oficinas, choferes, rutas, paradas, rutas, buses y usuarios. cada migración establece los tipos de datos, relaciones claves y foráneas y restricciones necesarias.

**/badabase/seeders**

Se utilizo para cargar información inicial indispensable para ejecutar datos en el sistema. Los seeders poblaron datos con lista de oficinas disponibles, rutas operativas, roles de usuarios, configuraciones por defecto para pruebas funcionales.

**/database/factories**

Facilita la creación de datos ficticios mediante la definición de plantillas para modelos con usuarios, encomiendas o movimientos. Utilizados para pruebas automatizadas, ayudando a generar pruebas automatizadas, permitiendo generar escenarios completos en segundos para validar los controladores.

**/routes/api.php**

Define todos los endpoints que el frontend en React consume mediante la API REST. Esta estructura funciona como un mapa completo de comunicación en las funcionalidades visuales y la lógica del backend. En este archivo se encuentran rutas organizadas por módulos, cada una enlazada con su controlador dentro de /app/Http/Controllers/Api

- **Frontend React**

**/src/components/ui**

Estos componentes encapsulan la lógica visual relacionada con la presentación del estado de los paquetes, formularios de registro y elementos de interacción con el usuario, permitiendo que las páginas principales se mantengan limpias y enfocadas en la lógica de negocio.

- Navbar.tsx: este define la barra de navegación superior del sistema. Su función es permitir el acceso rápido a secciones principales y mostrar información contextual del usuario.
- Toaster.tsx: gestiona la visualización de notificaciones emergentes. Su función es mostrar mensajes de error o advertencia como respuesta a acciones del usuario.
- LocationSearch.tsx: define un componente de búsqueda de ubicaciones. Permite al usuario buscar y seleccionar direcciones o puntos geográficos, integrándose con mapas.
- mapPicker.tsx: permite al usuario selecciona coordenadas geográficas de manera visual, utilizando modulos como la sección de rutas y el seguimiento de la encomienda.

**/src/features**

Esta sección organiza las funcionalidades del sistema, alineándose directamente con las historias de usuarios definidas. Cada carpeta representa una funcionalidad independiente, facilitando el desarrollo incremental, las pruebas y refactorización. Contiene carpetas definiendo los siguientes módulos:

- auth: define la autenticación del usuario dentro del sistema, incluye funciones como el formulario de inicio de sesión, manejo de tokens de autenticación y el control de acceso según los roles definidos en el sistema.
- clientes: se encarga del registro de clientes y la consulta de información de clientes asociados con la encomienda. Además, de visualizar el historial de envíos por cliente.
- dashboard: presenta los módulos principales, en base al rol del usuario, contiene resúmenes estadísticos de las encomiendas registradas.
- encomiendas: permite el registro de encomienda, el consultar detalles de una encomienda, la actualización del envió. Además, de asociar rutas, oficinas y clientes.
- home: define la pantalla principal del sistema, visible para todos los usuarios, incluye información general del servicio, y el acceso al tracking público de encomiendas.
- oficinas: gestiona información de oficinas, teniendo funciones como el registro, la visualización de oficinas y la asociación de oficias a tutas y encomiendas.
- reportes: genera y visualiza los reportes por fechas, incluye dentro de un listado las encomiendas entregadas o pendientes, permitiendo la descarga de reportes generador por el backend en formato PDF.
- rutas: gestiona las rutas de transporte de la cooperativa, permite registrar rutas, visualizar y asociar rutas a buses y encomiendas.
- tracking: modulo destinado al seguimiento de encomiendas por parte del cliente. Permite consultar el estado de la encomienda mediante un código de guía única.
- usuarios: gestiona la administración de usuarios del sistema. Permite el registro y edición de usuarios, asigna roles y la activación y desactivación de cuentas.

**/src/lib**

Contiene utilidades y configuraciones compartidas que sirven de soporte al funcionamiento del sistema. Incluye configuraciones para el consumo de la API REST

desarrollada en Laravel, como instancias de Axios y funciones para manejar solicitudes HTTP, gestión de encabezados de autenticación y tratamiento de errores de comunicación. Además, se incluyen utilidades para formateo de fechas, manejo de tokens de sesión y normalización de datos antes de ser presentados en la interfaz.

- **Interfaces de pantalla**

**Tipografía**

Para mantener la legibilidad del contenido, se ha considerado utilizar la familia de tipografía de fuentes (Sans-serif). Predeterminada en React, tiende a ser más legibles en pantalla y permite la lectura más fluida, crea un enfoque de una interfaz minimalista.

- **Logo**

**Figura 14.**

*Logo de la Cooperativa de Transportes Flota Bolívar*

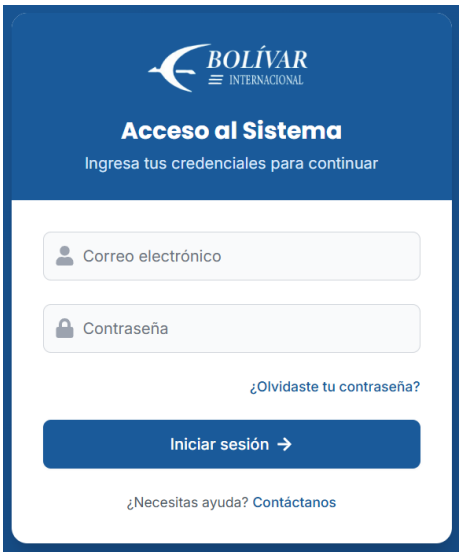



**Nota.** Representación del logo corporativo y emblema oficinal de la institución. Adaptado de [Figura], por Cooperativa de Transporte Flota Bolívar, 2025.

- **Loguin para gerente y oficinista**

**Figura 15.**

*Login para ingreso al interfaz gerente y oficinista*

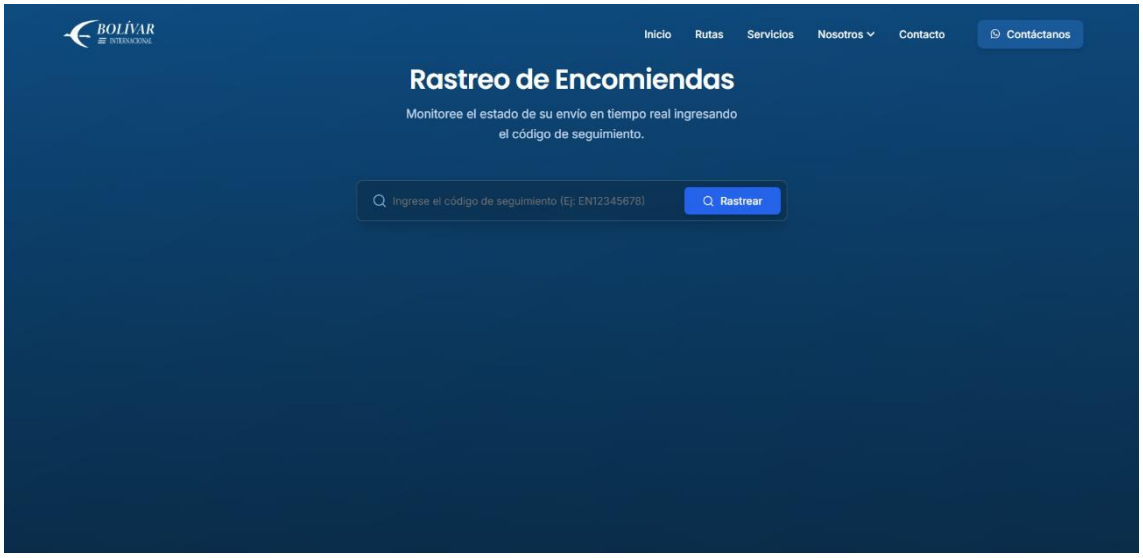



***Nota.*** Permiso de solo acceso a usuarios gerente y oficinista.

- **Interfaz de seguimiento de encomienda**

**Figura 16.**

*Interfaz de seguimiento de encomienda*




***Nota.*** Barra de búsqueda que permite ingresar el código de seguimiento, para visualizar el estado de su encomienda.

- **Interfaz principal**

**Figura 17.**

*Interfaz principal*

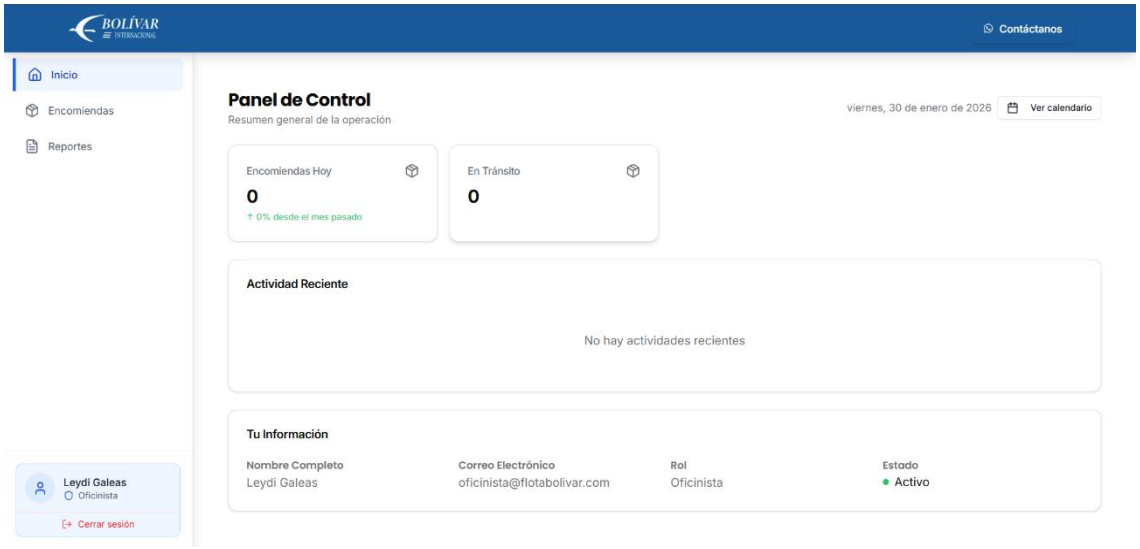



***Nota.*** Interfaz principal, con su barra de navegación ubicado en la parte superior, y su botón del rastreo de encomienda.

- **Interfaz oficinista**

**Figura 18.**

*Panel de control.*

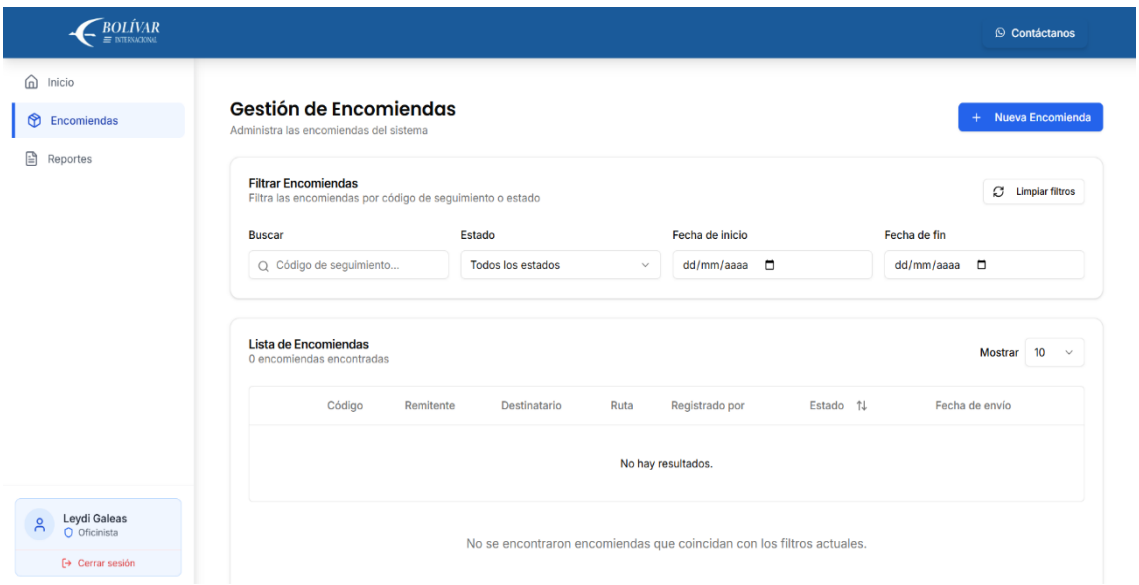



***Nota.*** Panel de control, con datos informativos de encomiendas registrados por día, en tránsito y la actividad reciente.

- **Interfaz Gestión de encomiendas**

**Figura 19.**

*Gestión de encomiendas*

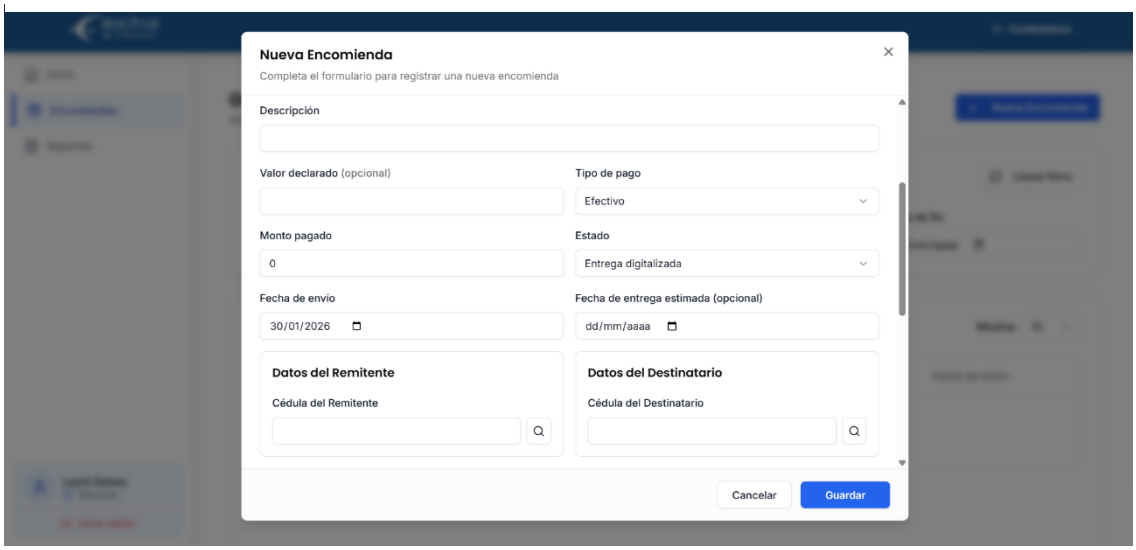



***Nota.*** Datos de encomiendas registradas por el oficinista, y gestión de cambio de estado de la misma.

- **Interfaz Formulario de registro de encomienda.**

**Figura 20.**

*Formulario de registro de encomienda.*

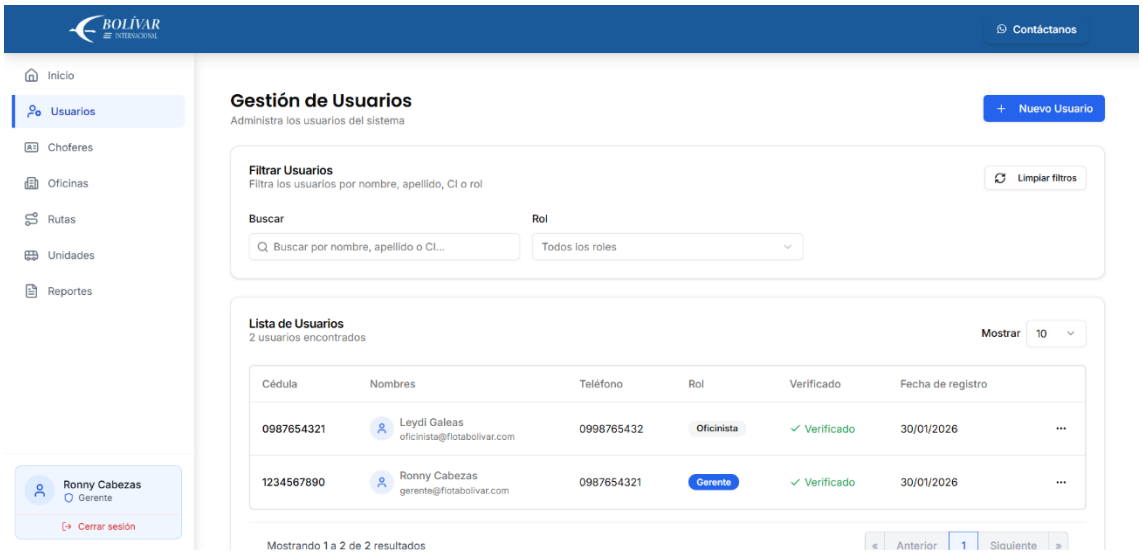



***Nota.*** Formulario de registro de los datos de la encomienda, con gestión integrada de remitente, destinatario y detalles del paquete de envío.

- **Interfaz Gerente**

**Figura 21.**

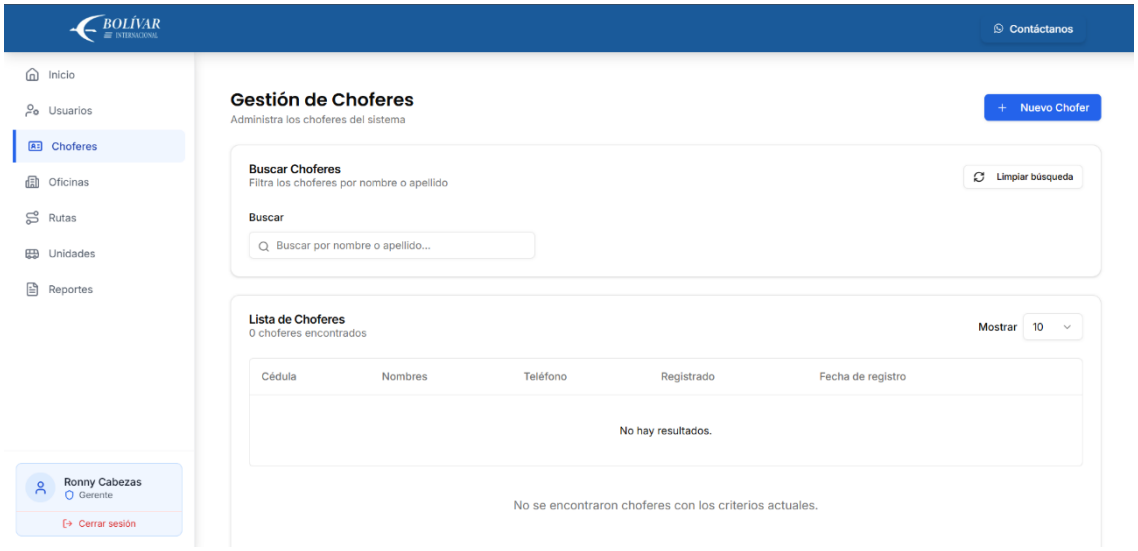
*Gestión de usuarios.*




***Nota.*** Módulo de gestión de usuario, donde se visualizará el gerente y oficinistas registrados en el sistema.

**Figura 22.**

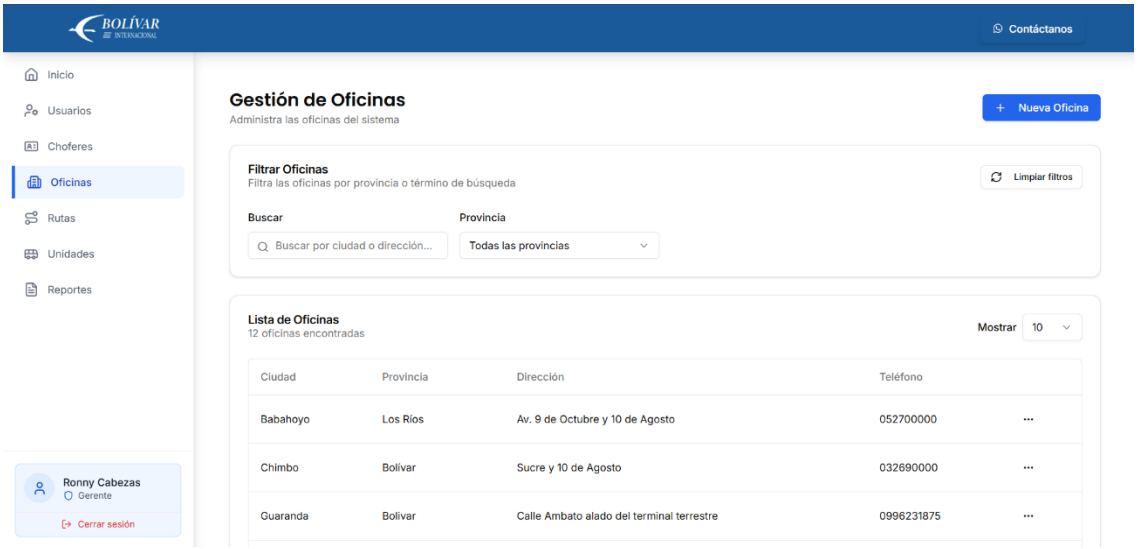
*Gestión de choferes.*




***Nota.*** Módulo de gestión de usuarios, visualizándose datos de los choferes registrados.

**Figura 23.**

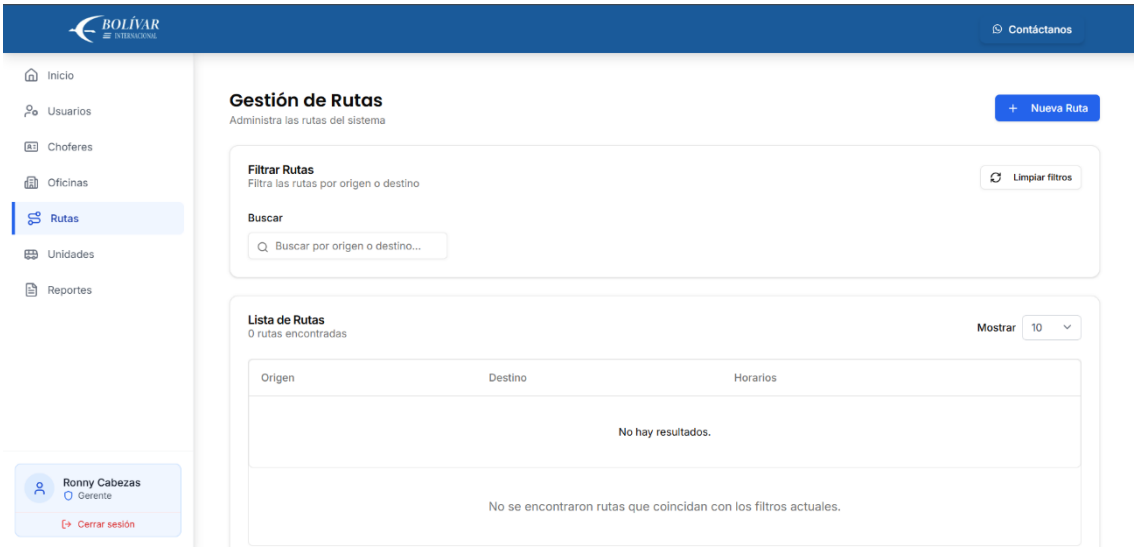
*Gestión de oficinas.*




***Nota.*** Módulo de gestión de oficinas, donde se visualizará los datos de las oficinas registradas en el sistema.

**Figura 24.**

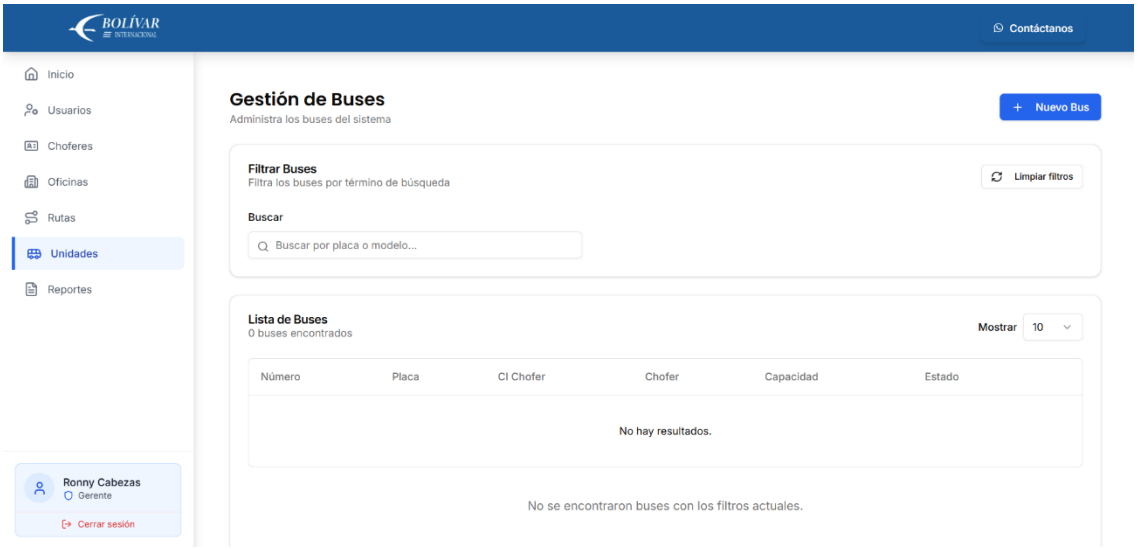
*Gestión de rutas.*




***Nota.*** Módulo de gestión de rutas, donde se visualizará los datos registrados de las rutas registradas en el sistema.

**Figura 25.**

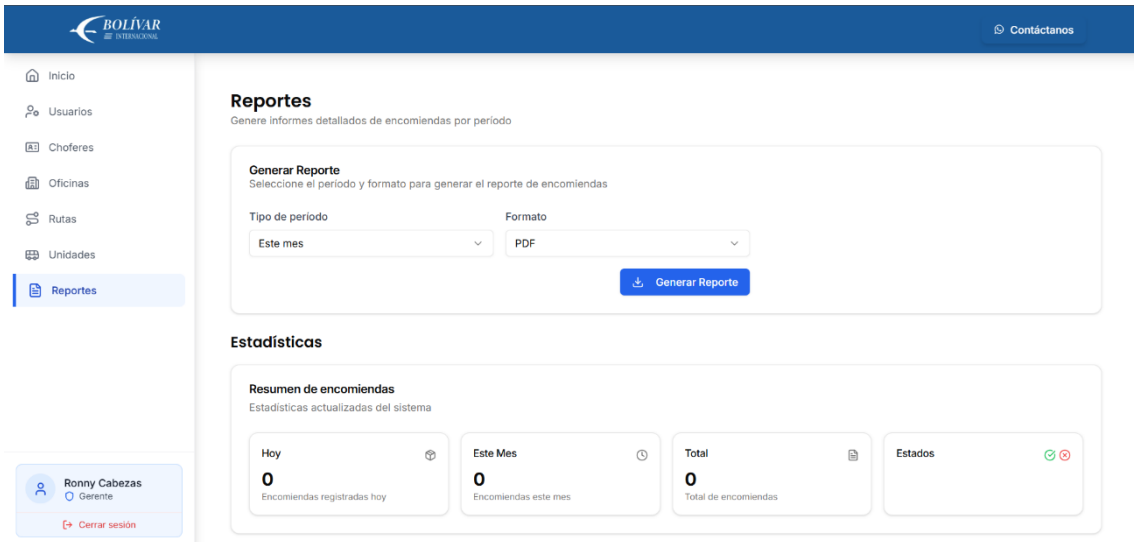
*Gestión de buses.*




***Nota.*** Módulo de gestión de buses, donde se visualizará los datos de los buses registrados en el sistema.

**Figura 26.**

*Reportes de encomiendas.*




***Nota.*** Módulo de reportes de las encomiendas, donde tanto el oficinista como el gerente podrán descargar un formato PDF con el detalle de las encomiendas registradas.

**Servicio**: Mapa de seguimiento de encomienda

**Descripción:** Implementación de API y conjunto de herramientas y servicios basados en el ecosistema OpenStreetMap.

```
import { format } from 'date-fns';
import { es } from 'date-fns/locale';
import { CheckCircle2, Clock, Package, Truck, Home, XCircle } from 'lucide-react';
import { cn } from '@/lib/utils';
const statusIcons = {
  'Entrega digitalizada': <Package className="h-5 w-5" />,
  'En ruta': <Truck className="h-5 w-5" />,
  'Recibido en oficina': <Home className="h-5 w-5" />,
  'Entregado al cliente': <CheckCircle2 className="h-5 w-5" />,
  'Cancelado': <XCircle className="h-5 w-5" />,
  'default': <Clock className="h-5 w-5" />
};
const statusColors = {
  'Entrega digitalizada': 'bg-blue-100 text-blue-800',
  'En ruta': 'bg-yellow-100 text-yellow-800',
  'Recibido en oficina': 'bg-indigo-100 text-indigo-800',
  'Entregado al cliente': 'bg-green-100 text-green-800',
  'Cancelado': 'bg-red-100 text-red-800',
  'default': 'bg-gray-100 text-gray-800'
};
export function TrackingTimeline({ movements }: { movements: any[] }) {
  if (!movements || movements.length === 0) {
```

```
    return (
      <div className="text-center py-8 text-gray-500">
        No hay movimientos registrados para este envío.
      </div>
    );
  }
  // Ordenar movimientos por fecha (más reciente primero)
  const sortedMovements = [...movements].sort(
    (a, b) => new Date(b.fecha_hora.replace(' ', 'T') + (b.fecha_hora.includes('Z') ? '' :
'Z')).getTime() - new Date(a.fecha_hora.replace(' ', 'T') + (a.fecha_hora.includes('Z') ? '' :
'Z')).getTime()
  );
  return (
    <div className="flow-root">
      <ul className="-mb-8">
        {sortedMovements.map((movement, index) => {
          const isCurrent = index === 0;
          const statusIcon = statusIcons[movement.estado as keyof typeof statusIcons] ||
statusIcons.default;
          const statusColor = statusColors[movement.estado as keyof typeof statusColors] ||
statusColors.default;

          return (
            <li key={movement.id_movimiento} className="relative pb-8">
              {index !== sortedMovements.length - 1 && (
                <span
                  className="absolute left-4 top-4 -ml-px h-full w-0.5 bg-gray-200"
                  aria-hidden="true"
                />
              )}
              <div className="relative flex items-start space-x-3">
                <div className="relative">
                  <span className={cn(
                    'h-8 w-8 rounded-full flex items-center justify-center ring-8 ring-white',
                    isCurrent ? statusColor : 'bg-gray-100 text-gray-500',
                    isCurrent ? 'ring-4 ring-opacity-30' : ''
                  )}>
                    {statusIcon}
                  </span>
                </div>
                <div className="min-w-0 flex-1 pt-1.5">
                  <div className="flex justify-between">
                    <div>
                      <p className="text-sm font-medium text-gray-900">
                        {movement.estado}
                      </p>
                      <p className="text-sm text-gray-500">
                        {movement.oficina?.ciudad}, {movement.oficina?.provincia}
                      </p>
                      {movement.observaciones && movement.observaciones !== 'Sin observaciones'
&& (
                        <p className="mt-1 text-sm text-gray-500">
```

```
                    {movement.observaciones}
                  </p>
                )}
              </div>
              <div className="whitespace-nowrap text-right text-sm text-gray-500">
                {format(new Date(movement.fecha_hora.replace(' ', 'T') +
(movement.fecha_hora.includes('Z') ? '' : 'Z')), "PPp", { locale: es })}
              </div>
            </div>
            {movement.oficinista && (
              <div className="mt-1 text-sm text-gray-500">
                <span className="font-medium">Oficinista:</span>{' '}
                {movement.oficinista.nombre_completo}
              </div>
            )}
          </div>
        </div>
      </li>
    );
  })}
 </ul>
</div>
);
}
```

## 4.4 Pruebas

### *4.4.1 Propósito*

En el sistema de Seguimiento de Encomiendas de la Cooperativa de Transporte Flota Bolívar, el enfoque principal en las pruebas es el valor que el sistema ofrecerá a los involucrados, debe cumplir con las necesidades, pudiendo detectar distintos aspectos de calidad del software y problemas de seguridad, para garantizar que el producto final es de calidad y está libre de bugs y es intuitivo para los usuarios.

### *4.4.2 Diseño y desarrollo de casos de prueba*

- **Pruebas de funcionalidad:** se debe verificar que los procesos centrales del seguimiento de encomiendas operen según lo establecido.
- **Pruebas de rendimiento:** se evalúa la respuesta del sistema bajo una carga operativa real de datos masivos de la cooperativa.
- **Pruebas de seguridad:** se analiza la resistencia del sistema antes posibles vulnerabilidades y amenazas de seguridad. Además, de la validación de autenticación del ingreso al sistema de los usuarios autorizados.

- **Pruebas de usabilidad:** se mide la facilidad de uso para el gerente, oficinistas y clientes. El diseño responsivo y el uso de tipografía ayudan a que el sistema tienda a ser más intuitivo para los involucrados.
- **Pruebas de integración:** se evalúa la interoperabilidad entre los diferentes componentes del sistema.

1. **Ejecución y registro**

**Tabla 20.**

*Stakeholders*

| **Stakeholders** | |
|---|---|
| Equipo de evaluación | Su tarea es testear el sistema y garantizar los resultados según el análisis. |
| Gerente y oficinistas de la Cooperativa de Transporte Flota Bolívar | El gerente desempeñando un rol del administrador podrá registrar nuevos, oficinistas, rutas y horarios, la impresión de reportes.<br>El oficinista podrá registrar la encomienda ingresando datos del remitente y destinatario. Además, de generar los reportes por un periodo establecido.<br>El cliente visualizara el estado de su encomienda al ingresar el código del seguimiento. |

***Nota.*** Involucrados en el sistema.

**Tabla 21.**

*Gestor de pruebas*

| **Gestor de pruebas** | |
|---|---|
| Navegadores | Brave, Google Chrome |
| Sistemas Operativos | Windows |
| Servidor Web | Apache |
| Lenguajes de Programación | PHP, JavaScript, TypeScript |
| Framework | Backend (Laravel), Frontend (React) |

| | |
|---|---|
| Gestor de base de datos | Postgres |

***Nota.*** Entorno de trabajo para realizar pruebas de desarrollo y funcionalidad.

## 2. Pruebas unitarias

**Tabla 22.**

*Pruebas de registro de oficinistas.*

| **Registro de oficinistas** | | | |
|---|---|---|---|
| **Componente** | **Ítem a evaluar** | **Cumplimiento** | |
| | | **Si** | **No** |
| Login | Verificar que el correo electrónico no conste como registrado. | X | |
| Login | La gestión de roles esta realizada de manera adecuada. | X | |
| Login | Verificar que el número de cedula sea único registrado en el sistema | X | |
| Login | La contraseña debe contener más de 8 dígitos para su registro | X | |

***Nota.*** Pruebas de registro de oficinistas, desde la interfaz cliente.

**Tabla 23.**

*Gestión de oficinas*

| **Gestión de oficinas** | | | |
|---|---|---|---|
| **Componente** | **Ítem a evaluar** | **Cumplimiento** | |
| | | **Si** | **No** |
| Filtros de búsqueda | Verificar el funcionamiento de la búsqueda por filtrada por provincia. | X | |
| Formulario | Permite crear nuevas oficinas, estableciendo en un mapa de latitud y longitud | X | |

***Nota.*** Pruebas seguridad y rendimiento en el módulo de oficinas.

**Tabla 24.**

*Gestión de rutas*

| Gestión de rutas | | | |
|---|---|---|---|
| **Componente** | **Ítem a evaluar** | **Cumplimiento** | |
| | | **Si** | **No** |
| Barra de búsqueda | Verificar el funcionamiento de la barra de búsqueda ciudad de origen o destino. | X | |
| Formulario | Permite crear nuevas rutas y horarios, y crear rutas intermedias. | X | |

***Nota.*** Pruebas de seguridad y rendimiento en el módulo de rutas.

**Tabla 25.**

*Reportes de encomiendas*

| Reportes de encomiendas | | | |
|---|---|---|---|
| **Componente** | **Ítem a evaluar** | **Cumplimiento** | |
| | | **Si** | **No** |
| Filtros | Verificar el funcionamiento del tiempo de periodo y su formato a ser descargado. | X | |
| PDF | Constará con datos de las encomiendas, dependiendo el tipo de periodo elegido. | X | |

***Nota.*** Pruebas de seguridad y rendimiento en el módulo de reportes e impresión de reportes.

**Tabla 26.**

*Gestión de encomiendas*

| Gestión de encomiendas | | | |
|---|---|---|---|
| **Componente** | **Ítem a evaluar** | **Cumplimiento** | |
| | | **Si** | **No** |
| Formulario de registro | Verificar que los campos estén correctamente llenados. | X | |

| | | |
|---|---|---|
| Datos de clientes | Permite la búsqueda del cliente en caso de haber sido registrado, caso contrario tendría que ser registrado. | X |
| Documentos y comprobantes | Permite el ingreso de imágenes del recibo firmado por el cliente y la cedula declarando por el tipo de imagen subida. | X |

***Nota.*** Pruebas de seguridad y rendimiento en el módulo de gestión y registro de encomiendas.

### 3. Pruebas de funcionalidad

**Tabla 27.**

*Pruebas de funcionalidad*

| **Pruebas de funcionalidad** | | |
|---|---|---|
| **Objetivo:** | | |
| **Aspecto evaluado** | **Cumplimiento** | |
| | **Si** | **No** |
| ¿El sistema permite el registro del gerente y oficinista? | X | |
| ¿El sistema permite registrar oficinas? | X | |
| ¿El sistema permite el registro de encomiendas? | X | |
| ¿El sistema permite la impresión de reportes? | X | |
| ¿El sistema permite el registro de rutas y horarios? | X | |
| ¿El sistema permite la actualización del estado de la encomienda? | X | |
| ¿El sistema permite a los clientes realizar el monitoreo de su encomienda, con el ingreso del código de seguimiento generado? | X | |

***Nota.*** Pruebas de funcionalidad en la actualización de estados, reportes y registros de encomiendas.

**Tabla 28.**

*Pruebas de disponibilidad*

| **Pruebas de disponibilidad** | | |
|---|---|---|
| **Objetivo:** | | |
| **Aspecto evaluado** | **Cumplimiento** | |
| | **Si** | **No** |
| ¿El sistema estará disponible las 24 horas del día? | X | |
| ¿El sistema cerrara su acceso mediante futuros mantenimientos? | X | |
| ¿El sistema cerrara sesiones activas mediante el mantenimiento? | X | |

***Nota.*** Pruebas de disponibilidad y seguridad en la gestión de inicio y cerrar sesión.

**4. Pruebas de rendimiento**

**Tabla 29.**

*Pruebas de rendimiento*

| **Pruebas de rendimiento** | | |
|---|---|---|
| **Objetivo:** | | |
| **Aspecto evaluado** | **Cumplimiento** | |
| | **Si** | **No** |
| ¿El sistema gestiona múltiples peticiones a la ves? | X | |
| ¿Si existe una sobre carga de datos, el sistema es capaz de conllevar a cabo las operaciones con total normalidad? | X | |

***Nota.*** Pruebas de rendimiento y posibles colapsos del sistema en futuras cargas masivas de información.

5. **Pruebas de seguridad**

**Tabla 30.**

*Pruebas de seguridad*

| **Pruebas de seguridad** | | |
|---|---|---|
| **Objetivo:** | | |
| **Aspecto evaluado** | **Cumplimiento** | |
| | **Si** | **No** |
| ¿El sistema gestiona de manera correcta las sesiones de los usuarios? | X | |
| ¿El sistema el vulnerable ante ataques de inyección SQL? | | X |
| ¿El sistema encripta las contraseñas? | X | |
| ¿Se cifran los datos al enviarse al servidor? | X | |
| ¿El sistema controla las vistas de usuario según el rol asignado? | X | |
| ***Nota.*** Pruebas de seguridad en la gestión de roles e información guardada en la base de datos. | | |

6. **Pruebas de usabilidad**

**Tabla 31.**

*Pruebas de usabilidad*

| **Pruebas de usabilidad** | | |
|---|---|---|
| **Objetivo:** | | |
| **Aspecto evaluado** | **Cumplimiento** | |
| | **Si** | **No** |
| ¿El sistema incorpora iconos adecuados para su funcionalidad? | X | |
| ¿La información expuesta está ubicada correctamente para su fácil acceso? | X | |
| ¿El sistema muestra en el encabezado en la sección donde se encuentra el usuario? | X | |
| ¿Los clientes pueden acceder con facilidad a la barra de búsqueda de la encomienda? | X | |

| | | |
|---|---|---|
| ¿El menú de acciones se establece según el rol de usuario? | X | |
| ¿El diseño de la interfaz es adecuado para los stakeholders? | X | |
| ¿El sistema ofrece varios módulos de interacción (canal de sonido)? | | X |
| ¿El sistema presenta documentación? | X | |
| ¿El sistema incorpora ayuda guiada al usuario? | | X |
| ¿El sistema responde dentro de un periodo adaptable? | X | |
| ¿El sistema ofrece la opción de configuración de cambio de apariencia? | X | |

***Nota.*** Gestión de pruebas, que evalúa la usabilidad del sistema por parte del usuario que interviene.

### 7. Pruebas de integración

**Tabla 32.**

*Pruebas de integración*

| **Pruebas de integración** | | | |
|---|---|---|---|
| Gestión de encomiendas<br>1. El oficinista registra las encomiendas del remitente.<br>2. El sistema genera un código único de seguimiento.<br>3. El oficinista y gerente puede imprimir el reporte de la encomienda.<br>4. El oficinista entrega la encomienda al cliente destinatario.<br>5. El oficinista sube al sistema la cedula y recibo firmado por el destinatario.<br>6. El cliente puede ingresar el código de seguimiento en la barra de búsqueda de la encomienda dentro del sistema. | | | |
| **Involucrados** | **Ítem a evaluar** | **Cumplimiento** | |
| | | **Si** | **No** |
| Oficinista | Verifica que los datos de la encomienda se han registrado sin alteración de datos. | X | |
| Oficinista | Permite cancelar él envió de una encomienda, antes de haber sido despachado. | X | |
| Cliente | El sistema permite al cliente monitorear el estado de su encomienda. | X | |

***Nota.*** Evaluación de módulos de registro, impresiones de DPFs, generación de códigos de seguimiento e interfaz de visualización del estado de la encomienda.

### 4.5 Entrega

Una vez finalizadas las fases de codificación y pruebas, se procede a la entrega del sistema como un incremento funcional validado, de acuerdo con los principios de la metodología XP y el flujo continuo, propuesto por Kanban. Esta entrega no se limita únicamente al software, sino que incluye la documentación técnica y funcional necesaria para su correcta utilización y mantenimiento.

**Entregables del proyecto**

Como resultado de esta fase, se entregarán los siguientes productos:

- Sistema web funcional para el seguimiento de encomiendas.
- Código fuente del sistema, organizado según la arquitectura N-Capas.
- Base de datos estructurada con sus respectivas tablas y relaciones.
- Manual de usuario, orientado al gerente, oficinista y cliente. Anexo 8.
- Manual técnico, que describe la arquitectura del sistema, tecnologías utilizadas y recomendaciones para mantenimiento. Anexo 9.

**Capacitación y retroalimentación**

Como parte de la entrega, se realizará una capacitación básica al personal de la Cooperativa de Transporte Flota Bolívar, enfocada en el uso correcto del sistema y en las principales funcionalidades operativas. Esta actividad permitirá a los usuarios familiarizarse con la plataforma y brindar retroalimentación inmediata, la cual podrá ser considerada para futuras mejoras o ajustes.

Finalmente, la entrega del sistema marca el cierre formal de la fase de ingeniería del proyecto, dejando implementada una solución tecnológica funcional, confiable y alineada con las necesidades operativas de la cooperativa.

## CONCLUSIONES

- Se logró implementar un sistema web funcional para el seguimiento de encomiendas de la Cooperativa de Transporte Flota Bolívar, permitiendo automatizar los procesos de registro, control y consulta del estado de las encomiendas, mejorando así la calidad del servicio brindado a los clientes.
- El análisis del proceso actual de la gestión de encomiendas permitió identificar errores relacionados con el manejo manual de la información y la limitada comunicación con los clientes, lo que facilitó la definición clara de los requerimientos funcionales y no funcionales del sistema.
- El diseño de la solución tecnológica se desarrolló tomando en cuenta las necesidades reales de la cooperativa, para ello se utilizó una arquitectura N-Capas y modelos UML, lo que nos permitió estructurar un sistema escalable, seguro y alineado a los procesos operativos de la gestión de encomiendas.
- La programación de los módulos frontend y backend permitió desarrollar de manera efectiva las funcionalidades del sistema, logrando una correcta integración de la mensajería por WhatsApp, mejorando así la comunicación con los clientes mediante notificaciones automáticas sobre el estado de sus encomiendas.
- Las pruebas realizadas ayudaron a comprobar que el sistema esté funcionando de manera correcta, verificando que las funcionalidades desarrolladas cumplan con los requerimientos establecidos y que el software sea estable, confiable y apto para su implementación en un entorno real.

## RECOMENDACIONES

- Se recomienda a la Cooperativa de Transporte Flota Bolívar adoptar el sistema como herramienta oficial de gestión de encomiendas, promoviendo su uso continuo y brindando capacitaciones periódicas al personal para asegurar su correcta utilización.
- Se recomienda ejecutar evaluaciones periódicas de los procesos internos de la cooperativa, con el fin de identificar nuevas necesidades permitiendo así realizar mejoras continuas en futuras versiones del sistema.
- Se recomienda mantener la documentación del diseño actualizada, ya que esto nos ayudara a facilitar ampliaciones del sistema a futuro y también va a servir como referencia para nuevos desarrollos o mejoras tecnológicas que se realicen más adelante.
- Se recomienda realizar un monitoreo constante del funcionamiento del servicio de mensajería por WhatsApp y se sugiere considerar la incorporación de otros canales de comunicación en futuras versiones del sistema para ampliar las opciones de notificación.
- Se recomienda implementar un plan de pruebas continuo y realizar mantenimientos preventivos del sistema, con el objetivo de detectar y corregir posibles fallos de manera oportuna y así garantizar la calidad del software a largo plazo.

**BIBLIOGRAFÍA**


Asamblea Nacional del Ecuador. (2016). *Código Orgánico de la Economía Social de los Conocimientos, Creatividad e Innovación.* https://www.gobiernoelectronico.gob.ec/wp-content/uploads/2018/10/Codigo-Organico-de-la-Economia-Social-de-los-Conocimientos-Creatividad-e-Innovacion.pdf

Asamblea Nacional del Ecuador. (2021*). Ley Orgánica de Protección de Datos Personales.* Registro Oficial Suplemento 459. https://www.finanzaspopulares.gob.ec/wp-content/uploads/2021/07/ley_organica_de_proteccion_de_datos_personales.pdf

Beekman, G. (2005). *Introducción a la informática* (6a ed.). PEARSON Prentice Hall.

Cepeda, R. P. (2010). *Trazabilidad*. Revista ReCiTeIA.

Egas, L., & Játiva, J. (2008). *Evolución de las Metodologías de Desarrollo de la Ingeniería de Software en el Proceso de la Ingeniería de Sistemas Software,* Creación:2008; Recuperado: 10 septiembre 2025. Retrieved from http://repositorio.espe.edu.ec/bitstream/21000/8771/1/AC-ESPEL-SOF-0004.pdf

Folgueiras Bertomeu, P. (2016). *La entrevista.*

García Rodríguez, M. J. (2015). *Estudio comparativo entre las metodologías ágiles y las metodologías tradicionales para la gestión de proyectos software* (Master's thesis).

García, I. M. (2015*). Metodologías de desarrollo software. ¿tradicional o ágil?* MoleQla: revista de Ciencias de la Universidad Pablo de Olavide, (19), 7.

Gobierno Autónomo Descentralizado del Cantón Guaranda. (2025). *Guaranda*. Obtenido de https://www.guaranda.gob.ec/newsiteCMT/

Gómez Valdelomar, D. (2012). Sistema de seguimiento de envíos.

LEON, M. A. G. (2018). *Servicio de tracking online para empresas de encomiendas que no tienen portal de información*. [Tesis de magíster inédita]. Universidad de Santiago de Chile.

Ley de Propiedad Intelectual, Codificación 13. (2006). *Registro Oficial Suplemento 426 de 28-dic.-2006*. https://www.gobiernoelectronico.gob.ec/wp-content/uploads/2018/10/Ley-de-Propiedad-Intelectual.pdf

López de Jiménez, R. E. (2015). Metodologías ágiles de desarrollo de software aplicadas a la gestión de proyectos empresariales. *Revista Tecnológica: no. 8.*

López, A. M. (2023). *La entrevista en profundidad y la observación directa: observaciones cualitativas para un enfoque holístico. Caminos de utopía: Las ciencias sociales en las nuevas sociedades inteligentes*, 1(33), 739-749.

Macario, E., & Macario, N. I. E. (2023). *Sistema web para el control administrativo y manejo de proyectos de la organización coordinadora nacional indígena y campesina CONIC, utilizando el lenguaje de programación PHP, framework laravel y el gestor de base datos MYSQL*.

Maida, E. G., & Pacienzia, J. (2015). *Metodologías de desarrollo de software* [Tesis de licenciatura, Universidad Católica Argentina]. Repositorio Institucional UCA. http://bibliotecadigital.uca.edu.ar/repositorio/tesis/metodologias-desarrollo-software.pdf

Maida, E. G., Pacienzia, J., & Casiello, F. A. (2015). *Metodologías de desarrollo de software.*

Martín, M. J. R., Rodríguez, F. M., & Martin, A. R. (2006). *Sistemas gestores de bases de datos. McGraw-Hill Interamericana.*

Merchán-Narváez, N. J., Palma-Peralta, E. E., & Poma-Japón, D. X. (2024). *Comparación de metodologías ágiles para el desarrollo de software.* MQRInvestigar, 8(1), 5052-5074.

Mestres, A. F. (2018). *Introducción al frontend y backend.* Revista Openaccess.

Pérez, J. C., Carballeira, F. G., de Miguel Anasagasti, P., & Costoya, F. P. (2021). *Sistemas operativos*. McGraw-Hill Interamericana.

Postigo Palacios, A. (2023). *Administración de sistemas gestores de bases de datos*. Ediciones Paraninfo, S.A.

Pressman, R. S. (2013). *Ingeniería de Software un enfoque práctico.* Journal of Chemical Information and Modeling (7ma ed., Vol. 53). Mc Graw Hill. https://doi.org/10.1017/CBO9781107415324.004

Vásquez Vega, S. P. (2023). *Aplicación web de trazabilidad para la gestión logística de encomiendas de la empresa "ServiQuito"* (Doctoral dissertation, Pontificia Universidad Católica del Ecuador Ibarra).

Vivas Mena, J. B. *Propuesta de implementación del sistema Web de venta de boletos de viaje y gestión de encomiendas para la empresa Transportes Montero SAC Piura*; 2018.

García Sánchez, E., Vite Chávez, O., Navarrate Sánchez, M. Á., García Sánchez, M. Á., & Torres Cosío, V. (2016). Metodología para el desarrollo de software multimedia educativo MEDESME. CPU-e, Revista de Investigación Educativa, (23), 216-226.

Subecz, Z. (2021). Desarrollo web con el framework Laravel. *Gradus*, *8* (1), 211-218.

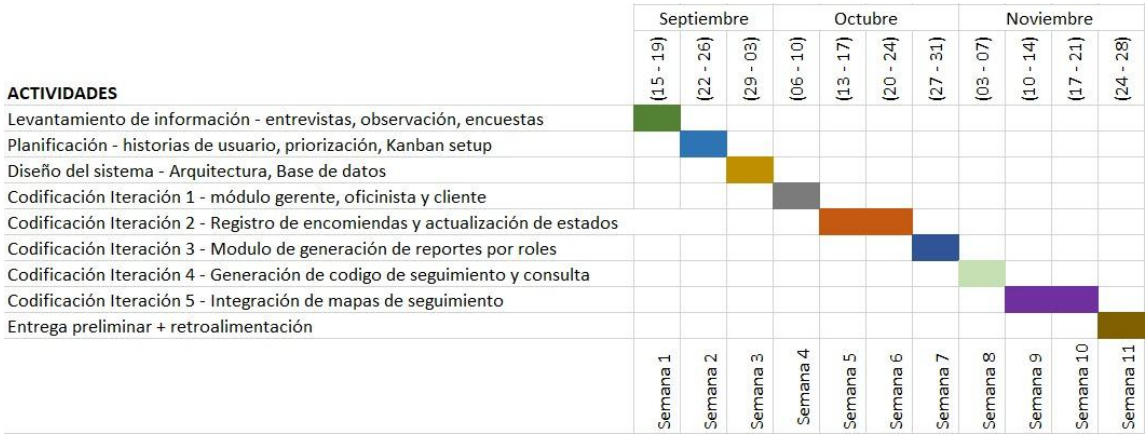
**ANEXOS**

# Anexo 1.

Cronograma (Gantt)

**Figura 27.**

*Diagrama de Gantt*




**Elaborado por:** Galeas L. & Cabezas R.

***Nota.*** Diagrama de seguimiento de actividades, organizadas por etapas.

# Anexo 2.

Presupuesto Ejecutado

**Tabla 33.**

*Tabla de presupuesto previsto para el desarrollo del proyecto.*

| Recursos | Subtotales |
|---|---|
| Internet | $20.00 |
| Impresiones | $5.00 |
| Pasajes | $7.00 |
| Computadora | $900.00 |
| Dominio | $30.00 |
| Teléfono | $150.00 |
| Subtotal | **$1,112.00** |
| Imprevisto 10% | **$111.20** |
| Total | **$1,223.20** |

***Nota.*** Presupuesto aproximado sobre los gastos a cubrir.

# Anexo 3.

Carta de aceptación de la organización donde se realizó el trabajo de integración curricular.

**Cooperativa de Transporte Flota Bolívar**

**Lic. Washington Allan Baño**

**Dir.** Av. Elisa Mariño de Carvajal frente al Terminal Terrestre, Guaranda, Ecúador.

---

**Guaranda, 1 de agosto del 2025**

Ing. Darwin Carrión Buenaño

**Coordinador de la Unidad de Integración Curricular**

**Carrera Software**

Con un cordial saludo y deseándole éxitos en sus funciones, solicito a usted autorice desarrollar el proyecto denominado Sistema Web para el Seguimiento de Encomiendas de la Cooperativa de Transporte Flota Bolívar, propuesto por la Srta. Galeas Armijo Leydi Marianela con CI: 0202321642 y el Sr. Cabezas Lechón Ronny Johel con C.I: 1727598888, certifico que se dio paso para el desarrolló del proyecto en la empresa en la cual represento y así poder apoyar en su proceso de titulación.

Agradezco su atención presente.

Saludos cordiales.



**Lic. Washington Allan Baño**

Gerente de la Cooperativa de Transporte Flota Bolívar

# Anexo 4.

Instrumentos de recopilación de datos (Entrevista).

**Universidad Estatal de Bolívar**

**Faculta de Ciencias Administrativas, Gestión Empresarial e Informática**

**Carrera de Ingeniería de Software**

**Proyecto de Titulación**

**Tema:** Sistema web para el seguimiento de encomiendas de la Cooperativa de Transporte Flota Bolívar.

**Objetivo:** Obtener información sobre los procesos actuales de envíos/recepción de encomiendas, problemas frecuentes y expectativas del nuevo sistema.

**Nombre del entrevistado:** Allan Baño Washington Rodrigo

**Cargo:** gerente

1) ¿Cómo se comunican actualmente las oficinas o sucursales al momento de coordinar una entrega o reclamo?

Se comunican mediante un grupo de WhatsApp donde están integrados todos los trabajadores oficinistas.

2) ¿Cómo gestionan actualmente el proceso de recepción y envío de encomiendas?

Actualmente se gestiona este proceso mediante hojas de Excel que son enviadas a sus correspondientes oficinas para que realicen el proceso de entrega del paquete al cliente destinatario.

3) ¿Qué sucede actualmente si se pierde la información en Excel o hay fallos en la comunicación por WhatsApp?

Cuando se presenta una pérdida de información en Excel o fallos en la comunicación por WhatsApp, se intenta recuperar los datos lo más rápido posible mediante la revisión de respaldos o consultando directamente con las oficinas involucradas. Esta situación puede generar retrasos en la entrega, confusión entre el personal y malestar en los clientes.

4) ¿Qué tipo de información es indispensable que se registre al recibir una encomienda?

Los datos del cliente que solicita el envío de una encomienda y del cliente que va a recibir la encomienda. También es indispensable registrar los detalles de la encomienda, para así saber qué tipo de paquete se está enviando.

5) ¿Cuáles son los tiempos promedio de entrega y registro de paquetes?

Tiene un tiempo aproximado de 10 minutos, contando desde cuando se recibe la encomienda, se pesa y se registra para su posterior envió.

6) ¿Qué problemas principales enfrentan con el sistema manual o semi automatizado que usan hoy?

El proceso actual semi automatizado, ha conllevado a confusiones de entregas de paquetes a los clientes destinatarios. Actualmente solo existe el sistema de cobro, pero el inconveniente es que el cliente no puede ver en donde está su encomienda.

7) ¿Existen limitaciones tecnológicas como internet, equipos, personal capacitado que debamos considerar?

Actualmente no existe limitaciones por el momento.

8) ¿Qué tan accesible es el sistema actual para el personal nuevo o en capacitación?

Un poco demoroso y poco eficaz, debido a que se demora en el proceso de registro y recepción de cada encomienda que llega a la oficina.

9) ¿Qué esperan lograr con la implementación del nuevo sistema?

Que los tiempos de registro y entrega de una encomienda a sus respectivos clientes sea mucho más rápido y fácil de realizar. También que el cliente pueda ver el estado de su encomienda, desde que salió de la oficina donde se registró hasta la oficina destino.

10) ¿Qué tipos de usuarios considera que debería estar constando en el nuevo sistema?

Principalmente el gerente, que sería el que sube la información al sistema y un módulo para los oficinistas, quienes deberían registrar la encomienda dentro del sistema y quien registra la entrega al cliente destinatario.

11) ¿Considera necesario que el nuevo sistema incorpore un módulo de reportes que permita generar consultas y estadísticas de las encomiendas registradas por cada oficinista, en un periodo determinado?

Si considero necesario, debido a que ahí se puede tener un control de cuantas encomiendas han sido registradas durante un periodo determinado por cada oficinista.

12) ¿Qué tipo de reportes le gustaría obtener del sistema?

Por el lado del gerente, un reporte general por todas las oficinas, y un reporte por oficina en específico. Por el lado del módulo del oficinista, sería un reporte general y uno por cada ruta.

13) ¿Cuál sería el impacto positivo esperado en la cooperativa si se implementa un sistema web?

Un impacto positivo seria reducir los tiempos de espera que realiza el cliente por cada proceso, así como la agilización de los procesos en todas las oficinas. Además, permitiría un mejor control y seguimiento de cada envío, mejorando el proceso de registro y entrega.

# Anexo 5.

Instrumentos de recopilación de datos (ficha de observación).

**Sistema Web para el seguimiento de encomiendas de la Cooperativa de Transporte Flota Bolívar**

**Datos generales**

| | | | |
|---|---|---|---|
| **Lugar de observación** | Oficina de la cooperativa agencia Guaranda. | | |
| **Empresa** | Cooperativa de Transporte Flota Bolívar. | | |
| **Área observada** | Área de encomiendas Flota Bolívar. | | |
| **Fecha** | 21/02/2026 | | |
| **Hora de inicio** | 09:30 am | **Hora de finalización** | 12:00 pm |
| **Observadores** | Ronny Cabezas | | |

**Objetivo general:**

- Observar y analizar el proceso de registro, despacho, recepción y entrega de encomiendas en la Cooperativa de Transporte Flota Bolívar, con el fin de identificar mejoras en el sistema de seguimiento de encomiendas.

**Objetivos específicos:**

- Identificar las actividades realizadas en cada etapa del proceso.
- Determinar los tiempos aproximados de cada procedimiento.
- Detectar posibles fallas o retrasos en el proceso.

**Actores que intervienen**

| Actor | Función | Observaciones |
|---|---|---|
| Cliente remitente | Entrega el paquete y proporciona datos. | Provee información de destino y realiza el pago. |
| Oficinistas (origen) | Pesaje y registro en el cuaderno de apuntes. | Registra datos en el libro de apuntes, cobro de envió. |
| Chofer | Recibe y Transporta las encomiendas. | Responsable del traslado físico de la encomienda. |
| Oficinista (destino) | Recepción de encomienda y entrega al remitente. | Actualiza estados y entrega el paquete al cliente destinatario. |

| | | |
|---|---|---|
| Cliente destinatario | Retiro de encomienda. | Solicita entrega, presenta documento de identidad y firma recibo de entrega. |

**Descripción del proceso de recepción**

| N. | Actividad | Responsable | Herramienta utilizada | Tiempo estimado | Observación |
|---|---|---|---|---|---|
| 1 | Recepción y pesaje de encomienda (pesaje en kg). | Cliente y oficinista origen | Báscula | 3 min | Se solicitan datos del remitente y destinatario, de forma verbal. |
| 2 | Cobro según el peso del paquete. | Cliente y oficinista origen | Recibos de pago | 1 min | El registro manual de datos puede generar lentitud si hay un flujo alto de clientes. |
| 3 | Despacho y traslado a la bodega del bus. | Oficinista y chofer | Manual | 9 min | La unidad de transporte recepta los paquetes físicamente y se llena una guía de envío. |

**Conclusiones:**

- En el proceso de registro y recepción de la encomienda conllevo un tiempo aproximado de 4 minutos. Aquí se identificó que la fluidez depende de la digitalización manal de datos del remitente, destinatario y detalles del paquete. Las herramientas utilizadas como la báscula fueron esencial para evitar atrasos en el registro de envío. El sistema guarda a los clientes que anteriormente hayan

sido registrados, esto con el fin de minimizar los tiempos de registros de datos y agilizar los procesos de recepción de encomiendas.

- Dentro del proceso de despacho de la encomienda se requirió un tiempo aproximado de 9 minutos para el traslado físico del paquete hacia la bodega del bus.

**Datos generales**

| | | | |
|---|---|---|---|
| **Lugar de observación** | Oficina de la cooperativa agencia San Miguel. | | |
| **Empresa** | Cooperativa de Transporte Flota Bolívar. | | |
| **Área observada** | Área de encomiendas Flota Bolívar. | | |
| **Fecha** | 21/02/2026 | | |
| **Hora de inicio** | 10:39 am | **Hora de finalización** | 14:25 pm |
| **Observadores** | Leydi Galeas | | |

**Descripción del proceso de entrega al cliente**

| N. | Actividad | Responsable | Herramienta utilizada | Tiempo estimado | Observación |
|---|---|---|---|---|---|
| 1 | Recepción física del paquete del bus. | Chofer y oficinista destino | Manual | 5 min | Ingreso físico del paquete a la oficina destino. |
| 2 | Verificación de llegada y organización. | Oficinista destino | Guía de envió. | 4 min | Se valida que el paquete físico coincida con la guía. |
| 3 | Verificación de identidad y firma | Cliente y oficinista destinatario | Cedula y recibo físico | 2 min | El cliente debe presentar documento original para la entrega. |
| 4 | Registro de entrega y archivo. | Oficinista destino | Cámara/ Archivador para recibos | 2 min | Se toma evidencia física y fotográfica de la identificación |

| | |
|---|---|
| | y recibo firmado. |

**Conclusiones:**

- El proceso de recepción en la oficina destino San Miguel conlleva un tiempo aproximado de 9 minutos. Esta etapa es crítica, ya que este proceso permite saber al cliente destinatario que ya llego al destino su encomienda.
- La entrega del paquete al cliente destinatario dura un tiempo aproximado de 4 minutos, el oficinista procedió a la captura fotográfica de documento como cedula de identidad y recibo firmado por el cliente, para subirlos al sistema y posteriormente ser guardados en el sistema con la nueva actualización de “Entregado al cliente”.

# Anexo 6.

Ficha de Observación de casos de prueba

**Tabla 34.**

*Ficha de observación de casos de prueba.*

| **Ficha de observación** | | | | |
|---|---|---|---|---|
| **Involucrados:** | Ronny Cabezas y Leydi Galeas | | | |
| **Fecha:** | 06/01/2026 | **Hora:** | 11:00 | |
| **Propósito:** | Describir el impacto de las pruebas realizadas en el sistema de seguimiento de encomiendas de la Cooperativa de Transporte Flota Bolívar. | | | |
| 1 = No cumple | 2 = Cumple parcialmente | 3 = Cumple totalmente | | |
| **N°** | **Características** | **Métricas internas** | **Valoración promedio** | **Porcentaje** |
| 1 | Funcionalidad | Su funcionalidad es correcta. | 2.7 | 90.00% |
| 2 | Rendimiento | Su rendimiento es adecuado. | 2.8 | 93.33% |
| 3 | Seguridad | Cumple con las métricas de seguridad. | 2.7 | 90.00% |
| 4 | Usabilidad | Cumple con el nivel de satisfacción y facilidad de uso. | 2.9 | 96.66% |
| 5 | Integración | Sus componentes de integran al flujo adecuado. | 2.8 | 93.33% |
| **PROMEDIO TOTAL** | | | **2.78** | **92.66%** |

***Nota.*** Ficha de observación de todos los puntos establecidos evaluados del sistema.

# Anexo 7.

Certificado de conformidad por parte de la organización.

**Cooperativa de Transporte Flota Bolívar**

**Lic. Washington Allan Baño**

**Dir.** Av. Elisa Mariño de Carvajal frente al Terminal Terrestre, Guaranda, Ecuador.

---

Guaranda, 23 de febrero de 2026

A petición de los interesados, certifico que:

De acuerdo con lo solicitado, certifico que el señor Ronny Cabezas y la señorita Leydi Marianela Galeas Armijo, egresados de la carrera de Software de la Universidad Estatal de Bolívar, realizaron la socialización del proyecto de "SISTEMA WEB PARA EL SEGUIMIENTO DE ENCOMIENDAS DE LA COOPERATIVA DE TRANSPORTE FLOTA BOLÍVAR", correspondiente al proyecto de integración curricular. En mi calidad de gerente de la cooperative de buses 'Flota Bolívar', me encuentro conforme con lo presentado por los responsables.

Es todo cuanto puedo mencionar en honor a la verdad, pudiendo los interesados hacer uso del presente documento como consideren pertinente.

Agradezco la atención prestada.

Saludos cordiales,

WASHINGTON RODRIGO ALLAN BAÑO

Firmado digitalmente por WASHINGTON RODRIGO ALLAN BAÑO
Fecha: 2026.02.23 12:04:45 -05'00'

---

**Li. Washington Allan Baño**

Gerente de la Cooperativa de Transporte Flota Bolívar

# Anexo 8.

Manual de Usuario




# Manual de Usuario del Sistema de Encomiendas

## Flota Bolívar

Autores:

Leydi Marianela Galeas Armijo

Ronny Johel Cabezas Lechón

Versión: 01.01

Fecha:15/01/2026

# Índice

Objeto del documento ..... 24
Elaborado ..... 24
Objetivo ..... 24
Requerimientos ..... 24
Manual de usuario ..... 25
1. Pantalla Inicial ..... 25
2. Iniciar sesión ..... 25
Registro de usuario ..... 27
3. Usuario con Rol de Gerente ..... 28
Presentación de la pantalla principal “Panel de Control” ..... 28
Gestión de usuarios ..... 29
Gestión de Oficinas ..... 31
Gestión de Rutas ..... 33
Reportes ..... 35
4. Usuario con Rol de Oficinista ..... 37
Presentación de la pantalla principal “Panel de Control” ..... 37
Gestión de Encomiendas ..... 38
Filtros ..... 40
Rastreo de Encomiendas ..... 43

## Índice de figuras

Figura 1. *Página Principal* ..... 25
Figura 2. *Página de Inicio de Sesión* ..... 26
Figura 3. *Mensaje cuando falta llenar un campo* ..... 27
Figura 4. *Pantalla de registro de un nuevo usuario* ..... 28
Figura 5. *Pantalla principal Panel de Control* ..... 29
Figura 6. *Pantalla Gestión de Usuarios* ..... 30
Figura 7. *Pantalla Eliminar Usuarios* ..... 31
Figura 8. *Pantalla para la Gestión de Oficinas* ..... 32
Figura 9. *Pantalla para Agregar una Nueva Oficina* ..... 33
Figura 10. *Pantalla Gestión de Rutas* ..... 34
Figura 11. *Pantalla para Crear una Nueva Ruta* ..... 35
Figura 12. *Pantalla para Visualizar Reportes* ..... 36
Figura 13. *Pantalla para Visualizar Reporte de Encomiendas* ..... 37
Figura 14. *Pantalla principal Panel de Control del Oficinista* ..... 38
Figura 15. *Pantalla Gestión de Encomiendas* ..... 39
Figura 16. *Pantalla de Registro de una Nueva Encomienda* ..... 40
Figura 17. *Pantalla de los Filtros de la Gestión de Encomienda* ..... 41
Figura 18. *Pantalla de los Detalles de la Encomienda* ..... 42
Figura 19. *Pantalla para Editar la Encomienda* ..... 43
Figura 20. *Pantalla de Rastreo de Encomiendas* ..... 44
Figura 21. *Pantalla de los Detalles del Envió* ..... 45
Figura 22. *Pantalla Historial de Movimientos* ..... 46

**Hoja de control de modificaciones**

| | |
|---|---|
| **Titulo** | Manual de usuario del sistema |
| **Versión** | 01.01 |
| **Realizado** | Leydi Marianela Galeas Armijo<br>Ronny Johel Cabezas Lechón |
| **Fecha** | 06/01/2026 |

| **CONTROL DE VERSIONES** | | |
|---|---|---|
| Versión | Descripción/Motivo versión | Fecha de presentación |
| 01.00 | Documento inicial | |
| 01.01 | Documento corregido | |
| | | |

## Objeto del documento

El presente documento tiene como propósito principal servir de guía práctica para los usuarios del sistema para el seguimiento de encomiendas. Está diseñado para brindar instrucciones claras y detalladas sobre como operar de manera eficiente las diferentes funcionalidades con las que cuenta el sistema.

## Elaborado

| | |
|---|---|
| **Participante** | Leydi Marianela Galeas Armijo<br>Ronny Johel Cabezas Lechón |
| **Área/Departamento** | Carrera de Software-UEB |
| **Teléfono** | 0982828932 / 0983603497 |
| **Rol** | Desarrolladores |

## Objetivo

Proporcionar a los usuarios una guía completa y fácil en la navegación dentro del sistema de seguimiento de encomiendas.

## Requerimientos

Los requerimientos mínimos para el sistema web de seguimiento de encomiendas funcione correctamente, son las siguientes:

1. Computador o laptop con conexión a internet
2. Navegadores web Firefox, Chrome, Edge
3. Cuenta de Usuario creada por el Administrador

**Manual de usuario**

1. **Pantalla Inicial**

Digitar en el navegador la dirección del dominio https://ctflotabolivar.com se presentará una pantalla como la que se muestra en la **Figura 1.** Aquí se puede navegar utilizando el menú presentado en la parte superior de la pantalla o con los accesos rápidos mostrados en la parte central de la pantalla.

Solo podrán navegar en esta página los usuarios que el administrador del sistema haya creado sus credenciales.

**Figura 1**

*Página Principal*




*Figura 4 Página Principal*

**Elaborado por:** Galeas L. & Cabezas R.

**Fuente:** https://ctflotabolivar.com

2. **Iniciar sesión**

La opción ***iniciar sesión*** se encuentra en la parte superior derecha, habilitada para los usuarios que se encuentran registrados en el Sistema, los clientes crearan su acceso en la opción de ***Registrarse*** identificando sus datos personales como su usuario y contraseña.

El usuario debe ingresar su correo y contraseña con los que creo su cuenta.

1. Si da clic en **INICIAR SESIÓN** se redirige a otra pantalla según el rol del usuario.
2. Si da clic en el botón **CANCELAR** el usuario se redirige a la página principal "Fig. 1"

**Figura 2**

*Página de Inicio de Sesión*




*Figura 5 Página de Inicio de Sesión*

**Elaborado por:** Galeas L. & Cabezas R.

**Fuente:** https://ctflotabolivar.com

Si el usuario da clic en **INICIAR SESIÓN** sin llenar algún campo, la ventana mostrara el siguiente mensaje “Completa este campo” en la sección que está vacía.

**Figura 3**

*Mensaje cuando falta llenar un campo*




*Figura 6 Mensaje cuando falta llenar un campo*

**Elaborado por:** Galeas L. & Cabezas R.

**Fuente:** https://ctflotabolivar.com

Si los datos ingresados son incorrectos la pantalla muestra un mensaje de **"Error al iniciar sesión revise sus datos".**

Si los datos son correctos, el sistema redireccionara a los diferentes apartados con los que cuenta la aplicación web, se muestran según el tipo de rol de cada usuario.

**Registro de usuario**

Al ingresar al sitio un nuevo usuario necesitara registrarse en el sistema para poder tener acceso, para esto necesitamos que se llene de manera adecuada el formulario que su contraseña sea mayor a 8 caracteres.

**Figura 4**

*Pantalla de registro de un nuevo usuario*

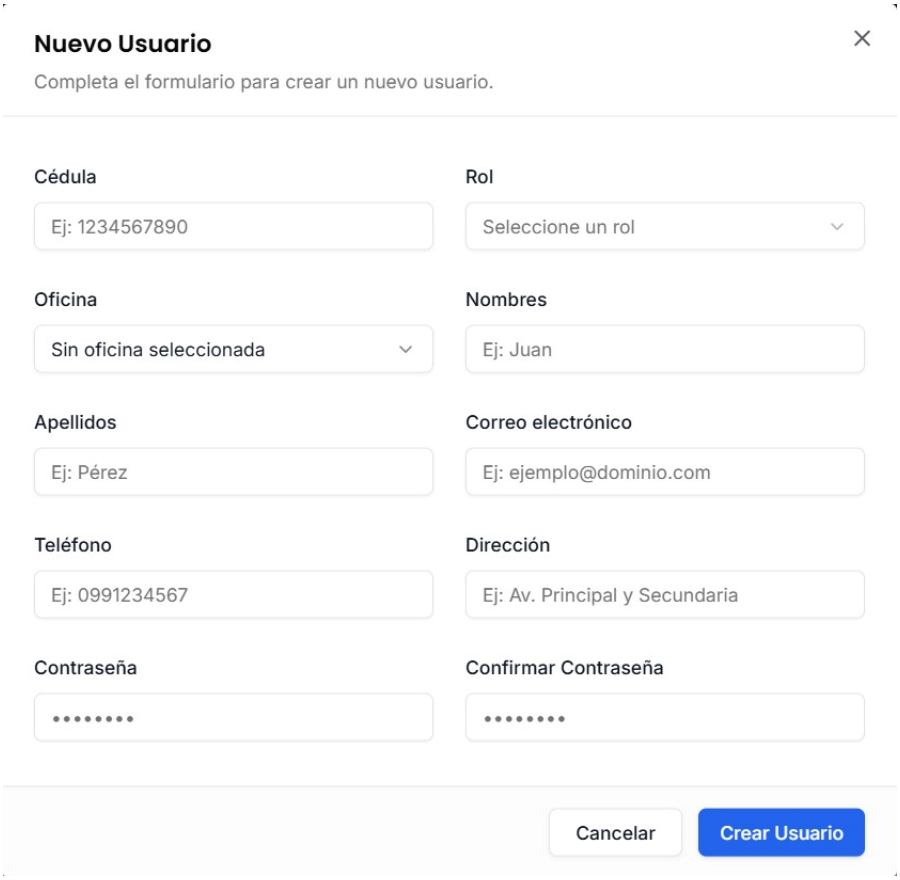



*Figura 7 Pantalla de registro de un nuevo usuario*

**Elaborado por:** Galeas L. & Cabezas R.

**Fuente:** https://ctflotabolivar.com

3. **Usuario con Rol de Gerente**

**Presentación de la pantalla principal "Panel de Control"**

Cuando el usuario gerente ingrese su correo y contraseña se presentará la siguiente pantalla en donde puede registrar usuarios, oficinas, rutas y visualizar reportes.

**Figura 5**

*Pantalla principal Panel de Control*

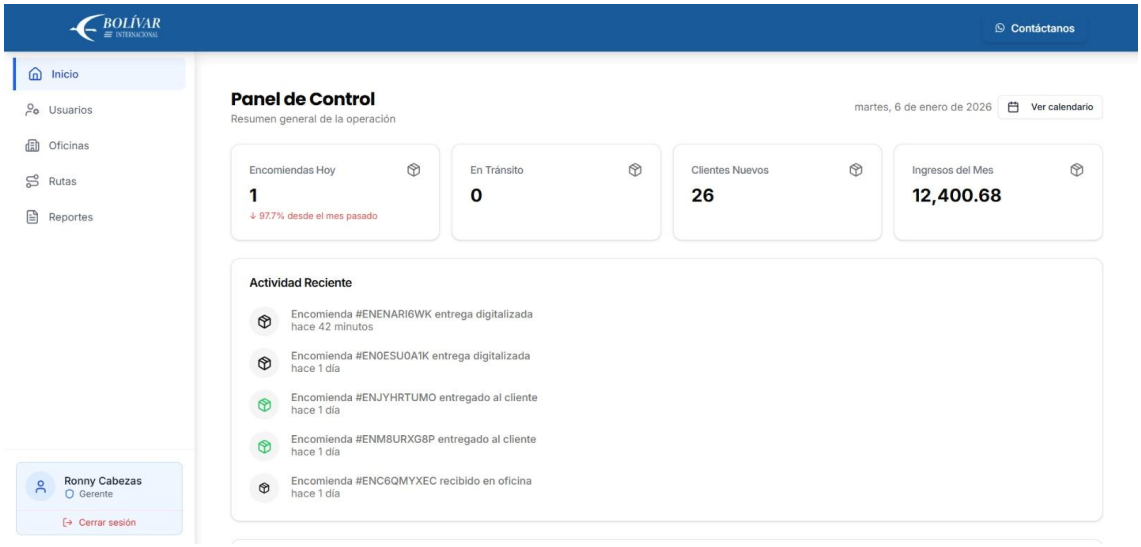



*Figura 8 Pantalla principal Panel de Control*

**Elaborado por:** Galeas L. & Cabezas R.

**Fuente:** https://ctflotabolivar.com

El gerente puede acceder a las opciones habilitadas usando el menú que se muestra en el lado izquierdo de la pantalla o los accesos rápidos que se encuentran en la pantalla principal mostrando un dashboard con información como la cantidad de usuarios, las oficinas, las rutas, los reportes, también cuenta con un resumen general de la operación para identificar la cantidad de encomiendas del día, se puede visualizar cuando las encomiendas están en tránsito y los clientes nuevos.

### Gestión de usuarios

En esta sección se puede acceder a la opción nuevo usuario para poder registrar un nuevo usuario al sistema como también se puede eliminar a un usuario ya existente.

**Figura 6**

*Pantalla Gestión de Usuarios*

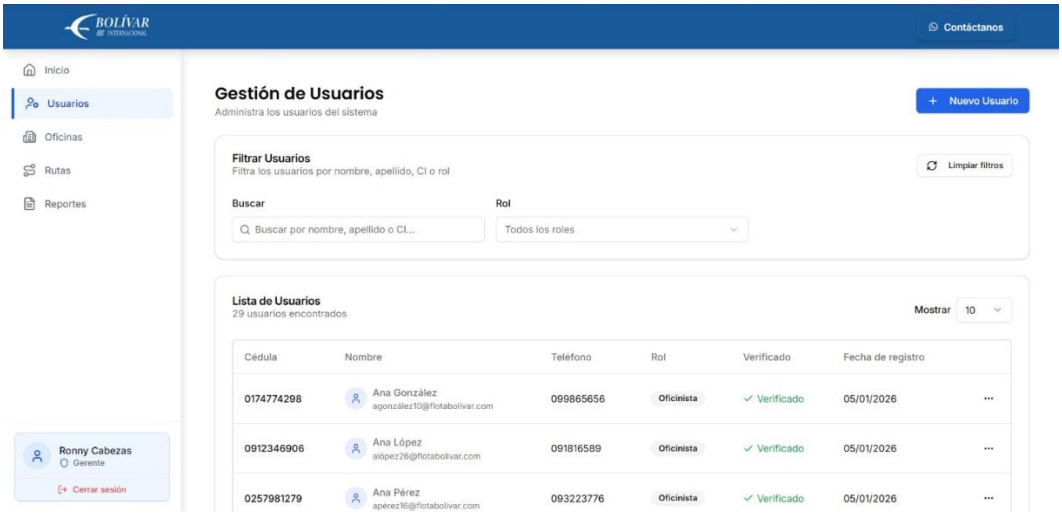



*Figura 9 Pantalla Gestión de Usuarios*

**Elaborado por:** Galeas L. & Cabezas R.

**Fuente:** https://ctflotabolivar.com

**Figura 7**

*Pantalla Eliminar Usuarios*

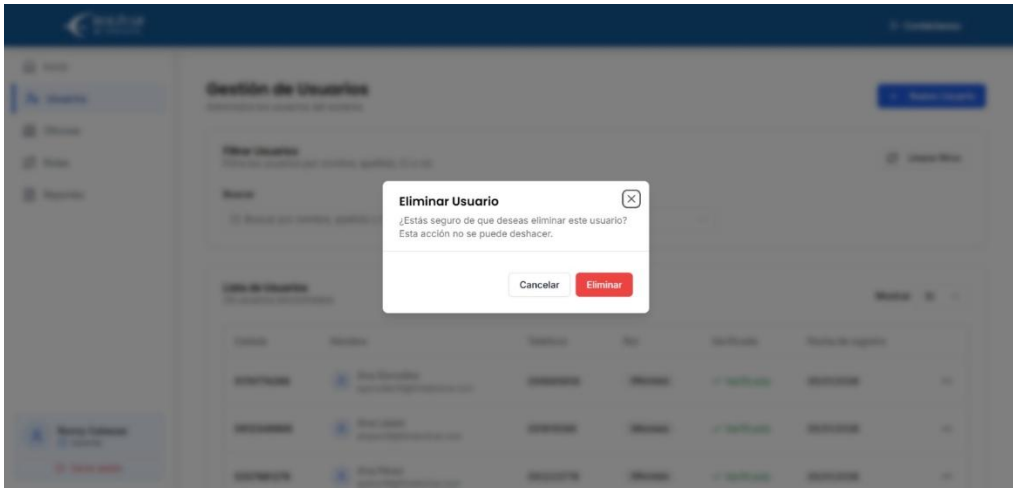



*Figura 10 Pantalla Eliminar Usuarios*

**Elaborado por:** Galeas L. & Cabezas R.

**Fuente:** https://ctflotabolivar.com

**Gestión de Oficinas**

En esta sección se puede acceder a la opción nueva oficina para poder registrar una nueva oficina completando el formulario.

**Figura 8**

*Pantalla para la Gestión de Oficinas*

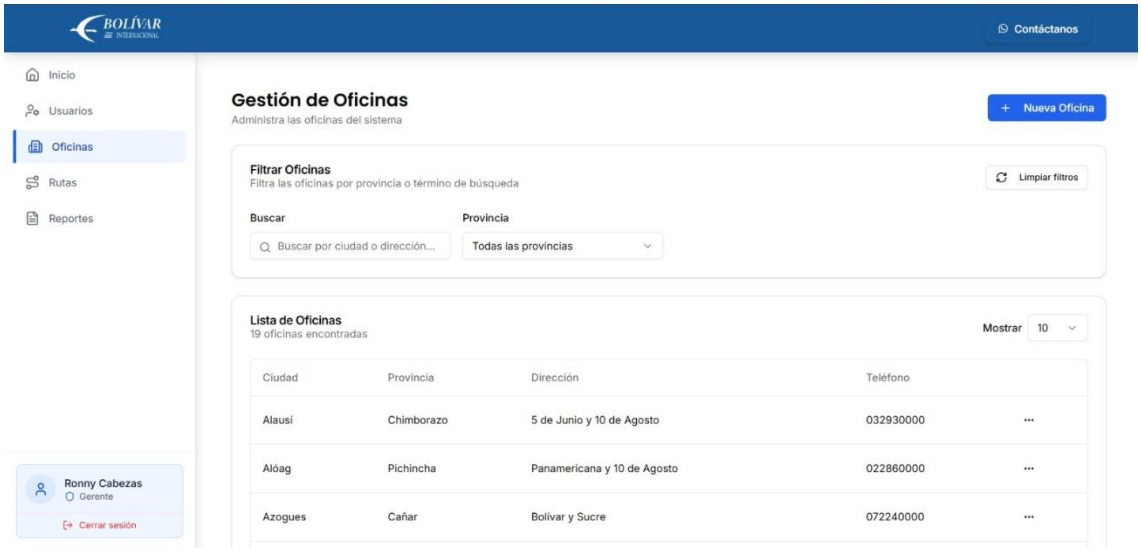



*Figura 11 Pantalla para la Gestión de Oficinas*

**Elaborado por:** Galeas L. & Cabezas R.

**Fuente:** https://ctflotabolivar.com

**Figura 9**

*Pantalla para Agregar una Nueva Oficina*

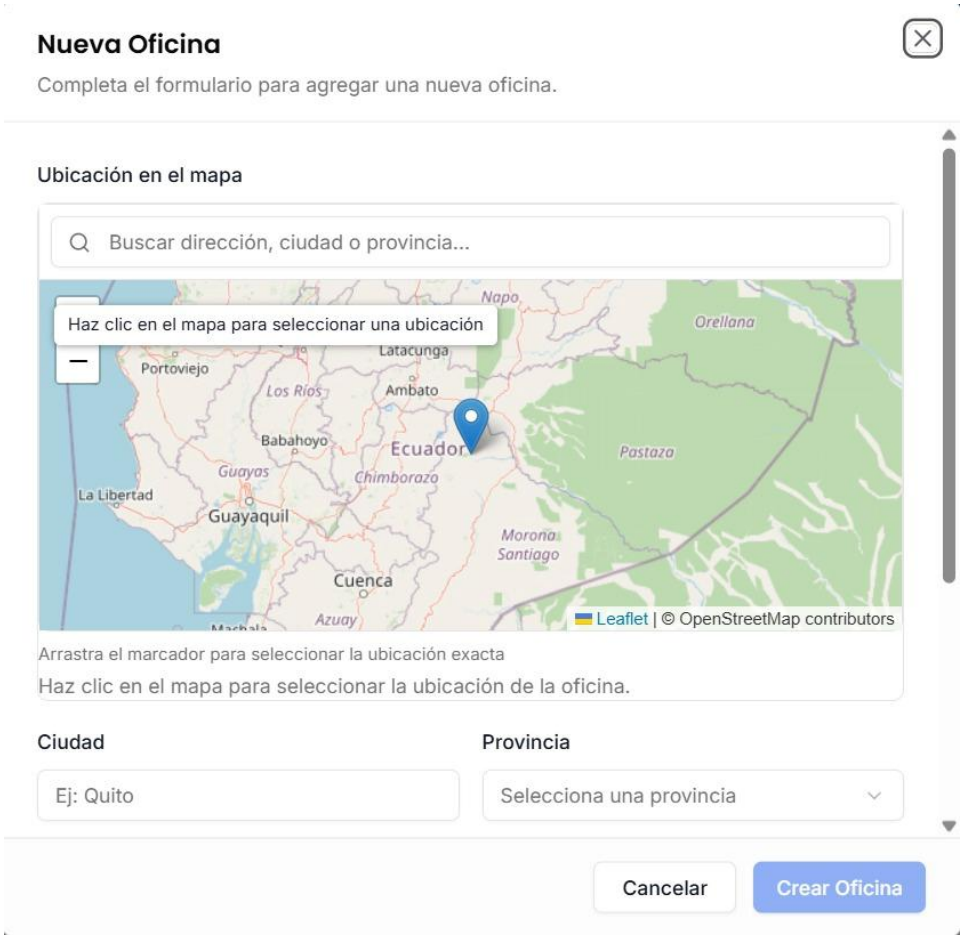



*Figura 12 Pantalla para Agregar una Nueva Oficina*

**Elaborado por:** Galeas L. & Cabezas R.

**Fuente:** https://ctflotabolivar.com

**Gestión de Rutas**

En esta sección se puede acceder a la opción nuevo usuario para poder registrar una nueva ruta al sistema llenando el formulario.

**Figura 10**

*Pantalla Gestión de Rutas*

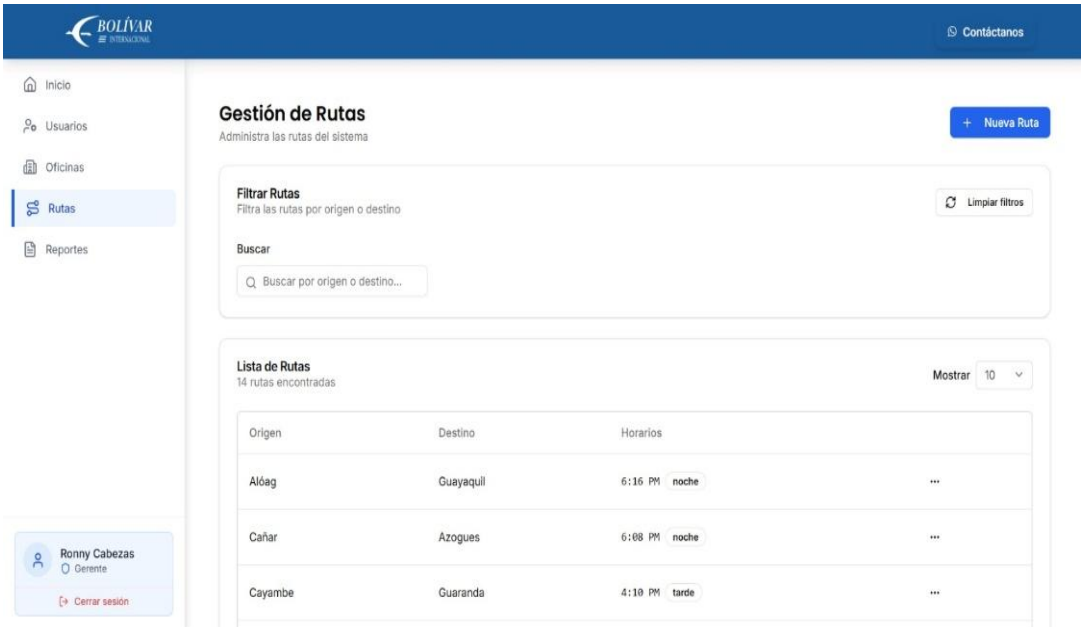



*Figura 13 Pantalla Gestión de Rutas*

**Elaborado por:** Galeas L. & Cabezas R.

**Fuente:** https://ctflotabolivar.com

**Figura 11**

*Pantalla para Crear una Nueva Ruta*

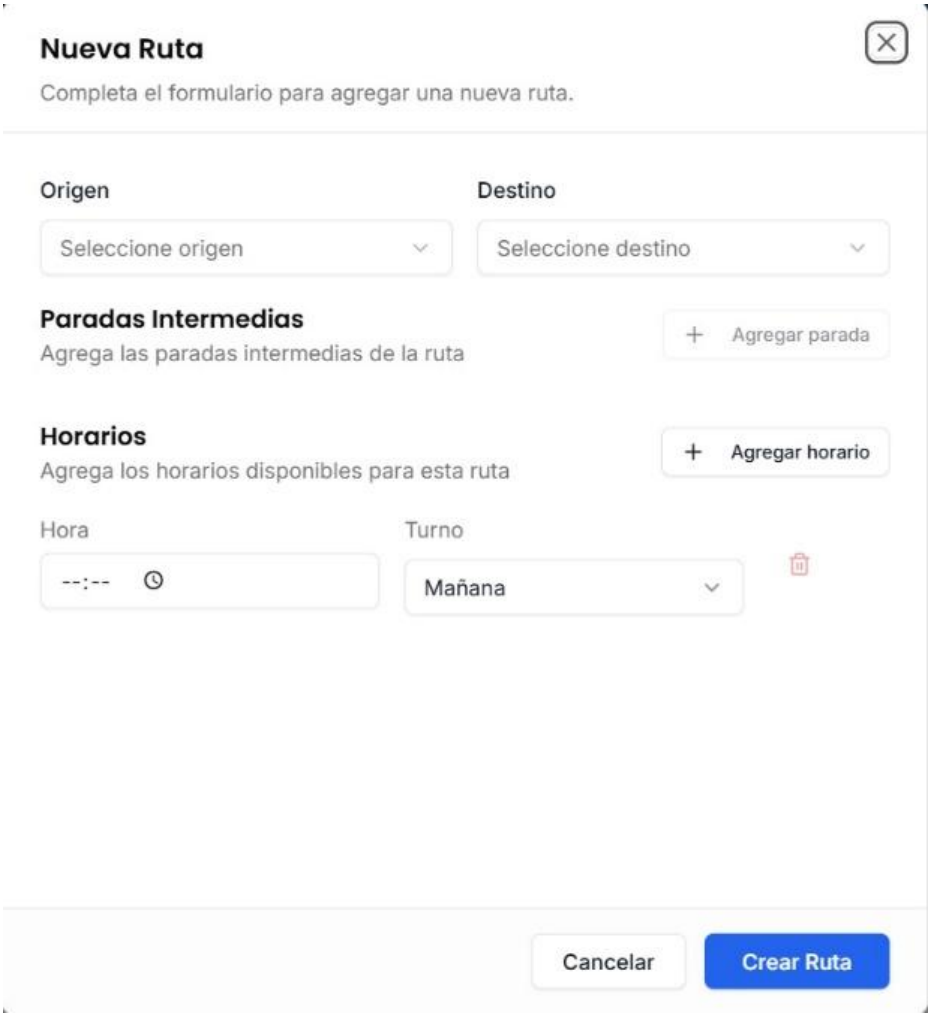



*Figura 14 Pantalla para Crear una Nueva Ruta*

**Elaborado por:** Galeas L. & Cabezas R.

**Fuente:** https://ctflotabolivar.com

**Reportes**

El administrador puede acceder a las opciones habilitadas usando el menú que se muestra en el lado izquierdo de la pantalla o los accesos rápidos que se encuentran en la pantalla principal mostrando un dashboard con información y podrá ingresar a la opción de reportes donde podrá generar informes detallados por periodo.

**Figura 12**

*Pantalla para Visualizar Reportes*

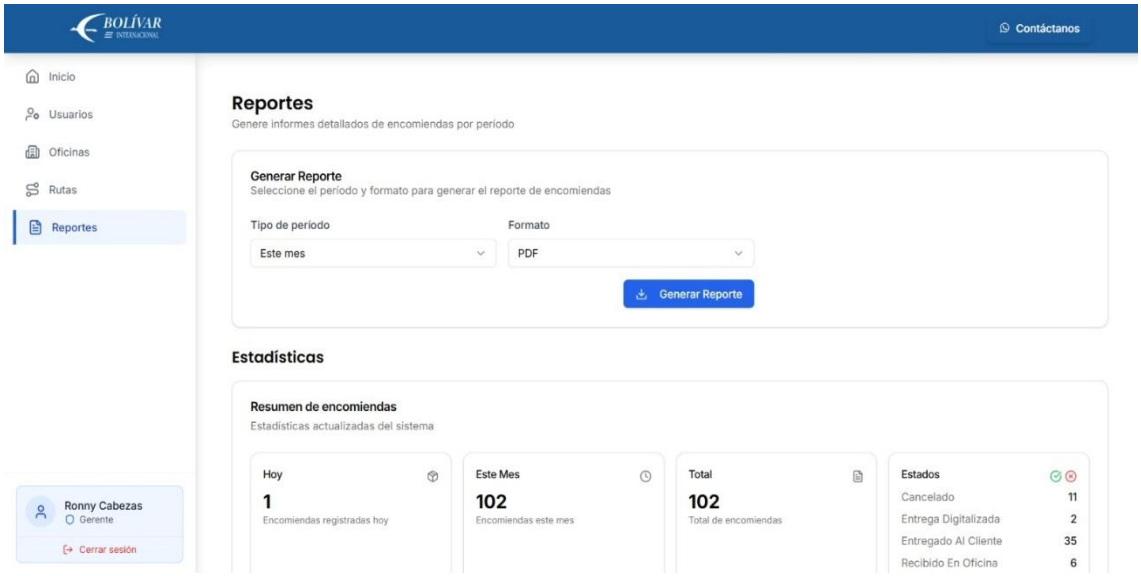



*Figura 15 Pantalla para Visualizar Reportes*

**Elaborado por:** Galeas L. & Cabezas R.

**Fuente:** https://ctflotabolivar.com

**Figura 13**

*Pantalla para Visualizar Reporte de Encomiendas*

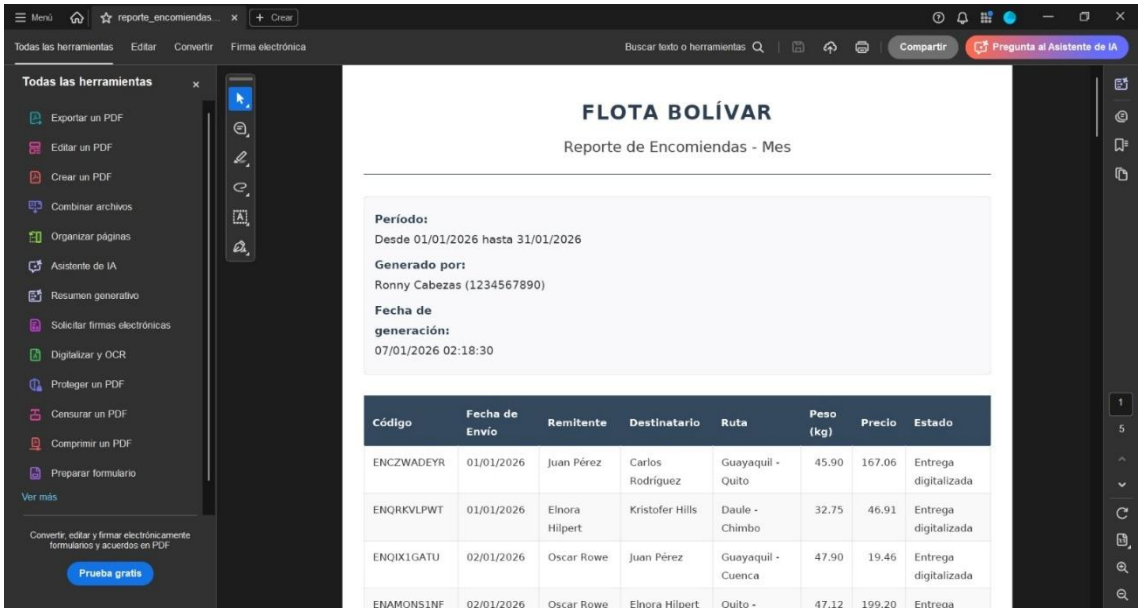



*Figura 16 Pantalla para Visualizar Reporte de Encomiendas*

**Elaborado por:** Galeas L. & Cabezas R.

**Fuente:** https://ctflotabolivar.com

4. **Usuario con Rol de Oficinista**

**Presentación de la pantalla principal "Panel de Control"**

Cuando el usuario administrador ingrese su correo y contraseña se presentará la siguiente pantalla en donde puede registrar encomiendas y reportes.

**Figura 14**

*Pantalla principal Panel de Control del Oficinista*

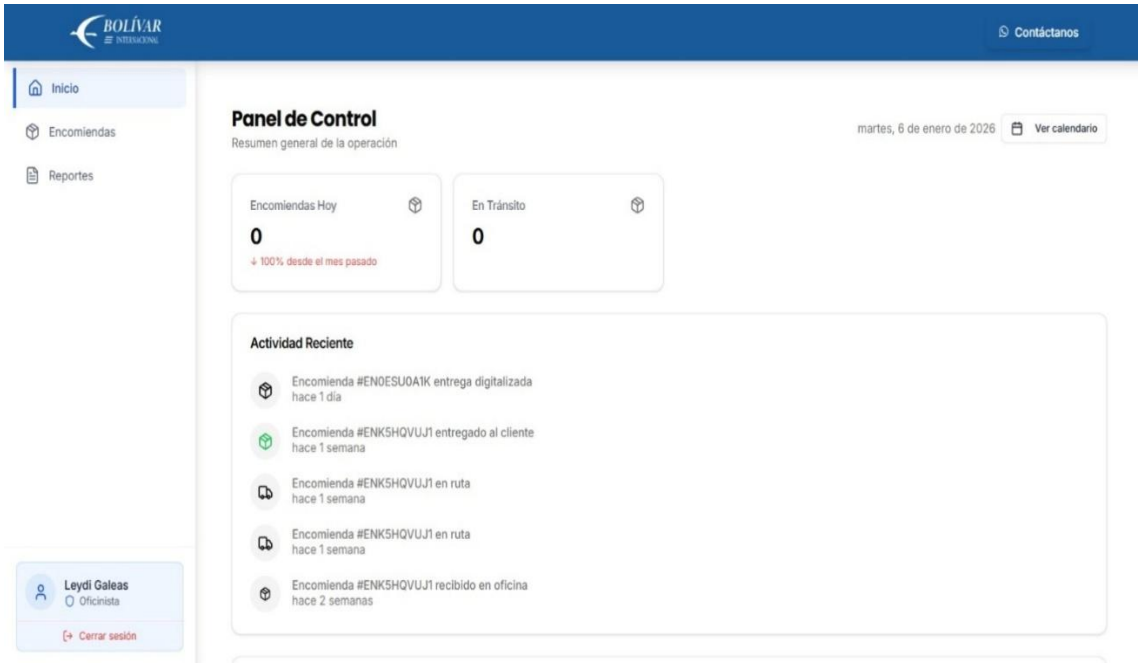



*Figura 17 Pantalla principal Panel de Control del Oficinista*

**Elaborado por:** Galeas L. & Cabezas R.

**Fuente:** https://ctflotabolivar.com

El oficinista puede acceder a las opciones habilitadas usando el menú que se muestra en el lado izquierdo de la pantalla o los accesos rápidos que se encuentran en la pantalla principal mostrando un dashboard con información como las encomiendas y los reportes, también cuenta con un resumen general de la operación para identificar la cantidad de encomiendas del día, se puede visualizar cuando las encomiendas están en tránsito.

**Gestión de Encomiendas**

En esta sección se puede administrar las encomiendas del sistema, al acceder a la opción nueva encomienda se puede registrar una nueva encomienda al sistema como también se puede filtrar la búsqueda de las encomiendas a través del código de seguimiento o estado, y se puede visualizar la lista de las encomiendas.

**Figura 15**

*Pantalla Gestión de Encomiendas*

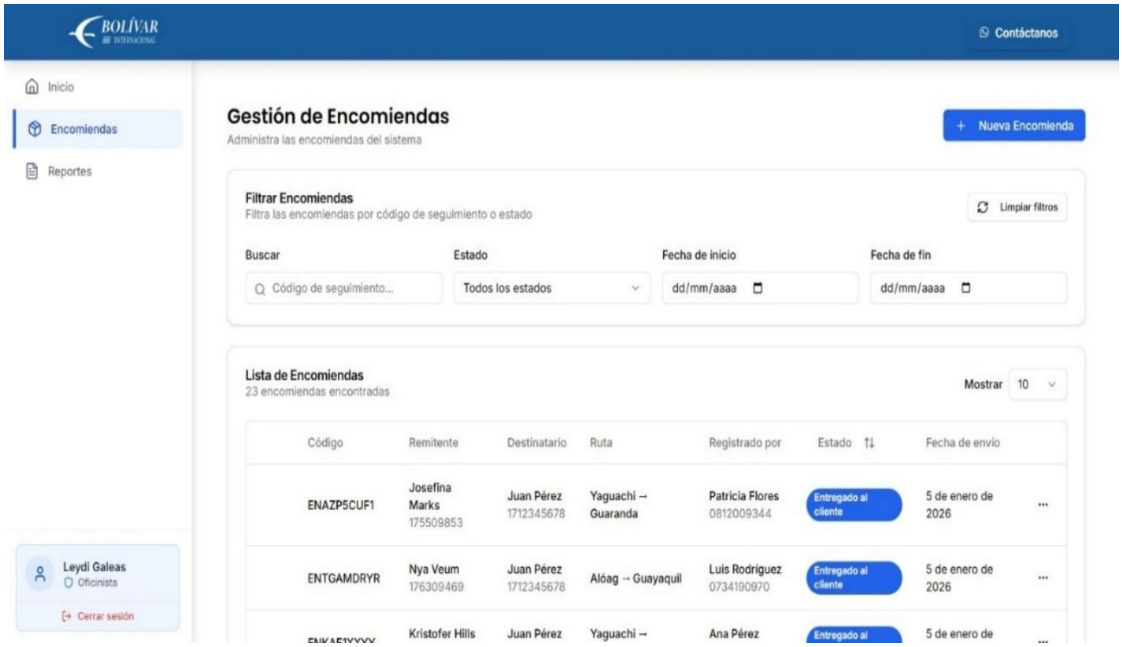



*Figura 18 Pantalla Gestión de Encomiendas*

**Elaborado por:** Galeas L. & Cabezas R.

**Fuente:** https://ctflotabolivar.com

**Figura 16**

*Pantalla de Registro de una Nueva Encomienda*

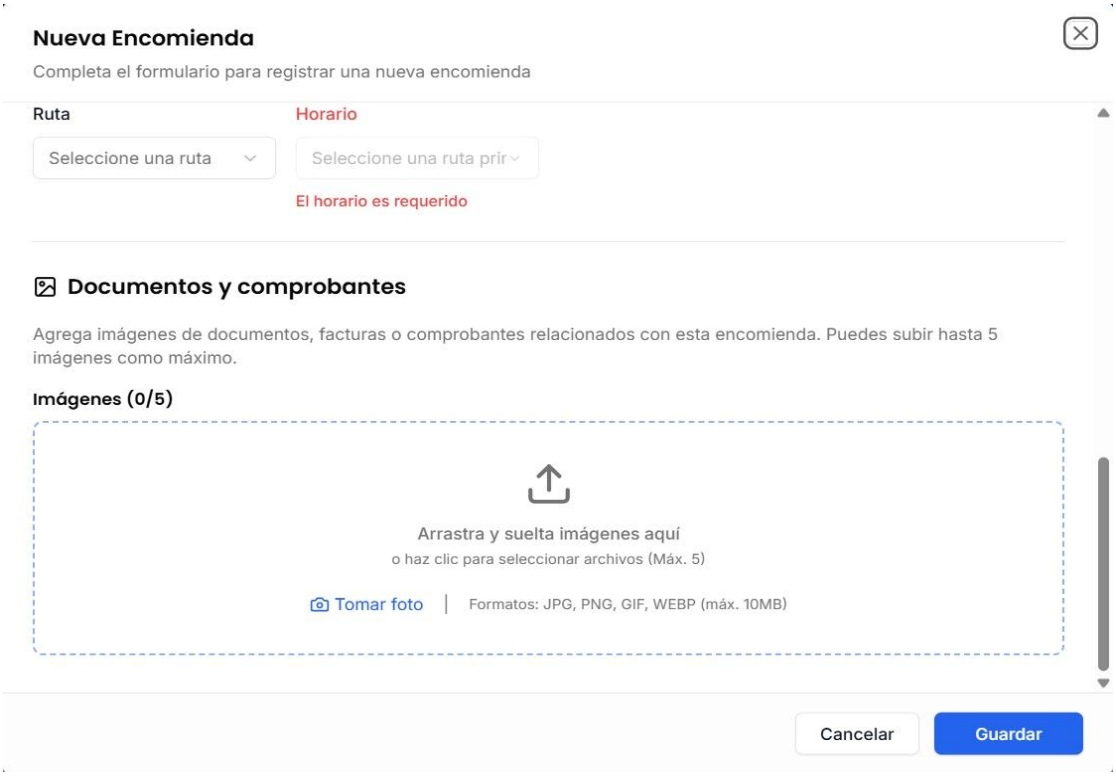



*Figura 19 Pantalla de Registro de una Nueva Encomienda*

**Elaborado por:** Galeas L. & Cabezas R.

**Fuente:** https://ctflotabolivar.com

**Filtros**

En esta sección se puede ver los detalles de la encomienda como la información del envió, información del pago, la ruta, información del remitente y destinatario.

**Figura 17**

*Pantalla de los Filtros de la Gestión de Encomienda*

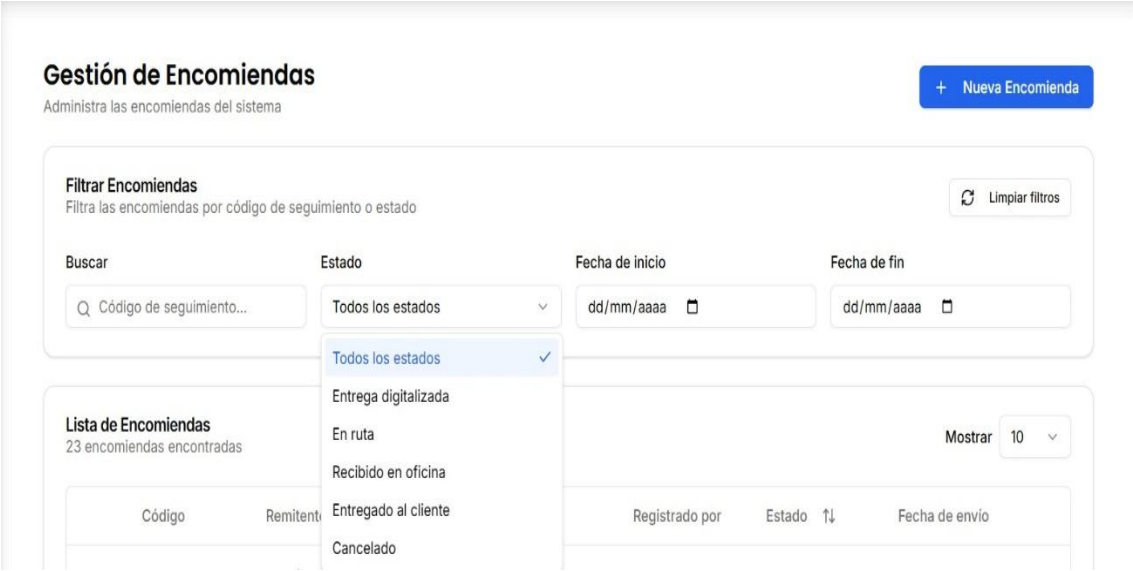



*Figura 20 Pantalla de los Filtros de la Gestión de Encomienda*

**Elaborado por:** Galeas L. & Cabezas R.

**Fuente:** https://ctflotabolivar.com

**Figura 18**

*Pantalla de los Detalles de la Encomienda*

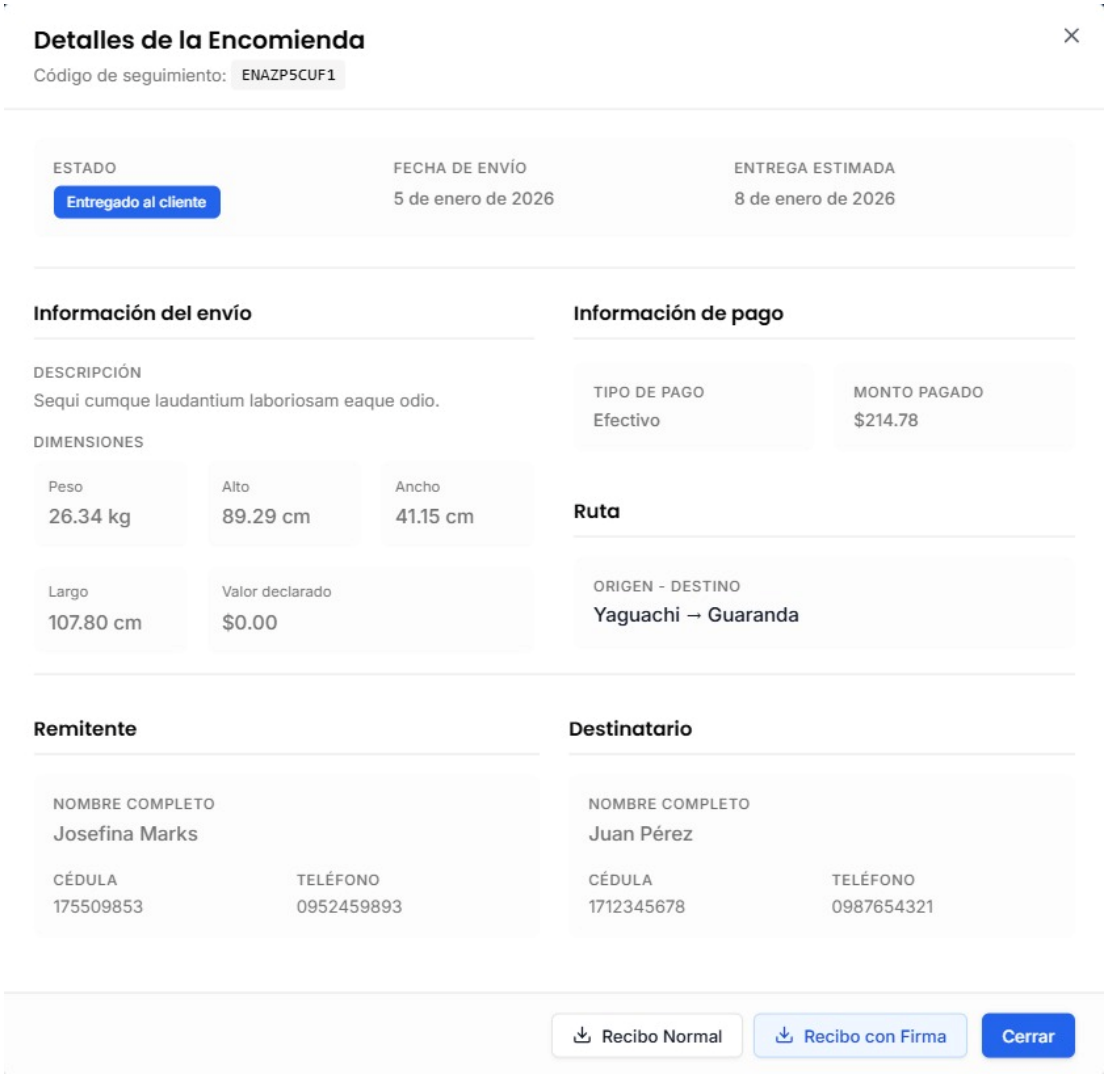



*Figura 21 Pantalla de los Detalles de la Encomienda*

**Elaborado por:** Galeas L. & Cabezas R.

**Fuente:** https://ctflotabolivar.com

**Figura 19**

*Pantalla para Editar la Encomienda*

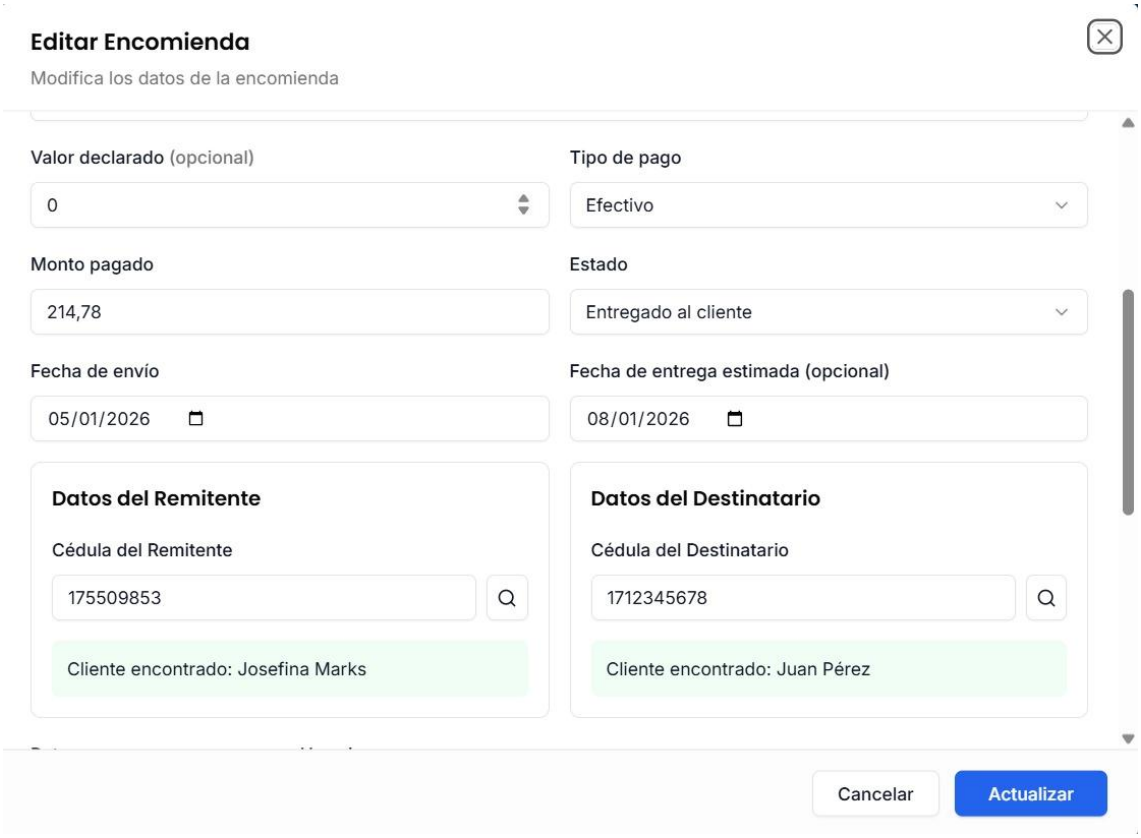



*Figura 22 Pantalla para Editar la Encomienda*

**Elaborado por:** Galeas L. & Cabezas R.

**Fuente:** https://ctflotabolivar.com

### Rastreo de Encomiendas

En esta sección se puede rastrear la encomienda a través de un código designado al momento de registrar la encomienda como la información del envió, información del pago, la ruta, información del remitente y destinatario.

**Figura 20**

*Pantalla de Rastreo de Encomiendas*

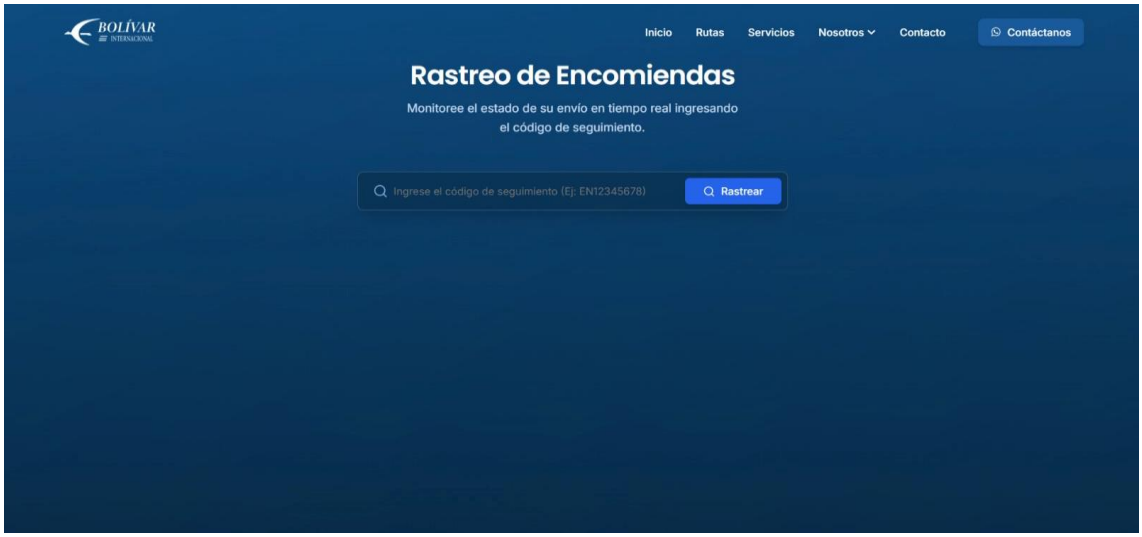



*Figura 23 Pantalla de Rastreo de Encomiendas*

**Elaborado por:** Leydi G. & Ronny C.

**Fuente:** https://ctflotabolivar.com

**Figura 21**

*Pantalla de los Detalles del Envió*

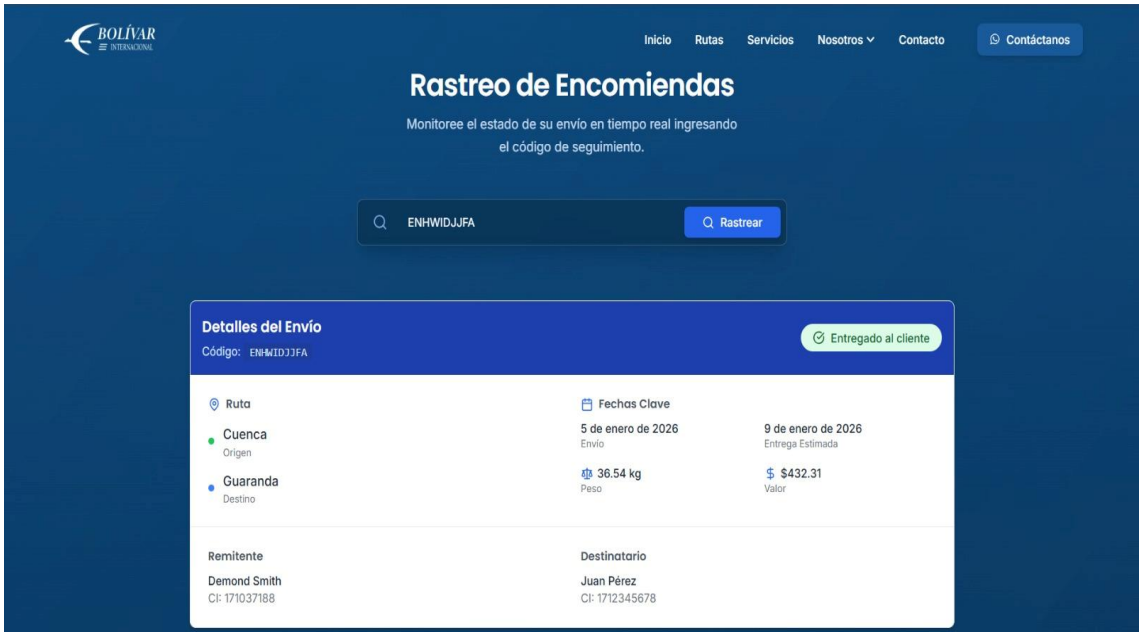



*Figura 24 Pantalla de los Detalles del Envió*

**Elaborado por:** Galeas L. & Cabezas R.

**Fuente:** https://ctflotabolivar.com

**Figura 22**

*Pantalla Historial de Movimientos*

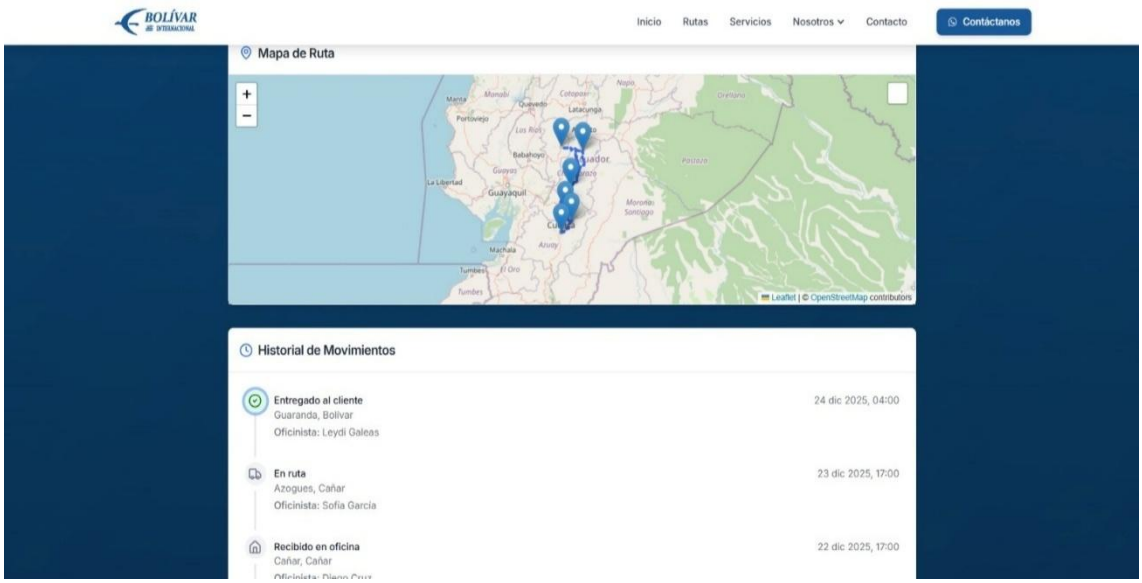



*Figura 25 Pantalla Historial de Movimientos*

**Elaborado por:** Galeas L. & Cabezas R.

**Fuente:** https://ctflotabolivar.com

# Anexo 9.

Manual Técnico




# Manual de Técnico del Sistema de Encomiendas

## Flota Bolívar

Autores:

Ronny Johel Cabezas Lechón

Leydi Marianela Galeas Armijo

Versión: 1.0

Fecha:07/01/2026

# Índice

Introducción ..... 52

Objetivo ..... 52

Manual técnico ..... 52

1. Especificaciones técnicas ..... 52
2. Instalación de dependencias ..... 53
3. Ejecución en modo de desarrollo ..... 54
4. Alojamiento del sistema en Render ..... 56

Alojamiento del backend Laravel en Web Service ..... 56

Alojamiento del frontend React en Static site. ..... 58

Crear la base de datos en Postgres. ..... 59

5. Conexión y configuración al dominio www.ctflotabolivar.com ..... 60
6. Estructura del sistema ..... 61

Frontend React ..... 61

Backend Laravel ..... 61

## Índice de figuras

Figura 1. *Instalación de dependencias composer.* ..... 53

Figura 2. *Instalación del paquete npm en frontend React.* ..... 53

Figura 3. *Estructura del backend Laravel* ..... 54

Figura 4. *Estructura del archivo .env* ..... 55

Figura 5. *Estructura del frontend React* ..... 56

Figura 6. *Configuración de dependencias para producción.* ..... 57

Figura 7. *Configuración de las variables de entorno.* ..... 57

Figura 8. *Conexión con backend y configuración de dependencias para producción...* 58

Figura 9. *Configuración de la variable de entorno.* ..... 58

Figura 10. *Configuración de nuestra base de datos en el servicio de Postgres.* ..... 59

Figura 11. *Sistema Tracking Flota Bolívar en producción.* ..... 59

Figura 12. *Configuración DNS en squarespace.* ..... 60

Figura 13. *Inicio del sitio web con el dominio personalizado.* ..... 61

| Hoja de control de modificaciones | |
|---|---|
| **Titulo** | Manual Técnico del Sistema Flota Bolívar |
| **Versión** | 1.0 |
| **Realizado** | Ronny Johel Cabezas Lechón |
| | Leydi Marianela Galeas Armijo |
| **Fecha** | 07/01/2026 |

| Versiones publicadas | | |
|---|---|---|
| **Versión** | **Descripción** | **Fecha publicada** |
| 1.0 | Primer documento | 07/01/2026 |

**Introducción**

El manual técnico describe detalladamente las sentencias, códigos, ligas que se deben efectuar con la ayuda de la herramienta de diseño, para llevar a cabo la incorporación de los elementos multimedia y generar el software, así como para la generación de la interfaz. El manual técnico es útil para efectuar alguna modificación del software multimedia educativo o darle mantenimiento (García Sánchez et al., 2016, p.222).

El sistema surge como respuesta a la necesidad de optimizar la gestión logística y la trazabilidad de los envíos dentro de la cooperativa, reemplazando procesos manuales por una solución sistemática basada en una arquitectura de N-capas. En los distintos puntos se detallan las especificaciones técnicas que garantizan un funcionamiento optimo, incluyendo el uso de un framework backend Laravel, y React para la interfaz de usuario y la integración de mensajería mediante API de WhatsApp para las notificaciones en tiempo real.

**Objetivo**

Establecer una guía técnica que describa su arquitectura, componentes, configuración del software Flota Bolívar, facilitando a los responsables del sistema las herramientas necesarias para ejecutar actualizaciones, soporte técnico y mantenimiento preventivo del sistema.

**Manual técnico**

1. **Especificaciones técnicas**

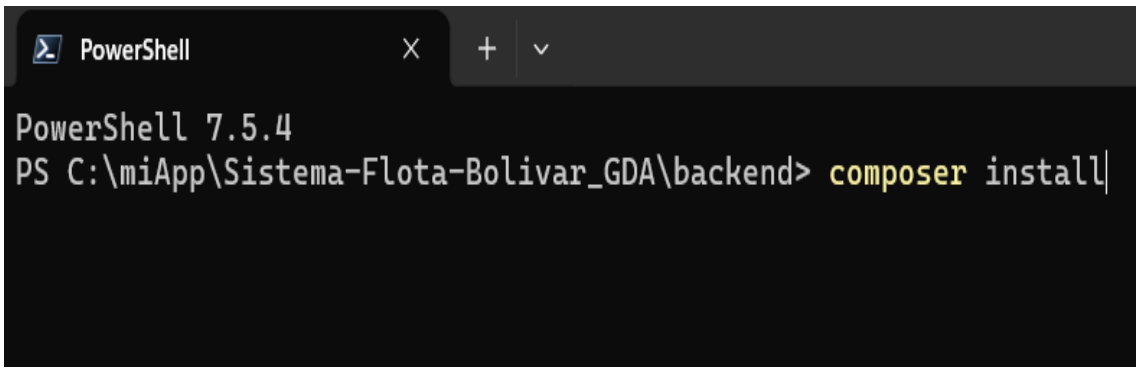
| **Backend** | | **Frontend** | | **Base de datos** | |
|---|---|---|---|---|---|
| **Framework:** | Laravel 12.34.0. | **Librería:** | React.js | **Motor:** | PostgreSQL |
| **Lenguaje:** | PHP 4.2.12 | **Gestión de rutas:** | React Router Dom | **Modelado:** | Uso de migraciones de laravel para la estructura de las tablas. |
| **Autenticación:** | Laravel Sanctum para la protección de las rutas API. | **Estilos:** | Tailwind CSS. | | |

## 2. Instalación de dependencias

Para desarrollar un sistema con backend Laravel es necesario instalar composer, ya que este gestor de dependencias permite instalar el framework, actualizar los paquetes y administración de todas las librerías que Lavarel necesita para tu correcto funcionamiento.

**Figura 26.**

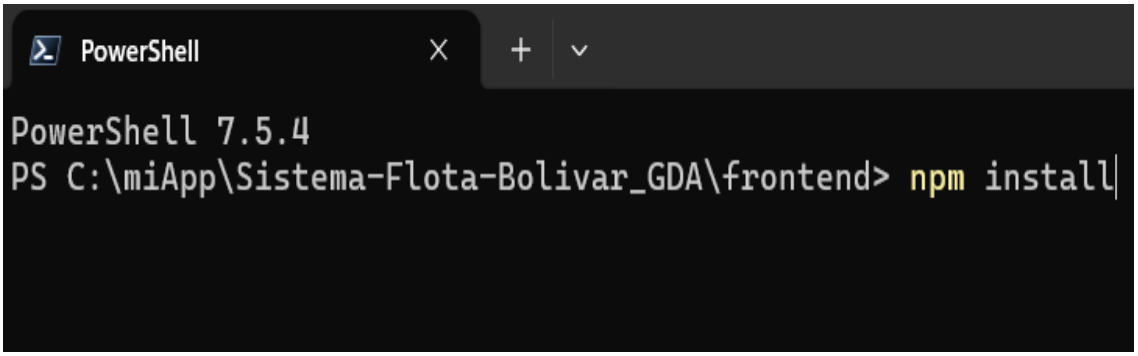
*Instalación de dependencias composer.*




Para trabajar como frontend React es indispensable instalar Node.js, debido a que proporciona npm, el gestor de paquetes encargados de instalar actualizar y administrar las librerías y dependencias que nuestro sistema necesitara para su correcto funcionamiento.

**Figura 27.**

*Instalación del paquete npm en frontend React.*

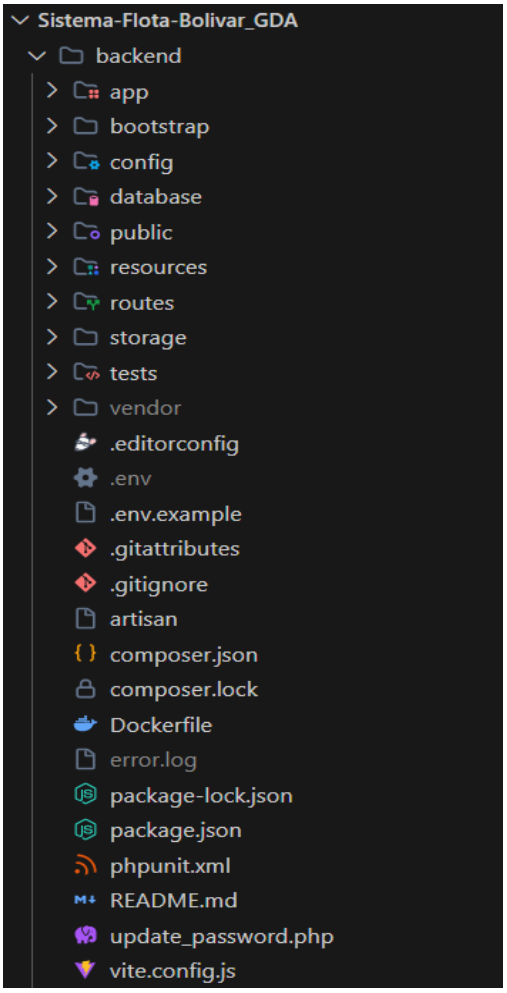


### 3. Ejecución en modo de desarrollo

El proyecto contiene una sección de backend desarrollado en el framework Laravel.

**Figura 28.**

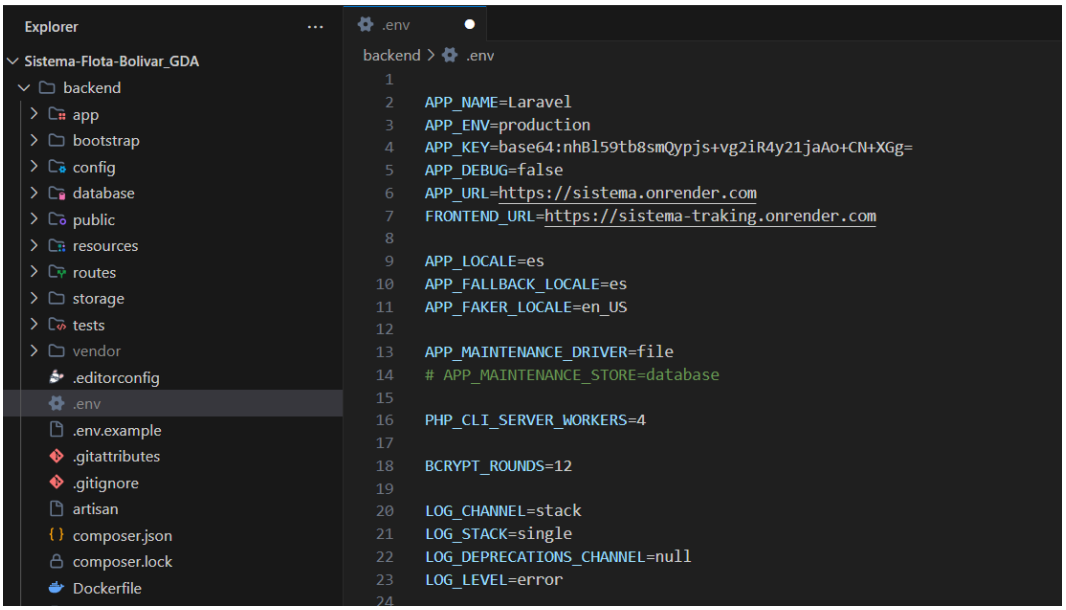
*Estructura del backend Laravel*




En el backend configuramos nuestro archivo .env que proporciona diversas configuraciones de nuestro proyecto, en caso de querer mantener el archivo .env original se puede realizar una copia y nombrar al original como .env.example y al archivo copia nombrarlo como .env y configurarlo según su entorno de desarrollo.

**Figura 29.**

*Estructura del archivo .env*

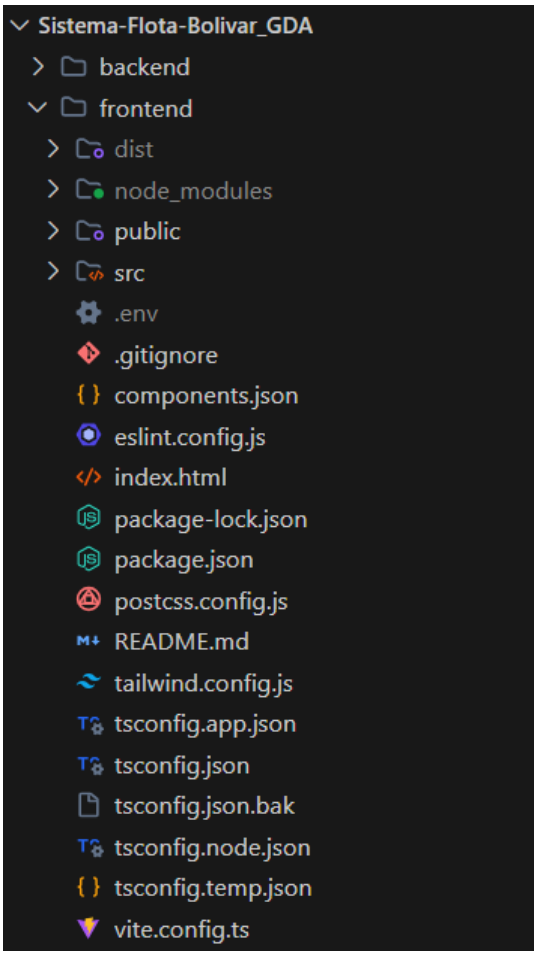



- Para la configuración del envío de correos electrónicos, se estableció el uso de la configuración el Mailtrap, colocamos en usuario y la contraseña, conjunto con el puerto.
- Para nuestra carpeta raíz creamos un enlace de la carpeta storage a la carpeta public y private usando el comando **php artisan storage:link.**
- Para crear nuestras tablas en la base de datos Posgres ejecutamos en comando **php artisan migrate**, en caso de tener información en la base de datos y querer mantenerlos, ejecutaremos el comando **php artisan migrate:refresh –seed**.

Luego configuraremos nuestro frontend, el framework usado es React, en el proyecto se encuentra en el archivo frontend.

**Figura 30.**

*Estructura del frontend React*




## 4. Alojamiento del sistema en Render

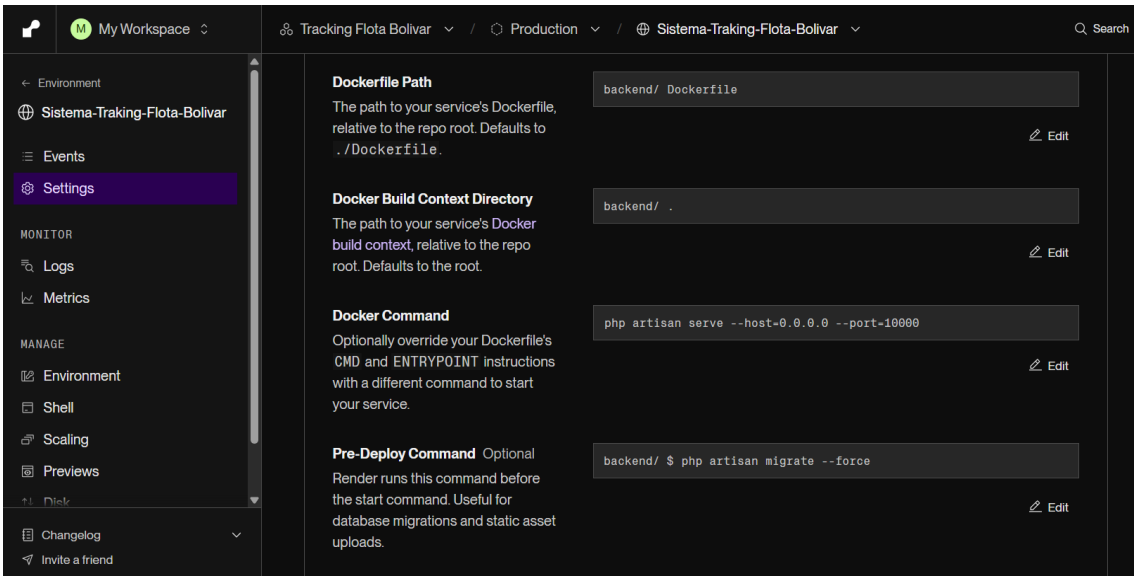
Para alojar el sistema en Render, es indispensable separar el frontend y backend para mayor compatibilidad y buenas prácticas de programación. Creamos un proyecto e alojamos en estos sitios.

### Alojamiento del backend Laravel en Web Service

Alojaremos nuestro backend en este servicio, permitiéndonos ejecutar nuestro backend 24/7, escuchas las peticiones y devuelve las respuestas al frontend de manera autónoma.

**Figura 31.**

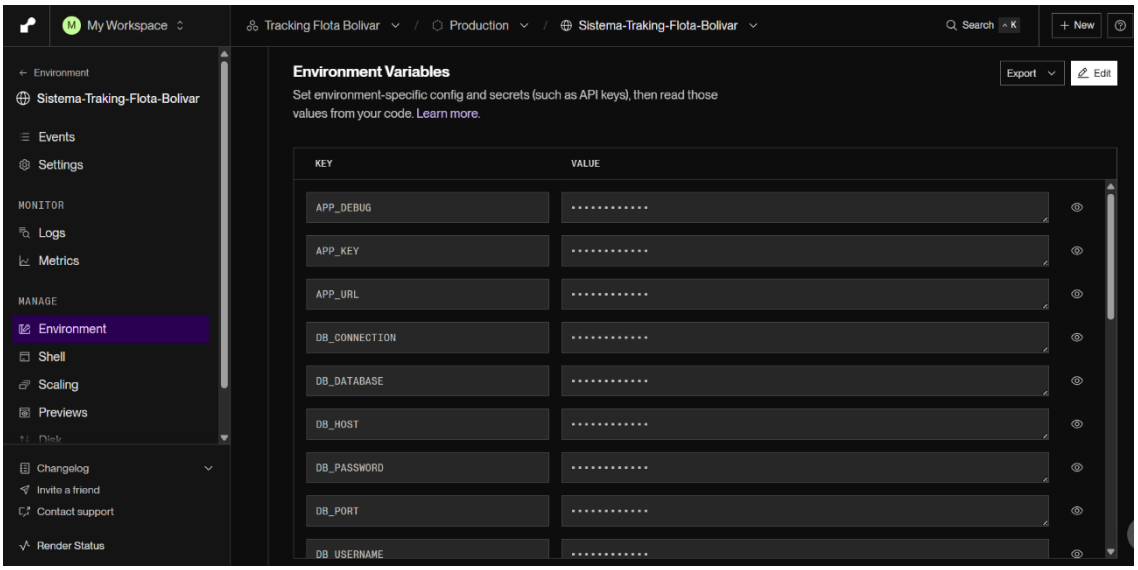
*Configuración de dependencias para producción.*




De la misma manera, configuraremos las variables de entorno que dentro de nuestro código lo tenemos configurado en el archivo backend .env para producción.

**Figura 32.**

*Configuración de las variables de entorno.*



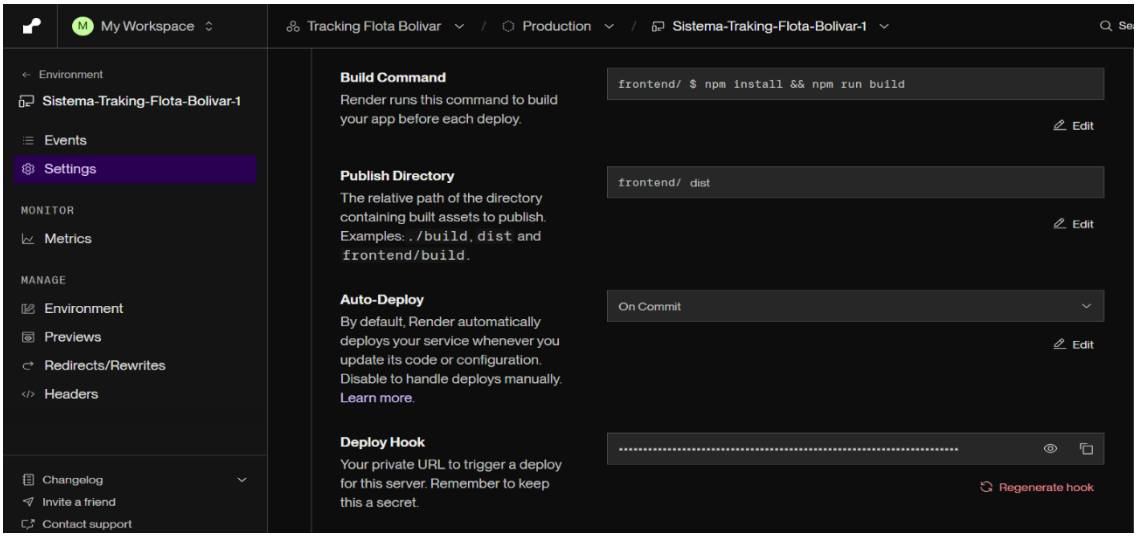

Al guardar los cambios, se nos desplegará el servicio backend, que como base nos permitirá conectarnos con el frontend y la base de datos.

**Alojamiento del frontend React en Static site.**

Este servicio nos permite desplegar nuestro sistema como un sitio estático o páginas que no necesitan un servidor ejecutándose, siendo así esencial para desplegar React.

**Figura 33.**

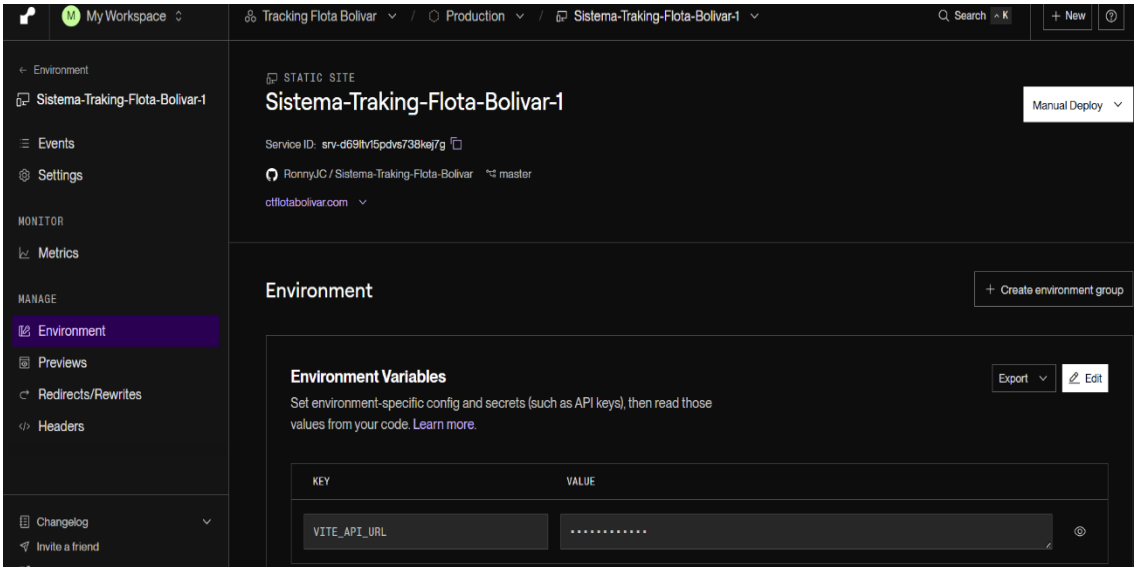
*Conexión con backend y configuración de dependencias para producción.*




También configuraremos las variables de entorno, el cual es VITE_API_URL que nos permitirá redireccionarnos a nuestro backend en producción para recibir los datos del backend.

**Figura 34.**

*Configuración de la variable de entorno.*

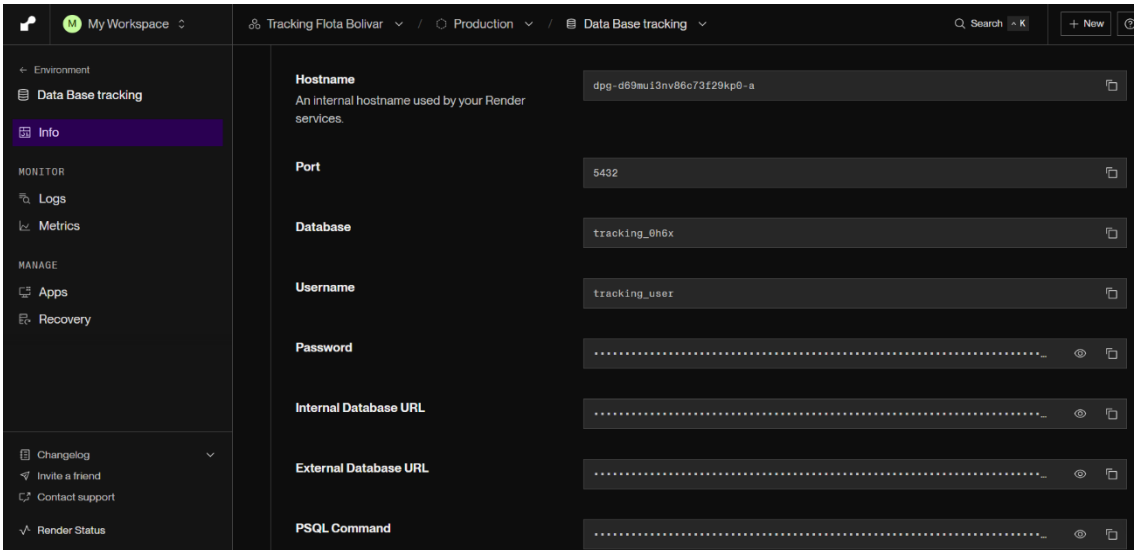


**Crear la base de datos en Postgres.**

Dentro de Render tenemos un servicio que nos permite crear y conectar nuestra base de datos dentro de Postgres, un gestor de base de datos gratuito y robusto.

**Figura 35.**

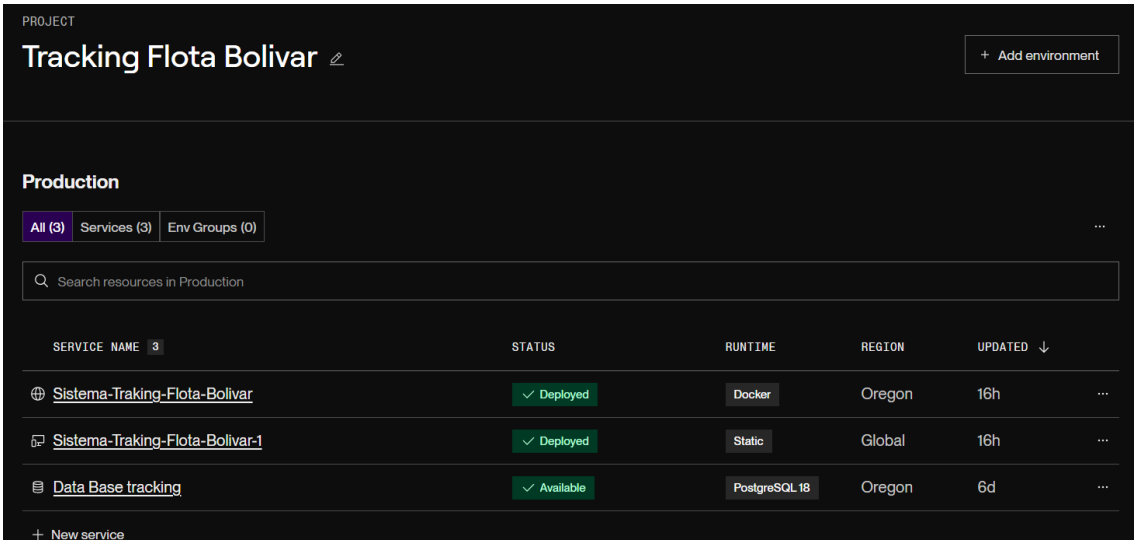
*Configuración de nuestra base de datos en el servicio de Postgres.*




Al alojar y configurar nuestro proyecto en servicios diferentes en Render, ya nos permitirá ver la página web la producción con el dominio preestablecido de Render.

**Figura 36.**

*Sistema Tracking Flota Bolívar en producción.*

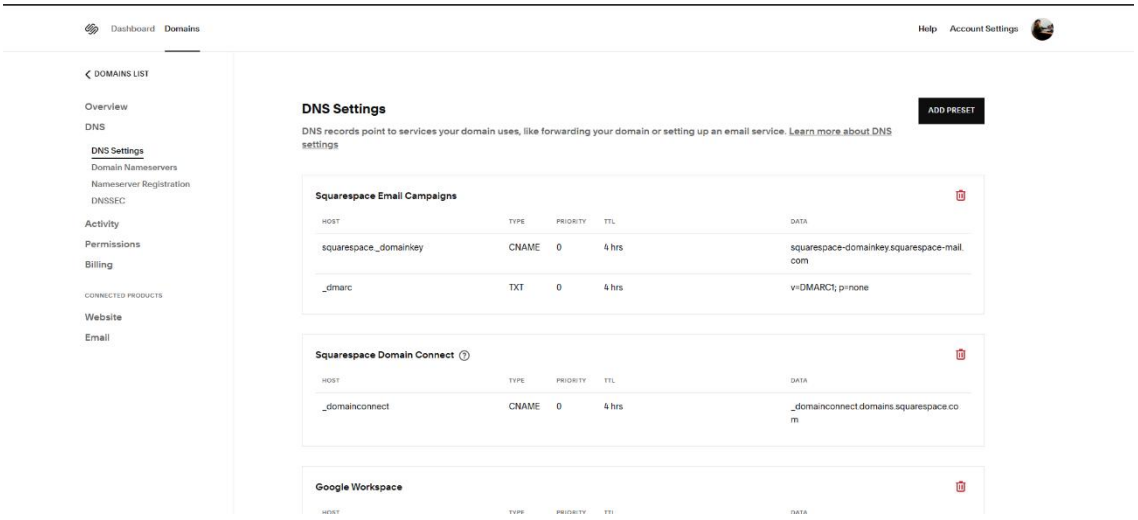


## 5. Conexión y configuración al dominio www.ctflotabolivar.com

Render nos ofrece un dominio predeterminado para nuestro sistema desplegado tanto como frontend y backend, lo cual, para poder un dominio propio, se requiere realizar la compra de uno, y configurarlo con los servicios alojados en Render.

**Figura 37.**

*Configuración DNS en squarespace.*

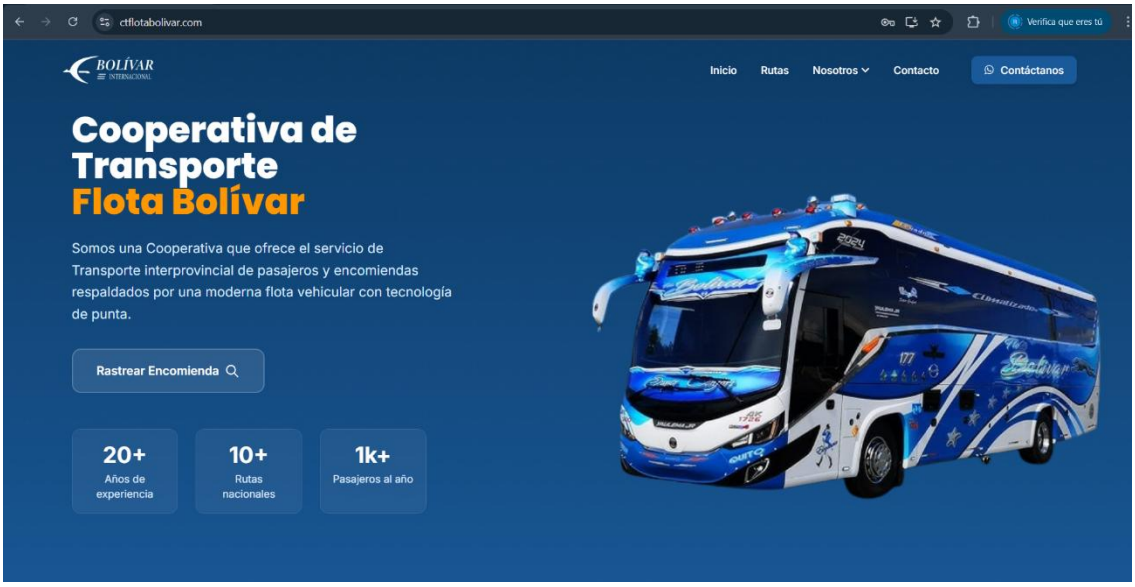



Se establece una configuración como personalizado en Custom records, donde se configuraremos lo siguiente:

- Host: es el subdominio que se requiere para configurar a nuestro dominio principal, tengo conexiones. Esta configurado con @ que representa el dominio principal, el www. configuración se establece como el dominio asociado al frontend, y el api. es un subdominio en donde representa el backend, para establecer la conexión con el frotend.
- Type: indica el tipo de dato que esta configurado. Los tipos establecidos es A, que apunta a las direcciones IP, y el tipo CNAME, que es el que apunta a otros dominios.
- TTL: es el tiempo de los servidores DNS guardan esta configuración en cache antes de actualizarla.
- Data: este es el destino al que apunta a un dominio. En nuestros servicios Render tenemos como preestablecidos dominios tanto para frontend como para backend, y se configura para apuntar a estos subdominios personalizados.

**Figura 38.**

*Inicio del sitio web con el dominio personalizado.*




## 6. Estructura del sistema

### Frontend React

- src/components/: contiene componentes visuales reutilizables para los botones, cuadros de diálogos y tablas compartidas.
- src/feactures/: se utiliza una arquitectura basada en características. Se estructura en archivos distribuidos por cada módulo del sistema, cada característica tiene dentro de si sus propios componentes, páginas y servicios que llaman a la API.
- src/layouts/: define las estructuras del menú de la barra lateral, el encabezado y el área donde se muestra el contenido principal.
- src/router/: aquí se define las rutas de navegación dentro del sistema, contiene enlaces a los que el usuario puede acceder en el navegador.
- src/assets/: se encuentran archivos estáticos como imágenes, logos de la Flota Bolívar o iconos que el sistema carga directamente.

### Backend Laravel

- app/Http/Controllers/: los controladores contienen los archivos que reciben, procesa y devuelve las peticiones del frontend.
- app/Models/: los modelos con representaciones de las tablas de la base de datos.
- routes/api.php: define los puntos de acceso del sistema. Guarda todas las URLs que el frontend consume para comunicarse con el servidor.

- database/migrations/: las migraciones son archivos donde se establece que tablas se va a crear y que columnas llevan.
- storage/: esta carpeta se usa para guardar archivos temporal o permanentemente.
- public/index.php: este es el directorio raíz, siendo la puerta de entrada a todas las peticiones.

# Anexo 10.

Certificado Anti-plagio

**ING. MÓNICA ELIZABETH BONILLA MANOBANDA EN CALIDAD DE DIRECTORA DEL TRABAJO DE INTEGRACIÓN CURRICULAR,**

# CERTIFICA

Que el trabajo de integración curricular denominado "**SISTEMA WEB PARA EL SEGUIMIENTO DE ENCOMIENDAS DE LA COOPERATIVA DE TRANSPORTE FLOTA BOLÍVAR**", presentado por: LEYDI MARIANELA GALEAS ARMIJO y RONNY JOHEL CABEZAS LECHÓN estudiantes de la **carrera de Software** pasó el análisis de coincidencia no accidental en la herramienta COMPILATIO, reflejando un **porcentaje de similitud del 8%**, como se puede evidenciar en el documento adjunto.

Guaranda, 23 de febrero del 2026

Atentamente,

Firmado electrónicamente por:
MONICA ELIZABETH
BONILLA MANOBANDA
Validar únicamente con FirmaEC

Ing. Mónica Bonilla Manobanda
**Directora**

**CERTIFICADO DE ANÁLISIS**
*magister*

# PROYECTO_TESIS

**8%** Textos sospechosos

- **8%** Similitudes
  - < 1 % similitudes entre comillas (ignorado)
  - 0 % entre las fuentes mencionadas (ignorado)
- **2%** Idiomas no reconocidos (ignorado)
- **13%** Textos potencialmente generados por la IA (ignorado)

**Nombre del documento:** PROYECTO_TESIS.docx
**ID del documento:** 281bb3983cea15a3e35efdf1a21a0d77764827ad
**Tamaño del documento original:** 17,28 MB
**Autor:** Leydi Marianela Galeas Armijo

**Depositante:** Leydi Marianela Galeas Armijo
**Fecha de depósito:** 22/2/2026
**Tipo de carga:** url_submission
**fecha de fin de análisis:** 22/2/2026

**Número de palabras:** 20.863
**Número de caracteres:** 141.952

Ubicación de las similitudes en el documento:



## Fuentes principales detectadas

| Nº | Descripciones | Similitudes | Ubicaciones | Datos adicionales |
|---|---|---|---|---|
| 1 | **www.creatividad.gob.ec**<br>https://www.creatividad.gob.ec/wp-content/uploads/downloads/2020/03/Código-O-Eco-Soc-C...<br>18 fuentes similares | 3% | | Palabras idénticas: 3% (720 palabras) |
| 2 | **dspace.ucacue.edu.ec** \| Comparación de metodologías ágiles para el desarrollo ...<br>https://dspace.ucacue.edu.ec/handle/ucacue/17091 | 1% | | Palabras idénticas: 1% (264 palabras) |
| 3 | **localhost** \| "Análisis del uso de prácticas de seguridad en el ciclo de vida de desar...<br>http://localhost:8080/xmlui/bitstream/redug/59824/3/B-CINT-PTG-N. 801 Rodríguez Méndez ...<br>1 fuente similar | 1% | | Palabras idénticas: 1% (262 palabras) |
| 4 | **repositorio.uladech.edu.pe** \| Propuesta de implementación del sistema Web de ...<br>https://repositorio.uladech.edu.pe/handle/20.500.13032/23570<br>1 fuente similar | 1% | | Palabras idénticas: 1% (246 palabras) |
| 5 | **repositorio.uladech.edu.pe**<br>https://repositorio.uladech.edu.pe/bitstream/20.500.13032/23570/1/CALIDAD_PROCESOS_VIV... | 1% | | Palabras idénticas: 1% (245 palabras) |

## Fuentes con similitudes fortuitas

| Nº | Descripciones | Similitudes | Ubicaciones | Datos adicionales |
|---|---|---|---|---|
| 1 | **Documento de otro usuario** #af4c8<br>Viene de de otro grupo | < 1% | | Palabras idénticas: < 1% (33 palabras) |
| 2 | **Documento de otro usuario** #c58a9e<br>Viene de de otro grupo | < 1% | | Palabras idénticas: < 1% (32 palabras) |
| 3 | **Documento de otro usuario** #b12ffb<br>Viene de de otro grupo | < 1% | | Palabras idénticas: < 1% (25 palabras) |
| 4 | **www.boe.es** \| BOE-A-2021-8806 Ley Orgánica 7/2021, de 26 de mayo, de protecci...<br>https://www.boe.es/buscar/doc.php?id=BOE-A-2021-8806 | < 1% | | Palabras idénticas: < 1% (26 palabras) |
| 5 | **Documento de otro usuario** #1683c9<br>Viene de de otro grupo | < 1% | | Palabras idénticas: < 1% (18 palabras) |

## Fuentes ignoradas

Estas fuentes han sido retiradas del cálculo del porcentaje de similitud por el propietario del documento.

| Nº | Descripciones | Similitudes | Ubicaciones | Datos adicionales |
|---|---|---|---|---|
| 1 | **derechoecuador.com**<br>https://derechoecuador.com/registro-oficial-no-899-viernes-09-de-diciembre-de-2016-suplem... | 3% | | Palabras idénticas: 3% (605 palabras) |
| 2 | **repositorio.puce.edu.ec** \| Aplicación web de trazabilidad para la gestión logística...<br>https://repositorio.puce.edu.ec/handle/123456789/39726 | 1% | | Palabras idénticas: 1% (290 palabras) |
| 3 | **hdl.handle.net** \| Propuesta de implementación del sistema Web de venta de bole...<br>https://hdl.handle.net/20.500.13032/23570 | 1% | | Palabras idénticas: 1% (245 palabras) |
| 4 | **www.privacy-regulation.eu** \| Artículo 35 UE Reglamento general de protección d...<br>https://www.privacy-regulation.eu/es/35.htm | < 1% | | Palabras idénticas: < 1% (116 palabras) |

## Fuentes mencionadas (sin similitudes detectadas)

Estas fuentes han sido citadas en el documento sin encontrar similitudes.

1 https://api.smsmobileapi.com/sendsms/
2 http://repositorio.espe.edu.ec/bitstream/21000/8771/1/AC-ESPEL-SOF-0004.pdf
3 https://www.guaranda.gob.ec/newsiteCMT/

4 https://www.gobiernoelectronico.gob.ec/wp-content/uploads/2018/10/Ley-de-Propiedad-Intelectual.pdf
5 http://bibliotecadigital.uca.edu.ar/repositorio/tesis/metodologias-desarrollo-software.pdf

# Anexo 11.

Link del repositorio digital de biblioteca donde fue subido el proyecto

# Anexo 12.

Evidencia de reuniones con el director, pares académicos del proyecto y personal de la Flota Bolívar.



