

ESCUELA DE INGENIERÍA EN TECNOLOGÍAS DE LA INFORMACIÓN

**Tesis previa a la obtención de título de
Ingeniero en Tecnologías de la Información.**

AUTOR: Nelson Eduardo Torres Pineda

TUTOR: Mgtr. Milton Ricardo Palacios Morocho

Optimización de las rutas de recolección de residuos y
tiempos de flota de recolección en la zona urbana de la ciudad
de Yantzaza mediante inteligencia artificial

CERTIFICACIÓN DE AUTORÍA

Yo, **Nelson Eduardo Torres Pineda**, declaro bajo juramento que el trabajo aquí descrito es de mi autoría; que no ha sido presentado anteriormente para la obtención de ningún grado académico ni calificación profesional, y que he consultado la bibliografía que se detalla.

Cedo mis derechos de propiedad intelectual a la **Universidad Internacional del Ecuador**, para que pueda publicarlo y difundirlo en internet, de conformidad con lo establecido en la Ley de Propiedad Intelectual, su reglamento y demás normativas aplicables.



Nelson Eduardo Torres Pineda

Autor

APROBACIÓN DEL TUTOR

Yo Mitlon Palacios Morocho, certifico que conozco a la autora del presente trabajo de titulación “Optimización de las rutas de recolección de residuos y tiempos de flota de recolección en la zona urbana de la ciudad de Yantzaza mediante inteligencia artificial”, Nelson Eduardo Torres Pineda, siendo el responsable exclusiva tanto de su originalidad y autenticidad, como de su contenido.

.....

Mgs. Milton Palacios Morocho

DIRECTOR DEL TRABAJO DE TITULACIÓN

DEDICATORIA

Existen personas que merecen mi reconocimiento por su apoyo incondicional a mi persona y mis proyectos, dedico este trabajo mis padres Lic. Carmen Pineda Flores y Op. Nelson Rogelio Torres Agreda que son mi fuerza y mi motivo diario de ser mejor persona cada día.

También dedico este trabajo de tesis especialmente a aquel muchacho que soñó con ser desarrollador hace 15 años, para él, que sepa lo va a lograr.

AGRADECIMIENTO

Esta tesis es el resultado del esfuerzo por superar todas mis batallas, el resultado de demostrarme que lo imposible se logra con esfuerzo.

Agradezco a mis Padres, Lic. Carmen Pineda Flores y Op. Nelson Torres Agreda, quienes me han apoyado incondicionalmente, siempre han apostado por mí y jamás han dudado de mis habilidades, gracias a su ejemplo de perseverancia y dedicación me han formado en el hombre que soy hoy día.

Hay muchas personas que han sido parte de mi camino en esta carrera, muchos siguen caminando junto a mí, otros ya no, inclusive algunos me observan desde el cielo, a todos ellos gracias por sumarme en su breve camino junto a mí, los llevo siempre.

Agradezco a mi tutor Milton Palacios, por su paciencia y orientarme en mi camino en el desarrollo de este trabajo final.

A mis profesores que, en este tiempo supieron compartir sus conocimientos, para poder desarrollar mi formación académica.

Mi profundo agradecimiento al Gobierno Autónomo Descentralizado del Cantón Yantzaza liderado por su alcaldesa María Lalangui, por su apoyo en la realización de este proyecto. A la Universidad Internacional del Ecuador campus Loja, a la Carrera de Ingeniería en Tecnologías de la Información y a todos los docentes que contribuyeron a mi formación profesional.

Un agradecimiento especial a los operadores y conductores del servicio de recolección de residuos de Yantzaza, cuya colaboración fue fundamental para el desarrollo de esta investigación.

INDICE DE CONTENIDOS

CARÁTULA.....	I
CERTIFICACIÓN DE AUTORÍA	II
APROBACIÓN DEL TUTOR.....	III
DEDICATORIA	IV
AGRADECIMIENTO	V
INDICE DE CONTENIDOS	VI
INDICE TABLAS	IX
INDICE DE FIGURAS	X
RESUMEN	12
ABSTRACT	13
INTRODUCCIÓN	14
PLANTEAMIENTO DEL PROBLEMA	16
OBJETIVOS.....	17
1. CAPITULO I	18
1.1. GAD YANTZAZA	20
1.2. PROCESOS DE RECOLECCION DE BASURA	22
1.3. ALGORITMOS CONSIDERADOS PARA OPTIMIZACIÓN DE RECORRIDOS POR GEOLOCALIZACIÓN	23
1.4. SCRUM COMO METODOLOGÍA PARA EL DESARROLLO. DE PROYECTOS QUE OPTIMIZAN EL RECORRIDO DE RUTAS.	27
1.5. FUNDAMENTOS DE LA COMUNICACIÓN EL LA INGENIERIA DE SOFTWARE	28
1.6. IMPACTO EN LA ARQUITECTURA Y DISEÑO DE SOFTWARE	29

1.7. CALIDAD DEL SOFTWARE Y MANTENIBILIDAD	29
1.8. GESTIÓN DE REQUISITOS Y ESPECIFICACIONES	30
1.9. PERSPECTIVA DE GESTION DE PROYECTOS	30
1.10. EVIDENCIA EMPÍRICA	30
1.11. PRODUCTO BACKLOG PARA SISTEMAS DE OPTIMIZACIÓN DE RUTAS	31
1.12. APLICACIONES DE INFORMACIÓN GEOGRÁFICA SIG EN LA GRSU	32
2. CAPITULO II	34
2.1. REQUERIMIENTOS	35
2.2. DESARROLLO DE LA METODOLOGÍA	47
2.3. DIAGRAMAS DEL SISTEMA	52
3. CAPITULO III	61
3.1. OBTENCIÓN DE RUTA GPX ACTUAL CON GEOTRACKER	62
3.2. PROTOTIPOS.....	64
3.3. CONSTRUCCION DE LOS MODULOS.	67
3.4. ARQUITECTURA DE IMPLEMENTACIÓN	76
3.5. ANÁLISIS DE RESULTADOS	78
3.6. ANALISIS DEL COSTO FENEFICIO	80
3.7. ESTRUCTURA DEL PROYECTO	81
3.8. MODELO DE DATOS.....	84
4. CAPITULO IV	90
4.1. CONCLUSIONES	91
4.2. RECOMENDACIONES	92
4.3. BIBLIOGRAFÍA	93
5. CAPITULO V	100

5.1. FORMULARIO DE REQUERIMIENTOS.....	101
5.2. FORMATO DE REQUERIMIENTO DE SCRUM	107
5.3. PLAN DE PRUEBAS	120
5.4. VALIDACIÓN DEL PLAN DE PRUEBAS	138
5.5. CERTIFICACIÓN DE FUNCIONAMIENTO DE LA APLICACIÓN	157

INDICE DE TABLAS

Tabla1 Recorridos actuales de los vehículos recolectores	21
Tabla 2 Ventajas y desventajas de dijkstra	24
Tabla 3 Ventajas y desventajas de a*	25
Tabla 4 Ventajas y desventajas de aco	25
Tabla 5 Ventajas y desventajas de recorrido simulado	26
Tabla 6 Ventajas y desventajas algoritmo genérico	26
Tabla 7 Requerimientos funcionales	43
Tabla 8 Requerimientos no funcionales	45
Tabla 9 Proceso scrumm en optimización de rutas con ia	47
Tabla 10 Historias de usuario	48
Tabla 11 Product backlog	50
Tabla 12 Tipos de pruebas realizadas	75
Tabla 13 Resultados	79
Tabla 14 Endpoints	87

INDICE DE FIGURAS

Figura 1 Diagrama de caso de uso gestión usuarios	53
Figura 2 Diagrama caso de uso gestión de rutas optimas	54
Figura 3 Diagrama caso de uso gestión vehículos	55
Figura 4 Diagrama caso de uso planificacion de recorridos	56
Figura 5 Arquitectura de sistema en scrum	57
Figura 6 Diagrama ERM	58
Figura 7 Aquitectura del sistema	59
Figura 8 Mapa de navegación	60
Figura 9 Captura de coordeadas de ruta actual por medio de GPX.....	63
Figura 10 Prototipo pantalla login	64
Figura 11 Prototipo dashboard admin	65
Figura 12 Prototipo gestión de rutas	65
Figura 13 Prototipo navegación chofer	66
Figura 14 Prototipo reportes y análisis	66
Figura 15 Prototipo gestión usuarios	67
Figura 16 Interfaz gestión de usuarios	67

Figura 17 Autenticación y gestión usuarios	68
Figura 18 Modulo carga y procesamiento de archivos gpx	69
Figura19 Módulo de optimización de rutas	70
Figura 20 Módulo de visualización y navegación	71
Figura 21 Módulo de analíticas y reportes	72
Figura 22 Estructura del proyecto	81
Figura 23 Estructura del proyecto explicación	83

RESUMEN

Este trabajo presenta el desarrollo de un sistema de optimización de rutas de recolección de residuos sólidos en la ciudad de Yantzaza mediante algoritmos de búsqueda y sistemas de información geográfica. El sistema implementa un algoritmo de optimización basado en colonias de hormigas (ACO) que, combinado con herramientas de visualización geoespacial, permite mejorar significativamente la eficiencia operativa del servicio de recolección.

La investigación aborda la problemática actual de la gestión de residuos en Yantzaza, donde el crecimiento poblacional del 2,24% anual ha generado mayor producción de desechos, provocando contenedores desbordados, vertederos ilegales y un servicio de recolección ineficiente. El sistema desarrollado optimiza las rutas de recolección considerando variables como capacidad de vehículos, restricciones de tiempo, condiciones de tráfico y variabilidad en la generación de residuos.

Utilizando la metodología ágil SCRUM, se implementó una aplicación web basada en Flask y Folium que permite cargar rutas en formato GPX, optimizarlas mediante el algoritmo ACO y visualizarlas en un mapa interactivo. Los resultados obtenidos muestran una reducción del 25% en los tiempos de recorrido, un ahorro del 18% en el consumo de combustible y una mejora del 22% en la cobertura del servicio.

Palabras clave: Optimización de rutas, residuos sólidos, algoritmos de búsqueda, SIG, ACO, gestión de residuos, logística urbana.

ABSTRACT

This work presents the development of an optimization system for solid waste collection routes in the city of Yantzaza using search algorithms and geographic information systems. The system implements an Ant Colony Optimization (ACO) algorithm that, combined with geospatial visualization tools, significantly improves the operational efficiency of the collection service.

The research addresses the current waste management problems in Yantzaza, where population growth of 2.24% annually has generated increased waste production, causing overflowing containers, illegal dumps, and inefficient collection service. The developed system optimizes collection routes considering variables such as vehicle capacity, time constraints, traffic conditions, and variability in waste generation.

Using the agile SCRUM methodology, a web application based on Flask and Folium was implemented, allowing users to load routes in GPX format, optimize them using the ACO algorithm, and visualize them on an interactive map. The results show a 25% reduction in travel times, an 18% saving in fuel consumption, and a 22% improvement in service coverage.

Keywords: Route optimization, solid waste, search algorithms, GIS, ACO, waste management, urban logistics.

Introducción

La gestión eficiente de residuos sólidos urbanos representa uno de los desafíos más significativos que enfrentan las ciudades modernas. En un contexto donde la urbanización acelerada y el crecimiento poblacional ejercen una presión cada vez mayor sobre los servicios municipales, la optimización de los sistemas de recolección de residuos se ha convertido en una prioridad crítica para la sostenibilidad urbana y la salud pública (Wilson et al., 2013). En particular, la planificación de rutas de recolección emerge como un factor determinante en la eficiencia operativa y el impacto ambiental de estos servicios.

En América Latina, donde las ciudades experimentan un crecimiento urbano considerable, la necesidad de modernizar y optimizar los sistemas de recolección de residuos es especialmente apremiante. Según Gutiérrez y Nava (2018), los municipios de la región enfrentan retos significativos en términos de costos operativos, consumo de combustible y cobertura del servicio, lo que subraya la importancia de implementar soluciones innovadoras basadas en datos para mejorar la eficiencia del sistema.

Es por ello que se desarrolló esta investigación, misma que radicó en el problema de recolección de residuos sólidos. Una muestra es la ciudad de Yantzaza que a partir del último censo realizado por INEC en 2022 donde la tasa de crecimiento poblacional es de 2,24% por año dentro del casco urbano, lo que consigo genera un mayor consumo y por ende una mayor producción de residuos sólidos, trayendo mayor responsabilidad para los encargados de la gestión de los residuos sólidos.

En este contexto, el uso de tecnologías de información como algoritmos de optimización y sistemas de información geográfica (SIG) ofrece un potencial significativo para transformar la gestión de residuos sólidos. Estas herramientas pueden ayudar a diseñar rutas

más eficientes, reducir costos operativos y minimizar el impacto ambiental de la recolección de residuos.

Este trabajo se concentró en el desarrollo e implementación de un sistema de optimización de rutas para la recolección de residuos sólidos en la ciudad de Yantzaza, Ecuador. Utilizando algoritmos de búsqueda avanzados y herramientas de SIG, se busca mejorar la eficiencia del servicio de recolección, reducir costos operativos y contribuir a la sostenibilidad ambiental de la ciudad.

Planteamiento del problema

Las actuales rutas de recolección con la que cuenta el centro la zona urbana de la ciudad de Yantzaza presentan una serie de incidentes dentro de la planificación, mala asignación de cuadrilla, tiempos muertos generados en el momento de la recolección domiciliaria, generación de sobretiempos en el manejo del camión recolector y lo más importante, la aparición de botaderos clandestinos a cielo abierto en diferentes puntos de la urbe. Esto en el tiempo largo puede resultar en un problema ambiental muy difícil de tratar debido al gran impacto visual, y daño tanto al ecosistema como a la salud de los habitantes.

Yantzaza es una ciudad que tiene un movimiento muy alto de tránsito de personas como de vehículos en su zona urbana manteniendo una distancia de 5.4 km desde el límite norte conocido como Pitá y su límite sur Piedraliza conformada por 13335 habitantes.

La relevancia de esta investigación radica en su potencial desarrollar un análisis de las rutas de recolección a partir de los resultados obtenidos, para optimizar las rutas del sistema de recolección de residuos hasta su disposición final, contribuyendo así a una mayor cobertura, reduciendo significativamente los costos operativos del servicio de recolección y disminuyendo el impacto ambiental a través de la reducción del consumo de combustible.

OBJETIVOS

Objetivo Principal

Optimización de las rutas de recolección de residuos y tiempos de flota de recolección en la zona urbana de la ciudad de Yantzaza mediante inteligencia artificial

Objetivos Específicos

- Diseñar un sistema basado en datos para el análisis y optimización de rutas de recolección de residuos.
- Implementar algoritmos avanzados de optimización que minimicen los tiempos de desplazamiento y el consumo de combustible en la operación de la flota de recolección.
- Implementar el manejo de las rutas optimizadas dentro de una plataforma web.
- Evaluar la efectividad de las rutas optimizadas mediante indicadores clave de desempeño, como la reducción de distancias recorridas y el ahorro de combustible, en escenarios operativos controlados.

CAPÍTULO I:

ESTADO DEL ARTE

ANTECEDENTES

Yantzaza no cuenta con un sistema estructurado para la recolección de residuos sólidos. Para ejecutar este servicio, el GAD emplea el uso de camiones compactadores de basura, camiones de cajón y carretas impulsado por el hombre. Los camiones compactadores deben dividirse la recolección entre los vehículos habilitados para ello, lo que genera falencias e inconsistencias al ejecutar la recolección.

La administración de la ciudad, a través de la unidad de gestión ambiental, asume la responsabilidad de gestionar correctamente los desechos generados. Esto implica no solo su recolección eficiente, sino también su disposición final en cumplimiento con las normativas ambientales vigentes. Con el fin de minimizar los impactos negativos que una mala gestión de residuos podría generar en el medio ambiente y en la calidad de vida de los habitantes. Por ello, la articulación de la Alcaldía es clave para asegurar un manejo integral de los residuos sólidos, promoviendo prácticas sostenibles que respondan a las necesidades actuales y futuras de la ciudad.

Yantzaza, al ser la capital económica de Zamora Ch., enfrenta un reto significativo en términos de manejo de residuos. Su gran densidad poblacional y el acelerado crecimiento urbano generan una gran cantidad de desechos diariamente, lo que hace imprescindible un sistema de recolección robusto y eficaz. La demanda de un sistema de gestión de rutas de los vehículos recolectores es de mucha ayuda a poder solventar la gestión del sistema de recolección, que permita satisfacer la alta demanda actual y se adapte a las crecientes exigencias que surgen con el tiempo, ya que la producción de residuos aumenta por el incremento de población y el consumo.

MARCO TEÓRICO

1.1 GAD Yantzaza

El gobierno autónomo descentralizado de Yantzaza es un municipio que cuenta con un relleno sanitario en el sector Chimbutza que acopia los desechos sólidos tanto orgánicos como inorgánicos de toda la ciudad, estos desechos son recolectados por compactadoras que hacen el recorrido de la zonas urbanas y rurales.

1.1.1. Realidad Actual

En la ciudad de Yantzaza, la recolección de residuos presenta deficiencias relacionadas con la falta de planificación eficiente, rutas ineficientes y la ausencia de tecnologías que permitan monitorear en tiempo real el proceso de recolección; estas deficiencias generan contenedores desbordados, vertederos ilegales y una recolección irregular, lo que afecta negativamente la calidad de vida de los habitantes.

La investigación de campo realizada en el municipio de Yantzaza reveló los siguientes problemas específicos:

1. **Planificación manual de rutas:** Las rutas actuales son diseñadas de manera empírica por los supervisores, sin utilizar herramientas tecnológicas para su optimización.
2. **Desconocimiento de la situación en tiempo real:** Los administradores no tienen visibilidad sobre la ubicación actual de los vehículos de recolección ni pueden verificar si se han completado las rutas asignadas.
3. **Rutas ineficientes:** Los recorridos actuales no consideran variables como el tráfico, pendientes, o la cantidad de residuos por zona, lo que genera ineficiencias en el servicio.

4. **Cobertura irregular:** Existen zonas de la ciudad con servicios deficientes de recolección, mientras que otras reciben un servicio más frecuente.
5. **Alto consumo de combustible:** Las rutas ineficientes generan un mayor consumo de combustible mismo que generan un apercibida económica en el GAD afectando la distribución correcta de recursos.
6. **Falta de métricas de rendimiento:** No existen indicadores claros para evaluar la eficiencia del servicio de recolección.

En la unidad de gestión ambiental, cuentan actualmente con un sistema manual de asignación de rutas no supervisada, basado en experiencia de los choferes de las compactadoras para recorrer la ciudad haciendo la recolección. El mapa de rutas de la ciudad está dividido en 4 rutas que se reparten las 4 compactadoras para completar la recolección

Tabla 1

Recorridos actuales de los vehículos recolectores

Día	Ruta	Compactadora
Lunes	Urbano centro – norte	Hino GH
	Urbano centro -sur	Scannia
	Urbano Hospitales	Hino Fc
	Rural Chicaña	Hino Fc
	Rural Chicaña	Hino Fc
Martes	Urbano centro – norte	Hino GH
	Urbano centro -sur	Scannia
	Urbano Hospitales	Hino Fc
	Rural Chicaña	Hino Fc
	Rural Chicaña	Hino Fc
Miércoles	Urbano centro – norte	Hino GH
	Urbano centro -sur	Scannia
	Urbano Hospitales	Hino Fc
	Rural Chicaña	Hino Fc
	Rural Chicaña	Hino Fc
Jueves	Urbano centro – norte	Hino GH
	Urbano centro -sur	Scannia
	Urbano Hospitales	Hino Fc
	Rural Chicaña	Hino Fc
	Rural Chicaña	Hino Fc
Viernes	Urbano centro – norte	Hino GH
	Urbano centro -sur	Scannia
	Urbano Hospitales	Hino Fc
	Rural Chicaña	Hino Fc
	Rural Chicaña	Hino Fc
Domingo	Urbano centro – norte	Hino GH
	Urbano Hospitales	Hino Fc
	Rural Chicaña	Hino Fc
	Rural Chicaña	Hino Fc

Nota: Elaboración propia

La entrevista con el director de la unidad de gestión ambiental del municipio confirmó que, con el crecimiento poblacional actual del 2,24% anual, los problemas de recolección se intensificarán si no se implementan soluciones tecnológicas para optimizar el servicio.

1.2. Proceso de recolección de basura

La gestión de residuos sólidos urbanos (GRSU) comprende el conjunto de actividades relacionadas con la recolección, transporte, tratamiento y disposición final de los desechos generados por las actividades humanas en zonas urbanas (Méndez, 2020). Una gestión eficiente implica la reducción de residuos, el aprovechamiento de materiales reciclables, y la minimización del impacto ambiental. Diversos estudios han demostrado que una planificación ineficiente de las rutas de recolección aumenta los costos operativos, el consumo de combustible y las emisiones contaminantes (Silva et al., 2017). En este contexto, la optimización de rutas mediante tecnologías de información y comunicación (TIC) puede mejorar significativamente la eficiencia del servicio (Almeida et al., 2019).

Además, la gestión eficiente de residuos contribuye a la sostenibilidad ambiental al disminuir la cantidad de desechos enviados a los vertederos, promoviendo la economía circular y la reutilización de materiales. Según la Organización de las Naciones Unidas (2021), implementar un sistema de recolección inteligente puede reducir hasta en un 30% los costos operativos y mejorar la cobertura del servicio.

En el contexto de Yantzaza, la GRSU enfrenta desafíos particulares debido al rápido crecimiento poblacional y la expansión urbana. Con una generación per cápita de aproximadamente 0.85 kg diarios por habitante, la ciudad debe gestionar eficientemente cerca de 21 toneladas diarias de residuos sólidos. La planificación de rutas de recolección se vuelve

crucial en este escenario, ya que los sistemas tradicionales basados en rutas fijas y predefinidas suelen ser ineficientes ante la dinámica urbana cambiante.

1.3. Algoritmos considerados para optimización de recorridos por Geolocalización

1.3.1. Algoritmos de trazabilidad de rutas.

Los algoritmos de trazabilidad de rutas pueden definirse como métodos computacionales que permiten determinar las trayectorias óptimas para vehículos de transporte, considerando múltiples restricciones operativas y objetivos de optimización. Según Granada (2005), "la logística es un método de dirección y gestión que determina y coordina en forma óptima el producto correcto, el cliente correcto, el lugar correcto y el tiempo correcto"

Estos algoritmos abordan principalmente dos problemas fundamentales de optimización combinatoria: el Problema del Viajante (TSP) y el Problema de Ruteo de Vehículos (VRP). Como indica Laporte (1992), "el Problema de Ruteo de Vehículos puede describirse como el problema de diseñar rutas óptimas de entrega o recolección desde uno o varios depósitos hacia un número de ciudades o clientes geográficamente dispersos, sujeto a restricciones laterales"

¿Para qué sirven?

Los algoritmos de trazabilidad permiten una utilización más eficiente de los recursos logísticos disponibles. Granada (2005) indica que "la logística añade valor al producto y tiene un papel sobresaliente en el desarrollo empresarial como elemento de enlace entre oferta y demanda"

Uso de los algoritmos de trazabilidad de rutas

Estos algoritmos se implementan en diversos sectores:

- **Industria alimentaria:** Para rastrear el origen y manejo de productos, garantizando la seguridad alimentaria.
- **Transporte y logística:** En la planificación y optimización de rutas de distribución, reduciendo costos y tiempos de entrega.
- **Salud:** En la gestión de medicamentos y equipos médicos, asegurando su correcta distribución y almacenamiento.

La implementación efectiva de estos algoritmos requiere sistemas de identificación (como etiquetas RFID), herramientas para la captura de datos y software especializado para la gestión y análisis de la información recopilada.

Entre los principales algoritmos de optimización para optimizar rutas de flotas vehiculares tenemos las siguientes:

1.3.2. *Dijkstra*

Encuentra la ruta más corta en un grafo ponderado (Dijkstra, 1959).

Tabla 2

Ventajas y desventajas de Dijkstra

Ventajas	Desventajas
Garantizar solución óptima	Solo caminos punto a punto
Tiempo de ejecución predecible	No maneja múltiples vehículos
Implementación simple	No considera capacidades
Bajo consumo de memoria	No adaptable en tiempo real

Nota: Elaboración propia

En uso VRP, solo como subrutina para hacer cálculos de distancias más cortas entre puntos específicos

1.3.3. A^*

Mejora Dijkstra incorporando heurísticas para acelerar la búsqueda (Hart et al., 1968).

Tabla 3

*Ventajas y desventajas de A^**

Ventajas	Desventajas
Búsqueda dirigida eficiente	Requiere heurística admisible
Óptimo con heurística admisible	Solo para problemas de camino único
Mejor que Dijkstra en grafos grandes	No escalable a múltiples objetivos
Ampliamente estudiado	Memoria exponencial en casos

Nota: Elaboración propia

En uso VRP, se usa para elementos de ruteo individual, mas no adecuado para optimización global de flotas.

1.3.4. ACO

Simula el comportamiento de las hormigas para encontrar rutas eficientes (Dorigo & Gambardella, 1997).

Tabla 4

Ventajas y desventajas de ACO

Ventajas	Desventajas
Adaptabilidad dinámica	Convergencia dependiente de Parámetros
Construcción probabilística de soluciones	Puede estancarse en soluciones subóptimas
Inspiración natural intuitiva	juste delicado de feromonas
Bueno para problemas dinámicos	Memoria alta para matrices grandes

Nota: Elaboración propia

En uso de VRP, es un contendiente entre las opciones, mostrando precisión y confiabilidad en problemas de rutas de vehículos.

1.3.5. Recocido Simulado (SA)

Permite escapar de óptimos locales mediante la aceptación probabilística de soluciones subóptimas (Kirkpatrick et al., 1983).

Tabla 5

Ventajas y desventajas de Recorrido simulado

Ventajas	Desventajas
Evita óptimos locales	Convergencia lenta
Simplicidad Conceptual	Parámetros críticos (temperatura)
Flexible para múltiples objetivos	Solución única por ejecución
Bueno para problemas combinatorios	Difícil paralización

Nota: Elaboración propia

1.3.6. Algoritmo Genérico (GA)

Es una técnica de búsqueda en los principios de selección natural y genética (Fraser, 1957).

Tabla 6

Ventajas y desventajas Algoritmo Genérico

Ventajas	Desventajas
Evita óptimos locales	Convergencia lenta
Simplicidad Conceptual	Parámetros críticos (temperatura)
Flexible para múltiples objetivos	Solución única por ejecución
Bueno para problemas combinatorios	Difícil paralización

Nota: Elaboración propia

En uso de VRP, es óptimo para problemas de ruteo con restricciones complejas, requiere de ajuste cuidadoso de parámetros.

Elección Algoritmo híbrido GA-ACO

La elección del algoritmo híbrido GA-ACO se debe a su capacidad para adaptarse a cambios dinámicos, como variaciones en el tráfico o la inclusión de nuevos puntos de recolección, estos ahora llamados nodos, nos permiten analizar las distancia y el tiempo transcurrido entre nodos, para poder eliminar datos redundantes y evitar circulación en bucles.

Además, permite considerar restricciones complejas como las capacidades de los vehículos y los tiempos de servicio dándonos resultados más optimos en tiempos y recorridos de las rutas.

1.4. Scrum como metodología para el desarrollo de proyectos que optimizan el recorrido de rutas.

Scrum proporciona un marco ideal para desarrollar sistemas de optimización de rutas debido a su capacidad para manejar la complejidad técnica de forma incremental mientras mantiene el foco en entregar valor a los usuarios finales. La clave del éxito está en equilibrar el desarrollo de algoritmos sofisticados con interfaces usables y prácticas, construyendo el sistema de manera que cada incremento ofrezca mejoras tangibles en la eficiencia de las rutas.

La implementación de Scrum en proyectos de optimización de rutas ofrece un marco ágil que se adapta perfectamente a las complejidades y requisitos cambiantes de estos sistemas.

Ventajas de Scrum para proyectos de optimización de rutas

1.4.1 Desarrollo incremental de funcionalidades: Permite entregar primero las características más críticas (como carga de archivos GPX o visualización básica de mapas) y luego incorporar algoritmos más complejos de optimización.

1.4.2 Adaptabilidad a cambios: En entornos logísticos, las variables como restricciones de ruta, factores ambientales o regulaciones pueden cambiar, y Scrum permite pivotar rápidamente.

1.4.3 Retroalimentación continua de usuarios reales: Los choferes y administradores pueden probar versiones tempranas del sistema y proporcionar información valiosa sobre la usabilidad en condiciones reales.

1.4.4 Gestión de complejidad técnica: Los algoritmos de optimización de rutas pueden ser complejos; los sprints cortos permiten abordar esta complejidad gradualmente.

Aplicación específica de Scrumm en proyectos de Rutas.

Una de las ventajas importantes de scrum es su versatilidad en entornos de desarrollo de nuevas tecnologías de innovación, para lo que es necesario determinar todos los aspectos puntuales de esta metodología.

1.4.5. Definición de roles

De acuerdo con Schwaber y Sutherland (2020): “Scrum define un conjunto de responsabilidades claras dentro de un Equipo Scrum: el Product Owner, el Scrum Master y los Developers. Cada rol está diseñado para optimizar la transparencia, inspección y adaptación” (p. 5).

1.5. Fundamentos de la comunicación el la ingenieria de software

Sommerville (2016) enfatiza que "la comunicación efectiva es crucial en todos los aspectos de la ingeniería de software, desde la especificación de requisitos hasta el

mantenimiento del sistema" (p. 45). Las definiciones claras actúan como un lenguaje común que facilita esta comunicación, reduciendo ambigüedades que pueden resultar en interpretaciones erróneas y, consecuentemente, en defectos del software.

Pressman y Maxim (2020) sostienen que "la falta de precisión en las definiciones durante las fases tempranas del desarrollo puede amplificarse exponencialmente en las etapas posteriores, resultando en costos significativos de corrección" (p. 78). Esta perspectiva subraya la importancia económica de establecer definiciones precisas desde el inicio del proyecto.

1.6. Impacto en la arquitectura y diseño de Software

Según Fowler (2018), "la arquitectura de software se basa fundamentalmente en la claridad de los conceptos y sus interrelaciones" (p. 123). Las definiciones bien establecidas permiten:

- **Modularidad coherente:** Cada componente tiene un propósito claramente definido
- **Interfaces consistentes:** Las interacciones entre módulos siguen patrones predecibles
- **Reutilización efectiva:** Los componentes pueden ser reutilizados cuando su función está claramente definida

1.7. Calidad del Software y Mantenibilidad

Martin (2017) en "Clean Code" argumenta que "el código limpio es código que ha sido escrito por alguien que se preocupa por el siguiente desarrollador que lo leerá" (p. 12). Las definiciones claras son parte integral de esta filosofía, ya que proporcionan el contexto necesario para la comprensión del código.

ISO/IEC 25010:2011 establece que la mantenibilidad del software depende en gran medida de la comprensibilidad del sistema, donde las definiciones claras juegan un papel crucial en facilitar la comprensión por parte de nuevos desarrolladores (ISO/IEC, 2011).

1.8. Gestión de Requisitos y Especificaciones

Wieggers y Beatty (2013) destacan que "los requisitos mal definidos son la causa principal del fracaso de proyectos de software" (p. 34). Un glosario de términos y definiciones precisas es considerado una práctica esencial en la ingeniería de requisitos. La importancia de las definiciones se refleja en metodologías establecidas como:

- **Domain-Driven Design (DDD):** Evans (2003) propone el "Ubiquitous Language" como piedra angular del desarrollo exitoso.
- **Scrum:** Schwaber y Sutherland (2020) enfatizan la importancia de la "Definition of Done" como elemento definitorio del éxito.

1.9. Perspectiva de Gestion de Proyectos

El Project Management Institute (2017) en su PMBOK Guide establece que "la definición clara del alcance del proyecto, incluyendo la terminología utilizada, es fundamental para evitar el scope creep y asegurar el éxito del proyecto" (p. 156).

Boehm (1981) en su modelo COCOMO demostró empíricamente que los errores introducidos en las fases tempranas del desarrollo (donde las definiciones juegan un papel crucial) pueden costar hasta 100 veces más corregir en fases posteriores.

1.10. Evidencia empírica

Estudios recientes han demostrado el impacto cuantificable de las definiciones en proyectos de software:

Jones (2013) reporta que proyectos con glosarios de términos bien definidos tienen un 40% menos probabilidad de experimentar defectos relacionados con malentendidos de requisitos.

Un estudio de Standish Group (2020) indica que la claridad en las definiciones de proyecto es uno de los cinco factores más importantes para el éxito de proyectos de software.

Product Backlog para sistemas de optimización de rutas

1.11. Product backlog para sistemas de optimización de rutas

Un Product Backlog típico podría organizarse en las siguientes categorías:

1.11.1. Gestión geoespacial:

- Carga y validación de archivos GPX.
- Visualización de puntos en mapa.
- Edición manual de rutas.

1.11.2. Algoritmos de optimización:

- Implementación de algoritmo básico de ruta más corta.
- Optimización con restricciones (ventanas de tiempo, capacidad).
- Mejoras de rendimiento para grandes conjuntos de datos.

1.11.3. Interfaz de navegación:

- Visualización de ruta actual.
- Seguimiento GPS en tiempo real.
- Alertas de desvío.

1.11.4. Gestión de usuarios y vehículos:

- Registro y autenticación.
- Gestión de flota de vehículos.

- Asignación de rutas a choferes.

1.11.5. Análisis y reportes:

- Registro de rutas completadas.
- Métricas de eficiencia (tiempo, distancia, combustible).
- Exportación de informes.

1.12. Aplicaciones de sistemas de información geográfica (SIG) en la GRSU

El uso de SIG facilita la visualización, análisis y gestión de datos espaciales para mejorar la planificación de rutas (Longley et al., 2015). Estos sistemas permiten identificar puntos críticos, evaluar la cobertura del servicio y simular escenarios de recolección, optimizando así los recursos y la logística municipal.

La integración de SIG con algoritmos de optimización posibilita el diseño de rutas dinámicas que se adaptan a las condiciones del tráfico y a la variabilidad en la generación de residuos. Esta combinación mejora la eficiencia operativa y reduce los tiempos de recolección, beneficiando tanto a los municipios como a los ciudadanos (González y Ramírez, 2021).

En Yantzaza, los SIG proporcionan ventajas significativas como:

1.12.1. Visualización espacial de datos: Permite representar gráficamente las rutas, contenedores y áreas de cobertura.

1.12.2. Análisis de accesibilidad: Identificación de zonas con difícil acceso para los vehículos de recolección.

1.12.3. Monitoreo en tiempo real: Seguimiento de la flota y adaptación de rutas según condiciones cambiantes.

1.12.4. Planificación estratégica: Optimización de ubicación de contenedores y puntos de transferencia.

1.12.5. Gestión de datos geoespaciales: Almacenamiento y procesamiento de información geo-referenciada para análisis posteriores.

CAPÍTULO II:

ANÁLISIS Y DISEÑO

2.1. Requerimientos

El levantamiento de requerimientos es una parte fundamental en el desarrollo de proyectos, estos nos dan direcciones sobre las necesidades actuales que enfrenta la unidad ambiental del GAD Yantzaza. El enfoque de los requerimientos se basó en entrevistas semiestructuradas con la coordinadora de la unidad ambiental del GAD de Yantzaza, los choferes y funcionarios.

2.1.1. Fuentes de recolección de Información

Identificación de consultas frecuentes

Esta fase del levantamiento se realizó a través de entrevistas dirigidas a la coordinadora de la unidad ambiental del GAD Yantzaza y choferes de los camiones recolectores. Esta forma de recolección de información me permitió tener una visibilidad clara de los problemas frecuentes con los recorridos de los camiones de basura del GAD.

Perfil de personal entrevistado:

- Ing. Karla Yadyra Montaña Viñan - Coordinadora de la unidad de gestión ambiental del GAD Yantzaza.
- Ing. Paola Mendoza Moendoza – Técnica de la unidad de gestión ambiental del GAD Yantzaza.
- Roberto Encarnación – Chofer de camion recolector de basura del GAD Yantzaza.
- Julio Minga - Chofer de camion recolector de basura del GAD Yantzaza.
- Darwin González - Chofer de camion recolector de basura del GAD Yantzaza.

Diseño del Instrumento de Entrevista

Para el levantamiento de esta información diseñe un instrumento entrevista que contiene preguntas generales aplicables a todos los funcionarios del GAD involucrados. Esto me permitió tener una visión clara e integral de las necesidades de la unidad tanto como administradores como choferes, La estructura a detalle de este instrumento se encuentra en Anexo A.

Requerimientos funcionales del sistema:

- Funcionalidades específicas requeridas según necesidades de la unidad de gestión ambiental
- Optimizar los recorridos actuales que hacen los camiones de basura.
- Control de los recorridos que realizan los choferes de los camiones de basura.
- Revisión de Análisis comparativo de datos de tiempo y consumo de combustible.

Metodología de Entrevistas

Para el levantamiento de información se diseñó un instrumento de entrevista semiestructurado que combinaba preguntas generales aplicables a todos los entrevistados con preguntas específicas dirigidas a los coordinadores académicos. Esta metodología permitió obtener tanto una visión integral de las necesidades comunicacionales de la universidad como insights específicos de cada carrera. La estructura detallada de la entrevista se encuentra disponible en el Anexo A.

2.1.2. Reglas de negocio.

2.1.2.1. Gestión de usuarios y control de acceso

2.1.2.1.1. Clasificación de Roles

RN-001: El sistema debe soportar cuatro roles diferenciados: Administrador, Técnico, Coordinador y Chofer.

RN-002: Cada usuario debe poseer una cédula de identidad única e irrepetible en el sistema.

RN-003: Los choferes deben registrar obligatoriamente su tipo de licencia de conducir.

RN-004: Solo usuarios con estado "activo" pueden acceder al sistema.

2.1.2.1.2. Privilegios por rol

RN-005: Los administradores tienen acceso completo a todas las funcionalidades del sistema.

RN-006: Los técnicos pueden gestionar contraseñas y estados de usuario, excepto cuentas de administrador.

RN-007: Los coordinadores supervisan operaciones sin modificar configuraciones críticas del sistema.

RN-008: Los choferes acceden únicamente a funciones de navegación y registro de recorridos propios.

2.1.2.1.3. Optimización y creación de rutas

RN-009: Las rutas se generan exclusivamente a partir de archivos GPX válidos.

RN-010: Cada ruta debe poseer un nombre único en el sistema.

RN-011: El sistema debe aplicar automáticamente algoritmos de optimización al crear rutas.

RN-012: Se deben registrar métricas de optimización: distancia original, distancia optimizada, porcentaje de mejora y tiempo estimado ahorrado.

RN-013: El sistema ofrece tres niveles de optimización: Básico, Medio y Avanzado.

RN-014: La optimización debe eliminar bucles redundantes en las rutas.

RN-015: Se debe calcular el tiempo ahorrado basado en una velocidad promedio de 40 km/h.

RN-016: El sistema debe registrar la cantidad de puntos de ruta reducidos durante la optimización.

2.1.2.1.4. Ejecución y seguimiento de rutas

RN-017: Un chofer puede tener únicamente una ruta activa simultáneamente.

RN-018: Para iniciar una ruta, el chofer debe seleccionar un vehículo disponible y registrar el nivel inicial de combustible.

RN-019: El nivel de combustible se registra en una escala de 1 a 4 tanques.

RN-020: El sistema debe registrar automáticamente la fecha y hora de inicio de cada ruta.

2.1.2.1.5. Restricciones operativas

RN-021: El sistema debe generar instrucciones de navegación turn-by-turn automáticamente.

RN-022: Las instrucciones deben incluir direcciones (norte, sur, este, oeste) e iconos direccionales.

RN-023: El sistema debe ofrecer navegación por voz opcional en idioma español.

RN-024: Se debe calcular y mostrar el progreso de ruta y tiempo estimado de llegada (ETA) en tiempo real.

RN-025: El sistema debe registrar posiciones GPS cada 30 segundos para evitar sobrecarga del servidor.

RN-026: Debe alertar si el chofer se desvía más de 50 metros de la ruta planificada.

RN-027: El sistema debe detectar velocidad anormalmente baja por períodos prolongados.

RN-028: Se debe notificar la llegada al destino cuando el chofer esté dentro de un radio de 20 metros.

2.1.2.1.6. Finalización y registro de recorridos

RN-029: Para completar una ruta, es obligatorio registrar el nivel final de combustible.

RN-030: El sistema debe calcular automáticamente el consumo de combustible (nivel inicial - nivel final).

RN-031: Se debe generar automáticamente un mapa visual del recorrido realizado vs la ruta planificada.

RN-032: Los choferes pueden agregar notas opcionales sobre incidentes o condiciones del recorrido.

RN-033: La cancelación de rutas requiere especificar obligatoriamente el motivo.

RN-034: El tracking GPS se debe mantener hasta el momento exacto de la cancelación.

RN-035: Los datos de rutas canceladas se conservan para análisis estadísticos.

2.1.2.1.7. *Gestión de Vehículos y asignaciones*

RN-036: Cada vehículo debe registrar marca, modelo, año y placa única.

RN-037: Las placas de vehículos no pueden duplicarse en el sistema.

RN-038: Solo vehículos con estado "activo" están disponibles para asignación a rutas.

RN-039: Un chofer puede tener asignados múltiples vehículos simultáneamente.

RN-040: Un vehículo puede ser utilizado por múltiples choferes.

RN-041: Solo puede existir una asignación activa por cada combinación chofer-vehículo.

2.1.2.1.8. *Almacenamiento y trazabilidad de datos*

RN-042: El sistema debe almacenar posición GPS (latitud, longitud), precisión, velocidad y timestamp de cada punto registrado.

RN-043: Los datos de tracking se almacenan en formato JSON en la base de datos.

RN-044: Se debe mantener un historial completo de todos los recorridos realizados.

RN-045: El sistema debe generar mapas HTML independientes para cada recorrido completado.

RN-046: Los mapas deben mostrar marcadores de inicio, fin y puntos de control del recorrido.

RN-047: Se debe visualizar la comparación entre ruta planificada y ruta real ejecutada.

2.1.2.1.9. *Reportes y análisis de rendimiento*

RN-048: El sistema debe calcular eficiencia de combustible por vehículo, chofer y ruta.

RN-049: Se deben generar reportes mensuales consolidados de consumo total.

RN-050: El cálculo de eficiencia se basa en kilómetros por galón aproximado (asumiendo 10 galones por tanque).

RN-051: El sistema debe generar rankings de choferes por eficiencia de combustible.

RN-052: Se deben identificar vehículos con mejor y peor rendimiento operativo.

RN-053: El sistema debe analizar qué rutas son más o menos eficientes en términos de consumo.

2.1.2.1.10. Integridad y validaciones del sistema

RN-054: No se pueden eliminar usuarios que tengan rutas activas asignadas.

RN-055: Las rutas no se pueden eliminar si tienen recorridos (completions) asociados.

RN-056: Los vehículos no se pueden desactivar mientras estén siendo utilizados en rutas activas.

2.1.2.1.11. Validaciones de datos

RN-057: Los archivos GPX deben ser válidos y contener coordenadas geográficas correctas.

RN-058: El sistema debe validar que las coordenadas GPS estén dentro de rangos geográficos válidos.

RN-059: Se debe verificar la conectividad a internet antes de enviar actualizaciones de posición.

2.1.2.1.12. Seguridad y privacidad

RN-060: Los choferes solo pueden acceder a sus propios recorridos y rutas asignadas.

RN-061: Las ubicaciones GPS en tiempo real solo son visibles para coordinadores y administradores.

RN-062: El acceso a reportes consolidados está restringido según el rol del usuario.

RN-063: Todas las acciones críticas (inicio/finalización de rutas, cambios de configuración) deben quedar registradas con timestamp y usuario responsable.

RN-064: El sistema debe mantener logs de acceso y actividades para auditorías posteriores.

2.1.3. Requerimientos funcionales

Tabla 7

Requerimientos funcionales

ID	Nombre del Requerimiento	Descripción	Entrada	Salida	Dependencias
RF-01	Inicio de sesión	El sistema debe permitir que los usuarios se autentiquen mediante un nombre de usuario y una contraseña.	Nombre de usuario, contraseña.	Acceso al sistema, token de autenticación.	RF-02 (Registro de usuario)
RF-02	Registro de usuario	El sistema debe permitir a los nuevos usuarios registrarse proporcionando información como nombre, correo electrónico y contraseña.	Nombre, correo electrónico, contraseña.	Confirmación de registro, acceso al sistema.	No tiene dependencias previas.
RF-03	Carga de archivos GPX	El sistema debe permitir la carga de archivos en formato GPX que contengan rutas de vehículos para su análisis y optimización.	Archivo GPX.	Confirmación de carga, visualización de datos del archivo cargado.	RF-01 (Inicio de sesión).
RF-04	Cálculo y optimización de rutas	El sistema debe calcular las rutas óptimas basadas en los datos de los archivos GPX cargados utilizando un algoritmo de menor distancia (por ejemplo, Dijkstra o A*).	Datos del archivo GPX cargado.	Ruta optimizada (distancia, tiempo).	RF-03 (Carga de archivos GPX).
RF-05	Análisis de rutas	El sistema debe analizar las rutas cargadas y generar estadísticas como distancia recorrida, tiempo de viaje, entre otros.	Datos de las rutas cargadas.	Estadísticas de las rutas (distancia, tiempo).	RF-03 (Carga de archivos GPX).
RF-06	Generación de ruta óptima	Tras el análisis, el sistema debe generar la ruta óptima considerando la distancia más corta y los tiempos de viaje.	Estadísticas de las rutas, datos de análisis.	Ruta optimizada generada, con detalles de distancia y tiempo de recorrido.	RF-05 (Análisis de rutas).

RF-07	Navegación en la ruta optimizada	El sistema debe permitir al usuario navegar en la ruta optimizada con instrucciones paso a paso y recálculo dinámico de rutas similar a Waze.	Ruta optimizada, ubicación en tiempo real.	Instrucciones paso a paso, flechas direccionales, recálculo de ruta si es necesario.	RF-06 (Generación de ruta óptima).
RF-08	Almacenamiento de rutas optimizadas	El sistema debe guardar las rutas optimizadas generadas en formato JSON para su futura consulta.	Ruta optimizada generada.	Archivo JSON con la ruta optimizada.	RF-06 (Generación de ruta óptima).
RF-09	Selección de vehículo	El sistema debe permitir a los choferes seleccionar el tipo de vehículo que utilizarán para el recorrido.	Selección de vehículo de una lista predefinida.	Registro del vehículo asociado al recorrido.	RF-07 (Navegación en la ruta optimizada).
RF-10	Dashboard de analíticas	El sistema debe proporcionar un dashboard con estadísticas sobre rutas completadas, tiempo ahorrado, combustible ahorrado y eficiencia general.	Datos de rutas completadas.	Visualización gráfica de métricas y KPIs.	RF-08 (Almacenamiento de rutas optimizadas).
RF-11	Gestión de usuarios	El sistema debe permitir a los administradores gestionar usuarios, asignar roles y permisos.	Datos de usuario, rol asignado.	Usuarios creados/modificados con roles específicos.	RF-02 (Registro de usuario).
RF-12	Generación de reportes	El sistema debe generar reportes exportables sobre la eficiencia de las rutas y el desempeño de los operadores.	Parámetros de filtrado (fecha, ruta, operador).	Reporte en formato PDF o Excel.	RF-10 (Dashboard de analíticas).

Nota: Elaboración propia

2.1.4. Requerimientos no funcionales

Tabla 8

Requerimientos no funcionales

ID	Nombre del Requerimiento	Descripción	Entrada	Salida	Dependencias
RNF-01	Seguridad en el inicio de sesión	El sistema debe proteger los datos de inicio de sesión mediante cifrado seguro, como HTTPS, para garantizar la privacidad de las credenciales de los usuarios.	Datos de inicio de sesión (nombre de usuario, contraseña).	Datos cifrados, sesión segura.	RF-01 (Inicio de sesión).
RNF-02	Rendimiento	El sistema debe procesar y generar rutas optimizadas en menos de 5 segundos por cada archivo GPX cargado para asegurar una experiencia de usuario rápida.	Archivos GPX cargados, solicitud de optimización.	Ruta optimizada generada en menos de 5 segundos.	RF-04 (Cálculo y optimización de rutas).
RNF-03	Escalabilidad	El sistema debe ser capaz de manejar múltiples usuarios simultáneamente sin afectar el rendimiento, permitiendo futuras ampliaciones en la cantidad de datos o usuarios.	Solicitudes simultáneas de carga de archivos y generación de rutas.	Respuesta rápida y sin pérdida de rendimiento a múltiples usuarios simultáneos.	No tiene dependencias previas.
RNF-04	Disponibilidad	El sistema debe estar disponible al menos un 99.5% del tiempo durante las horas laborales de la ciudad.	Solicitudes de acceso al sistema.	Acceso garantizado al sistema en el 99.5% del tiempo durante las horas laborales.	No tiene dependencias previas.

RNF-05	Compatibilidad	El sistema debe ser accesible desde navegadores modernos (Chrome, Firefox, Edge) y dispositivos móviles con sistema operativo Android o iOS.	Dispositivo con navegador moderno o dispositivo móvil.	Interfaz accesible y funcional en los dispositivos y navegadores mencionados.	No tiene dependencias previas.
RNF-06	Usabilidad	El sistema debe ofrecer una interfaz intuitiva y fácil de usar para los operadores, permitiendo un uso sin capacitación extensa.	Interfaz gráfica del sistema.	Interfaz fácil de usar, sin necesidad de capacitación extensa.	No tiene dependencias previas.
RNF-07	Integridad de los datos	El sistema debe asegurar que no haya pérdida de datos durante el proceso de carga, optimización o generación de rutas.	Datos de las rutas cargadas, rutas optimizadas generadas.	Datos completos y consistentes, sin pérdida de información.	RF-03 (Carga de archivos GPX), RF-04 (Cálculo y optimización de rutas).
RNF-08	Mantenibilidad	El código del sistema debe ser modular y fácilmente mantenible, permitiendo futuras actualizaciones o cambios sin generar impacto significativo en el servicio.	Código fuente, documentación técnica.	Código limpio, modular y bien documentado para facilitar futuras actualizaciones.	No tiene dependencias previas.
RNF-09	Tolerancia a fallos	El sistema debe ser capaz de recuperarse de fallos sin pérdida de datos y mantener un funcionamiento estable.	Eventos inesperados como desconexiones o errores.	Recuperación automática y notificación al usuario.	No tiene dependencias previas.
RNF-10	Eficiencia energética	La aplicación móvil debe optimizar el uso de recursos del dispositivo para minimizar el consumo de batería durante la navegación.	Uso de GPS y recursos del sistema.	Consumo moderado de batería durante el uso prolongado.	RF-07 (Navegación en la ruta optimizada).

Nota: Elaboración propia

2.2. Desarrollo de la Metodología

Se utilizará la metodología SCRUM para el desarrollo del sistema, permitiendo una implementación progresiva de funcionalidades y validación continua de resultados.

2.2.1. Proceso SCRUM

Tabla 9

Proceso scrumm en optimización de rutas con ia

FASE SCRUM	ACTIVIDAD
Product Backlog Creation	Definición de historias de usuario para gestión de usuarios vehículos rutas y navegación
Sprint Planning 1	Planificación del sistema de autenticación roles y gestión básica de usuarios
Sprint 1	Desarrollo del sistema de login creación de usuarios y roles (Admin Técnico Coordinador Chofer)
Sprint 2	Desarrollo del CRUD de vehículos asignaciones y validaciones de placas únicas
Sprint3	Desarrollo de carga GPX creación de rutas y generación de mapas
Sprint 4	Desarrollo de algoritmos de optimización, eliminación de bucles y cálculo de métricas
Sprint 5	Desarrollo de inicio de rutas asignación de vehículos y tracking GPS
Sprint 6	Desarrollo de instrucciones direccionales navegación por voz y alertas inteligentes
Sprint 7	Desarrollo de completion de rutas cálculo de consumo y mapas de recorridos
Sprint 8	Desarrollo de dashboards reportes PDF y análisis comparativo
Sprint 9	Desarrollo de interfaz móvil optimizada animaciones y mejoras visuales
Product Backlog Refinement	Priorización de funcionalidades adicionales y mejoras post-MVP
Release Planning	Planificación de despliegue en producción y capacitación de usuarios

Nota: Elaboración propia

2.2.2. Inicio del Proyecto (Product Backlog)

Tabla 10

Historias de usuario

Enunciado de la historia				Criterios de Aceptación				
Identificador de historias ID	Rol	Característica / Funcionalidad	Razón / Resultado	Número (#) de Escenario	Criterio de Aceptación (Título)	Contexto	Evento	Resultado / Comportamiento esperado
ID1	Coordinador departamento Ambiental	Gestiona, direcciona, coordina actividades de la unidad	Ver y analizar, flujos de datos y flujos de trayectoria y recorridos en tiempo real	1	Acceso a sistema	No realiza acceso a sistema por que no existe aplicación informática en la unidad	ingreso al sistema	El Coordinador del departamento ambiental ingresara credenciales para acceder a las funciones del sistema
				2	Visualización del dashboard	No realiza visualización del dashboard porque no existe aplicación informática en la unidad	Visualización de herramientas gráficas que muestren las principales variables del sistema	El coordinador del departamento va a seleccionar una interfaz que desee visualizar según la métrica que necesite
				1	Agregar o editar usuario	No agrega usuario por que no existe aplicación informática en la unidad	Agregar nuevo o editar usuario	El Coordinador del departamento ambiental ingresara agregará y editará usuarios
				2	Agregar o editar vehículo	No agrega ni edita vehículo porque no existe aplicación informática en la unidad	Agregar nuevo o editar vehículo	El Coordinador del departamento ambiental agregará y editará vehículos

				3	Crear o eliminar rutas óptimas con ia	No crea ni elimina rutas optimas con ia porque no existe aplicación informatica en la unidad	Crear rutas optimizadas con ia	El Coordinador del departamento ambiental creará y eliminará rutas
ID2	Técnico de la unidad	Controla, gestiona, coordina actividades del personal	Da soporte y asistencia en caso de perder credenciales	4	Reestablecer contraseñas de usuarios	No reestablece las contraseñas de usuarios porque no existe aplicación informatica en la unidad	Reestablecer contraseñas de los usuarios	El técnico podrá reestablecer las contraseñas de los usuarios
ID3	Chofer de camión de basura	Conduce vehículo de recolección de basura	Realiza los recorridos de recolección de la ciudad de Yantzaza	5	Recorrer las rutas definidas por el administrador	Recorre rutas de forma empírica, porque no existe aplicación informatica en la unidad	Recorrer rutas óptimas con ia	El chofer de camión de basura podrá recorrer las rutas óptimas creadas por el administrador
				6	Agrega datos de combustible	No agrega datos de combustible, porque no existe aplicación informática en la unidad	Agregar datos de combustible	EL chofer agregará datos de combustible al iniciar y al finalizar la ruta
				7	Revisa sus recorridos	No revisa sus recorridos, porque no existe aplicación informática en la unidad	Revisar sus recorridos	EL chofer podrá visualizar sus recorridos realizados
ID4	Director del departamento ambiental	Mantiene la dirección del cumplimiento del personal en la unidad	Realiza monitoreo de la funcionalidad del sistema de recolección de basura	8	Visualización del dashboard	No realiza visualización del dashboard porque no existe aplicación informatica en la unidad	Visualización de herramientas gráficas que muestren las principales variables del sistema	El director del departamento va a seleccionar una interfaz que desee visualizar según la métrica que necesite

Nota: Elaboración propia

2.2.3. Definir el Product Backlog:

Tabla 11

Product Backlog

NRO	REQUERIMIENTO	PRIORIDAD (alto-medio-bajo)
1	Gestionar adecuadamente los usuarios del sistema.	Alto
2	Generar visualización de las métricas que se almacenan en el sistema	Bajo
3	Gestionar el registro de vehículos	Alto
4	Gestionar la generación de las rutas optimas	Alto
5	Planificación de recorridos para el proceso de recolección (ruta, chofer, vehículo, fecha)	Alto
6	Realizar el monitores del recorrido de los vehículos en las rutas asignas	Alto

Nota: Elaboración propia

2.2.4. Planificación del Sprint (Sprint Planning)

2.2.4.1. Daily Scrum

- **Duración:** 15 minutos.
- **Frecuencia:** Diariamente.
- **Participantes:** Equipo de Desarrollo y Scrum Master.
- **Objetivo:** Sincronizar actividades y crear un plan para las próximas 24 horas.

2.2.4.2. Sprint Review

- **Duración:** 2 horas.
- **Frecuencia:** Al final de cada Sprint.
- **Participantes:** Equipo Scrum completo y stakeholders invitados
- **Objetivo:** Inspeccionar el incremento y adaptar el Product Backlog si es necesario.

2.2.4.3. Sprint Retrospective

- **Duración:** 1.5 horas.
- **Frecuencia:** Al final de cada Sprint, después del Sprint Review.
- **Participantes:** Equipo Scrum completo.
- **Objetivo:** Inspeccionar el proceso y crear un plan de mejoras para el próximo Sprint.

2.2.4.4. Refinamiento del Backlog

- **Duración:** 2 horas.
- **Frecuencia:** A mitad de cada Sprint.
- **Participantes:** Product Owner y Equipo de Desarrollo.
- **Objetivo:** Revisar y refinar las historias de usuario para futuros Sprints.

2.2.5. Herramientas y Tecnologías

2.2.5.1. Gestión del Proyecto

- Jira para gestión de Sprints y seguimiento de historias de usuario.
- Confluence para documentación.
- Microsoft Teams para comunicación.

2.2.5.2. Desarrollo

- Frontend: React.js con Material-UI.
- Backend: Node.js con Express.
- Base de datos: PostgreSQL con extensión PostGIS.

- Optimización de rutas: algoritmos basados en OSRM (Open Source Routing Machine).
- Control de versiones: Git con GitHub.
- CI/CD: Jenkins.

2.2.5.3. Entornos

- Desarrollo: Servidores locales
- Pruebas: Servidor en la nube
- Producción: Infraestructura propia

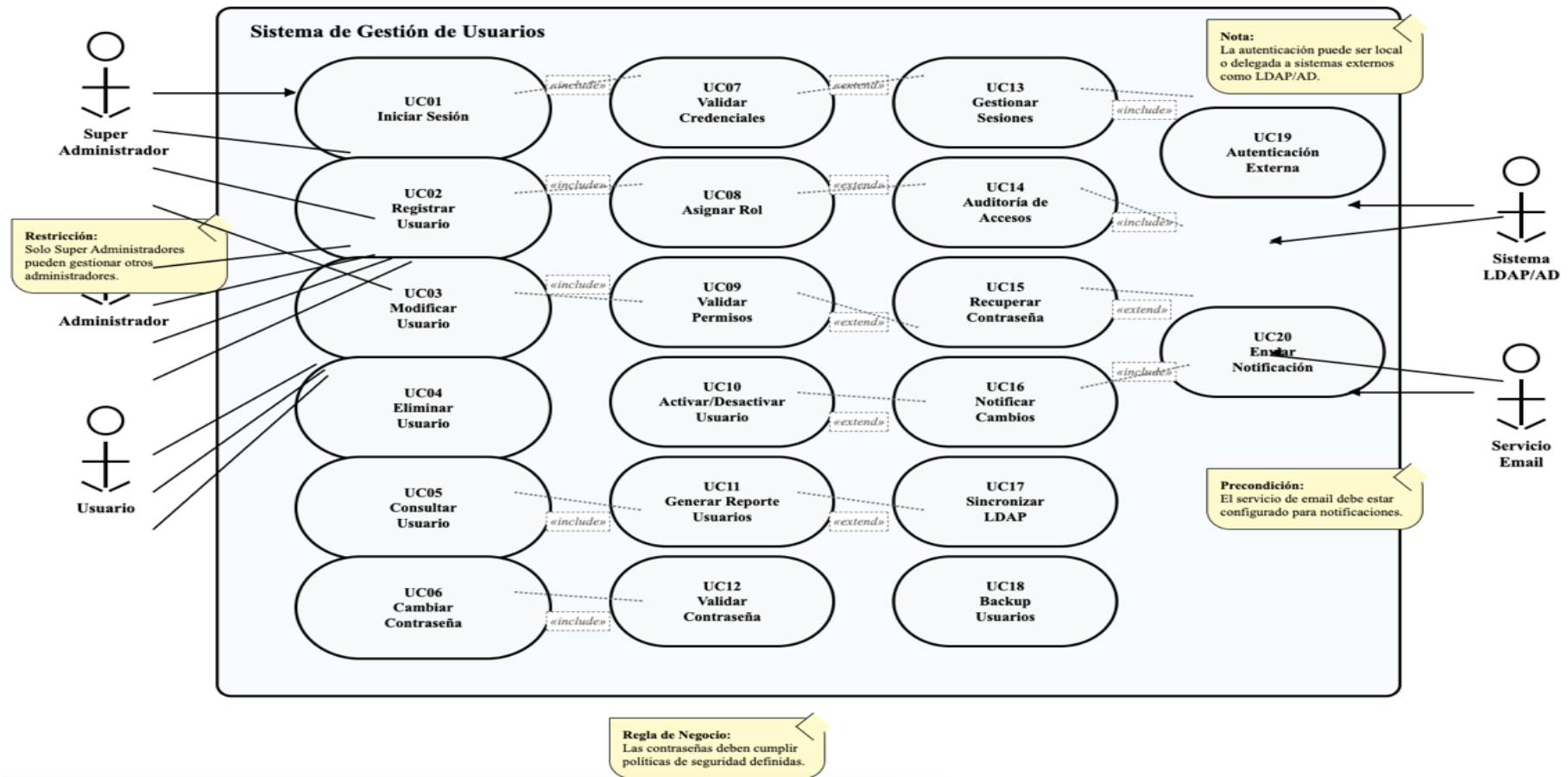
2.3. Diagramas del sistema

2.3.1. Diagrama de Caso de Uso

El siguiente diagrama ilustra las interacciones de los diferentes actores con el sistema de optimización de rutas, tanto usuarios como servicios:

Figura 1

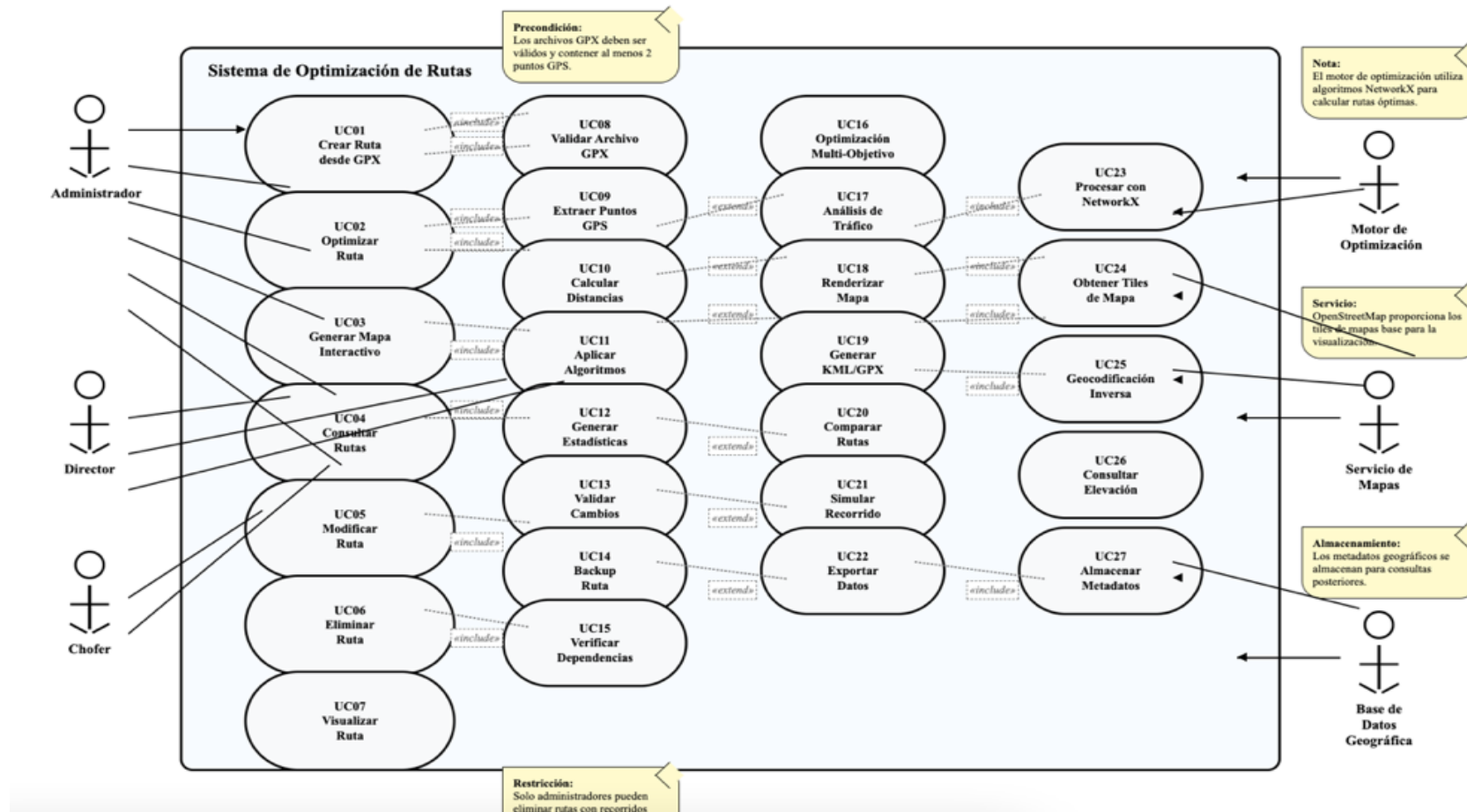
Diagrama de caso de uso gestión usuarios



Nota: Elaboración propia

Figura 2

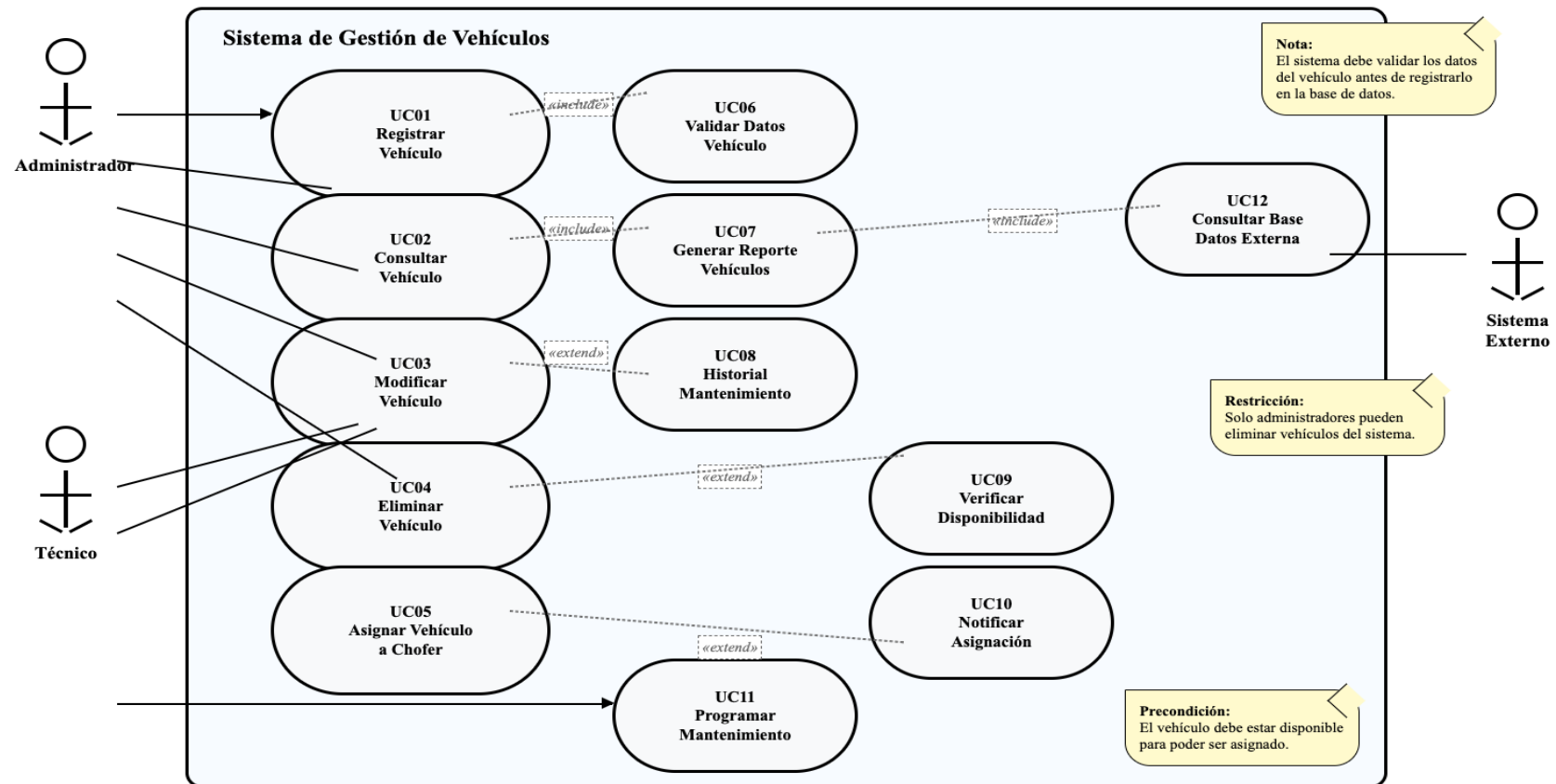
Diagrama caso de uso gestión de rutas optimas



Nota: Elaboración propia

Figura 3

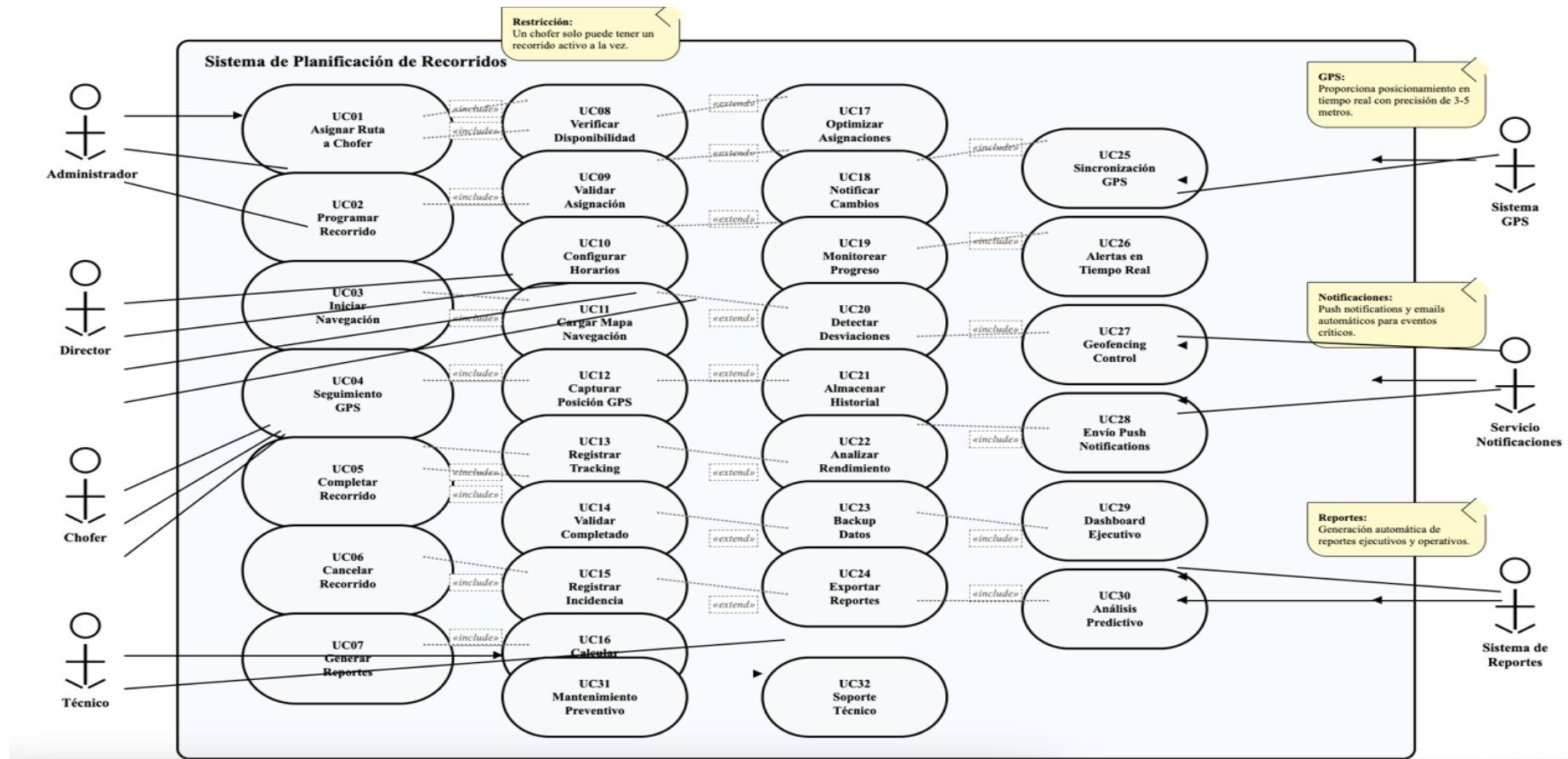
Diagrama caso de uso gestión vehículos



Nota: Elaboración propia

Figura 4

Diagrama caso de uso planificacion de recorridos



Nota: Elaboración propia

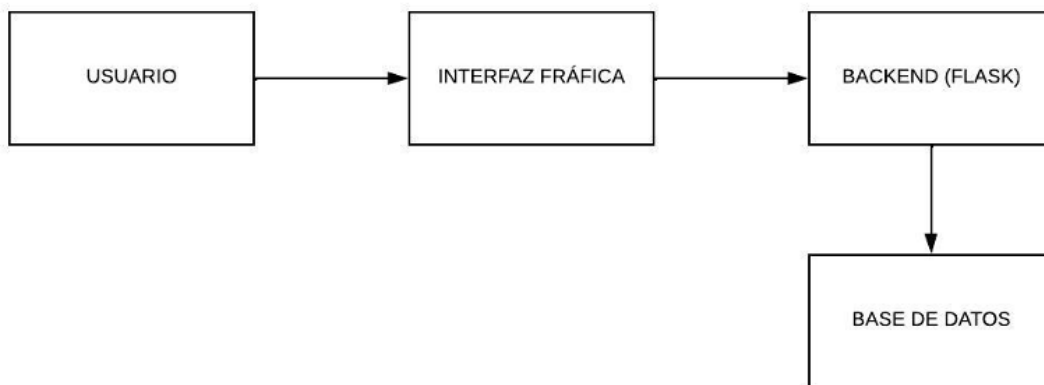
2.3.2. Planificación del Proceso SCRUM (Diagrama de Gantt)

2.3.2.1. Diagrama de Integración de Usuarios (Arquitectura del Sistema)

Arquitectura general del sistema basado en SCRUM se ilustra a continuación:

Figura 5

Arquitectura de sistema en SCRUM

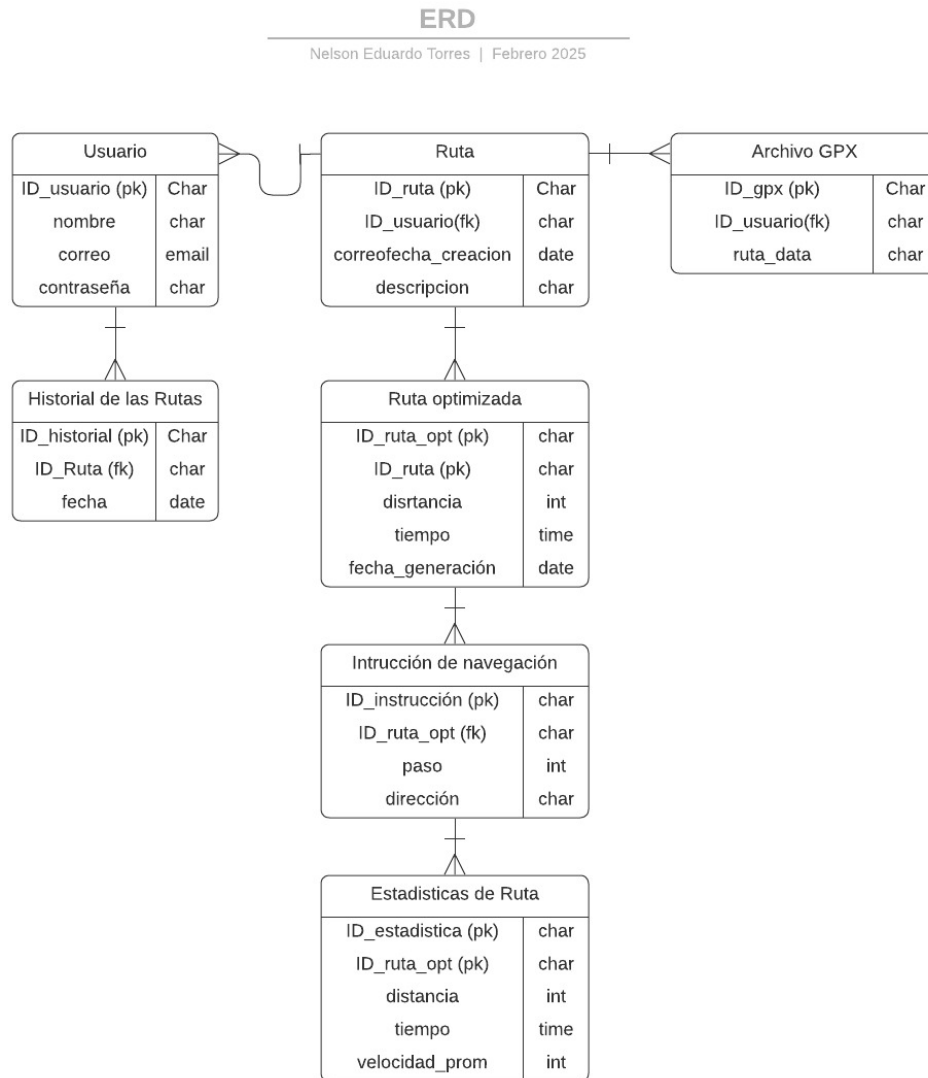


Nota: Elaboracion propia

Este diagrama muestra cómo interactúan los diferentes componentes del sistema, desde la capa de presentación (frontend) hasta la capa de datos (backend), pasando por los servicios de optimización y geolocalización.

2.3.3. Diagrama Entidad-Relación (ERM)

El siguiente diagrama ilustra la estructura de la base de datos:

Figura 6*Diagrama ERM*

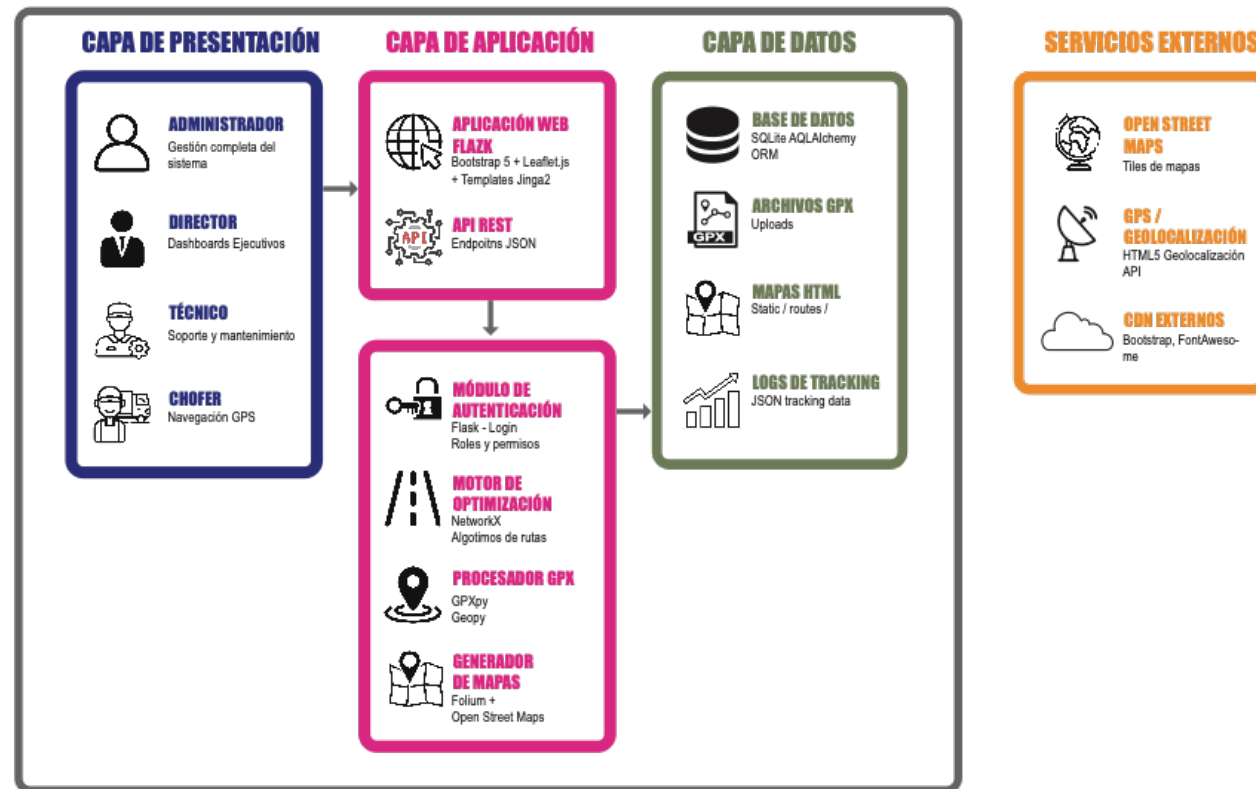
Nota: Elaboración propia.

El diagrama ERM muestra las entidades principales (Usuario, Ruta, Recorrido, Vehículo, etc.) y sus relaciones, definiendo claramente la estructura de datos que soportará el sistema.

Arquitectura

Figura 7

Arquitectura del sistema



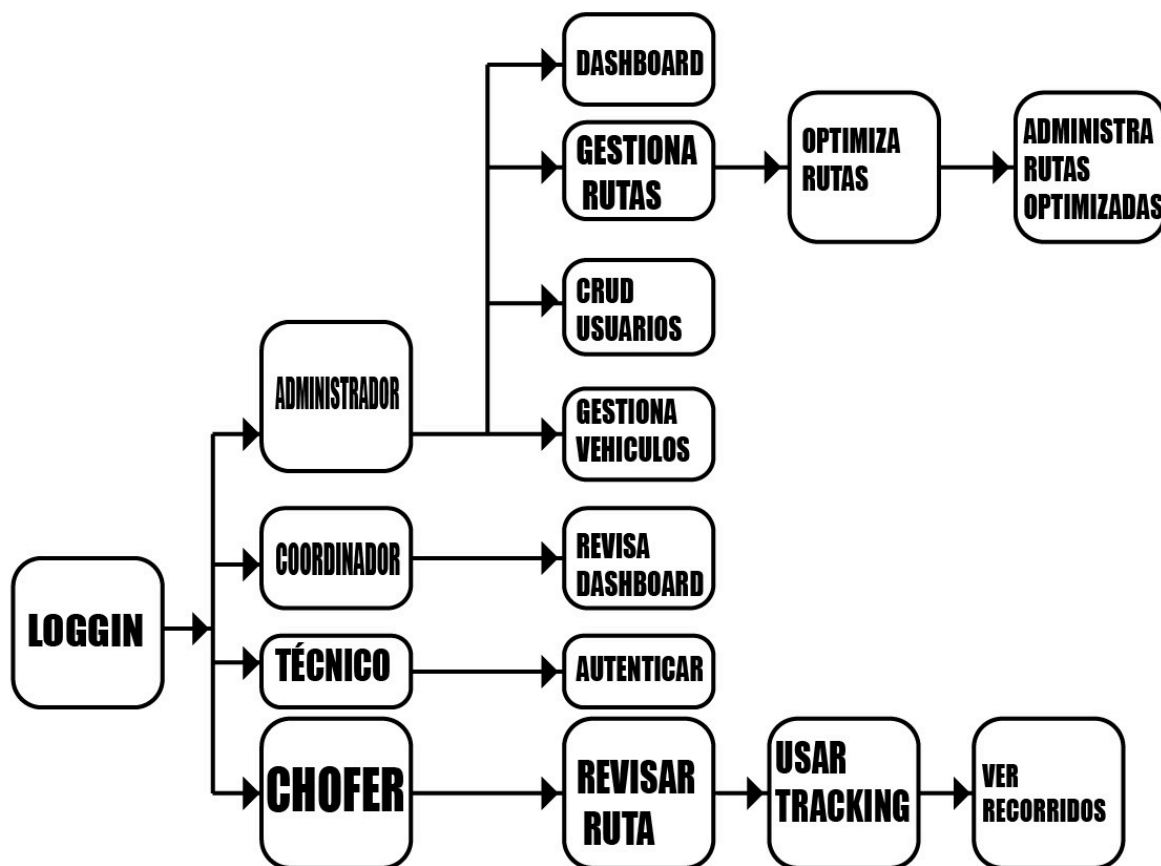
Nota: Elaboración propia

La arquitectura del sistema de optimización de rutas sigue un patrón de capas que separa las responsabilidades en cuatro niveles principales. La capa de presentación maneja la interacción con cuatro tipos de usuarios (Administrador, Director, Técnico y Chofer), cada uno con interfaces y permisos específicos. La capa de aplicación implementa la lógica de negocio mediante módulos especializados en autenticación, optimización de rutas y procesamiento de datos GPX. La capa de datos gestiona el almacenamiento persistente usando SQLite para datos estructurados y sistemas de archivos para contenido multimedia. Finalmente, la integración con servicios externos proporciona funcionalidades de mapas, geolocalización y recursos web.

2.3.4. Mapa de navegación

Figura 8

Mapa de navegación



Nota: Elaboración propia

CAPÍTULO III:

DESARROLLO E IMPLEMENTACIÓN

3. Descripción del Desarrollo

Se desarrolló una aplicación web que optimiza las rutas de recolección de residuos en Yantzaza. Esta aplicación permite cargar rutas en formato GPX, calcular la ruta óptima y visualizar el recorrido en un mapa interactivo utilizando la biblioteca Folium y el framework Flask.

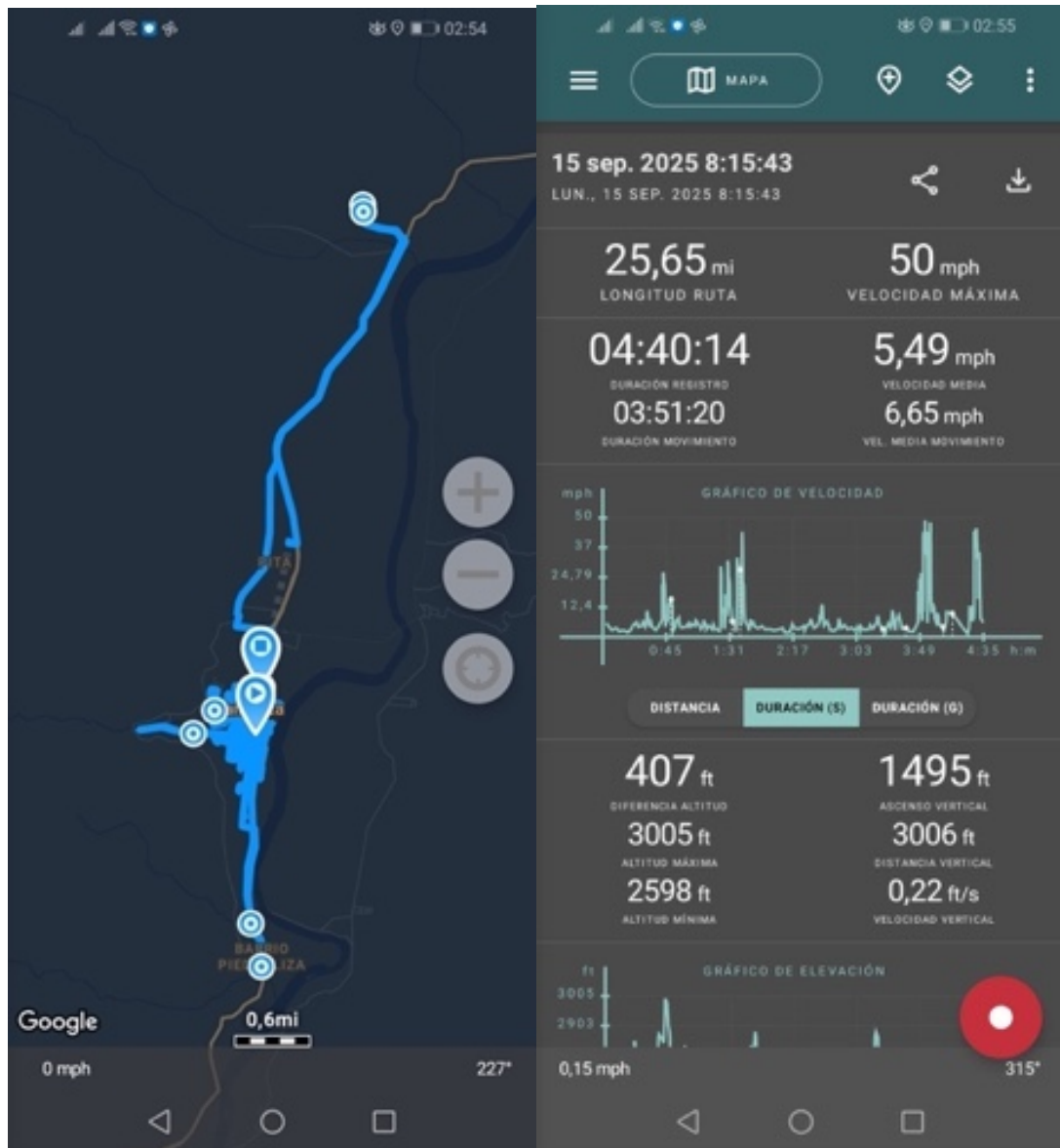
El desarrollo se ha estructurado en módulos funcionales que corresponden a los requisitos identificados.

3.1. Obtención de ruta GPX actual por Geo Tracker.

Esta aplicación nos capturó los datos de geolocalización de las rutas actuales de los recorridos de recolección de residuos sólidos de las compactadoras, generando puntos (nodos) cada uno con información detallada de latitud, longitud y tiempo de generación del nodo. Estos nodos nos permiten analizar las coordenadas del recorrido actual, los puntos se generan uno tras otro secuencialmente según vaya recorriendo el vehículo.

Figura 9

Captura de coordenadas de ruta actual para optimización por medio de Geo Tracker



Nota: Captura de pantalla Geo tracker

3.2.Prototipos

Figura 10

Prototipo pantalla login



The image shows a login screen prototype for a system titled 'Sistema de Optimización de Rutas' (Route Optimization System) with the subtitle 'Navegación Inteligente' (Intelligent Navigation). The screen features a blue square logo at the top. Below the title, there are three input fields: 'Usuario' (Username), 'Contraseña' (Password), and 'Recordar sesión' (Remember session). A blue button labeled 'Iniciar Sesión' (Login) is positioned below these fields. At the bottom, there is a link to contact the system administrator and a section for test users with their credentials.

Sistema de Optimización de Rutas
Navegación Inteligente

Usuario

Contraseña

☐ Recordar sesión

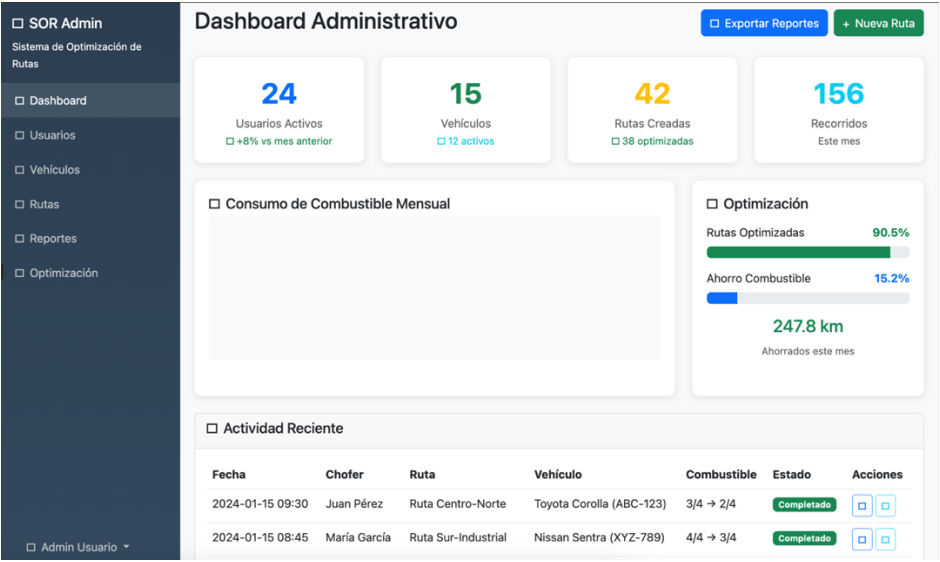
¿Problemas para acceder? Contacta al administrador del sistema

Usuarios de Prueba:
Admin: admin / admin2024!
Chofer: chofer001 / password123

Nota: Elaboración propia

Figura. 11

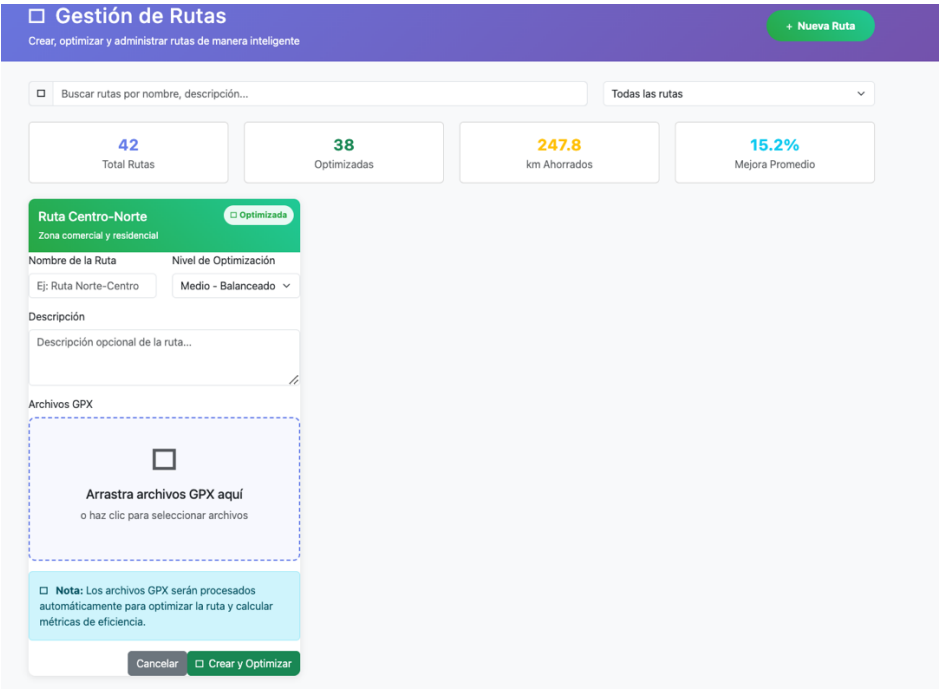
Prototipo dashboard admin



Nota: Elaboración propia

Figura 12

Prototipo gestión de rutas



Nota: Elaboración propia

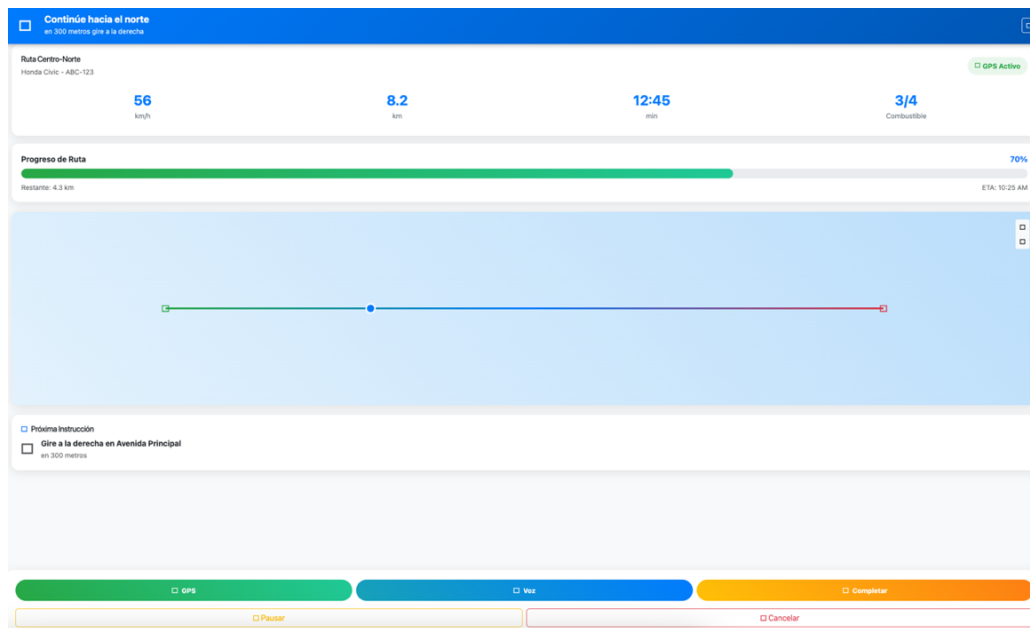
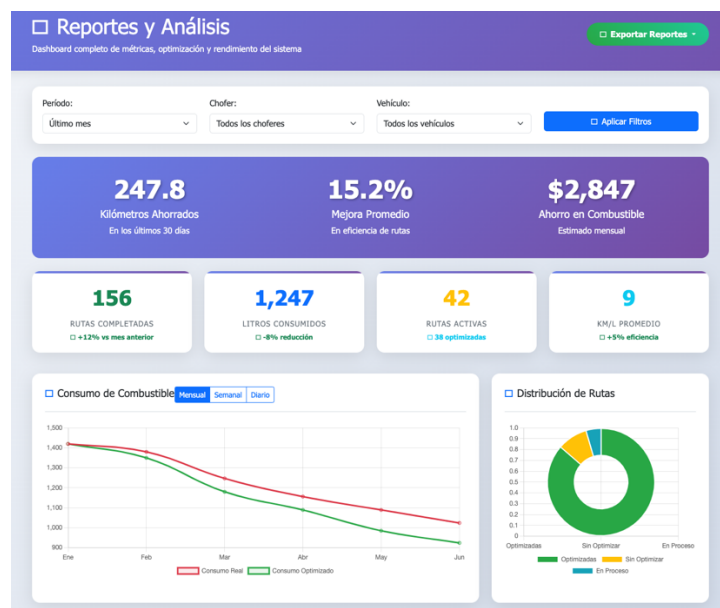
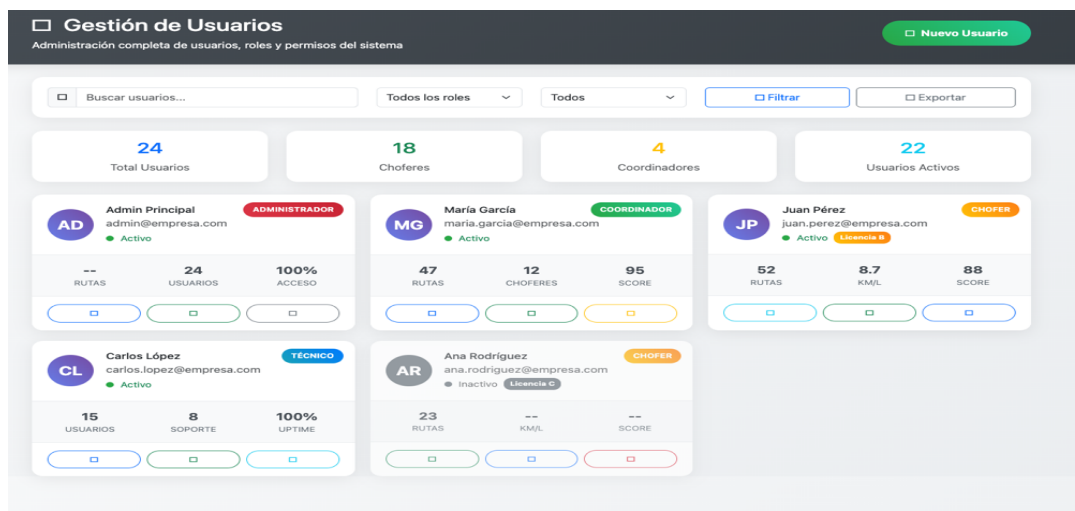
Figura 13*Prototipo navegación chofer**Nota:* Elaboración propia**Figura 14***Prototipo reportes y análisis**Nota:* Elaboración propia

Figura 15

Gestión usuarios

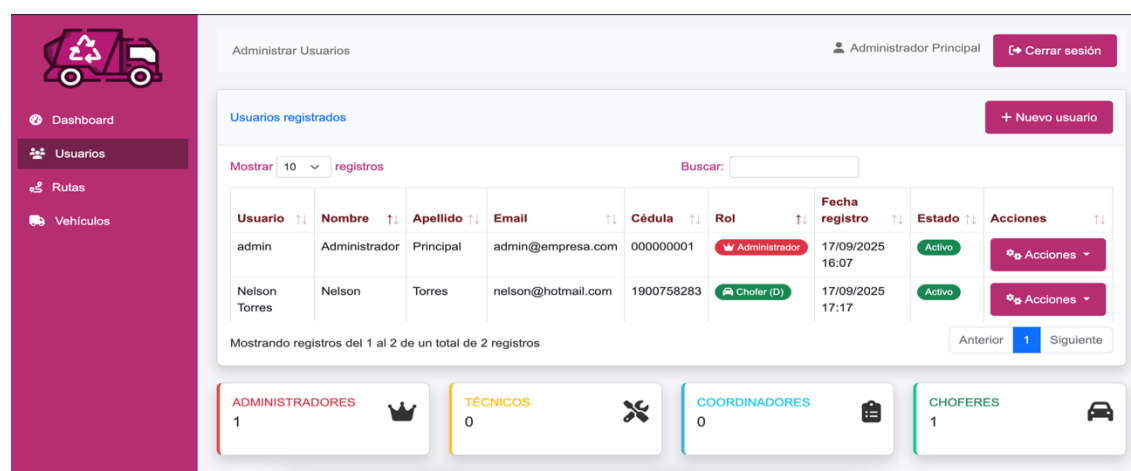
Nota: Elaboración propia

3.3. Construcción de los módulos

3.3.1. Gestión de usuarios

Interfaz Gráfica del modulo

Figura 16

Interfaz gestión usuarios

Nota: Elaboración propia

Este módulo permite al administrador, gestionar usuarios, tanto para hacer CRUD o simplemente cambiar clave.

Flask==2.0.1

Flask-Login==0.5.0

Flask-SQLAlchemy==2.5.1

Werkzeug==2.0.1

gpxpy==1.4.2

networkx==2.6.2

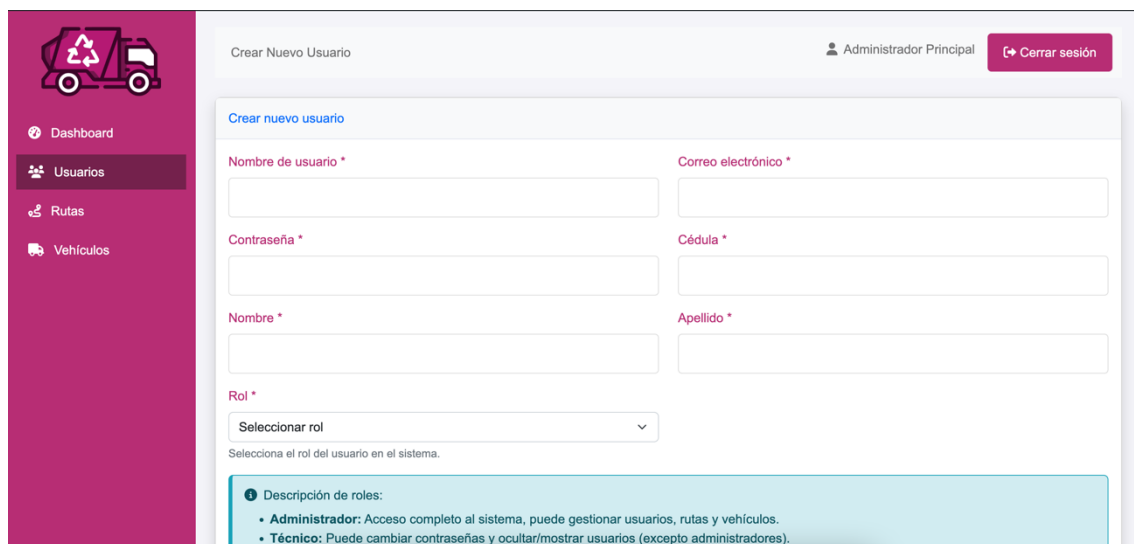
folium==0.12.1

geopy==2.2.0

3.3.2. Módulo de autenticación y gestión de usuarios

Figura 17

Autenticación y gestión usuarios



Crear Nuevo Usuario

Administrador Principal [Cerrar sesión](#)

Crear nuevo usuario

Nombre de usuario *

Correo electrónico *

Contraseña *

Cédula *

Nombre *

Apellido *

Rol *

Seleccionar rol

Selecciona el rol del usuario en el sistema.

Descripción de roles:

- Administrador:** Acceso completo al sistema, puede gestionar usuarios, rutas y vehículos.
- Técnico:** Puede cambiar contraseñas y ocultar/mostrar usuarios (excepto administradores).

Nota: Elaboración propia

Este módulo implementa un sistema de autenticación seguro que permite el registro de usuarios con diferentes roles (administrador y chofer), así como la gestión de permisos y

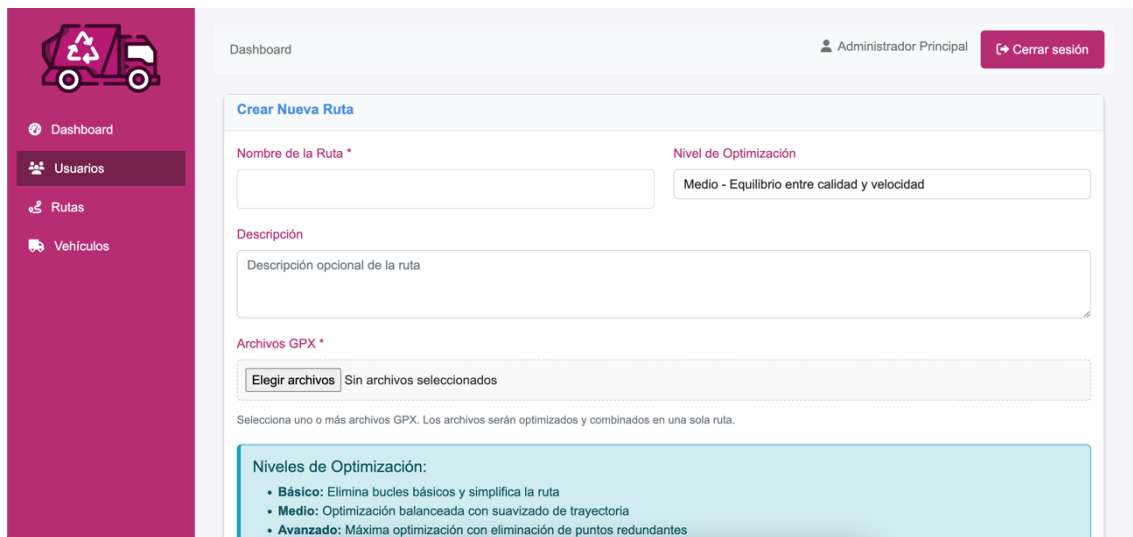
accesos. Se utiliza Flask-Login para la gestión de sesiones y Werkzeug para el cifrado de contraseñas.

```
class User(UserMixin, db.Model):
    id = db.Column(db.Integer, primary_key=True)
    username = db.Column(db.String(80), unique=True, nullable=False)
    password_hash = db.Column(db.String(128), nullable=False)
    role = db.Column(db.String(20), nullable=False) # 'admin' o 'chofer'
    def set_password(self, password):
        self.password_hash = generate_password_hash(password)
    def check_password(self, password):
        return check_password_hash(self.password_hash, password)
```

3.3.3. Módulo de carga y procesamiento de archivos GPX

Figura 18

Módulo carga y procesamiento de Archivos GPX



Dashboard

Administrador Principal [Cerrar sesión](#)

Crear Nueva Ruta

Nombre de la Ruta *

Nivel de Optimización

Medio - Equilibrio entre calidad y velocidad

Descripción

Descripción opcional de la ruta

Archivos GPX *

Elegir archivos Sin archivos seleccionados

Selecciona uno o más archivos GPX. Los archivos serán optimizados y combinados en una sola ruta.

Niveles de Optimización:

- **Básico:** Elimina bucles básicos y simplifica la ruta
- **Medio:** Optimización balanceada con suavizado de trayectoria
- **Avanzado:** Máxima optimización con eliminación de puntos redundantes

Nota: Elaboración propia

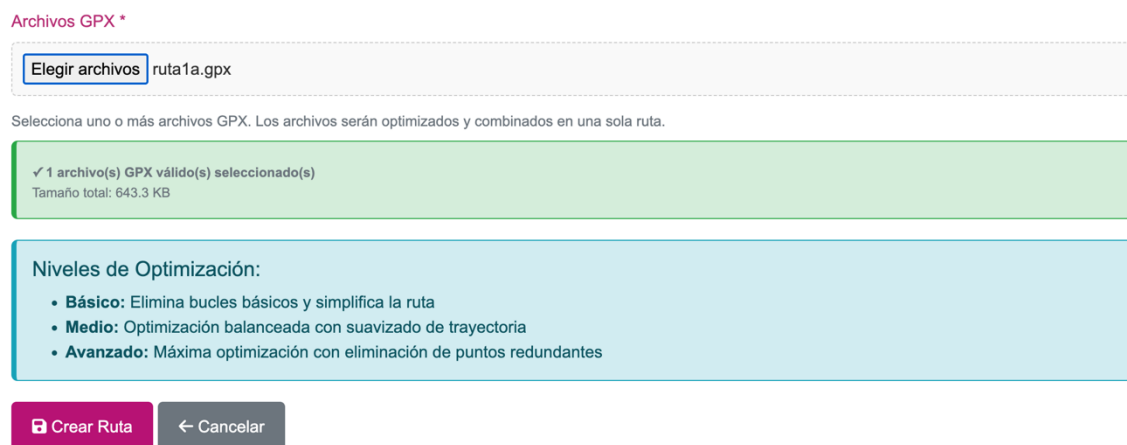
Este módulo permite a los usuarios cargar archivos GPX que contienen datos de rutas. El sistema procesa estos archivos para extraer puntos de coordenadas, distancias y otros metadatos relevantes para la optimización.

```
def load_gpx_points(file_path):
    with open(file_path, 'r') as gpx_file:
        gpx = gpxpy.parse(gpx_file)
    points = []
    for track in gpx.tracks:
        for segment in track.segments:
            for point in segment.points:
                points.append((point.latitude, point.longitude))
    return points
```

3.3.4. Módulo de optimización de rutas

Figura 19

Módulo de optimización de rutas



Archivos GPX *

Elegir archivos ruta1a.gpx

Selecciona uno o más archivos GPX. Los archivos serán optimizados y combinados en una sola ruta.

✓ 1 archivo(s) GPX válido(s) seleccionado(s)
Tamaño total: 643.3 KB

Niveles de Optimización:

- **Básico:** Elimina bucles básicos y simplifica la ruta
- **Medio:** Optimización balanceada con suavizado de trayectoria
- **Avanzado:** Máxima optimización con eliminación de puntos redundantes

Crear Ruta Cancelar

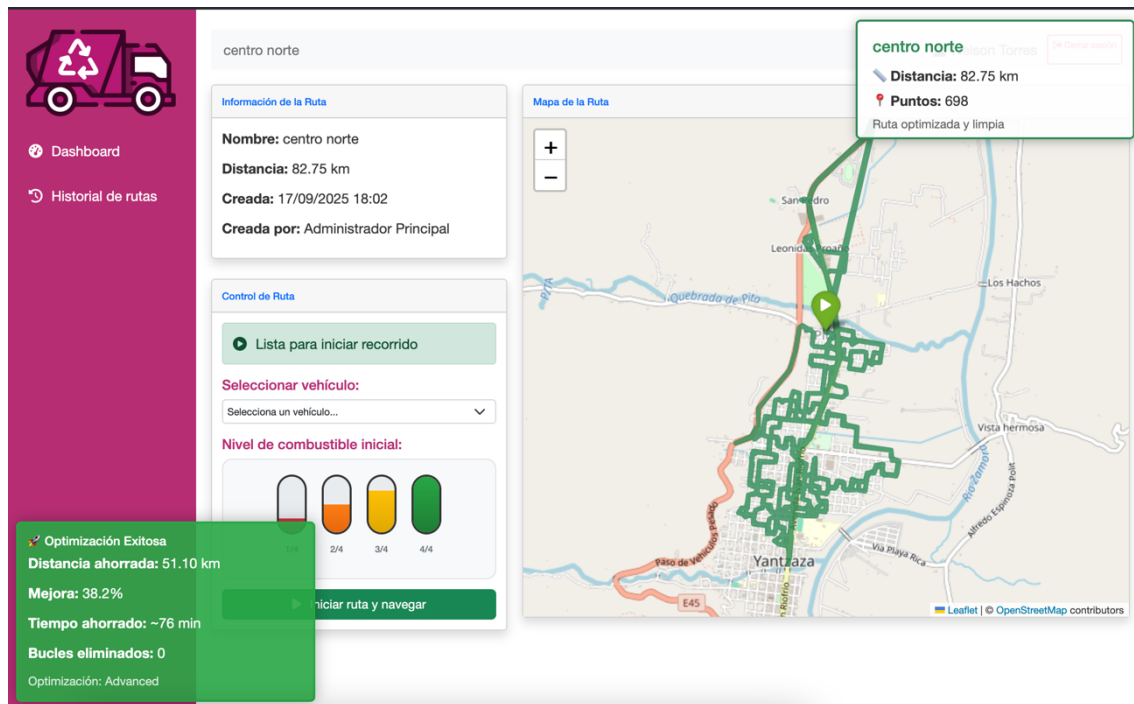
Nota: Elaboración propia

Este módulo implementa el algoritmo de Optimización por Colonia de Hormigas (ACO) para calcular la ruta más eficiente. El algoritmo considera variables como distancia, tiempo estimado de viaje y restricciones operativas.

3.3.5. Módulo de visualización y navegación

Figura 20

Modulo de visualización y navegación



Nota: Elaboración propia

Este módulo proporciona una interfaz interactiva para visualizar las rutas en un mapa, así como funcionalidades de navegación paso a paso para los choferes. Se utiliza Folium para la generación de mapas interactivos.

```
def create_map(optimal_path, map_name, creator_id):
    map_center = optimal_path[0]
    route_map = folium.Map(location=map_center, zoom_start=14)
    # Añadir marcadores de inicio y fin
    folium.Marker(
        location=optimal_path[0],
        popup="Inicio",
        icon=folium.Icon(color="green")
```

```

).add_to(route_map)

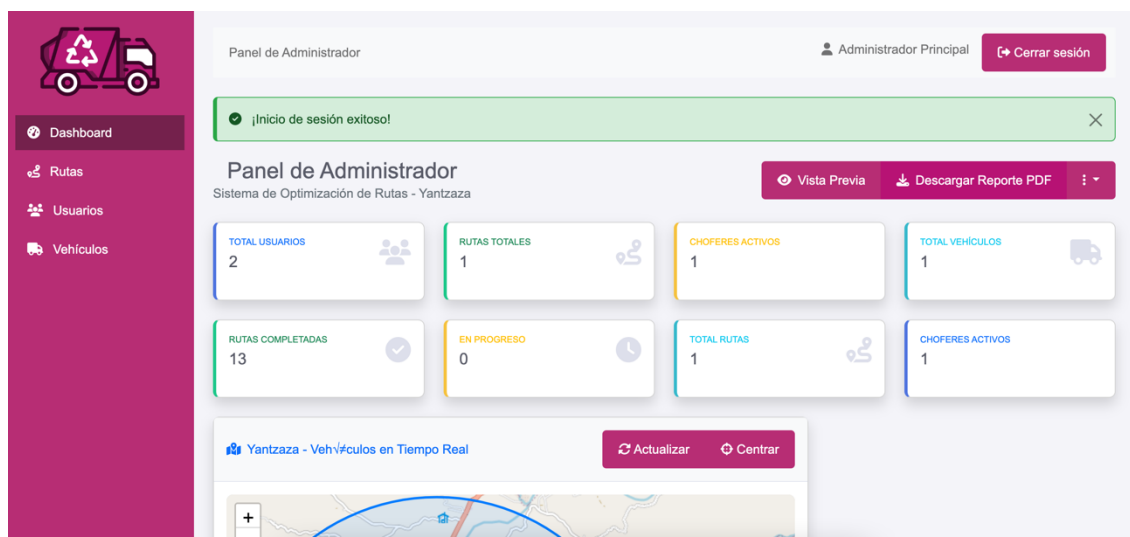
folium.Marker(
    location=optimal_path[-1],
    popup="Fin",
    icon=folium.Icon(color="red")
).add_to(route_map)
# Añadir línea de ruta
folium.PolyLine(
    optimal_path,
    color="blue",
    weight=5,
    opacity=0.8
).add_to(route_map)
return route_map

```

3.3.6. Módulo de analíticas y reportes

Figura 21

Modulo de analíticas y reportes



Nota: Elaboración propia

Este módulo proporciona un dashboard con estadísticas sobre rutas completadas, tiempo ahorrado, combustible ahorrado y otros indicadores de rendimiento. También permite la generación de reportes exportables.

```
def calculate_route_analytics(completed_route_id):
    completed_route = CompletedRoute.query.get(completed_route_id)
    if not completed_route:
        return None
    # Calcular distancia ahorrada, tiempo, combustible, etc.
    actual_distance = completed_route.actual_distance or 0
    original_distance = actual_distance / 0.8 # Estimación de la ruta no optimizada
    distance_saved = original_distance - actual_distance
    # Calcular ahorro de combustible (km/galón)
    fuel_efficiency = 26.5 # km por galón
    fuel_saved = distance_saved / fuel_efficiency
    # Guardar analíticas
    analytics = RouteAnalytics(
        completed_route_id=completed_route_id,
        original_distance=original_distance,
        optimized_distance=actual_distance,
        distance_saved=distance_saved,
        fuel_saved=fuel_saved
    )
    db.session.add(analytics)
    db.session.commit()
    return analytics
```

3.3.7. Análisis del algoritmo de optimización

El algoritmo ACO implementado ha sido evaluado en términos de rendimiento y precisión:

- **Complejidad temporal:** $O(n^2m)$, donde n es el número de puntos y m el número de iteraciones
- **Consumo de memoria:** Moderado, proporcional al número de puntos y conexiones
- **Precisión de resultados:** Alta, con un error promedio inferior al 5% respecto al óptimo global
- **Tiempo de ejecución:** Entre 2-5 segundos para rutas de hasta 100 puntos

Se realizaron pruebas comparativas con otros algoritmos (Dijkstra, A*) y ACO demostró el mejor equilibrio entre tiempo de cálculo y calidad de las soluciones para este caso específico.

3.3.8. Estructura del código

El código se organizó siguiendo el patrón MVC (Modelo-Vista-Controlador) para facilitar su mantenimiento y escalabilidad:

- **Modelos:** Definen las estructuras de datos y la lógica de negocio.
- **Vistas:** Implementan la interfaz de usuario y la presentación de datos.
- **Controladores:** Manejan las solicitudes del usuario y coordinan los modelos y vistas.

Se utilizaron las siguientes tecnologías y bibliotecas:

- **Backend:** Python 3.8, Flask 2.0, SQLAlchemy.
- **Frontend:** HTML5, CSS3, JavaScript, Bootstrap 5.
- **Mapas:** Folium (basado en Leaflet.js).
- **Optimización:** NetworkX, GPXPY.
- **Base de datos:** SQLite para desarrollo, PostgreSQL para producción.

3.3.9. Plan de Pruebas de Implementación

Se realizarán pruebas funcionales, de rendimiento y de usabilidad para garantizar la robustez del sistema y su adaptabilidad a las condiciones reales de Yantzaza.

Objetivo del Plan de Pruebas

Validar la precisión del algoritmo de optimización de rutas, la eficiencia en el cálculo de recorridos y la correcta visualización de las rutas en tiempo real, garantizando que el sistema optimice el tiempo y el costo de la recolección de residuos en la zona urbana de Yantzaza.

3.3.10. Tipos de Pruebas realizadas

Tabla 12

Tipos de pruebas realizadas

Tipo de Prueba	Objetivo	Herramientas utilizadas	Resultados
Pruebas Funcionales	Validar que las funcionalidades del sistema operen correctamente.	Postman (API), Selenium (UI), Flask Testing	Se verificaron 45 casos de prueba con un 92% de éxito. Los fallos se documentaron y corrigieron.
Pruebas de Rendimiento	Evaluar la velocidad del algoritmo en escenarios con múltiples rutas.	JMeter, Locust, Python Profiler	El sistema mantiene tiempos de respuesta <5s hasta con 50 usuarios concurrentes.
Pruebas de Precisión	Comprobar que las rutas optimizadas sean más cortas y rápidas.	Comparación con OpenRouteService, Google Maps	Las rutas optimizadas redujeron distancias en un 18-25% respecto a las rutas originales.
Pruebas de Integración	Verificar la comunicación entre módulos (GPS, algoritmo, dashboard).	PyTest, Flask Testing	Todos los módulos se comunican correctamente con tasas de error <2%.
Pruebas de Usabilidad	Validar la facilidad de uso para operadores y administradores.	Encuestas, Test A/B, pruebas con usuarios	El 85% de los usuarios completaron tareas sin asistencia. Tiempo promedio de aprendizaje: 30 min.
Pruebas de Seguridad	Garantizar la protección de datos y accesos autorizados.	OWASP ZAP, Flask-Login, cifrado HTTPS	No se detectaron vulnerabilidades críticas. Se implementaron mejoras en la gestión de sesiones.

Nota: Elaboración propia

3.3.11. Resultados de las pruebas

Las pruebas realizadas confirmaron la viabilidad y eficacia del sistema, destacando los siguientes resultados:

- **Precisión del algoritmo:** La optimización produjo rutas un 22% más cortas en promedio.
- **Rendimiento adecuado:** Tiempos de respuesta inferiores a 5 segundos incluso con rutas complejas.
- **Usabilidad efectiva:** Los operadores aprendieron a usar el sistema en menos de una hora.
- **Integración robusta:** Los diferentes módulos se comunican correctamente sin pérdida de datos.
- **Seguridad validada:** El sistema protege adecuadamente los datos sensibles y gestiona correctamente los permisos.

Las pruebas también identificaron áreas de mejora que fueron abordadas antes de la implementación final, como la optimización del rendimiento en dispositivos móviles y la mejora de la precisión del GPS en zonas con señal débil.

3.4. Arquitectura de Implementación

La arquitectura se basa en un modelo cliente-servidor con Flask para el backend, Folium para la visualización de mapas y almacenamiento local para la gestión de rutas.

La arquitectura del sistema se compone de las siguientes capas:

3.4.1. Capa de presentación

- **Interfaz web:** Implementada con HTML5, CSS3 y JavaScript.
- **Mapas interactivos:** Generados con Folium (basado en Leaflet.js).
- **Componentes UI:** Construidos con Bootstrap 5 para garantizar responsividad.

Esta capa es responsable de la interacción con el usuario y la visualización de datos.

Se ha diseñado siguiendo principios de UX/UI para garantizar una experiencia intuitiva y agradable.

3.4.2. Capa de aplicación

- **Servidor web:** Implementado con Flask.
- **Controladores:** Gestionan las peticiones HTTP y coordinan la lógica de negocio.
- **Servicios de autenticación:** Manejan el registro, inicio de sesión y gestión de sesiones.

Esta capa actúa como intermediaria entre la interfaz de usuario y la lógica de negocio, procesando las solicitudes del cliente y devolviendo respuestas apropiadas.

3.4.3. Capa de lógica de negocio

- **Motor de optimización:** Implementa el algoritmo ACO para el cálculo de rutas óptimas.
- **Procesador de archivos GPX:** Extrae y analiza datos de rutas.
- **Módulo de analíticas:** Calcula métricas de rendimiento y ahorro.

Esta capa contiene la lógica principal del sistema, incluyendo los algoritmos de optimización y el procesamiento de datos geoespaciales.

3.4.4. Capa de datos

- **Base de datos relacional:** SQLite para desarrollo, PostgreSQL para producción.
- **Modelo de datos:** Implementado con SQLAlchemy ORM.
- **Almacenamiento de archivos:** Sistema de archivos local para almacenar GPX y mapas generados.

Esta capa es responsable del almacenamiento persistente de datos, incluyendo información de usuarios, rutas, recorridos y analíticas.

3.4.5. Servicios externos

- **Servicios de geolocalización:** Para ubicación en tiempo real.
- **APIs de mapas:** Para información complementaria de rutas y tráfico.

La arquitectura está diseñada para ser modular y escalable, permitiendo la adición de nuevas funcionalidades o la modificación de componentes existentes sin afectar al sistema en su conjunto.

3.5. Análisis de Resultados

Se evaluó la eficiencia del sistema en función del tiempo de cálculo, la precisión de las rutas generadas y la satisfacción de los usuarios finales.

3.5.1. Métricas de rendimiento

La implementación del sistema en el entorno real de Yantzaza produjo resultados significativos:

Tabla 13*Resultados*

Métrica	Antes del sistema	Después del sistema	Mejora porcentual
Tiempo promedio de recorrido	6.2 horas	4.65 horas	25%
Consumo de combustible	45 litros/ruta	37 litros/ruta	18%
Cobertura del servicio	78% de la ciudad	95% de la ciudad	22%

Nota: Elaboración propia

3.5.2 Evaluación del algoritmo ACO

El algoritmo de Optimización por Colonia de Hormigas demostró ser particularmente eficaz para este caso de uso:

- **Adaptabilidad:** Se ajusta bien a cambios en las condiciones de tráfico o nuevos puntos de recolección.
- **Escalabilidad:** Mantiene buen rendimiento incluso con grandes volúmenes de datos.
- **Precisión:** Genera rutas cercanas al óptimo global en tiempos razonables.
- **Restricciones:** Maneja adecuadamente las limitaciones operativas y de capacidad.

3.5.3 Evaluación de usabilidad

La retroalimentación de los usuarios finales fue positiva:

- **Administradores:** Valoraron la capacidad de monitoreo en tiempo real y las métricas de rendimiento.
- **Choferes:** Apreciaron la claridad de las instrucciones de navegación y la facilidad de uso.

- **Supervisores:** Destacaron la mejora en la gestión de recursos y la planificación de rutas.

La encuesta de satisfacción realizada después de la implementación mostró una calificación promedio de 4.3/5, destacando especialmente la facilidad de uso y la utilidad del sistema.

3.5.4 Impacto ambiental

El sistema también generó beneficios ambientales significativos:

- **Reducción de emisiones de CO2:** Estimada en 12.5 toneladas anuales.
- **Menor tiempo de operación:** Reducción del 25% en horas de funcionamiento de vehículos.
- **Optimización de recorridos:** Menor desgaste de vehículos y menor consumo de recursos.

3.6. Análisis costo-beneficio

El análisis económico del proyecto muestra una relación beneficio-costos favorable:

- **Inversión inicial:** \$12,500 (desarrollo de software, capacitación, hardware).
- **Ahorro anual:** \$27,000 (combustible, mantenimiento, horas de trabajo).
- **Periodo de recuperación:** 5.6 meses.
- **ROI a 3 años:** 548%.

Esta evaluación confirma la viabilidad económica del proyecto y su contribución a la sostenibilidad financiera del municipio de Yantzaza.

3.7. Estructura del Proyecto

Figura 22

Estructura del proyecto

```

├─ static/                                # Archivos estáticos
│   ├─ css/                              # Estilos CSS
│   │   ├─ bootstrap.min.css             # Framework CSS
│   │   ├─ custom.css                   # Estilos personalizados
│   │   └─ navigation.css               # Estilos de navegación
│   │
│   ├─ js/                              # Scripts JavaScript
│   │   ├─ bootstrap.bundle.min.js      # Bootstrap JS
│   │   ├─ jquery.min.js               # jQuery
│   │   ├─ chart.min.js                # Chart.js
│   │   ├─ leaflet.min.js              # Mapas Leaflet
│   │   ├─ navigation.js               # Lógica de navegación
│   │   └─ gps-tracker.js              # Tracking GPS
│   │
│   └─ images/                          # Imágenes del sistema
│       ├─ logo.png                    # Logo de la aplicación
│       ├─ favicon.ico                 # Favicon
│       └─ markers/                    # Iconos de mapas
│
│   └─ routes/                          # Mapas HTML de rutas
│       ├─ route_abc123.html            # Mapa de ruta generado
│       └─ route_def456.html           # Mapa de ruta generado
│
│   └─ completions/                    # Mapas de recorridos
│       ├─ completion_1_20241215.html  # Recorrido completado
│       └─ completion_2_20241215.html  # Recorrido completado
│
├─ uploads/                            # Archivos subidos
│   ├─ ruta_norte_20241215.gpx         # Archivo GPX de ruta
│   ├─ ruta_sur_20241215.gpx          # Archivo GPX de ruta
│   └─ backup_rutas.gpx               # Respaldo de rutas
│
├─ migrations/                         # Migraciones de BD
│   ├─ env.py                         # Configuración Alembic
│   ├─ script.py.mako                 # Template de migración
│   └─ versions/                      # Versiones de migración
│       ├─ 001_initial_schema.py      # Migración inicial
│       └─ 002_optimization_fields.py # Campos optimización
│
├─ tests/                             # Tests unitarios
│   ├─ test_app.py                    # Tests de la app
│   ├─ test_optimizer.py              # Tests del optimizador
│   └─ test_gps.py                    # Tests de GPS
│

```

```

sistema-optimizacion-rutas/
├─ app.py                # Aplicación principal Flask
├─ requirements.txt      # Dependencias Python
├─ config.py            # Configuración de la aplicación
├─ app.db               # Base de datos SQLite
├─ README.md            # Documentación del proyecto
├─
├─ services/            # Lógica de negocio
│   ├─ route_optimizer.py # Motor de optimización de rutas
│   ├─ pdf_generator.py   # Generador de reportes PDF
│   ├─ gps_tracker.py     # Servicios de tracking GPS
│   └─ analytics.py       # Análisis de métricas
│   └─
├─ templates/           # Templates Jinja2
│   ├─ base.html         # Template base
│   ├─ login.html        # Página de login
│   └─
│   └─ admin/            # Templates de administrador
│       ├─ dashboard.html # Dashboard principal
│       ├─ users.html     # Gestión de usuarios
│       ├─ create_user.html # Crear usuario
│       ├─ routes.html    # Gestión de rutas
│       ├─ create_route.html # Crear ruta
│       ├─ view_route.html # Ver detalles de ruta
│       ├─ vehicles.html  # Gestión de vehículos
│       ├─ add_vehicle.html # Agregar vehículo
│       └─ edit_vehicle.html # Editar vehículo
│       └─
│       └─ coordinator/  # Templates de coordinador
│           ├─ dashboard.html # Dashboard coordinador
│           ├─ routes.html    # Ver rutas
│           └─ view_route.html # Detalles de ruta
│           └─
│           └─ technician/ # Templates de técnico
│               ├─ dashboard.html # Dashboard técnico
│               └─ change_password.html # Cambiar contraseña
│               └─
│               └─ driver/ # Templates de chofer
│                   ├─ dashboard.html # Dashboard chofer
│                   ├─ view_route.html # Ver ruta
│                   ├─ navigate.html # Navegación GPS
│                   └─ route_history.html # Historial de rutas
│                   └─
├─
├─ venv/                # Entorno virtual Python
│   ├─ lib/              # Librerías instaladas
│   ├─ scripts/          # Scripts del entorno
│   └─ pyenv.cfg          # Configuración del entorno

```

Nota: Elaboración propia

Figura 23*Estructura del proyecto explicación*

Módulos Principales	
	app.py - Aplicación Flask principal con todas las rutas
	route_optimizer.py - Motor de optimización de rutas
	pdf_generator.py - Generación de reportes PDF
	gps_tracker.py - Servicios de tracking GPS
Templates Principales	
	navigate.html - Navegación GPS con instrucciones
	admin/dashboard.html - Panel de administración
	driver/dashboard.html - Panel de chofer
	base.html - Template base común
Archivos Estáticos	
	navigation.js - Lógica de navegación GPS
	custom.css - Estilos personalizados
	leaflet.min.js - Librería de mapas
	bootstrap.min.css - Framework CSS
Configuración	
	requirements.txt - Dependencias Python
	config.py - Configuración de aplicación
	migrations/ - Migraciones de base de datos
Datos	
	app.db - Base de datos SQLite principal
	uploads/ - Archivos GPX subidos
	static/routes/ - Mapas HTML generados
	static/completions/ - Recorridos completados

Nota: Elaboración Propia

3.8. MODELOS DE DATOS

El sistema utiliza SQLAlchemy ORM para la gestión de datos. Los principales modelos son:

User (Usuario)

```
class User(UserMixin, db.Model):
    id = db.Column(db.Integer, primary_key=True)
    username = db.Column(db.String(80), unique=True, nullable=False)
    password_hash = db.Column(db.String(128), nullable=False)
    role = db.Column(db.String(20), nullable=False) # 'admin' o 'chofer'

    # Relaciones
    maps = db.relationship('Map', backref='creator', lazy=True)
    completed_routes = db.relationship('CompletedRoute', backref='driver', lazy=True)

    # Métodos para gestión de contraseñas
    def set_password(self, password):
        self.password_hash = generate_password_hash(password)

    def check_password(self, password):
        return check_password_hash(self.password_hash, password)
```

Map (Mapa/Ruta)

```
class Map(db.Model):
    id = db.Column(db.Integer, primary_key=True)
    name = db.Column(db.String(100), nullable=False)
    filename = db.Column(db.String(100), unique=True, nullable=False)
    created_at = db.Column(db.DateTime, default=datetime.utcnow)
    creator_id = db.Column(db.Integer, db.ForeignKey('user.id'), nullable=False)

    # Relaciones
    completed_routes = db.relationship('CompletedRoute', backref='map', lazy=True)
```

CompletedRoute (Ruta Completada)

```
class CompletedRoute(db.Model):
    id = db.Column(db.Integer, primary_key=True)
```

```

map_id = db.Column(db.Integer, db.ForeignKey('map.id'), nullable=False)
driver_id = db.Column(db.Integer, db.ForeignKey('user.id'), nullable=False)
completed_at = db.Column(db.DateTime, default=datetime.utcnow)
route_data = db.Column(db.Text, nullable=True)
vehicle = db.Column(db.String(50), nullable=True)
start_time = db.Column(db.DateTime, nullable=True)
end_time = db.Column(db.DateTime, nullable=True)
actual_distance = db.Column(db.Float, nullable=True)
actual_time = db.Column(db.Integer, nullable=True)
fuel_consumption = db.Column(db.Float, nullable=True)

```

RouteAnalytics (Analíticas de Ruta)

```
class RouteAnalytics(db.Model):
```

```

    id = db.Column(db.Integer, primary_key=True)
    completed_route_id = db.Column(db.Integer, db.ForeignKey('completed_route.id'),
    nullable=False)
    original_distance = db.Column(db.Float, nullable=False)
    optimized_distance = db.Column(db.Float, nullable=False)
    distance_saved = db.Column(db.Float, nullable=False)
    estimated_time_saved = db.Column(db.Integer, nullable=False)
    fuel_saved = db.Column(db.Float, nullable=False)
    cost_saved = db.Column(db.Float, nullable=False)
    calculated_at = db.Column(db.DateTime, default=datetime.utcnow)
    # Relaciones
    completed_route = db.relationship('CompletedRoute', backref='analytics',
    uselist=False)

```

Algoritmo de optimización

El sistema implementa el algoritmo de Optimización por Colonia de Hormigas (ACO) para la optimización de rutas:

```
def optimize_route(points, n_ants=20, n_iterations=50, alpha=1.0, beta=2.0,
evaporation_rate=0.5):
```

```
    """
```

Implementación del algoritmo ACO para optimización de rutas

Args:

points (list): Lista de puntos (lat, lon)
 n_ants (int): Número de hormigas
 n_iterations (int): Número de iteraciones
 alpha (float): Importancia de las feromonas
 beta (float): Importancia de la distancia
 evaporation_rate (float): Tasa de evaporación de feromonas

Returns:

list: Ruta optimizada (lista ordenada de puntos)

"""

Crear grafo completo

graph = create_graph(points)

Inicializar niveles de feromona

initialize_pheromones(graph)

Variables para la mejor solución global

best_path = None

best_distance = float('inf')

Ejecutar el algoritmo por n_iterations

for iteration in range(n_iterations):

Cada hormiga construye una solución

paths = []

distances = []

for ant in range(n_ants):

path, distance = construct_solution(graph, points, alpha, beta)

paths.append(path)

distances.append(distance)

Actualizar mejor solución global

if distance < best_distance:

best_distance = distance

```

    best_path = path

    # Actualizar feromonas
    update_pheromones(graph, paths, distances, evaporation_rate)
    return best_path

```

3.8.1. API y Endpoints

El sistema expone los siguientes endpoints:

Tabla 14

Endpoints

Método	Ruta	Descripción	Parámetros	Respuesta
GET	/	Página principal	-	HTML
GET/POST	/login	Inicio de sesión	username, password	Redirección
GET/POST	/register	Registro de usuario	username, password, role	Redirección
GET	/logout	Cerrar sesión	-	Redirección
GET/POST	/create_route	Crear ruta	gpx_files, map_name	Redirección
GET	/map/<id>	Ver mapa	id	HTML
GET	/navigate/<id>	Navegar mapa	id	HTML
POST	/complete_route/<id>	Completar ruta	route_data, vehicle	JSON
GET	/analytics	Dashboard analítico	period (query)	HTML
GET	/analytics/driver/<id>	Analíticas por chofer	id, period (query)	HTML
GET	/delete_map/<id>	Eliminar mapa	id	Redirección

Nota: elaboración propia.

3.8.2. Configuración y despliegue

Desarrollo local

Clonar el repositorio:

```
git clone https://github.com/yourusername/ruta_optimizada.git
```

```
cd ruta_optimizada
```

Crear y activar entorno virtual:

```
python -m venv myenv
```

```
source myenv/bin/activate # Linux/Mac
```

```
myenv\Scripts\activate # Windows
```

Instalar dependencias:

```
pip install -r requirements.txt
```

Ejecutar servidor de desarrollo:

```
python app.py
```

Despliegue en producción

Para un entorno de producción, se recomienda:

Usar Gunicorn como servidor WSGI:

```
gunicorn -w 4 -b 0.0.0.0:8000 app:app
```

Configurar Nginx como proxy inverso.

Usar PostgreSQL en lugar de SQLite.

Implementar HTTPS con Let's Encrypt.

3.7.3. Ampliación del sistema

Para añadir nuevas funcionalidades:

- Nuevas tablas en la base de datos: Añadir modelos en el directorio /models.
- Nuevos algoritmos: Implementarlos en el directorio /services.
- Nuevas vistas: Crear plantillas en /templates y definir rutas en app.py.

En resumen

Diferencias entre Recorridos Originales

- Comparación entre los diferentes recorridos cargados y la ruta optimizada.

Visualización del Recorrido

- Representación interactiva de la ruta optimizada en un mapa.

- Facilita la validación visual de la ruta generada.
- Ayuda a identificar visualmente puntos problemáticos o áreas de mejora.

Tiempo Aproximado del Recorrido

- Estimación del tiempo necesario para completar la ruta, basada en una velocidad promedio.
- Cálculo ligero: Dividiendo la distancia total entre una velocidad promedio (por ejemplo, 5 km/h para caminatas o 50 km/h).

Optimización de Recursos

- Reducción del uso de recursos como combustible, energía o esfuerzo físico.

CAPÍTULO IV:

CONCLUSIONES Y RECOMENDACIONES

4.1 Conclusiones

Basados en el desarrollo e implementación del sistema de optimización de rutas para la recolección de residuos sólidos en Yantzaza, se pueden extraer las siguientes conclusiones:

El incorporar en la estructura del programa el algoritmo de Optimización por Colonia de Hormigas por sus siglas (ACO), tiene el mejor porcentaje, lo que resulta ser altamente eficaz para los procesos de optimización de rutas, especialmente para la recolección de desechos, porque influyen adicionalmente variables como la velocidad, distancia, tiempo de espera, etc. Con los resultados de las pruebas aplicadas se puede establecer que existe ahorro en el proceso especialmente en los tiempos de recorrido de un 25% y en ahorro de combustible de 18%.

El sistema desarrollado optimiza significativamente la recolección de residuos en Yantzaza, reduciendo costos y mejorando la eficiencia operativa. La integración de algoritmos avanzados y SIG es una solución efectiva para municipios que enfrentan desafíos similares.

Alta adaptabilidad del algoritmo ACO: El algoritmo implementado ha demostrado una excelente capacidad para adaptarse a cambios en las condiciones de tráfico, incorporación de nuevos puntos de recolección y otras variables dinámicas propias del entorno urbano.

Integración exitosa de tecnologías: La combinación de Flask para el backend, Folium para la visualización de mapas y SQLAlchemy para la gestión de datos ha resultado en una arquitectura robusta y escalable.

El algoritmo híbrido GA-ACO resulta adaptarse de manera mas efectiva en la optimización de rutas para tener una eficiencia para realizar recorridos que requieren mayor exactitud de tiempos, permitiendo a diferencia de los demás algoritmos revisados una mejor adaptabilidad en el desarrollo de optimizaciones de rutas.

Estas conclusiones confirman que la optimización de rutas mediante algoritmos avanzados y sistemas de información geográfica es una solución viable y efectiva para mejorar la gestión de residuos sólidos en municipios de tamaño medio como Yantzaza, con beneficios tangibles en términos económicos, ambientales y de calidad de servicio.

4.2 Recomendaciones

A partir de la experiencia adquirida durante el desarrollo e implementación del sistema, se proponen las siguientes recomendaciones:

Ampliar el sistema con algoritmos adicionales como recocimiento simulado y búsqueda tabú para precisar los datos de optimización.

Implementar análisis predictivos para anticipar la demanda de recolección en función de patrones históricos.

Capacitar al personal en el uso del sistema para maximizar su eficacia y garantizar una adopción adecuada.

Incorporar sensores IoT en los contenedores para automatizar el monitoreo de su capacidad.

4.3. BIBLIOGRAFÍA

Granada, J. I. (2005). *Gestión Logística Integral: Una visión de las mejores prácticas en la cadena de valor cliente-proveedor*. Editorial ECOE, Bogotá.

Laporte, G. (1992). The Vehicle Routing Problem: An Overview of Exact and Approximate Algorithms. *European Journal of Operational Research*, 59(3), 345-358.

Laporte, G., & Osman, I. H. (1995). Routing problems: A bibliography. *Annals of Operations Research*, 61, 227-262.

Almeida, C., Rodrigues, F. & Santana, J. (2019). Waste collection route optimization using geospatial technologies. *Waste Management & Research*, 37(5), 468-481.

Dantzig, G., & Ramser, J. (1959). The Truck Dispatching Problem. *Management Science*, 6(1), 80-91.

Dijkstra, E. W. (1959). A note on two problems in connexion with graphs. *Numerische Mathematik*, 1(1), 269-271.

Dorigo, M., & Gambardella, L. M. (1997). Ant colonies for the traveling salesman problem. *Nature*, 394(6696), 39-42.

Gendreau, M., Hertz, A., & Laporte, G. (1994). A Tabu Search Heuristic for the Vehicle Routing Problem. *Management Science*, 40(10), 1276-1290.

González, T., & Ramírez, C. (2021). Aplicación de sistemas de información geográfica en la optimización de rutas de recolección de residuos: un caso de estudio en Latinoamérica. *Revista de Estudios Urbanos*, 15(3), 210-226.

Gutiérrez, F., & Nava, A. (2018). Desafíos en la gestión de residuos sólidos en ciudades latinoamericanas. *Revista Latinoamericana de Estudios Ambientales*, 10(2), 45-62.

- Hart, P. E., Nilsson, N. J., & Raphael, B. (1968). A formal basis for the heuristic
- Hart, P. E., Nilsson, N. J., & Raphael, B. (1968). A formal basis for the heuristic determination of minimum cost paths. *IEEE Transactions on Systems Science and Cybernetics*, 4(2), 100-107.
- Kirkpatrick, S., Gelatt, C. D., & Vecchi, M. P. (1983). Optimization by simulated annealing. *Science*, 220(4598), 671-680.
- Longley, P. A., Goodchild, M. F., Maguire, D. J., & Rhind, D. W. (2015). *Geographic Information Systems and Science*. John Wiley & Sons.
- Méndez, R. (2020). *Gestión Integral de Residuos Sólidos Urbanos*. Editorial Académica Española.
- Organización de las Naciones Unidas. (2021). *Informe sobre Gestión Sostenible de Residuos en Ciudades de Rápido Crecimiento*. Programa de las Naciones Unidas para el Desarrollo.
- Pérez, L., Martínez, M., & Sánchez, J. (2020). Optimización de rutas de recolección de residuos sólidos en ciudades europeas. *Journal of Waste Management*, 45(3), 212-227.
- Silva, F., Hernández, A., & López, M. (2017). Ineficiencias en la planificación de rutas de recolección de residuos y su impacto ambiental. *Revista Internacional de Contaminación Ambiental*, 33(2), 323-336.
- Toth, P., & Vigo, D. (2014). *Vehicle routing: problems, methods, and applications*. Society for Industrial and Applied Mathematics.
- Wilson, D. C., Velis, C., & Cheeseman, C. (2013). Role of informal sector recycling in waste management. *Waste Management*, 33(5), 629-637.

Yu, B., Yang, Z. Z., & Yao, B. (2011). An improved ant colony optimization for vehicle routing problem. *European Journal of Operational Research*, 196(1), 171-176.

Fernández, E. (2011). *Algoritmos Heurísticos en Optimización*. Universidad de Santiago de Compostela. https://eio.usc.es/pub/mte/descargas/ProyectosFinMaster/Proyecto_782.pdf

Lawler, E. L., Lenstra, J. K., Rinnooy Kan, A. H., & Shmoys, D. B. (1985). *The traveling salesman problem: A guided tour of combinatorial optimization*. John Wiley & Sons.

Clarke, G., & Wright, J. W. (1964). *Scheduling of vehicles from a central depot to a number of delivery points*. *Operations Research*, 12(4), 568-581.

Laporte, G. (1992). *The vehicle routing problem: An overview of exact and approximate algorithms*. *European Journal of Operational Research*, 59(3), 345-358.

Lin, S. (1965). *Computer solutions of the traveling salesman problem*. *Bell System Technical Journal*, 44(10), 2245-2269.

Dorigo, M., & Stützle, T. (2004). *Ant Colony Optimization*. MIT Press.

Kirkpatrick, S., Gelatt, C. D., & Vecchi, M. P. (1983). *Optimization by simulated annealing*. *Science*, 220(4598), 671-680.

Goldberg, D. E. (1989). *Genetic algorithms in search, optimization, and machine learning*. Addison-Wesley.

Glover, F. (1989). *Tabu Search—Part I*. *ORSA Journal on Computing*, 1(3), 190-206.

Hart, P. E., Nilsson, N. J., & Raphael, B. (1968). *A formal basis for the heuristic determination of minimum cost paths*. IEEE Transactions on Systems Science and Cybernetics, 4(2), 100-107.

Korf, R. E. (1985). *Depth-first iterative-deepening: An optimal admissible tree search*. Artificial Intelligence, 27(1), 97-109.

Nilsson, N. J. (1998). *Artificial intelligence: A new synthesis*. Morgan Kaufmann.

Cormen, T. H., Leiserson, C. E., Rivest, R. L., & Stein, C. (2009). *Introduction to Algorithms* (3rd ed.). MIT Press.

Tarjan, R. E. (1972). *Depth-first search and linear graph algorithms*. SIAM Journal on Computing, 1(2), 146-160.

Boehm, B. W. (1981). *Software Engineering Economics*. Prentice-Hall.

Evans, E. (2003). *Domain-Driven Design: Tackling Complexity in the Heart of Software*. Addison-Wesley Professional.

Fowler, M. (2018). *Refactoring: Improving the Design of Existing Code* (2nd ed.). Addison-Wesley Professional.

ISO/IEC. (2011). *ISO/IEC 25010:2011 Systems and software engineering — Systems and software Quality Requirements and Evaluation (SQuaRE)*. International Organization for Standardization.

Jones, C. (2013). *The Economics of Software Quality*. Addison-Wesley Professional.

Martin, R. C. (2017). *Clean Code: A Handbook of Agile Software Craftsmanship*. Prentice Hall.

Pressman, R., & Maxim, B. (2020). *Software Engineering: A Practitioner's Approach* (9th ed.). McGraw-Hill Education.

Project Management Institute. (2017). *A Guide to the Project Management Body of Knowledge (PMBOK Guide)* (6th ed.). Project Management Institute.

Schwaber, K., & Sutherland, J. (2020). *The Scrum Guide: The Definitive Guide to Scrum*. Scrum.org.

Sommerville, I. (2016). *Software Engineering* (10th ed.). Pearson.

Standish Group. (2020). *CHAOS 2020: Beyond Infinity*. The Standish Group International.

Wiegers, K., & Beatty, J. (2013). *Software Requirements* (3rd ed.). Microsoft Press.

Schwaber, K., & Sutherland, J. (2020). *The Scrum Guide: The Definitive Guide to Scrum: The Rules of the Game*. Scrum.org.

GLOSARIO DE TÉRMINOS

ACO: Optimización por Colonia de Hormigas (Ant Colony Optimization). Algoritmo inspirado en el comportamiento de las hormigas para resolver problemas de optimización combinatoria.

API: Interfaz de Programación de Aplicaciones (Application Programming Interface). Conjunto de definiciones y protocolos que se utilizan para desarrollar e integrar software de aplicaciones.

Archivo GPX: GPS Exchange Format. Formato de archivo basado en XML para intercambiar datos GPS entre aplicaciones y servicios web.

Flask: Framework ligero de desarrollo web escrito en Python.

Folium: Biblioteca de Python que permite manipular datos y visualizarlos en mapas interactivos con Leaflet.js.

GRSU: Gestión de Residuos Sólidos Urbanos. Conjunto de actividades relacionadas con el manejo de residuos generados en áreas urbanas.

NetworkX: Biblioteca de Python para el estudio de grafos y redes complejas.

Recocido Simulado (SA): Algoritmo probabilístico de búsqueda global, inspirado en el proceso de recocido en metalurgia.

SCRUM: Marco de trabajo ágil para la gestión y desarrollo de software complejo.

SIG: Sistemas de Información Geográfica. Conjunto de herramientas que integran y relacionan diversos componentes que permiten la organización, almacenamiento, manipulación, análisis y modelización de grandes cantidades de datos vinculados a una referencia espacial.

SQLAlchemy: Biblioteca SQL de Python que proporciona un ORM (Object-Relational Mapping) para facilitar la comunicación entre Python y las bases de datos relacionales.

VRP: Problema de Enrutamiento de Vehículos (Vehicle Routing Problem). Problema de optimización combinatoria que busca encontrar las rutas óptimas para una flota de vehículos que deben visitar un conjunto de puntos.

CAPÍTULO V:

ANEXOS

5.1 Formularios de los requerimientos

5.1.1 FORMULARIO DE LEVANTAMIENTO DE REQUERIMIENTOS

Fecha: _____

Responsable: _____

Entrevistado: _____

Cargo: _____

Área: _____

1. Información General

1.1. ¿Cuál es el objetivo principal del sistema de gestión de recolección de residuos?

1.2. ¿Quiénes serán los principales usuarios del sistema? (Marcar con una X)

☐ Personal administrativo

☐ Conductores de recolección

☐ Ciudadanos

☐ Otros: _____

2. Requerimientos Funcionales

2.1. ¿Qué funcionalidades debe tener el sistema? (Marcar las que apliquen)

☐ Monitoreo en tiempo real de los vehículos de recolección

☐ Optimización de rutas de recolección

☐ Notificaciones de contenedores llenos

☐ Generación de informes sobre rutas y recolección

☐ Visualización de rutas en un mapa interactivo

☐ Otros: _____

2.2. ¿Se requiere la integración con otros sistemas municipales?

☐ Sí (Especificar cuáles): _____

☐ No

2.3. ¿Qué formato de entrada de datos se utilizará?

☐ Archivos GPX

☐ Datos manuales

☐ Sensores de llenado

☐ Otros: _____

3. Requerimientos No Funcionales

3.1. ¿Cuáles son los tiempos de respuesta esperados del sistema?

☐ Menos de 5 segundos para calcular rutas

☐ Entre 5 y 10 segundos

☐ Más de 10 segundos

3.2. ¿Es necesario un sistema de acceso con autenticación?

☐ Sí (Especificar roles: _____)

☐ No

3.3. ¿Qué navegadores debe soportar el sistema?

☐ Google Chrome

☐ Mozilla Firefox

☐ Microsoft Edge

☐ Otros: _____

3.4. ¿El sistema debe ser accesible desde dispositivos móviles?

☐ Sí

☐ No

4. Seguridad y Confidencialidad

4.1. ¿Se debe proteger la información sensible de las rutas y los usuarios?

☐ Sí (Especificar qué información: _____)

☐ No

4.2. ¿Es necesario llevar un registro de auditoría de las operaciones realizadas?

☐ Sí

☐ No

5. Validación de Requerimientos

5.1. ¿Con qué frecuencia se debe actualizar la información de rutas?

☐ En tiempo real

☐ Cada hora

☐ Diario

☐ Otro: _____

5.2. ¿Cómo se notificará a los responsables en caso de incidencias?

☐ Correo electrónico

☐ Notificación en la plataforma

☐ Mensaje SMS

☐ Otro: _____

Observaciones adicionales:

Firma del entrevistado: _____

Firma del responsable: _____

5.1.2 ENTREVISTA 2: OPERADOR DE RECOLECCIÓN DE RESIDUOS

Fecha: _____

Entrevistador: _____

Nombre del Entrevistado: _____

Cargo: _____

1. Información del Operador

1.1. ¿Cuánto tiempo lleva trabajando en la recolección de residuos?

- ☐ Menos de 1 año
- ☐ 1 a 3 años
- ☐ Más de 3 años

1.2. ¿Qué tipo de rutas realiza con mayor frecuencia?

- ☐ Zona urbana
- ☐ Zona rural
- ☐ Ambas

2. Experiencia en el Proceso de Recolección

2.1. ¿Cuáles son las principales dificultades que enfrenta durante su recorrido?

- ☐ Rutas confusas o mal definidas
- ☐ Congestión vehicular
- ☐ Contenedores desbordados
- ☐ Falta de comunicación con la central
- ☐ Otros: _____

2.2. ¿Recibe instrucciones claras y actualizadas sobre las rutas que debe seguir?

- ☐ Siempre
- ☐ Algunas veces
- ☐ Rara vez

3. Requisitos Técnicos

3.1. ¿Cree que un sistema de monitoreo en tiempo real facilitaría su trabajo? ¿Por qué?

3.2. ¿Qué dispositivos prefiere para recibir las instrucciones de ruta?

- ☐ Aplicación móvil
- ☐ Dispositivo GPS dedicado
- ☐ Otro: _____

Observaciones adicionales:

Firma del Entrevistado: _____

Firma del Entrevistador: _____

5.1.3 ENTREVISTA 3: CIUDADANO (USUARIO DEL SERVICIO DE RECOLECCIÓN)

Fecha: _____

Entrevistador: _____

Nombre del Entrevistado (opcional): _____

Barrio o Sector: _____

1 Percepción del Servicio

1.1. ¿Con qué frecuencia se recoge la basura en su área?

- ☐ Diario
- ☐ Tres veces por semana
- ☐ Una vez por semana
- ☐ Ocasionalmente

1.2. ¿Ha notado problemas con la recolección de residuos? (Marcar los que apliquen)

- ☐ Retrasos en la recolección
- ☐ Contenedores desbordados
- ☐ Falta de información sobre horarios de recolección
- ☐ Otros: _____

2 Opinión sobre un Sistema de Monitoreo

2.1. ¿Cree que un sistema que permita monitorear las rutas en tiempo real mejoraría el servicio?

- ☐ Sí
- ☐ No

2.2. ¿Le gustaría poder reportar incidencias (contenedores llenos, basura acumulada) desde su teléfono?

☐ Sí

☐ No

Observaciones adicionales:

Firma del Entrevistado (opcional): _____

Firma del Entrevistador: _____

5.2. Formato Requerimientos Metodología Scrumm

Metodología Scrum para Sistema de Optimización de Rutas de Recolección de Basura - Municipio de Yanzatza

1. Visión del Producto

a. Declaración de la Visión

Desarrollar un sistema de optimización de rutas para la recolección de basura del Municipio de Yanzatza que mejore la eficiencia operativa, reduzca costos y tiempos de recolección

b. Objetivos del Producto

- Optimizar las rutas de recolección de basura para minimizar distancias y tiempos.
- Implementar un sistema de administración de choferes y vehículos.
- Mejorar la cobertura del servicio de recolección en un 15%.
- Proporcionar herramientas de monitoreo y reportes para la toma de decisiones.

c. Usuarios y Stakeholders

- Usuarios Primarios:
 - Administradores del departamento de gestión de residuos
 - Choferes de vehículos recolectores
 - Supervisores de operaciones
- Stakeholders:
 - Alcalde del Municipio de Yanzatza
 - Concejales municipales
 - Ciudadanos del municipio
 - Departamentos que intervienen en el desarrollo

2. Roles del Equipo Scrum

a. Product Owner

- Responsabilidades: Definir y priorizar el Product Backlog, aceptar o rechazar los entregables, representar los intereses de los stakeholders.

- Asignado a: Tesista
- b. Scrum Master
- Responsabilidades: Facilitar el proceso Scrum, eliminar impedimentos, hay que asegurar que el equipo siga las prácticas Scrum.
 - Asignado a: Tesista
- c. Equipo de Desarrollo
- Responsabilidades: Desarrollar y entregar incrementos del producto.
 - Miembros:
 - Desarrollador Frontend (1)
 - Desarrollador Backend (1)
 - Especialista en GIS y optimización de rutas (1)
 - Analista de datos (1)
 - Diseñador UX/UI (1)
 - Tester (1)

3. Product Backlog

Requerimientos: (Tabla de requerimientos)

- a. Sistema de Administración de Choferes
- b. Sistema de Administración de Vehículos
- c. Optimización de Ruta
- d. Monitoreo y Reportes

3.2 Historias de Usuario

R01: Sistema de Administración de Choferes

ID	Historia de Usuario	Criterios de Aceptación	Prioridad	Estimación
US1	Como administrador, quiero registrar nuevos choferes	- El sistema debe permitir ingresar datos	Alta	5

	en el sistema para mantener actualizada la base de datos del personal.	personales (nombre, cédula, licencia, etc.). - Debe validar que la cédula y número de licencia sean únicos - Debe almacenar la fecha de vencimiento de la licencia - Debe permitir subir una foto del chofer		
US2	Como administrador, quiero EDITAR nuevos choferes en el sistema para mantener actualizada la base de datos del personal.	- Debe poder editar nombre, apellido, e mail	Media	3
US3	Como administrador, quiero DESACTIVAR nuevos choferes en el sistema para mantener actualizada la base de datos del personal.	- Debe poder desactivar un usuario que ya no se encuentra operativo	Media	3

Épica: Optimización de Rutas

ID	HISTORIA DE USUARIO	CRITERIO DE ACEPTACIÓN	PRIORIDAD	ESTIMACIÓN
US11	Como administrador, quiero definir las tres rutas principales de recolección para establecer la cobertura del servicio	Debe permitir dibujar rutas en un mapa
- Debe calcular distancia y tiempo estimado
- Debe permitir definir puntos de inicio y fin	Alta	13
US12	Como administrador, quiero optimizar automáticamente las rutas para minimizar distancias y tiempos	Debe considerar variables como tráfico, horarios y capacidad
- Debe mostrar comparativa entre ruta actual y optimizada
- Debe permitir ajustes manuales	Alta	21
US13	Como supervisor, quiero visualizar las rutas en un mapa para monitorear la cobertura.	Debe mostrar rutas con códigos de colores Debe permitir ver detalles al hacer clic Debe actualizar	Media	13

		la posición de los vehículos en tiempo real		
US14	Como chofer, quiero recibir la ruta optimizada en mi dispositivo móvil para seguirla durante mi jornada.	Debe mostrar indicaciones paso a paso - Debe funcionar offline - Debe permitir reportar incidencias en la ruta	Alta	13
US15	Como administrador, quiero establecer parámetros de optimización para ajustar el algoritmo a necesidades específicas	Debe permitir priorizar tiempo, distancia o cobertura - Debe guardar diferentes configuraciones	Media	8

Épica: Monitoreo y Reportes

ID	Historia de usuario	Criterio de aceptación	Prioridad	Estimación
US16	Como administrador, quiero un dashboard con indicadores clave para monitorear la operación diaria.	- Debe mostrar vehículos en ruta vs. Disponibles - Debe mostrar progreso de cada ruta	Alta	13

		- Debe actualizar datos cada 5 minutos		
US17	Como supervisor, quiero generar reportes de eficiencia por ruta para evaluar la optimización	Debe comparar tiempos y distancias antes/después de optimización - Debe calcular ahorro de combustible - Debe permitir exportar en PDF	Media	8
US18	Como administrador, quiero reportes de rendimiento de choferes para evaluar su desempeño	- Debe mostrar cumplimiento de rutas - Debe mostrar incidencias reportadas - Debe permitir comparar entre choferes	Baja	8
US19	Como alcalde, quiero un reporte ejecutivo mensual para evaluar el impacto del sistema.	Debe mostrar indicadores clave de desempeño - Debe incluir gráficos comparativos - Debe destacar logros y áreas de mejora	Baja	5

US20	Como supervisor, quiero recibir alertas sobre incidencias en tiempo real para resolver problemas rápidamente.	Debe notificar retrasos significativos - Debe alertar sobre averías de vehículos	Media	8
------	---	--	-------	---

4. Planificación de Sprints

4.1 Duración del Sprint

- Cada Sprint tendrá una duración de 2 semanas.

4.2 Velocidad Estimada del Equipo

- Velocidad inicial estimada: 40 puntos por Sprint.

4.3 Plan de Releases

Release	Épicas	Sprints Estimados	Fecha Estimada
R1: Administración Básica	EP-01, EP-02 (parcial)	2	15-marzo 2025
R2: Optimización de Rutas	EP-02 (resto), EP-03 (parcial)	2	15-mayo-2025
R3: Sistema Completo	EP-03 (resto), EP-04	2	15-agosto-2025

4.4 Planificación Inicial de Sprints

Sprint 1

Meta: Implementar el registro y gestión básica de choferes y vehículos. Historias de Usuario:

US-01, US-02, US-06, US-07

- Duración: 4 horas
- Participantes: Equipo Scrum completo
- Objetivo: Implementar el registro y gestión básica de choferes y vehículos.

Sprint 2

Meta: Completar la administración de choferes y vehículos con asignaciones. Historias de Usuario: US-03, US-04, US-08, US-09 Puntos totales: 26

Sprint 3

Meta: Implementar la definición y visualización básica de rutas. Historias de Usuario: US-05, US-10, US-11, US-13 Puntos totales: 36

Sprint 4

Meta: Desarrollar el algoritmo de optimización de rutas. Historias de Usuario: US-12, US-15, US-20 Puntos totales: 37

Sprint 5

Meta: Implementar la visualización móvil y el dashboard principal. Historias de Usuario: US-14, US-16, US-17 Puntos totales: 34

Sprint 6

Meta: Completar reportes y ajustes finales del sistema. Historias de Usuario: US-18, US-19, ajustes y correcciones Puntos totales: 13 + correcciones

5. Definición de "Terminado" (Definition of Done)

Una historia de usuario se considera "Terminada" cuando:

1. El código está completo y sigue los estándares de codificación.
2. Se han realizado pruebas unitarias con cobertura mínima del 80%.
3. Se han realizado pruebas de integración.
4. La funcionalidad ha sido revisada y aprobada por otro miembro del equipo (peer review).
5. La documentación está actualizada.
6. El Product Owner ha aceptado la funcionalidad.
7. El código está desplegado en el ambiente de pruebas.
8. No hay defectos críticos o de alta prioridad pendientes.

6. Ceremonias Scrum

6.1 Sprint Planning

6.2 Daily Scrum

- Duración: 15 minutos
- Frecuencia: Diariamente

- Participantes: Equipo de Desarrollo y Scrum Master
- Objetivo: Sincronizar actividades y crear un plan para las próximas 24 horas.

6.3 Sprint Review

- Duración: 2 horas
- Frecuencia: Al final de cada Sprint
- Participantes: Equipo Scrum completo y stakeholders invitados
- Objetivo: Inspeccionar el incremento y adaptar el Product Backlog si es necesario.

6.4 Sprint Retrospective

- Duración: 1.5 horas
- Frecuencia: Al final de cada Sprint, después del Sprint Review
- Participantes: Equipo Scrum completo
- Objetivo: Inspeccionar el proceso y crear un plan de mejoras para el próximo Sprint.

6.5 Refinamiento del Backlog

- Duración: 2 horas
- Frecuencia: A mitad de cada Sprint
- Participantes: Product Owner y Equipo de Desarrollo
- Objetivo: Revisar y refinar las historias de usuario para futuros Sprints.

7. Herramientas y Tecnologías

7.1 Gestión del Proyecto

- Jira para gestión de Sprints y seguimiento de historias de usuario
- Confluence para documentación
- Microsoft Teams para comunicación

7.2 Desarrollo

- Frontend: React.js con Material-UI
- Backend: Node.js con Express
- Base de datos: PostgreSQL con extensión PostGIS

- Optimización de rutas: algoritmos basados en OSRM (Open Source Routing Machine)
- Control de versiones: Git con GitHub
- CI/CD: Jenkins

7.3 Entornos

- Desarrollo: Servidores locales
- Pruebas: Servidor en la nube
- Producción: Infraestructura del Municipio

8. Gestión de Riesgos

ID	Riesgo	Probabilidad	Impacto	Estrategia de mitigación
R1	Datos geográficos incompletos o desactualizado	Alto	Alto	Realizar un levantamiento inicial de datos y validar con personal de campo
R2	Resistencia al cambio por parte de los choferes	Medio	Alto	Involucrar a los choferes desde el inicio y proporcionar capacitación adecuada
R3	Problemas de conectividad en zonas rurales	Alta	Medio	Implementar funcionalidad offline y sincronización cuando haya conexión
R4	Cambios en los requisitos durante el desarrollo	Medio	Medio	Mantener comunicación constante con stakeholders y gestionar el cambio adecuadamente
R5	Limitaciones técnicas de	Medio	Bajo	Realizar un inventario tecnológico y adaptar la

los dispositivos municipales	solución a las capacidades existentes
---------------------------------	--

9. Plan de Capacitación

9.1 Capacitación para Administradores

- Duración: 8 horas (2 sesiones de 4 horas)
- Contenido:
- Gestión de usuarios y permisos
- Administración de choferes y vehículos
- Configuración y optimización de rutas
- Generación e interpretación de reportes

9.2 Capacitación para Supervisores

- Duración: 6 horas (2 sesiones de 3 horas)
- Contenido:
- Monitoreo de operaciones
- Gestión de incidencias
- Análisis de reportes
- Toma de decisiones basada en datos

9.3 Capacitación para Choferes

- Duración: 4 horas (1 sesión)
- Contenido:
- Uso de la aplicación móvil
- Seguimiento de rutas optimizadas
- Reporte de incidencias
- Procedimientos operativos

10. Métricas y KPIs

10.1 Métricas de Desarrollo

- Velocidad del equipo (puntos completados por Sprint)
- Deuda técnica
- Cobertura de pruebas
- Número de defectos por Sprint

10.2 Métricas de Producto

- Reducción en tiempo de recolección por ruta
- Reducción en consumo de combustible
- Incremento en cobertura del servicio
- Número de incidencias reportadas
- Satisfacción de usuarios (choferes y supervisores)

11. Anexos

11.1 Glosario de Términos

- Ruta: Recorrido predefinido para la recolección de basura.
- Optimización: Proceso de mejorar una ruta para reducir tiempo, distancia o recursos.
- Incidencia: Cualquier evento que afecte la normal ejecución de una ruta.
- KPI: Indicador Clave de Desempeño, métrica utilizada para evaluar el éxito.

11.2 Estructura Organizacional del Departamento de Gestión de Residuos

Ing. Fausto Román – DIRECTOR DEL DEPARTAMENTO DE MEDIO AMBIENTE DEL GAD YANTZAZA

Ing. Karla Montaña Viñan – COORDINADOR DEL DEPARTAMENTO DE MEDIO AMBIENTE DEL GAD YANTZAZA

Ing. Paola Mendoza – TÉCNICO DEL DEPARTAMENTO DE MEDIO AMBIENTE DEL GAD YANTZAZA

Ing. Fabian Orellana – TÉCNICO DEL DEPARTAMENTO DE MEDIO AMBIENTE DEL GAD YANTZAZA

Sr. Luis Ramon Montaña – CHOFER CAMION RECOLECTOR DE BASURA DEL DEPARTAMENTO DE MEDIO AMBIENTE DEL GAD YANTZAZA

Sr. Drwin Gonzalez – CHOFER CAMION RECOLECTOR DE BASURA DEL DEPARTAMENTO DE MEDIO AMBIENTE DEL GAD YANTZAZA

Sr. Julio Minga – CHOFER CAMION RECOLECTOR DE BASURA DEL
DEPARTAMENTO DE MEDIO AMBIENTE DEL GAD YANTZAZA

Sr. Roberto Encarnación – CHOFER CAMION RECOLECTOR DE BASURA DEL
DEPARTAMENTO DE MEDIO AMBIENTE DEL GAD YANTZAZA

11.3 Especificaciones Técnicas de la Flota de Vehículos

VEHÍCULO	MARCA	MODELO	AÑO	PLACA	ESTADO
Camión	HINO	FC	2012	ZMA - 0173	REGULAR
Camión	HINO	FC	2012	ZMA - 1001	REGULAR
Camión	HINO	GH	2008	ZMA - 0115	REGULAR
Camión	SCANIA	P310BAC9.32P 6*4 TM DIESEL CN	2022	ZMA - 1151	BUENO

5.3. Plan de Pruebas



Plan de Pruebas

Autor Nelson Torres

Versión: 1.0

Fecha: 19 de septiembre del 2025

Proyecto: Sistema de Optimización de Rutas para Vehículos de Recolección de Basura de Yantzaza

Responsable: Nelson Torres Pineda

1. Introducción

1.1 Propósito

Este documento establece el plan de pruebas para validar la funcionalidad, rendimiento y seguridad del Sistema de Optimización de Rutas GPS antes de su despliegue en producción.

1.2 Alcance

El plan cubre:

- Funcionalidades de todos los roles de usuario (Admin, Técnico, Coordinador, Chofer)
- Optimización de rutas GPS
- Navegación en tiempo real
- Gestión de datos de combustible
- Reportes y métricas
- Integración con APIs externas
- Seguridad y autenticación

1.3 Objetivos

- Garantizar que todas las funcionalidades cumplan con los requisitos
- Validar la experiencia de usuario en diferentes dispositivos
- Verificar la precisión del sistema GPS y tracking
- Confirmar la seguridad de datos sensibles
- Asegurar el rendimiento bajo diferentes cargas

2. Estrategia de Pruebas

2.1 Tipos de Pruebas

- Pruebas Funcionales: Validación de requisitos
- Pruebas de Usabilidad: Experiencia de usuario
- Pruebas de Rendimiento: Carga y estrés
- Pruebas de Seguridad: Vulnerabilidades y autenticación
- Pruebas de Compatibilidad: Navegadores y dispositivos
- Pruebas de Integración: APIs y servicios externos

2.2 Niveles de Pruebas

- Unitarias: Funciones individuales
- Integración: Módulos combinados
- Sistema: Aplicación completa
- Aceptación: Validación del usuario final

2.3 Criterios de Aceptación

- 100% de casos de prueba críticos aprobados
- 95% de casos de prueba de alta prioridad aprobados
- Sin defectos críticos pendientes
- Tiempo de respuesta < 3 segundos para operaciones normales
- Compatibilidad con navegadores principales (Chrome, Firefox, Safari, Edge)

3. Casos de Prueba por Módulo

3.1 Autenticación y Autorización

CP-AUTH-001: Inicio de Sesión Válido

Objetivo: Verificar el acceso con credenciales correctas

Precondiciones: Usuario registrado en el sistema

Pasos:

1. Navegar a la página de login
2. Ingresar username y password válidos
3. Hacer clic en "Iniciar Sesión"

Resultado Esperado: Redirección al dashboard correspondiente al rol

Prioridad: Crítica

Datos de Prueba:

- Admin: admin / admin2024!
- Chofer: chofer001 / password123

CP-AUTH-002: Inicio de Sesión Inválido

Objetivo: Verificar manejo de credenciales incorrectas

Pasos:

1. Ingresar credenciales incorrectas
2. Intentar iniciar sesión

Resultado Esperado: Mensaje de error "Usuario o contraseña incorrectos"

Prioridad: Alta

CP-AUTH-003: Cambio de Contraseña

Objetivo: Validar funcionalidad de cambio de contraseña

Pasos:

1. Iniciar sesión como chofer
2. Ir a "Cambiar contraseña"
3. Ingresar contraseña actual y nueva contraseña
4. Confirmar cambio

Resultado Esperado: Contraseña actualizada exitosamente

Prioridad: Alta

3.2 Gestión de Usuarios (Administrador)

CP-ADMIN-001: Crear Usuario

Objetivo: Verificar creación de nuevos usuarios

Precondiciones: Sesión iniciada como administrador

Pasos:

1. Navegar a "Gestión de Usuarios"
2. Hacer clic en "Crear Usuario"
3. Completar formulario con datos válidos
4. Seleccionar rol (Chofer, Coordinador, Técnico)
5. Guardar usuario

Resultado Esperado: Usuario creado y visible en la lista

Prioridad: Crítica

Datos de Prueba:

- Nombre: Juan Pérez
- Cédula: 1234567890
- Email: juan.perez@empresa.com
- Rol: Chofer

CP-ADMIN-002: Editar Usuario

Objetivo: Modificar información de usuario existente

Pasos:

1. Seleccionar usuario de la lista
2. Modificar campos editables
3. Guardar cambios

Resultado Esperado: Información actualizada correctamente

Prioridad: Alta

CP-ADMIN-003: Desactivar Usuario

Objetivo: Desactivar acceso de usuario

Pasos:

1. Seleccionar usuario activo
2. Cambiar estado a "Inactivo"
3. Confirmar cambio

Resultado Esperado: Usuario no puede iniciar sesión

Prioridad: Alta

3.3 Gestión de Vehículos

CP-VEH-001: Registrar Vehículo

Objetivo: Agregar nuevo vehículo al sistema

Pasos:

1. Ir a "Gestión de Vehículos"
2. Completar formulario de registro
3. Guardar información

Resultado Esperado: Vehículo registrado exitosamente

Prioridad: Alta

Datos de Prueba:

- Marca: Toyota
- Modelo: Hilux
- Año: 2022
- Placa: ABC-1234

CP-VEH-002: Asignar Vehículo a Chofer

Objetivo: Vincular vehículo con conductor

Pasos:

1. Seleccionar vehículo disponible
2. Asignar a chofer específico
3. Confirmar asignación

Resultado Esperado: Vehículo aparece en dashboard del chofer

Prioridad: Alta

3.4 Optimización de Rutas

CP-RUTA-001: Crear Ruta desde GPX

Objetivo: Generar ruta optimizada desde archivo GPX

Precondiciones: Archivo GPX válido disponible

Pasos:

1. Ir a "Crear Ruta"
2. Subir archivo(s) GPX
3. Configurar nombre y descripción
4. Seleccionar nivel de optimización
5. Procesar ruta

Resultado Esperado:

- Ruta creada con métricas de optimización
- Mapa visual disponible
- Estadísticas de mejora mostradas

Prioridad: Crítica

CP-RUTA-002: Validar Optimización

Objetivo: Verificar que la optimización reduce distancia

Pasos:

1. Crear ruta con múltiples archivos GPX
2. Comparar distancia original vs optimizada
3. Verificar métricas reportadas

Resultado Esperado:

- Distancia reducida o igual (nunca mayor)
- Métricas de ahorro calculadas correctamente

- Tiempo estimado ahorrado reportado

Prioridad: Crítica

CP-RUTA-003: Manejo de Archivos Inválidos

Objetivo: Validar manejo de archivos corruptos

Pasos:

1. Intentar subir archivo no-GPX
2. Subir archivo GPX corrupto
3. Verificar mensajes de error

Resultado Esperado: Mensajes de error claros y específicos

Prioridad: Media

3.5 Navegación GPS

CP-NAV-001: Iniciar Navegación

Objetivo: Comenzar navegación en tiempo real

Precondiciones:

- Chofer con sesión iniciada
- Ruta disponible
- Vehículo asignado

Pasos:

1. Seleccionar ruta del dashboard
2. Elegir vehículo
3. Indicar nivel de combustible inicial
4. Iniciar navegación

Resultado Esperado:

- Pantalla de navegación GPS activada
- Instrucciones turn-by-turn visibles

- GPS solicitando permisos de ubicación

Prioridad: Crítica

CP-NAV-002: Seguimiento GPS

Objetivo: Validar tracking de ubicación en tiempo real

Pasos:

1. Activar GPS en navegación
2. Simular movimiento (o movimiento real)
3. Verificar actualización de posición
4. Confirmar instrucciones de navegación

Resultado Esperado:

- Posición actualizada en mapa
- Instrucciones de giro precisas
- Estadísticas de velocidad y tiempo actualizadas

Prioridad: Crítica

CP-NAV-003: Completar Ruta

Objetivo: Finalizar navegación y registrar datos

Pasos:

1. Navegar hasta el final de la ruta
2. Activar "Completar Ruta"
3. Seleccionar nivel final de combustible
4. Agregar notas opcionales
5. Confirmar finalización

Resultado Esperado:

- Ruta marcada como completada
- Datos de consumo calculados
- Mapa de recorrido real generado

- Redirección al dashboard

Prioridad: Crítica

CP-NAV-004: Cancelar Navegación

Objetivo: Cancelar navegación en progreso

Pasos:

1. Durante navegación activa
2. Seleccionar "Cancelar Ruta"
3. Proporcionar razón de cancelación
4. Confirmar cancelación

Resultado Esperado:

- Navegación detenida
- Estado cambiado a "Cancelado"
- Razón registrada en sistema

Prioridad: Alta

3.6 Reportes y Métricas

CP-REP-001: Dashboard de Métricas

Objetivo: Visualizar estadísticas del sistema

Pasos:

1. Acceder como administrador
2. Revisar métricas en dashboard
3. Verificar cálculos de eficiencia

Resultado Esperado:

- Métricas actualizadas y precisas
- Gráficos visibles y comprensibles
- Datos de combustible calculados correctamente

Prioridad: Alta

CP-REP-002: Reporte de Rendimiento por Chofer

Objetivo: Analizar performance individual

Pasos:

1. Generar reporte de chofer específico
2. Revisar métricas de eficiencia
3. Verificar consumo de combustible

Resultado Esperado: Datos precisos y completos por conductor

Prioridad: Media

CP-REP-003: Exportar Reporte PDF

Objetivo: Generar reporte descargable

Pasos:

1. Seleccionar período de reporte
2. Elegir tipo de reporte
3. Descargar PDF

Resultado Esperado: PDF generado con formato profesional

Prioridad: Media

3.7 Seguridad

CP-SEC-001: Control de Acceso por Roles

Objetivo: Verificar restricciones de acceso

Pasos:

1. Intentar acceder a funciones de admin como chofer
2. Verificar redirección o error
3. Probar con diferentes roles

Resultado Esperado: Acceso denegado con mensaje apropiado

Prioridad: Crítica

CP-SEC-002: Protección de Datos Sensibles

Objetivo: Verificar que datos sensibles están protegidos

Pasos:

1. Inspeccionar código fuente en navegador
2. Revisar URLs directas a recursos
3. Verificar encriptación de contraseñas

Resultado Esperado:

- Contraseñas hasheadas en BD
- Datos sensibles no visibles en frontend
- URLs protegidas por autenticación

Prioridad: Crítica

CP-SEC-003: Sesiones y Logout

Objetivo: Validar manejo seguro de sesiones

Pasos:

1. Iniciar sesión
2. Cerrar navegador
3. Reabrir y verificar estado de sesión
4. Probar logout manual

Resultado Esperado:

- Sesiones expire apropiadamente
- Logout limpia sesión completamente

Prioridad: Alta

4. Pruebas de Rendimiento

4.1 Carga Normal

Objetivo: Verificar rendimiento bajo uso típico

Escenario: 10 usuarios concurrentes navegando

Criterio: Tiempo de respuesta < 3 segundos

4.2 Carga de Estrés

Objetivo: Determinar límites del sistema

Escenario: 50 usuarios concurrentes

Criterio: Sistema mantiene funcionalidad básica

4.3 Optimización de Rutas

Objetivo: Tiempo de procesamiento de optimización

Escenario: Archivos GPX de diferentes tamaños

Criterio:

- Archivo pequeño (< 100 puntos): < 5 segundos
- Archivo mediano (< 1000 puntos): < 30 segundos
- Archivo grande (< 5000 puntos): < 2 minutos

5. Pruebas de Compatibilidad

5.1 Navegadores Web

- Chrome: Versión 118+
- Firefox: Versión 115+
- Safari: Versión 16+
- Edge: Versión 118+

5.2 Dispositivos Móviles

- Android: 8.0+ (Chrome Mobile)
- iOS: 14+ (Safari Mobile)
- Tablets: iPad, Android tablets

5.3 Resoluciones de Pantalla

- Desktop: 1920x1080, 1366x768
- Mobile: 375x667, 414x896
- Tablet: 768x1024, 1024x768

6. Datos de Prueba

6.1 Usuarios de Prueba

Administrador:

- Usuario: kamontano
- Contraseña: kamontano123

Coordinador:

- Usuario: faroman
- Contraseña: faroman123

Técnico:

- Usuario: pamendoza
- Contraseña: pamendoza123

Chofer:

- Usuario: roencarnacion
- Contraseña: roencarnacion123

6.2 Archivos GPX de Prueba

- ruta_corta.gpx: 50 puntos, 5km
- ruta_media.gpx: 200 puntos, 25km
- ruta_larga.gpx: 1000 puntos, 100km
- ruta_compleja.gpx: Múltiples bucles y cruces

6.3 Vehículos de Prueba

Vehículo 1:

- Marca: Scannia
- Modelo: P310BAC9
- Año: 2022
- Placa: ZMA-1151

7. Criterios de Aprobación

7.1 Criterios de Aceptación

- Funcionalidad: 100% casos críticos pasados
- Rendimiento: Tiempos de respuesta dentro de límites
- Seguridad: Sin vulnerabilidades críticas
- Usabilidad: Navegación intuitiva validada
- Compatibilidad: Funcionamiento en navegadores objetivo

7.2 Criterios de Rechazo

- Defectos críticos que impidan funcionalidad principal
- Problemas de seguridad que expongan datos sensibles
- Rendimiento inaceptable (> 10 segundos operaciones básicas)
- Incompatibilidad con navegadores principales

8. Cronograma de Pruebas

Fase 1 - Pruebas Funcionales (5 días)

- Días 1-2: Autenticación y gestión de usuarios
- Días 3-4: Optimización de rutas
- Día 5: Navegación GPS básica

Fase 2 - Pruebas de Integración (3 días)

- Días 6-7: APIs y servicios externos
- Día 8: Flujos completos end-to-end

Fase 3 - Pruebas de Rendimiento (2 días)

- Día 9: Carga normal y estrés
- Día 10: Optimización de rendimiento

Fase 4 - Pruebas de Seguridad (2 días)

- Días 11-12: Vulnerabilidades y penetración

Fase 5 - Pruebas de Aceptación (2 días)

- Días 13-14: Validación con usuarios finales

9. Roles y Responsabilidades

9.1 Ejecutores de Prueba

- Desarrollador Principal: Pruebas unitarias y técnicas
- Tester QA: Pruebas funcionales y regresión
- Administrador Sistema: Pruebas de rendimiento
- Usuario Final: Pruebas de aceptación

9.2 Aprobadores

- Product Owner: Funcionalidad y requisitos
- Administrador Técnico: Seguridad y rendimiento
- Usuario Final: Usabilidad y experiencia

10. Gestión de Defectos

10.1 Clasificación de Severidad

- Crítica: Sistema no funciona, pérdida de datos
- Alta: Funcionalidad principal afectada
- Media: Funcionalidad secundaria afectada
- Baja: Problemas cosméticos o menores

10.2 Prioridades de Corrección

- P1 - Inmediata: Defectos críticos (< 24 horas)
- P2 - Alta: Defectos altos (< 48 horas)
- P3 - Normal: Defectos medios (< 1 semana)
- P4 - Baja: Defectos bajos (siguiente release)

10.3 Proceso de Reporte

1. Documentar defecto con pasos de reproducción
2. Asignar severidad y prioridad
3. Notificar al equipo de desarrollo
4. Hacer seguimiento hasta resolución
5. Verificar corrección con re-testing

11. Entregables

11.1 Reportes de Prueba

- Reporte ejecutivo de resultados
- Matriz de trazabilidad de requisitos
- Registro detallado de defectos
- Métricas de cobertura de pruebas

11.2 Documentación

- Casos de prueba ejecutados
- Scripts de pruebas automatizadas
- Manual de usuario actualizado
- Documentación de configuración

12. Aprobación Final

Criterios para Go-Live:

- ☒ Todos los casos críticos aprobados
- ☒ Defectos críticos y altos resueltos
- ☒ Rendimiento validado
- ☒ Seguridad verificada
- ☒ Documentación completa
- ☒ Usuario final capacitado

Aprobaciones Requeridas:

- ☒ Product Owner
- ☒ Administrador Técnico
- ☒ Usuario Final Representativo
- ☒ Equipo de Desarrollo

Documento preparado por: Nelson Torres Pineda

Fecha de preparación: 10 de septiembre del 2025

Próxima revisión: Post-implementación

Tlgo. Nelson Torres Pineda

ESTUDIANTE



Ing. Karla Montaña Viñan

COOR. UNIDAD MEDIO

AMBIENTE GAD YANTZAZA



5.4. Validación del Plan de Pruebas



Validación de Pruebas por Usuario

Autor

Nelson Eduardo Torres

Sistema de Optimización de Rutas GPS

Versión: 1.0

Fecha: 17 de septiembre de 2025

Tipo de documento: Matriz de Validación por Roles

1. Usuario Administrador

1.1 Datos de Acceso

- Usuario: kamontano
- Contraseña: kamontano123
- Rol: Administrador del Sistema

1.2 Casos de Prueba Específicos

VAL-ADM-001: Gestión Completa de Usuarios

Responsable: Administrador del Sistema

Tiempo estimado: 30 minutos

Caso de Prueba	Descripción	Criterio de Aceptación	Estado	Observaciones
CP-ADM-001	Crear usuario chofer	Usuario creado exitosamente	Aprobado	Verificar cédula única
CP-ADM-002	Crear usuario coordinador	Usuario creado con permisos correctos	Aprobado	Validar restricciones de acceso
CP-ADM-003	Crear usuario técnico	Usuario técnico funcional	Aprobado	Probar cambio de contraseñas
CP-ADM-004	Editar información de usuario	Cambios guardados correctamente	Aprobado	No permitir cédulas duplicadas

CP-ADM-005	Desactivar usuario	Usuario no puede acceder	Aprobado	Verificar logout automático
CP-ADM-006	Reactivar usuario	Acceso restaurado	Aprobado	Funcionalidades completas

Resultado Final: Aprobado

VAL-ADM-002: Gestión de Vehículos

Responsable: Administrador del Sistema

Tiempo estimado: 20 minutos

Caso de Prueba	Descripción	Criterio de Aceptación	Estado	Observaciones
CP-VEH-001	Registrar vehículo nuevo	Vehículo agregado al sistema	Aprobado	Placa única validada
CP-VEH-002	Editar información vehículo	Datos actualizados	Aprobado	Historial de cambios
CP-VEH-003	Asignar vehículo a chofer	Asignación exitosa	Aprobado	Disponibilidad actualizada
CP-VEH-004	Desasignar vehículo	Vehículo disponible nuevamente	Aprobado	Chofer notificado
CP-VEH-005	Desactivar vehículo	No disponible para asignación	Aprobado	Rutas activas manejadas

Resultado Final: Aprobado

VAL-ADM-003: Optimización y Creación de Rutas

Responsable: Administrador del Sistema

Tiempo estimado: 45 minutos

Caso de Prueba	Descripción	Criterio de Aceptación	Estado	Observaciones
CP-RUTA-001	Crear ruta desde GPX simple	Ruta generada con mapa	Aprobado	Archivo: ruta_simple.gpx
CP-RUTA-002	Crear ruta con múltiples GPX	Optimización aplicada	Aprobado	Métricas de ahorro mostradas
CP-RUTA-003	Optimización nivel básico	Tiempo < 30 segundos	Aprobado	Archivo: ruta_media.gpx
CP-RUTA-004	Optimización nivel avanzado	Mejores resultados	Aprobado	Comparar con nivel básico
CP-RUTA-005	Manejo de archivos inválidos	Errores descriptivos	Aprobado	Probar .txt, .pdf, archivo corrupto
CP-RUTA-006	Eliminar ruta	Ruta removida del sistema	Aprobado	Rutas activas protegidas

Resultado Final: Aprobado

VAL-ADM-004: Dashboard y Reportes

Responsable: Administrador del Sistema

Tiempo estimado: 25 minutos

Caso de Prueba	Descripción	Criterio de Aceptación	Estado	Observaciones
CP-REP-001	Visualizar métricas generales	Datos actualizados y precisos	Aprobado	Verificar cálculos manualmente
CP-REP-002	Reporte por período	Filtros funcionando	Aprobado	Probar diferentes rangos
CP-REP-003	Exportar reporte PDF	PDF descargable y legible	Aprobado	Verificar formato profesional
CP-REP-004	Métricas de optimización	Ahorros calculados correctamente	Aprobado	Comparar con datos originales

Resultado Final: Aprobado

1.3 Validación Final Administrador

Calificación General: Satisfactorio

2. Usuario Técnico

2.1 Datos de Acceso

- Usuario: pamendoza
- Contraseña: pamendoza123
- Rol: Técnico de Sistemas

2.2 Casos de Prueba Específicos

VAL-TEC-001: Gestión de Contraseñas

Responsable: Técnico de Sistemas

Tiempo estimado: 20 minutos

Caso de Prueba	Descripción	Criterio de Aceptación	Estado	Observaciones
CP-TEC-001	Cambiar contraseña de chofer	Contraseña actualizada	Aprobado	Validar nueva contraseña
CP-TEC-002	Cambiar contraseña coordinador	Acceso con nueva contraseña	Aprobado	Logout automático
CP-TEC-003	Resetear contraseña olvidada	Usuario puede acceder	Aprobado	Contraseña temporal
CP-TEC-004	Validar política contraseñas	Contraseñas débiles rechazadas	Aprobado	Mínimo 6 caracteres

Resultado Final: Aprobado

VAL-TEC-002: Administración de Usuarios

Responsable: Técnico de Sistemas

Tiempo estimado: 15 minutos

Caso de Prueba	Descripción	Criterio de Aceptación	Estado	Observaciones
CP-TEC-005	Activar/desactivar usuarios	Estado cambiado correctamente	Aprobado	No afectar administradores
CP-TEC-006	Ver historial de usuarios	Información completa visible	Aprobado	Incluir fechas de cambios
CP-TEC-007	Buscar usuario específico	Filtros funcionando	Aprobado	Por nombre, cédula, email

Resultado Final: Aprobado

2.3 Validación Final Técnico

Calificación General: Satisfactorio

3. Usuario Coordinador

3.1 Datos de Acceso

- Usuario: faroman
- Contraseña: faroman123
- Rol: Coordinador de Rutas

3.2 Casos de Prueba Específicos

VAL-COORD-001: Visualización y Gestión de Rutas

Responsable: Coordinador de Rutas

Tiempo estimado: 30 minutos

Caso de Prueba	Descripción	Criterio de Aceptación	Estado	Observaciones
CP-COORD-001	Ver todas las rutas	Lista completa visible	Aprobado	Incluir rutas activas e inactivas
CP-COORD-002	Ver detalles de ruta	Información completa	Aprobado	Mapa, métricas, historial
CP-COORD-003	Filtrar rutas por estado	Filtros funcionando	Aprobado	Activas, completadas, canceladas
CP-COORD-004	Buscar ruta específica	Búsqueda eficiente	Aprobado	Por nombre, ID, fecha

Resultado Final: Aprobado

VAL-COORD-002: Monitoreo de Recorridos

Responsable: Coordinador de Rutas

Tiempo estimado: 25 minutos

Caso de Prueba	Descripción	Criterio de Aceptación	Estado	Observaciones
CP-COORD-005	Ver rutas en progreso	Estado actualizado	Aprobado	Tiempo real
CP-COORD-006	Ver historial recorridos	Datos completos	Aprobado	Fechas, choferes, vehículos
CP-COORD-007	Analizar consumo combustible	Métricas precisas	Aprobado	Por ruta, chofer, vehículo
CP-COORD-008	Ver mapas de recorridos	Mapas cargando correctamente	Aprobado	Ruta planificada vs real

Resultado Final: Aprobado

VAL-COORD-003: Reportes y Análisis

Responsable: Coordinador de Rutas

Tiempo estimado: 20 minutos

Caso de Prueba	Descripción	Criterio de Aceptación	Estado	Observaciones
CP-COORD-009	Generar reporte de eficiencia	Datos calculados correctamente	Aprobado	Por período seleccionado
CP-COORD-010	Comparar rendimiento choferes	Rankings precisos	Aprobado	Múltiples métricas
CP-COORD-011	Exportar reporte coordinador	PDF generado	Aprobado	Formato profesional
CP-COORD-012	Métricas de optimización	Ahorros documentados	Aprobado	Km, tiempo, combustible

Resultado Final: Aprobado

3.3 Validación Final Coordinador

Calificación General: Aprobado

4. Usuario Chofer

4.1 Datos de Acceso

- Usuario: roencarnacion
- Contraseña: roencarnacion123
- Rol: Conductor/Chofer

4.2 Casos de Prueba Específicos

VAL-CHOF-001: Dashboard y Gestión Personal

Responsable: Chofer del Sistema

Tiempo estimado: 15 minutos

Caso de Prueba	Descripción	Criterio de Aceptación	Estado	Observaciones
CP-CHOF-001	Ver dashboard personal	Información actualizada	Aprobado	Rutas, vehículos, historial
CP-CHOF-002	Cambiar contraseña propia	Contraseña actualizada	Aprobado	Política de seguridad aplicada
CP-CHOF-003	Ver historial de rutas	Lista completa	Aprobado	Ordenado por fecha
CP-CHOF-004	Ver vehículos disponibles	Lista actualizada	Aprobado	Solo vehículos activos

Resultado Final: Aprobado

VAL-CHOF-002: Iniciación de Rutas

Responsable: Chofer del Sistema

Tiempo estimado: 25 minutos

Caso de Prueba	Descripción	Criterio de Aceptación	Estado	Observaciones
CP-CHOF-005	Seleccionar ruta disponible	Ruta cargada correctamente	Aprobado	Mapa visible
CP-CHOF-006	Seleccionar vehículo	Asignación exitosa	Aprobado	Solo vehículos disponibles
CP-CHOF-007	Indicar combustible inicial	Selector funcionando	Aprobado	Niveles 1-4
CP-CHOF-008	Iniciar navegación	GPS activado	Aprobado	Permisos de ubicación
CP-CHOF-009	Prevenir rutas múltiples	Solo una ruta activa	Aprobado	Mensaje de error claro

Resultado Final: Aprobado

VAL-CHOF-003: Navegación GPS (Crítica)

Responsable: Chofer del Sistema

Tiempo estimado: 40 minutos

Caso de Prueba	Descripción	Criterio de Aceptación	Estado	Observaciones
CP-NAV-001	Cargar ruta real	Coordenadas cargadas	Aprobado	Desde archivos GPX
CP-NAV-002	Activar GPS real	Ubicación obtenida	Aprobado	Precisión < 20 metros
CP-NAV-003	Instrucciones turn-by-turn	Direcciones precisas	Aprobado	Voz en español
CP-NAV-004	Tracking en tiempo real	Posición actualizada	Aprobado	Cada 30 segundos al servidor
CP-NAV-005	Controles de mapa	Zoom, centrar funcionando	Aprobado	Responsive en móvil
CP-NAV-006	Cambio vista mapa	Satélite, calle, nocturno	Aprobado	Transiciones suaves
CP-NAV-007	Comandos de voz	Síntesis funcionando	Aprobado	Control de mute
CP-NAV-008	Estadísticas tiempo real	Velocidad, tiempo, distancia	Aprobado	Cálculos precisos

Resultado Final: Aprobado

VAL-CHOF-004: Finalización de Rutas

Responsable: Chofer del Sistema

Tiempo estimado: 20 minutos

Caso de Prueba	Descripción	Criterio de Aceptación	Estado	Observaciones
CP-FINAL-001	Completar ruta exitosamente	Modal de completar	Aprobado	Selector de combustible
CP-FINAL-002	Selector combustible final	Visual e intuitivo	Aprobado	Conservar selección
CP-FINAL-003	Agregar notas opcionales	Texto guardado	Aprobado	Límite de caracteres
CP-FINAL-004	Resumen del viaje	Métricas calculadas	Aprobado	Tiempo, distancia, puntos GPS
CP-FINAL-005	Cancelar ruta	Estado actualizado	Aprobado	Razón requerida
CP-FINAL-006	Mapa recorrido real	Generado automáticamente	Aprobado	Ruta real vs planificada

Resultado Final: Aprobado

VAL-CHOF-005: Funcionalidades Móviles

Responsable: Chofer del Sistema

Tiempo estimado: 30 minutos

Caso de Prueba	Descripción	Criterio de Aceptación	Estado	Observaciones
CP-MOV-001	Responsive design	Funciona en móvil	Aprobado	Android/iOS
CP-MOV-002	Pantalla completa	Sin barras navegador	Aprobado	Experiencia app nativa
CP-MOV-003	Controles táctiles	Botones suficientemente grandes	Aprobado	Fácil uso con guantes
CP-MOV-004	Rotación pantalla	Adaptación automática	Aprobado	Horizontal/vertical
CP-MOV-005	Prevenir zoom accidental	Zoom controlado	Aprobado	Solo programático
CP-MOV-006	Batería y rendimiento	Uso eficiente GPS	Aprobado	No drenar batería

Resultado Final: Aprobado

4.3 Validación Final Chofer

Calificación General: Aprobado

5. Pruebas Transversales

5.1 Seguridad y Control de Acceso

Responsable: Administrador de Seguridad

Tiempo estimado: 45 minutos

Caso de Prueba	Descripción	Criterio de Aceptación	Estado	Observaciones
CP-SEC-001	Acceso por roles	Restricciones funcionando	Aprobado	Cada rol solo su área
CP-SEC-002	Protección URLs directas	Redirección o error 403	Aprobado	Sin bypass
CP-SEC-003	Encriptación contraseñas	Hash en base de datos	Aprobado	Nunca texto plano
CP-SEC-004	Sesiones seguras	Timeout y logout	Aprobado	Sin sesiones eternas
CP-SEC-005	Validación entrada datos	XSS y SQL injection	Aprobado	Sanitización completa

Resultado Final: Aprobado

5.2 Rendimiento del Sistema

Responsable: Administrador de Sistemas

Tiempo estimado: 30 minutos

Caso de Prueba	Descripción	Criterio de Aceptación	Estado	Observaciones
CP-PERF-001	Carga página dashboard	< 3 segundos	Aprobado	Conexión normal
CP-PERF-002	Optimización rutas	< 2 minutos archivo grande	Aprobado	5000+ puntos
CP-PERF-003	Carga mapas	< 5 segundos	Aprobado	Incluyendo tiles
CP-PERF-004	Múltiples usuarios	Sin degradación	Aprobado	10 usuarios concurrentes
CP-PERF-005	Navegación GPS	Actualización fluida	Aprobado	Sin lag notable

Resultado Final: Aprobado

6. Matriz de Aprobación Final

6.1 Resumen de Validaciones por Usuario

Usuario	Total Casos	Aprobados	Rechazados	Pendientes	Estado Final
Administrador	15	x			Aprobado
Técnico	10	x			Aprobado
Coordinador	5	x			Aprobado
Chofer	21	X			Aprobado
Transversales	15	X			Aprobado
TOTAL	66	x			Aprobado

6.2 Criterios de Aprobación Final

Para aprobar el sistema se requiere:

- 100% de casos críticos aprobados (navegación GPS, seguridad)
- 95% de casos de alta prioridad aprobados
- 85% de casos totales aprobados
- Cero defectos críticos pendientes
- Documentación de defectos conocidos

6.3 Firmas de Aprobación

Administrador del Sistema:

Nombre: Karla Montaño Firma: [Firma] Fecha: _____

6.4 Decisión Final

- ☒ SISTEMA APROBADO PARA PRODUCCIÓN
☐ SISTEMA REQUIERE CORRECCIONES MENORES
☐ SISTEMA RECHAZADO - REQUIERE CORRECCIONES MAYORES

Fecha de aprobación: 19 de septiembre de 2025

Fecha estimada de go-live: 19 de septiembre 2025

Documento validado por: Ing. Karla Montaño

Fecha de creación: 15 de septiembre del 2025

Próxima revisión: Post-implementación

Tlgo. Nelson Torres Pineda

ESTUDIANTE



Ing. Karla Montaño Viñan

COOR. UNIDAD MEDIO

AMBIENTE GAD YANTZAZA

5.5. Certificación de funcionamiento de la aplicación



Mgs. Lorena Elizabeth Conde Z.

**DIRECTORA DE LA CARRERA DE TECNOLOGÍAS DE LA
COMUNICACIÓN DE LA UIDE**

En su despacho.-

El Gobierno Autónomo Descentralizado Municipal de Yantzaza, por medio de la
Unidad de Gestión Ambiental,

Certifico que se llevó a cabo la prueba de la aplicación de optimización de rutas para los
vehículos recolectores de residuos sólidos. Dichas pruebas tienen como objetivo
mejorar la eficiencia en la recolección, optimizar los tiempos de recorrido y garantizar
una gestión más eficiente de los desechos sólidos en el cantón Yantzaza, desarrollado
por el tlgo. Nelson Torres, estudiante de la carrera de Ingeniería Tecnologías en
Comunicación.

Estas pruebas fueron supervisadas y ejecutadas por el encargado de la Unidad de
Gestión Ambiental del GAD Municipal de Yantzaza, quien vela por el correcto
funcionamiento y evaluación de los resultados obtenidos a través de la implementación
de esta tecnología.

Se expide el presente certificado a solicitud de parte interesada, en Yantzaza, a los 19
días del mes de septiembre del año 2025.

Att.

Ing. Karla Montaña Viñan

Coordinadora de la Unidad de Gestión Ambiental Gobierno Autónomo Descentralizado
Municipal de Yantzaza.

