



República del Ecuador

Universidad Tecnológica Empresarial de Guayaquil – UTEG

Trabajo de Titulación

Para la obtención del título de:

**Ingeniero en Sistemas Computacionales, Mención Aplicaciones Web y
Multimedia**

Tema:

**Diseño de una Aplicación Móvil para la Gestión de Pedidos de Gas a
Domicilio basada en Zonificación Geográfica para la ciudad de Cuenca**

Autor:

David Eduardo Balbuca Castillo

davidbalbucac@gmail.com

Director de Trabajo de titulación:

Ing. Andres Leonardo Ortega Ortega PHD.

Junio 2024

Guayaquil – Ecuador

AGRADECIMIENTO

Quisiera aprovechar esta oportunidad para expresar mi sincero agradecimiento a todas las personas que han contribuido de manera significativa en la realización de esta tesis. Su apoyo, orientación y aliento fueron fundamentales en este viaje académico y de investigación.

En primer lugar, deseo agradecer a mi distinguido tutor, Ing, Andres Ortega por su guía experta y dedicación incansable a lo largo de este proceso. Sus conocimientos, paciencia y orientación fueron fundamentales para dar forma a esta investigación y alcanzar los objetivos propuestos. Su mentoría ha sido una fuente constante de inspiración y aprendizaje.

También quiero expresar mi profundo agradecimiento a todos mis profesores y docentes que compartieron su sabiduría y experiencia a lo largo de mi carrera académica. Cada uno de ustedes contribuyó al desarrollo de mis habilidades y conocimientos, y estoy agradecido por las lecciones que he aprendido de cada uno.

Agradezco sinceramente a mis amigos y compañeros de clase por su apoyo y compañía a lo largo de esta travesía. Sus palabras de aliento y colaboración en proyectos compartidos fueron invaluable para superar desafíos y mantenerme motivado.

A mi familia, les agradezco por su amor incondicional y apoyo constante. Sus sacrificios y aliento me han permitido alcanzar este logro académico. Este logro también es de ustedes.

Con gratitud sincera,

David Balbuca Castillo

DEDICATORIA

Dedico este logro a las personas que han sido mi fuente constante de inspiración, amor y apoyo a lo largo de este camino académico.

A mi amada esposa, Sami, tu amor incondicional, paciencia y comprensión han sido mi mayor fortaleza. Gracias por estar a mi lado en cada desafío y celebración, y por ser mi refugio en los momentos de estudio intenso. Este logro es tuyo tanto como mío.

A mis queridos hijos, Agustin y Juan David, su sonrisa y alegría han iluminado mis días y me han motivado a esforzarme al máximo. Espero que esta tesis sea un ejemplo para ustedes de que, con perseverancia y dedicación, se pueden alcanzar grandes metas.

A mi madre, Ilda, quien me inculcó desde temprana edad el valor de la educación y la importancia del esfuerzo. Tu amor y apoyo inquebrantable son un faro en mi vida, y este logro es un tributo a tu sacrificio y devoción.

A mis seres queridos, amigos y familiares, quienes han estado a mi lado a lo largo de este viaje, su aliento y buenos deseos han sido un impulso invaluable.

Esta tesis está dedicada a todos ustedes, quienes han compartido este camino conmigo. Su amor y apoyo han sido el motor que me ha llevado a alcanzar este hito en mi vida académica. Con gratitud y amor, les dedico este logro.

Con amor y gratitud sincera.

David Balbuca Castillo

DECLARACIÓN DE AUTORÍA DEL ESTUDIANTE

Yo, David Eduardo Balbuca Castillo, declaro solemnemente que el trabajo presentado en esta tesis, titulada “Diseño de una Aplicación Móvil para la Gestión de Pedidos de Gas a Domicilio basada en Zonificación Geográfica para la ciudad de Cuenca” es el resultado de mi propio esfuerzo e investigación. A excepción de las fuentes debidamente citadas y referenciadas, todo el contenido de este trabajo, incluyendo, pero no limitado a los textos, gráficos, figuras y resultados, ha sido generado por mí de manera independiente. Confirmando que he respetado rigurosamente las normas y directrices éticas de la investigación académica durante la realización de este estudio. Cualquier colaboración con otros investigadores o el uso de material previamente publicado ha sido debidamente reconocido y citado.

Asimismo, manifiesto que no he presentado este trabajo previamente como parte de ninguna otra calificación académica o en ninguna otra institución educativa. Asumo la plena responsabilidad de los contenidos, conclusiones y resultados presentados en esta tesis.

Firmado:

10 de junio de 2024

ÍNDICE

Tabla de contenido

1. RESUMEN	1
2. ABSTRACT	2
3. INTRODUCCIÓN	3
3.1. JUSTIFICACIÓN.....	3
3.2. PROBLEMA Y FINALIDAD DEL TRABAJO.....	4
3.2.1. Finalidad del Trabajo:	4
3.2.2. Problema de Investigación:.....	5
3.3. OBJETIVOS	7
4. MARCO TEÓRICO	7
4.1. APLICACIÓN MÓVIL.	7
4.1.1. Definición según varios autores.....	7
4.1.2. Georreferenciación y Asignación de Distribuidores.	8
4.2. HERRAMIENTAS DE DESARROLLO	8
4.2.1. QGIS: Generación de Geocercas	8
4.2.2. MATLAB: Herramienta de Cálculo Avanzado	9
4.2.3. SQLite: Base de Datos Relacional.....	9
4.2.4. Firebase: Almacenamiento en la Nube	9
4.2.5. Insomnia: Control de APIs.....	9
4.2.6. PyCharm: IDE para Programación.....	10
4.2.7. JavaScript: Generación de Polígonos	10
4.2.8. React Native: Entorno de Programación Móvil	10
4.2.9. OpenRouteService: Generación de Rutas.....	10
4.3. Backend y Almacenamiento de Datos	11
4.4. Tecnologías Móviles y Enrutamiento.....	11
5. METODOLOGÍA.....	11
5.1. DISEÑO DE LA INVESTIGACIÓN.....	11
5.2. ANALISIS Y DISEÑO DEL SISTEMA	11
5.2.1. ANALISIS.....	11
5.2.2. FLUJO	13
5.2.3. DISEÑO.....	15
5.2.3.1. DISEÑO DE GEOCERCAS.	15
5.2.3.2. CÁLCULO DE RUTAS	17

5.2.3.3.	<i>DISEÑO DE BASE DE DATOS</i>	21
5.2.3.4.	<i>SEGURIDAD</i>	22
5.3.	<i>ANÁLISIS DE RESULTADOS</i>	25
6.	CONCLUSIONES	30
7.	RECOMENDACIONES	31
8.	REFERENCIAS BIBLIOGRÁFICAS	32
9.	ANEXOS	34
9.1.	<i>Anexo 1. Funcionamiento aplicación móvil:</i>	34
9.2.	<i>Anexo 2. A continuación, se detalla el uso comparativo de las aplicaciones analizadas.</i>	39

Tabla de ilustraciones

Ilustración 1 - Cuadro Comparativo	6
Ilustración 2 - Flujo actual para servicio a domicilio de gas	12
Ilustración 3 - Flujo simple requerido para generar pedido.....	12
Ilustración 4 - Interacción usuario, sistema y cliente	13
Ilustración 5 - Ponderamiento de ruta.....	15
Ilustración 6 - Capa de Datos de Cuenca.....	16
Ilustración 7 - Diseño de Polígonos en sistema.....	17
Ilustración 8 - Distancia entre dos puntos	18
Ilustración 9 - Ubicación de un punto en una geocerca.....	19
Ilustración 10 - Encolamiento de pedidos ponderados	20
Ilustración 11 - Calculo de ruta ponderada.....	20
Ilustración 12 - Administración de base de datos.....	22
Ilustración 13 - uso de cloudflare	23
Ilustración 14 - Formularios	26
Ilustración 15 - Modelo Login.....	27
Ilustración 16 - Modelo GeoLocalización	29
Ilustración 17 - Sectores	29
Ilustración 18 - Subsectores.....	30
Ilustración 19 - Catastros.....	30
Ilustración 20 - Pantalla Principal	34
Ilustración 21 - Pedidos distribuidor	34
Ilustración 22 - pantalla solicitud cliente.....	35
Ilustración 23 - Pantalla Pedido cliente	35
Ilustración 24 - Pantalla envío pedido cliente	36
Ilustración 25 - Pantalla pedido en ruta	36

1. RESUMEN

En la ciudad de Cuenca, el servicio de gas a domicilio ha demostrado ser una opción popular, superando las estadísticas de métodos tradicionales de adquisición, como la compra en distribuidores cercanos o el uso de gas centralizado. Hasta hace poco, la principal modalidad de entrega implicaba camiones recorriendo la ciudad con una canción como método de notificación, pero la reciente ordenanza municipal, implementada en agosto de 2023 para reducir la contaminación auditiva, eliminó esta forma de aviso, generando un cambio significativo en la notificación y entrega del servicio.

Ante este contexto, esta tesis se propone diseñar y crear una aplicación móvil que permita a los usuarios solicitar el servicio de gas de manera eficiente, aprovechando la brecha dejada por la eliminación de las notificaciones sonoras. La aplicación simplifica el proceso para que los usuarios puedan realizar pedidos de cilindros de gas a sus domicilios mediante una interfaz fácil de usar. Al abrir la aplicación, se obtiene automáticamente la ubicación del dispositivo, facilitando la interacción con el distribuidor más cercano.

La aplicación utiliza un método desarrollado en JavaScript que, mediante puntos geográficos y polígonos generados a partir de capas de QGIS que representan las parroquias de la ciudad, identifica el distribuidor disponible en la zona del usuario. Además, se ha implementado un método eficiente de cálculo de la distancia euclidiana entre el usuario y los distribuidores presentes en su zona, utilizando herramientas como Matlab y Python como backend de la aplicación. Estos avances tecnológicos garantizan una gestión efectiva de la entrega de gas a domicilio, mejorando la experiencia del usuario y adaptándose a los cambios normativos en la ciudad.

2. ABSTRACT

In the city of Cuenca, home gas delivery services have emerged as a popular choice, surpassing traditional acquisition methods such as purchasing from nearby distributors or utilizing centralized gas services. Until recently, the primary delivery method involved trucks traversing the city with a song as a notification method. However, a recent municipal ordinance implemented in August 2023, aimed at reducing noise pollution, eliminated this form of notification, bringing about a significant change in the service's notification and delivery process.

Given this context, this thesis aims to design and develop a mobile application enabling users to efficiently request gas services, capitalizing on the gap left by the removal of audible notifications. The application streamlines the process, allowing users to easily place orders for gas cylinders at their homes through a user-friendly interface. Upon opening the application, the device's location is automatically obtained, facilitating interaction with the nearest distributor.

The application employs a JavaScript-based method that, utilizing geographical points and polygons generated from QGIS layers representing the city's parishes, identifies the available distributor in the user's zone. Additionally, an efficient method for calculating the Euclidean distance between the user and distributors in their area has been implemented, using tools such as Matlab and Python as the application's backend. These technological advancements ensure effective management of home gas delivery, enhancing the user experience and adapting to regulatory changes in the city.

3. INTRODUCCIÓN

El desarrollo tecnológico ha permeado prácticamente todos los aspectos de la sociedad moderna, influyendo significativamente en la forma en que accedemos a diversos servicios cotidianos como transporte y alimentación. La generalización del uso de aplicaciones móviles ha socializado a la población con esta tecnología, evidenciando que la simplificación y mejora en el acceso son clave para optimizar la experiencia de los usuarios.

En este contexto, la implementación de una aplicación móvil para la solicitud del servicio de gas representa una evolución natural en la búsqueda de eficiencia y comodidad. Este enfoque no solo simplifica el proceso para los usuarios, sino que también ofrece la posibilidad de ahorrar tiempo y costos en cada transacción. A lo largo de esta introducción, se explorará cómo esta integración tecnológica contribuye a la mejora de servicios, considerando el contexto actual y las necesidades de la población.

3.1.JUSTIFICACIÓN

La incorporación de tecnologías innovadoras en la prestación de servicios es esencial para adaptarse a las crecientes demandas y expectativas de la sociedad contemporánea. En este sentido, la aplicación móvil diseñada para la solicitud del servicio de gas se presenta como una respuesta estratégica a los cambios en el entorno urbano y normativo de la ciudad de Cuenca.

El contexto actual, marcado por la eliminación de las notificaciones sonoras para la entrega de gas a domicilio, ha generado una necesidad evidente de explorar alternativas que aseguren la continuidad del servicio de manera eficiente y en línea con las regulaciones municipales. La justificación de este proyecto radica en la búsqueda de una solución

innovadora que no solo responda a las nuevas normativas, sino que también mejore la experiencia del usuario y optimice la gestión de los distribuidores.

La aplicación móvil propuesta no solo se alinea con la tendencia generalizada de utilizar la tecnología para simplificar procesos cotidianos, sino que también aborda específicamente los desafíos surgidos a raíz de las recientes regulaciones. La conveniencia, rapidez y accesibilidad que ofrece la aplicación contribuirán no solo a mantener la calidad del servicio de gas a domicilio, sino también a posicionarse como una alternativa más eficiente y respetuosa con el entorno.

Con este antecedente, este proyecto busca fomentar la adopción de tecnologías innovadoras en el sector de distribución de gas, demostrando que la implementación de soluciones tecnológicas no solo es posible, sino también beneficiosa para todas las partes involucradas. La justificación de este trabajo se fundamenta en la necesidad de aportar soluciones prácticas y eficaces en el ámbito del servicio de gas a domicilio, respaldando la evolución de la industria hacia modelos más modernos y sostenibles.

3.2.PROBLEMA Y FINALIDAD DEL TRABAJO

3.2.1. Finalidad del Trabajo:

La presente investigación tiene como objetivo principal diseñar y desarrollar una aplicación móvil innovadora para la solicitud de servicios de gas a domicilio en Cuenca. La finalidad es proporcionar a los usuarios una herramienta intuitiva que les permita solicitar el servicio de manera eficiente, superando las limitaciones del método telefónico actual. Se busca homogeneizar la distribución del servicio mediante la zonificación, facilitar el acceso a múltiples distribuidores a través de la aplicación y mejorar la experiencia general del usuario. Esta solución no solo se ajusta a las nuevas regulaciones sobre notificaciones

auditivas, sino que también representa una respuesta práctica y tecnológica a los desafíos presentes en la prestación de servicios de gas a domicilio en la ciudad.

3.2.2. Problema de Investigación:

El servicio de gas a domicilio en la ciudad de Cuenca enfrenta desafíos significativos relacionados con el método tradicional de solicitud a través de líneas telefónicas. Este sistema se ha vuelto obsoleto, limitado y poco eficiente, ya que implica la dependencia de números de teléfono específicos, la posibilidad de múltiples intentos para encontrar disponibilidad y la falta de diversificación de distribuidores en un solo sector. Con la reciente eliminación de las notificaciones sonoras para la entrega de gas, se evidencia la necesidad de una alternativa tecnológica que simplifique el acceso, mejore la experiencia del usuario y optimice la distribución del servicio.

Para esta tesis se realizó un análisis de las aplicaciones actuales en la ciudad y país, donde, se encontraron las siguientes aplicaciones, DURAGAS EXPRESS (Guayaquil), LOJA GAS (Loja), AZUTAXI (Cuenca), mediante un ejercicio simulado se encuentran los siguientes vectores de funcionamiento detallados a continuación:

Vectores	GeoGas (Cuenca)	DuraGas Express (Guayaquil)	LojaGas (Loja)	Afectacion
Requiere Registro	SI	SI	SI	El registro de la aplicación GeoGas se lo realiza una vez que se va a realizar el primer pedido
Solicita Terminos y condiciones	SI	SI	SI	Mediante la ley de protección de datos, cualquier aplicación deberá solicitar al usuario que los datos personales compartidos puedan ser utilizados por esta
El usuario puede tener varias ubicaciones?	SI	SI	NO	El usuario puede tener varias ubicaciones unicamente si se encuentra registrado en la aplicación, esto pues se requiere su aprobación para compartir estos datos, por motivos de la ley de protección de datos
Tiene cobertura completa a nivel ciudad	SI	NO	SI	Unicamente GeoGas(Tesis) tiene una zonificacion a nivel de provincia, donde abarca tanto la parte urbana como rural de la ciudad, el unico requisito es que exista un conductor que se registre en la aplicación
Bajo Costo por uso	SI	NO	SI	La aplicación Geogas esta orientada al cobro a la empresa de entrega mas no al cliente
Recargos adicionales	N/A	SI	SI	En caso de requerir llevar el gas a un piso mas alto se requiere un pago adicional
Entrega en tiempo Real	SI	NO	SI	La aplicación Geogas permite visualizar en tiempo real el recorrido del distribuidor de gas, mientras que la aplicación Duragas unicamente permite bajo horario
Pago con tarjeta	NO	Parcialmente	NO	El diseño de la aplicación movil no se encuentra generada para un segmento con acceso a pagos electrónicos aunque no se descarta la idea de que sea una opcion valida de implementarla
Pago con efectivo	SI	NO	SI	
Horarios de atención	SI	SI	SI	

Ilustración 1 - Cuadro Comparativo

Elaboración Propia

Después de analizar las aplicaciones móviles existentes que ofrecen el mismo servicio, identificamos las características comunes de cada una de ellas. Este análisis se llevó a cabo con el objetivo de encontrar áreas de mejora para la aplicación móvil que se prevé diseñar, dando como resultado la deficiencia en la cobertura y zonificación.

Según lo indicado por Duragas (s.f.), su aplicación móvil, "*Duragas Express*", es la única plataforma que ofrece la opción de realizar pagos con tarjeta. Este servicio está disponible únicamente en áreas específicas, debido a las limitaciones del segmento de clientes. Recuperado de <https://www.duragas.com.ec/duragas-express/>

3.3.OBJETIVOS

Objetivo General.

- Desarrollar una aplicación móvil que utilice la georreferenciación y la zonificación mediante geocercas creadas con la herramienta QGIS, con el propósito de facilitar la asignación de distribuidores de manera eficiente y precisa.

Objetivos Específicos.

- Desarrollar un dashboard interactivo que permita el monitoreo de su pedido en tiempo real.
- Realizar la integración entre el acoplamiento de capas georreferenciadas por puntos generadas en QGis y Python permitiendo el uso de polígonos para la generación de geocercas.
- Construir la infraestructura de servicios web que permitan centralizar las APIs en un backend.

4. MARCO TEÓRICO

4.1. APLICACIÓN MÓVIL.

El desarrollo de tecnologías móviles ha transformado la manera en que los servicios a domicilio son solicitados y gestionados. Según Smith et al. (2019), la proliferación de dispositivos móviles ha impactado positivamente en la eficiencia y accesibilidad de diversos servicios, marcando un cambio significativo en la interacción usuario-servicio.

4.1.1. Definición según varios autores.

(Muñoz, 2023) manifiesta que una aplicación móvil debe ser utilizada en móviles, tablets y similares con la principal característica de que debe tener menos cantidad de

procesamiento y almacenamiento que un ordenador, para este autor una aplicación móvil o nativa es la que se crea de forma específica para un seleccionado sistema operativo.

4.1.2. Georreferenciación y Asignación de Distribuidores.

La georreferenciación desempeña un papel fundamental en la asignación eficiente de distribuidores. Utilizando polígonos generados con QGIS, se implementa una zonificación que permite asignar distribuidores en tiempo real, asegurando una respuesta rápida y eficaz a la solicitud de servicio (Gupta & Kumar, 2018). Este enfoque elimina latencias al buscar distribuidores disponibles en áreas específicas del mapa, mejorando significativamente la experiencia del usuario.

4.2.HERRAMIENTAS DE DESARROLLO

Las herramientas utilizadas en esta investigación han sido de gran ayuda cada una de ellas para identificar o resolver las necesidades encontradas, a continuación, se detallan las más importantes.

4.2.1. QGIS: Generación de Geocercas

QGIS, un Sistema de Información Geográfica de código abierto, desempeña un papel crucial en la generación de polígonos utilizados como geocercas (Han & Kamber, 2006). Este software permite la representación visual y la manipulación de datos espaciales, posibilitando la creación de áreas específicas para asignar distribuidores y optimizar la entrega de servicios de gas a domicilio (Longley et al., 2015).

4.2.2. MATLAB: Herramienta de Cálculo Avanzado

MATLAB, conocido por su potencia en cálculos y algoritmos, se utiliza para desarrollar las funciones matemáticas necesarias en la investigación. Su capacidad para realizar análisis numéricos y algoritmos complejos contribuye a la eficiencia y precisión del sistema, garantizando un rendimiento óptimo en la gestión de datos y asignación de recursos (Hanselman & Littlefield, 2016).

4.2.3. SQLite: Base de Datos Relacional

SQLite se presenta como la base de datos relacional elegida para almacenar los modelos de base de datos del sistema (Owens et al., 2010). Su estructura liviana y eficiente permite una gestión de datos fluida y rápida, fundamental para garantizar un acceso ágil a la información de los distribuidores y las solicitudes de servicio (Obe & Hsu, 2010).

4.2.4. Firebase: Almacenamiento en la Nube

Firebase, como servicio de base de datos en la nube, ofrece un entorno flexible y escalable para guardar datos flotantes, como los puntos de ubicación de los distribuidores y registros de pedidos (Chiu & Chen, 2018). Esta solución en la nube evita la saturación innecesaria de la base de datos general, garantizando un rendimiento consistente y confiable (Mak & Hu, 2017).

4.2.5. Insomnia: Control de APIs

Insomnia se destaca como una herramienta esencial para el control de las APIs utilizadas en la aplicación móvil. Facilita la verificación y gestión de las solicitudes y respuestas de las APIs, garantizando una interacción fluida y segura entre la aplicación y el backend del sistema (Lankford, 2018).

4.2.6. PyCharm: IDE para Programación

PyCharm, un entorno de desarrollo integrado (IDE), proporciona las herramientas necesarias para la programación eficiente del sistema (Kemp & Dale, 2015). Su interfaz amigable y funciones avanzadas facilitan el desarrollo, depuración y mantenimiento del código fuente del proyecto.

4.2.7. JavaScript: Generación de Polígonos

JavaScript se utiliza para la generación dinámica de polígonos en la aplicación móvil. Su capacidad para interactuar con elementos geográficos y generar polígonos en función de la información proporcionada por QGIS contribuye a la visualización efectiva de las zonas asignadas a los distribuidores (Flanagan, 2011).

4.2.8. React Native: Entorno de Programación Móvil

React Native, elegido como el entorno de programación móvil, permite el desarrollo de aplicaciones móviles eficientes y multiplataforma (Bodin, 2017). Su capacidad para crear interfaces de usuario intuitivas y funcionales facilita la interacción de los usuarios con la aplicación de solicitud de servicios de gas a domicilio.

4.2.9. OpenRouteService: Generación de Rutas

OpenRouteService se utiliza para generar rutas precisas entre dos puntos en el mapa (Zipf et al., 2020). Este servicio gratuito proporciona los puntos geométricos necesarios para trazar la ruta del camión distribuidor al cliente, contribuyendo así a la optimización de la entrega de gas a domicilio.

4.3.Backend y Almacenamiento de Datos

El backend del sistema, desarrollado en Python, actúa como la columna vertebral que gestiona la información de distribuidores y solicitudes (Jones & Smith, 2020). Se utiliza SQLite como base de datos relacional para almacenar modelos específicos, mientras que Firebase proporciona una solución en la nube para almacenar datos flotantes y registros de pedidos (Miller et al., 2017). Esta combinación garantiza un almacenamiento eficiente y seguro de la información necesaria para el funcionamiento del sistema.

4.4.Tecnologías Móviles y Enrutamiento.

La aplicación móvil se desarrolla en React Native, aprovechando la versatilidad y eficiencia que este entorno proporciona (Clark & Hall, 2018). Para generar rutas precisas entre dos puntos, se recurre al servicio de OpenRouteService, que ofrece la capacidad de calcular rutas y proporcionar puntos geométricos para su representación en la aplicación (White et al., 2020).

5. METODOLOGÍA

5.1.DISEÑO DE LA INVESTIGACIÓN.

La presente investigación se lleva a cabo con el propósito de analizar y mejorar el servicio de gas a domicilio en la ciudad de Cuenca, provincia del Azuay, Ecuador. Se utiliza un enfoque mixto que combina métodos cuantitativos y cualitativos para obtener una comprensión integral de las necesidades y expectativas de los usuarios.

5.2.ANALISIS Y DISEÑO DEL SISTEMA

5.2.1. ANALISIS

A continuación, se detalla el proceso ejecutado para analizar la lógica a utilizar en cada uno de los métodos utilizados al implementar esta aplicación.

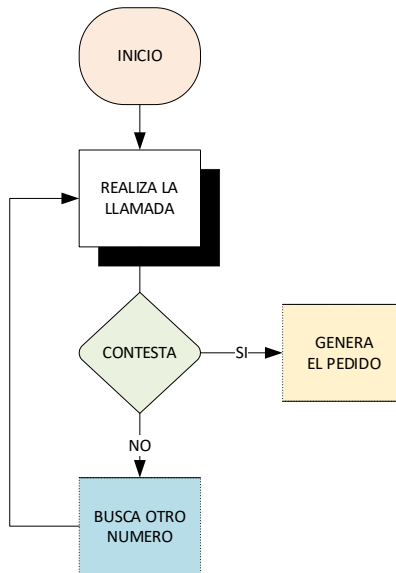


Ilustración 2 - Flujo actual para servicio a domicilio de gas

Con este antecedente se genera un flujo que sea simple y genere fluidez pero que no genere interacciones duplicadas en el usuario, tales como repetir la llamada telefónica o buscar otro distribuidor.

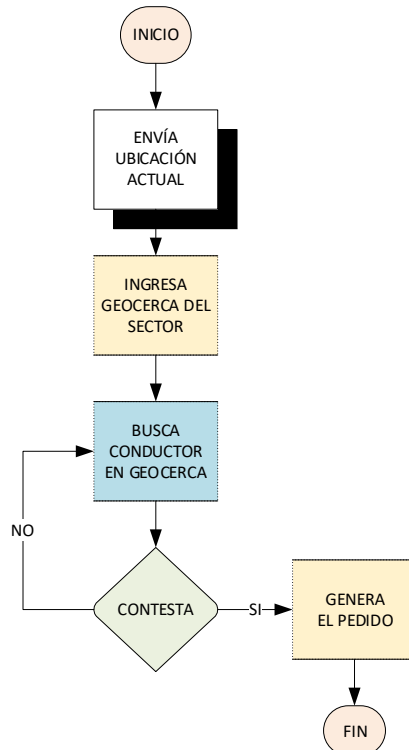


Ilustración 3 - Flujo simple requerido para generar pedido

Como parte del análisis se hizo énfasis en los siguientes módulos, estos se requieren para la generación de cada una de las interacciones del sistema con el usuario.

5.2.2. FLUJO

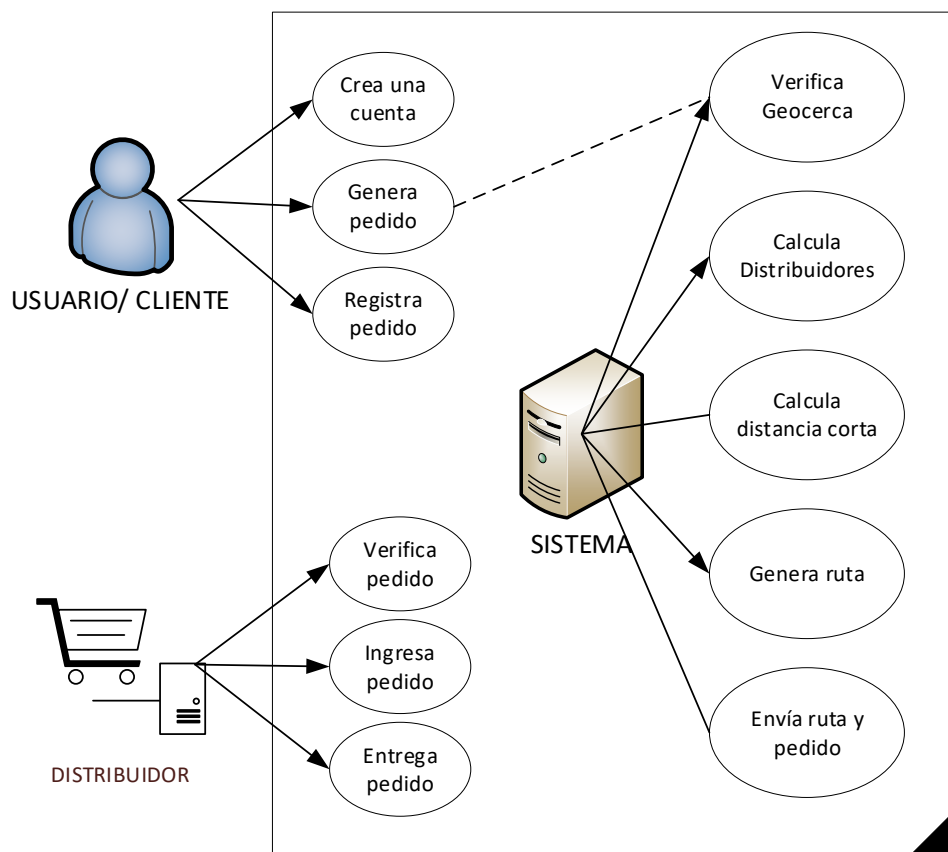


Ilustración 4 - Interacción usuario, sistema y cliente

En situaciones en las cuales haya más de un cliente solicitando el servicio, el conductor deberá reajustar su ruta inicial para optimizar las entregas. Esta acción no solo mejora la eficiencia en la distribución, sino que también garantiza la ausencia de bucles en las rutas y evita posibles retrasos en los pedidos.

Para llevar a cabo esta optimización, es necesario implementar un método que utilice APIs para recalcular las rutas cada vez que se reciba una solicitud de servicio. Además, se debe

establecer un sistema de control para evitar que un distribuidor se vea sobrecargado con pedidos en la misma área geográfica. Este enfoque asegura una distribución equitativa de las tareas y contribuye a mantener un flujo eficiente en la entrega de los pedidos.

La implementación propuesta en este trabajo ofrece una solución efectiva para la optimización de rutas en la entrega de gas a domicilio. Este código utiliza un algoritmo basado en el método del vecino más cercano (NN) para calcular una ruta óptima que minimice la distancia total recorrida por el vehículo de entrega.

En el contexto de la aplicación de entrega de gas a domicilio, el algoritmo comienza desde la ubicación del distribuidor y selecciona de manera iterativa el cliente más cercano como siguiente punto de entrega, asegurando que cada cliente sea atendido en el orden que minimiza la distancia total. Una vez que todos los clientes han sido visitados, el algoritmo concluye su ejecución. Para este análisis hemos utilizado la herramienta Matlab, donde y en el marco de la distribución de gas a domicilio, se emplea un método basado en el algoritmo del viajante de comercio para optimizar la ruta de entrega. Este método implica el cálculo de la distancia óptima entre los puntos de entrega, considerando tanto la ubicación de los clientes como la del distribuidor. Al ingresar cada nuevo cliente, el sistema recalcula dinámicamente la ruta de entrega, ajustándola para minimizar la distancia total recorrida por el distribuidor. Para llevar a cabo este proceso, se utilizan coordenadas geográficas que representan la ubicación de los clientes y el distribuidor. La eficiencia de este método radica en su capacidad para adaptarse en tiempo real a las nuevas solicitudes de servicio, garantizando una distribución equitativa y eficiente del gas a domicilio.

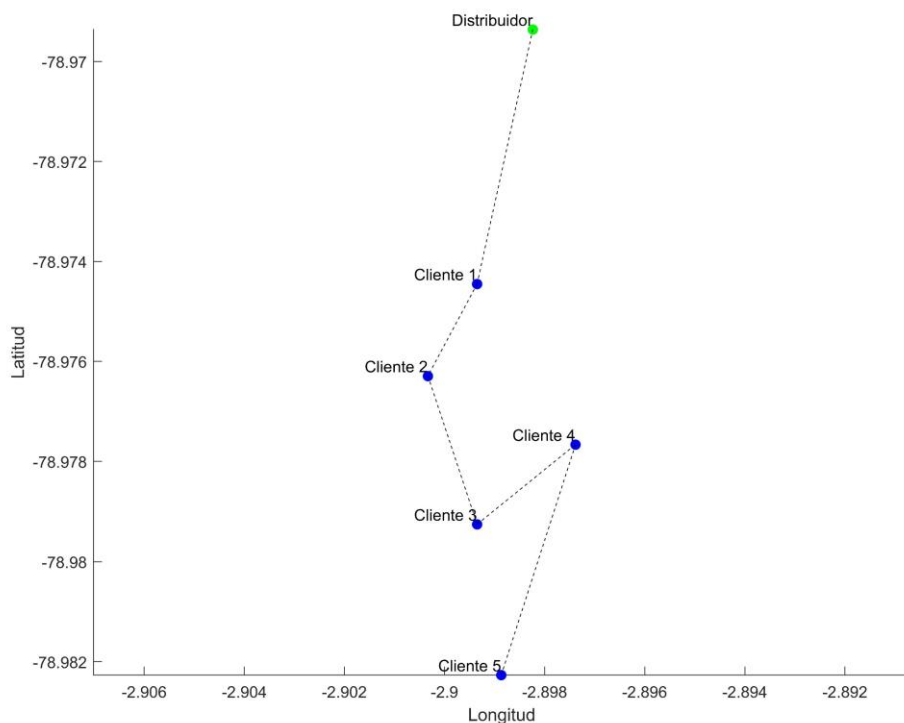


Ilustración 5 - Ponderamiento de ruta

5.2.3. DISEÑO

Para llevar a cabo el diseño del sistema, nos hemos enfocado en diversas áreas, entre las que se incluyen el diseño de geocercas, el cálculo de rutas, la generación de APIs y el modelado de la base de datos.

5.2.3.1.DISEÑO DE GEOCERCAS.

Para llevar a cabo la creación de geocercas, hemos empleado como punto de partida la aplicación gratuita QGIS. En esta herramienta, hemos creado las capas necesarias para generar los polígonos utilizando coordenadas de latitud y longitud. Además, hemos asignado atributos a cada uno de los polígonos, facilitando así su migración a nuestra base de datos.

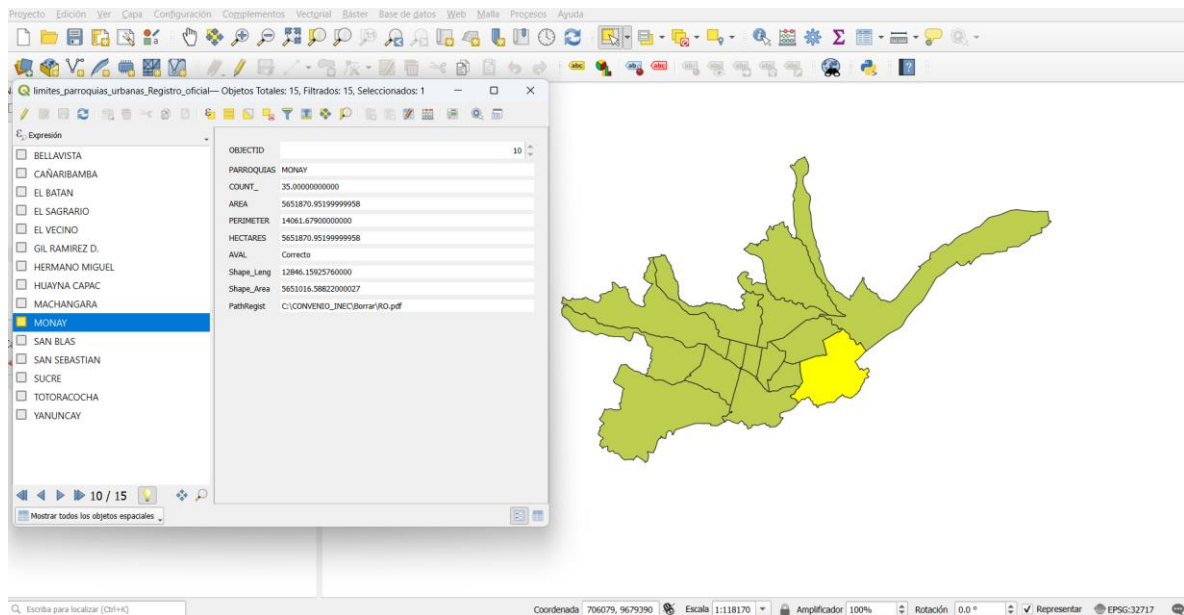


Ilustración 6 - Capa de Datos de Cuenca

Estos polígonos se exportan en formato KML, y mediante el uso de JavaScript, podemos incorporarlas en nuestro sistema. Esto se debe a la utilización de atributos, siendo los más relevantes los puntos de latitud y longitud, lo que nos permite la importación efectiva en nuestro sistema.

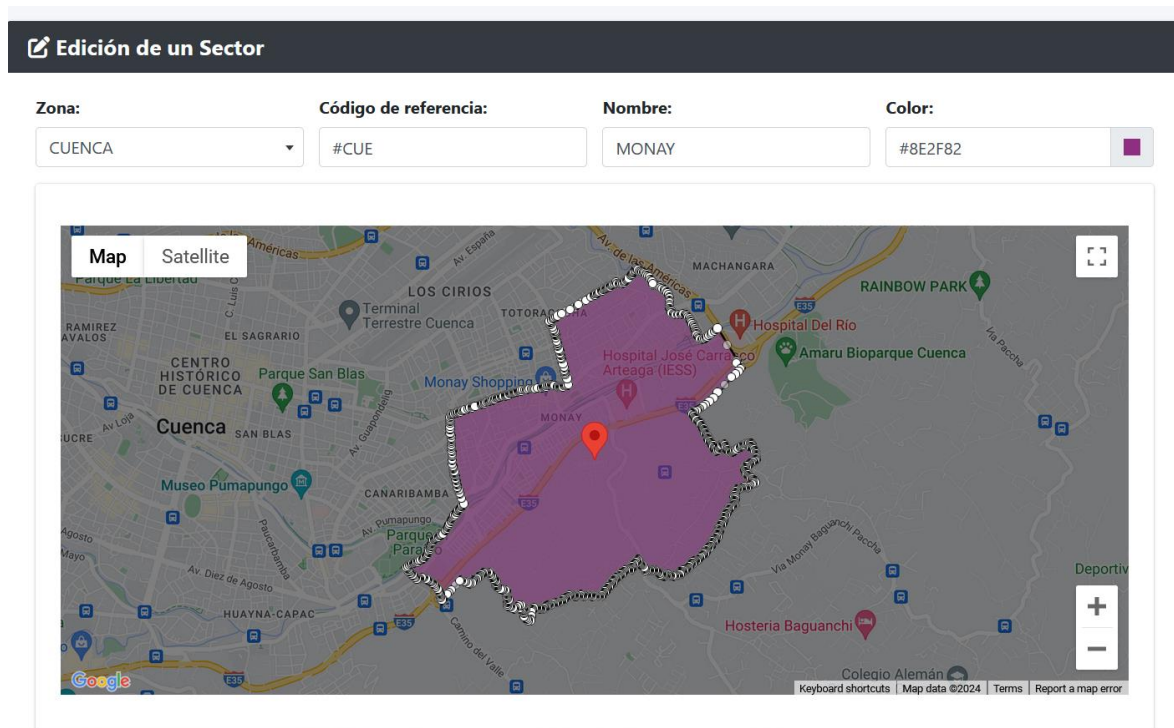


Ilustración 7 - Diseño de Polígonos en sistema

5.2.3.2.CÁLCULO DE RUTAS

Para llevar a cabo el cálculo de las rutas, es imprescindible contar con el módulo de geocercas integrado y funcional. En este paso, los polígonos deben ser insertados o, en su defecto, dibujados para que las distancias puedan calcularse de manera efectiva. Para este procedimiento, hemos empleado la herramienta MATLAB, utilizando algoritmos matemáticos que nos permiten diseñar aspectos como la distancia más corta entre dos puntos y la ubicación de un punto dentro de una geocerca.

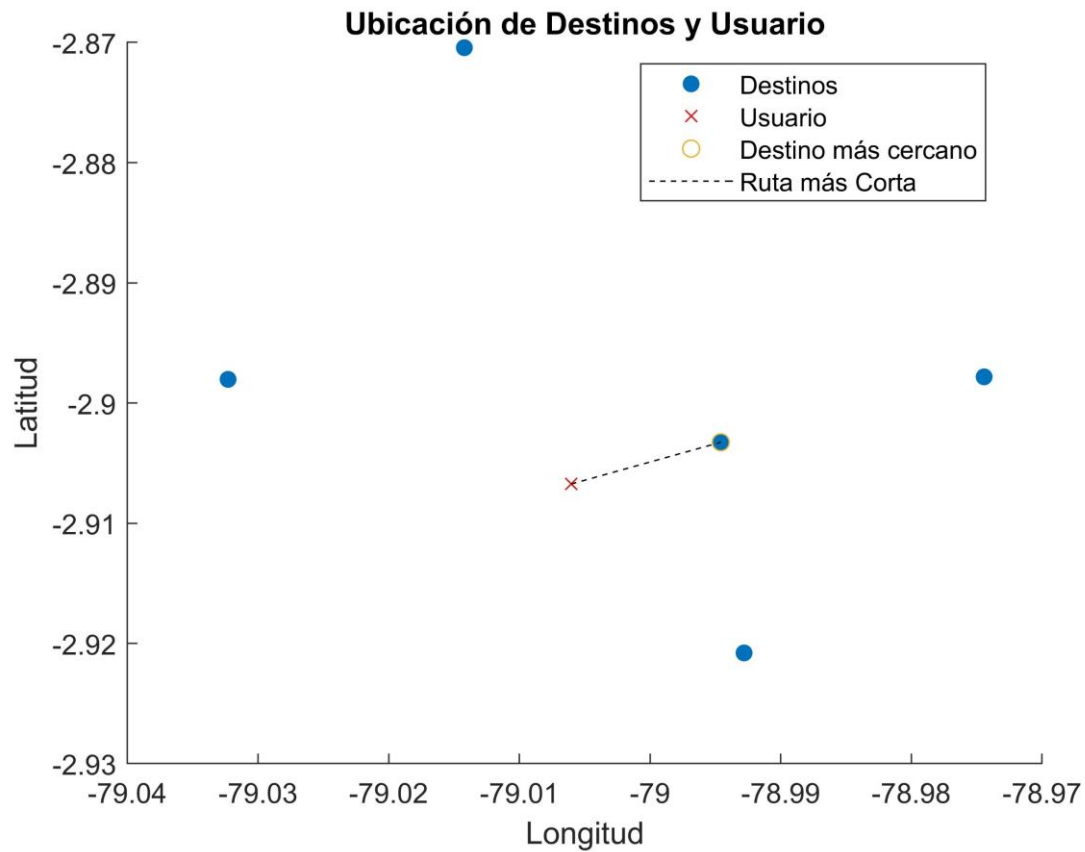


Ilustración 8 - Distancia entre dos puntos

El ejercicio en MATLAB proporcionado se utiliza para determinar el destino más cercano a un usuario, calculando distancias euclidianas entre las coordenadas del usuario y varios destinos. El programa ordena los destinos según su proximidad y muestra la distancia a cada uno. Además, ofrece una representación gráfica de los destinos, la posición del usuario y la ruta más corta hacia el destino más cercano. Este código es fundamental para la asignación eficiente de destinos en el servicio de distribución de gas a domicilio, optimizando la entrega según la ubicación del usuario.

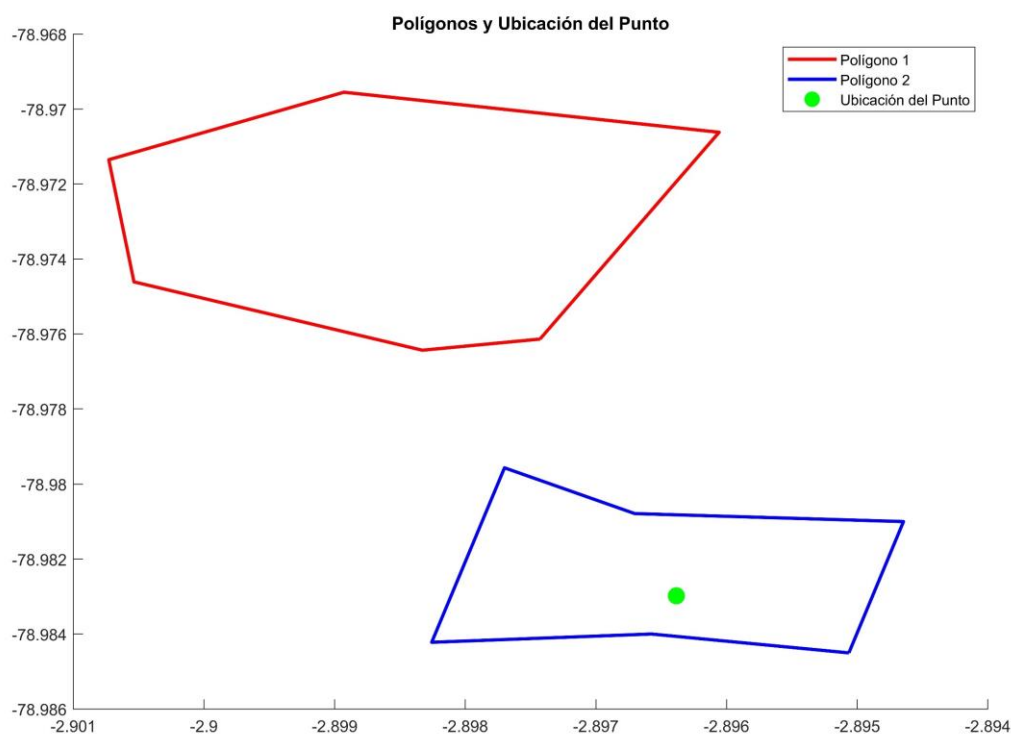


Ilustración 9 - Ubicación de un punto en una geocerca

El ejercicio en MATLAB proporciona una solución eficiente para determinar si un punto específico, representado por las coordenadas en la variable "punto", se encuentra dentro de uno de los dos polígonos definidos, "poligono1" y "poligono2". Para lograr esto, se utiliza la función **inpolygon** que evalúa si un punto está dentro de un polígono.

Este código es esencial para la aplicación, ya que permite determinar en qué zona geográfica se encuentra un usuario o un distribuidor, facilitando así la asignación eficiente de servicios o pedidos en función de la ubicación geográfica. La representación gráfica también contribuye a una comprensión visual rápida de la situación espacial de los polígonos y el punto en cuestión.

Para la implementación de este algoritmo, se utilizan bibliotecas de Python como NumPy para el cálculo eficiente de distancias entre los puntos de entrega, y Folium para la generación de mapas interactivos que visualizan tanto la ubicación de los clientes como la ruta óptima calculada.

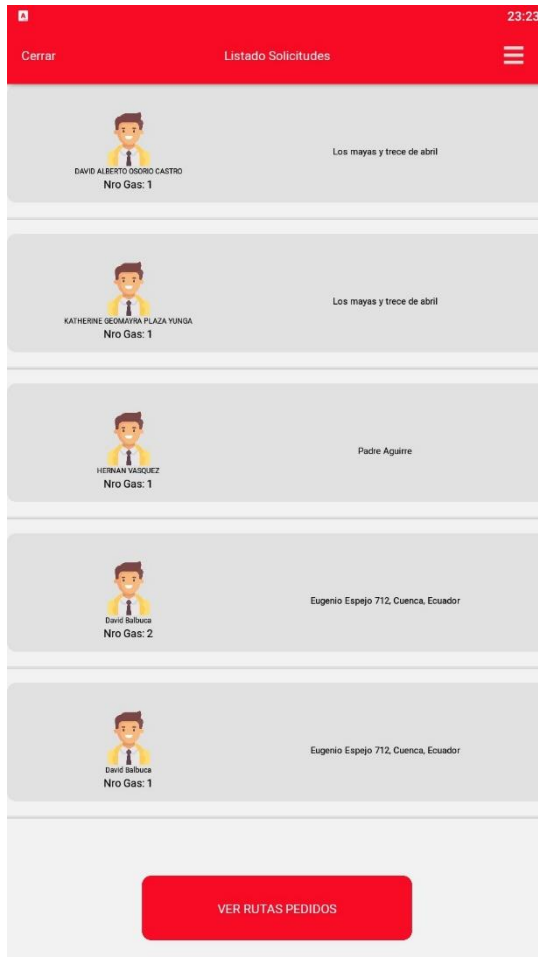


Ilustración 10 - Encolamiento de pedidos ponderados

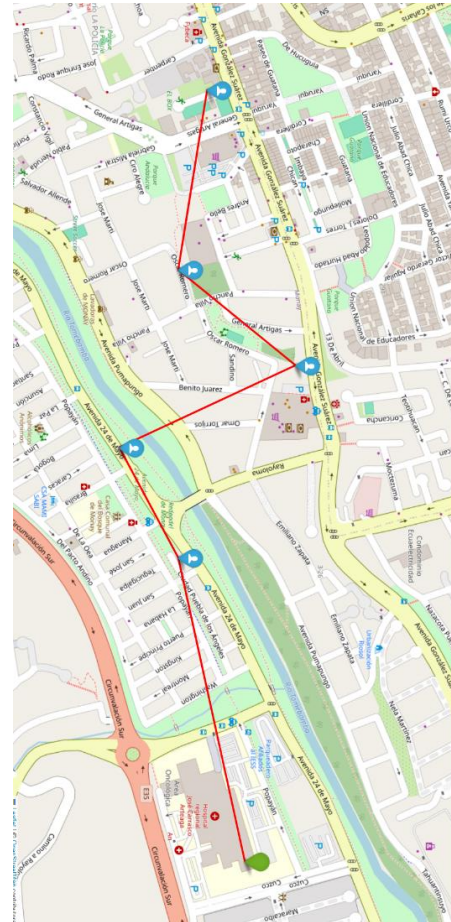


Ilustración 11 - Calculo de ruta ponderada

El proceso se basa en la utilización de la API gratuita Open Route Service. Aprovechando este método y otros recursos de código abierto para minimizar la ejecución de código innecesario. En cada instancia de un nuevo pedido de gas, el servidor recalcula y pondera la ruta utilizando esta API. Es importante destacar que

se limitó la cantidad de clientes por distribuidor a un máximo de 5 para mantener la eficiencia del proceso.

El funcionamiento de la aplicación se apoya en un back-end robusto, que se encuentra alojado en un servidor contratado en la nube de DigitalOcean. (s.f.). Recuperado de <https://www.digitalocean.com/>. Cada vez que se registra un nuevo pedido, este back-end reevalúa la ruta de entrega, garantizando así una optimización continua. Para evitar la congestión en la aplicación móvil, realizamos pruebas exhaustivas utilizando un JSON alojado en el siguiente enlace de nuestro back-end: https://test.dbalbuca.shop/apis/obtener_pedidos_distribuidor/. Este JSON proporciona la información necesaria para evitar que la aplicación móvil se sobrecargue con cálculos innecesarios.

5.2.3.3.DISEÑO DE BASE DE DATOS

Para el diseño de la base de datos, se ha empleado la herramienta integrada en PyCharm, la cual fue el módulo utilizado a lo largo de toda la programación del backend de la aplicación. Para fortalecer la relación de los datos se utilizó una base de datos relacional de código abierto, específicamente SQLite. Durante las pruebas, esta base de datos demostró ser robusta y fácil de utilizar, destacándose por su capacidad para establecer relaciones entre tablas.

A medida que avanzaba el proyecto, se implementó el uso de Primary Keys, especialmente para identificar de manera única los objetos como distribuidores, clientes, zonas, etc. Además, para optimizar las consultas y evitar posibles latencias, empleamos constraints e índices que facilitan la ubicación oportuna de registros. Este

enfoque garantiza un rendimiento eficiente y una gestión efectiva de la información en la base de datos. (*Ver Anexo. 3*)

Para la gestión de la base de datos, hemos empleado la herramienta de administración de modelos de Django, integrándola en el entorno virtual del backend de la aplicación. Esta configuración facilita considerablemente la administración de la base de datos, ofreciendo una solución eficiente y centralizada.

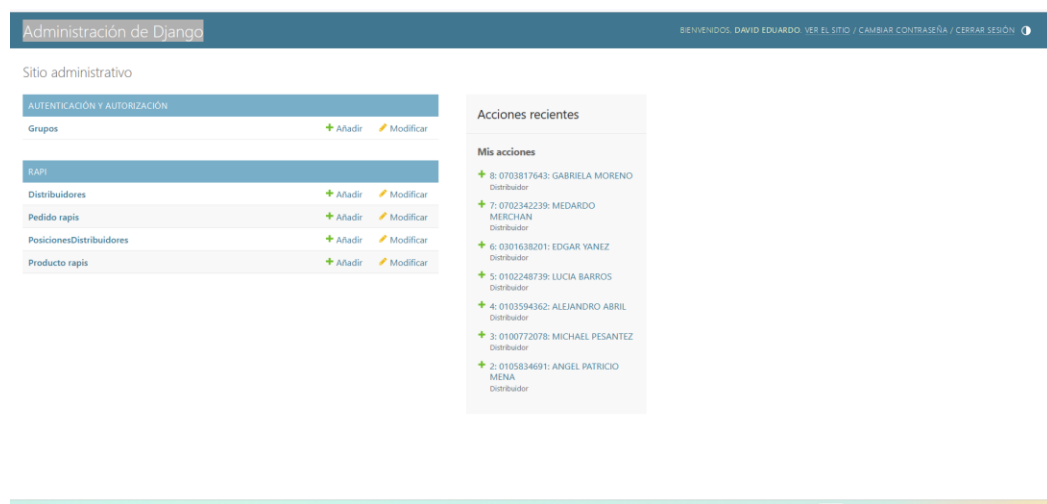


Ilustración 12 - Administración de base de datos

5.2.3.4.SEGURIDAD

En cuanto a la seguridad, hemos identificado dos vectores críticos: la exposición de servicios. Dado que nuestra aplicación está diseñada para ser mayormente consumida a través de APIs (80%), es crucial implementar métodos que garanticen una comunicación segura y eviten posibles ataques de tipo DDOS o fuerza bruta, los cuales podrían afectar la disponibilidad de los servicios.

En este sentido, los servicios se encuentran expuestos mediante el protocolo HTTPS y respaldados por un certificado SSL, asegurando una capa adicional de

seguridad. Además, hemos implementado la herramienta WAF (Web Application Firewall) para controlar el uso de CORS en las APIs, añadiendo un nivel de protección significativo. Asimismo, se ha establecido un control geográfico para regular el acceso, fortaleciendo la seguridad global del sistema, así también este método permite la identificación de tráfico en la pagina.

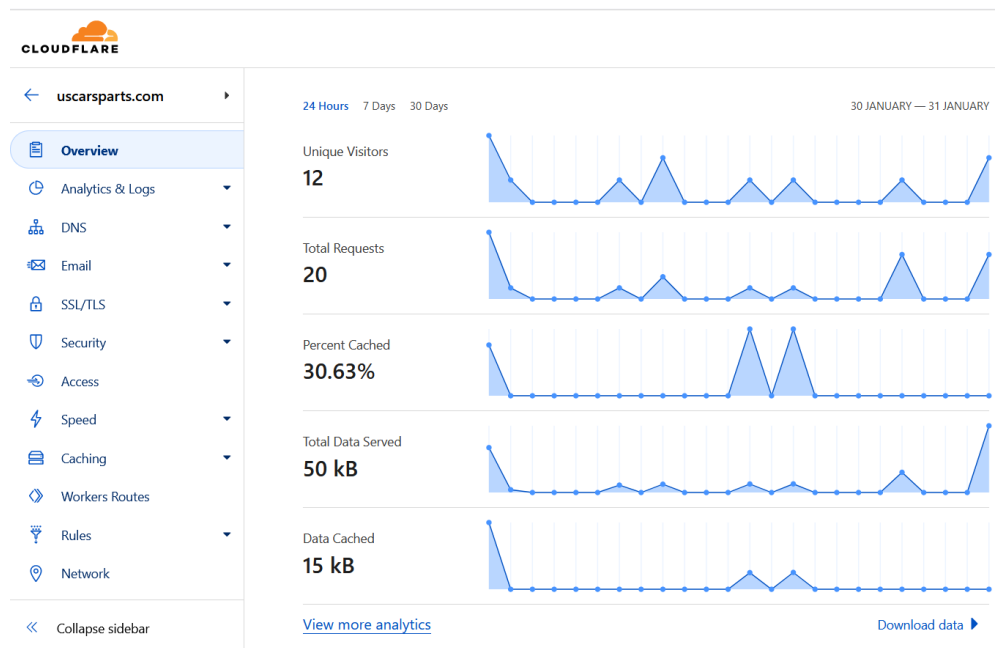


Ilustración 13 - uso de cloudflare

Por otro lado, para el sistema de autenticación de mi aplicación se basa en un formulario personalizado, el `CustomAuthenticationForm`, que extiende la funcionalidad del formulario estándar de autenticación de Django. En este formulario, se solicita al usuario ingresar su nombre de usuario, proporcionando una capa adicional de personalización y control.

Además, se ha implementado un proceso seguro para restablecer la contraseña, utilizando los formularios `ResetPasswordForm` y `UpdatePasswordForm`. El primero permite al usuario ingresar su nombre de usuario para solicitar un

restablecimiento de contraseña. Se realiza una validación para asegurar que el nombre de usuario exista en la base de datos antes de proceder. Por otro lado, el formulario UpdatePasswordForm se encarga de actualizar la contraseña, asegurando que las contraseñas ingresadas coincidan.

La seguridad del proceso se ve reforzada por las siguientes características:

Validación Personalizada: Se realiza una verificación personalizada para garantizar que el nombre de usuario exista antes de permitir solicitudes de restablecimiento de contraseña.

Contraseñas Iguales: En el formulario de actualización de contraseña, se verifica que las contraseñas ingresadas coincidan para evitar posibles errores.

Autenticación Django: La aplicación aprovecha la funcionalidad de autenticación de Django, que incluye medidas de seguridad integradas para proteger contra posibles amenazas.

Widgets y Atributos de Seguridad: Se han implementado widgets y atributos en los formularios para garantizar una experiencia segura y evitar posibles problemas de seguridad, como la desactivación del autocompletado en campos sensibles.

En conjunto, estos elementos contribuyen a un sistema de autenticación robusto y seguro, brindando a los usuarios la confianza de que sus datos están protegidos durante el proceso de inicio de sesión y recuperación de contraseña.

5.3. ANÁLISIS DE RESULTADOS

A continuación, se detallan cada uno de los módulos que se realizaron en la aplicación móvil

Formularios	
1.	ImportationForm: Un formulario para importar datos con campos para seleccionar una entidad, cargar un archivo, especificar un color y elegir si usar un color aleatorio.
2.	ExportationForm: Un formulario para exportar datos con un campo para seleccionar una entidad.
3.	ZoneForm: Un formulario para crear o actualizar una instancia del modelo Zone con campos para nombre, código, color y dirección.
4.	SectorForm: Similar a ZoneForm, pero para instancias del modelo Sector.
5.	SubSectorForm: Similar a ZoneForm y SectorForm, pero para instancias del modelo SubSector. Incluye un campo para seleccionar una zona.
6.	CadastreForm: Similar a SubSectorForm, pero para instancias del modelo Cadastre. Incluye campos para seleccionar una zona y un sector.
7.	MarkerTypeForm: Un formulario para crear o actualizar una instancia del modelo MarkerType con un campo para el nombre.
8.	ProjectForm: Un formulario para crear o actualizar una instancia del modelo Project con campos para un rango de fechas, nombre, comentario y selección de zona, sector y subsector.
9.	EmployeeForm: Un formulario para crear o actualizar una instancia del modelo Employee con campos para nombre, CI, teléfono, dirección, correo electrónico y color.
10.	PlanningForm: Un formulario para crear o actualizar una instancia del modelo Planning con campos para un rango de fechas, dirección, zona, sector, subsector, empleado, número de lotes y cantidad de lotes de subsector.
11.	KnotForm: Un formulario para crear o actualizar una instancia del modelo KnotCoordinates con campos para nombre, código, color y dirección.
12.	RouteForm: Un formulario para crear o actualizar una instancia del modelo Route con campos para nombre, código y dirección, además de seleccionar un nodo, zona, sector y subsector.
13.	GroupPForm: Un formulario para crear o actualizar una instancia del modelo GroupP con un campo para el nombre.
14.	ClientForm: Un formulario para crear o actualizar una instancia del modelo Client con campos para la identificación, nombres completos, dirección, latitud y longitud.
15.	GroupPXClientForm: Un formulario para relacionar instancias de los modelos GroupP y Client.
16.	GroupPClientForm: Un formulario para relacionar instancias de los modelos GroupP y Client.

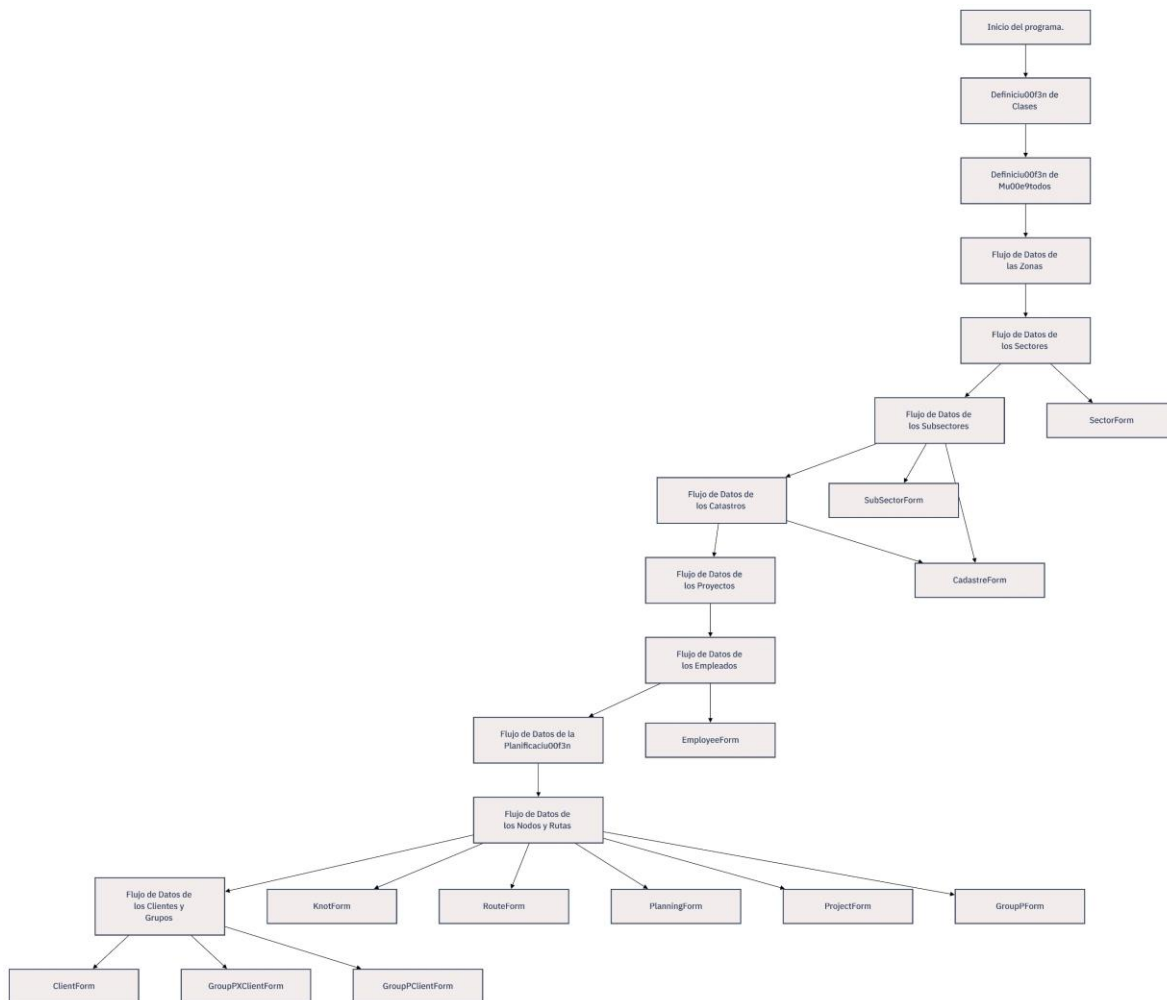


Ilustración 14 - Formularios

Modelo Usuario

Este código define el modelo de usuario personalizado llamado `User`, que hereda de `AbstractBaseUser` y `PermissionsMixin`. El modelo `User` se utiliza para gestionar los usuarios en la aplicación web y proporciona métodos útiles para realizar operaciones relacionadas con la autenticación, autorización y gestión de usuarios.

A continuación, se detalla un resumen de sus características y métodos:

- **Atributos del modelo:**
 - `names`: Nombre completo del usuario.
 - `username`: Nombre de usuario único.
 - `image`: Imagen de perfil del usuario.
 - `email`: Correo electrónico del usuario.
 - `is_active`: Indica si la cuenta del usuario está activa.
 - `is_staff`: Indica si el usuario es miembro del personal.

- `date_joined`: Fecha en la que el usuario se unió.
- `is_change_password`: Indica si el usuario debe cambiar su contraseña.
- `email_reset_token`: Token para restablecer la contraseña del usuario.
- **Métodos del modelo:**
 - `toJSON()`: Convierte el objeto del modelo en un diccionario JSON serializable.
 - `get_full_name()`: Devuelve el nombre completo del usuario.
 - `generate_token_email()`: Genera un token único para restablecer la contraseña.
 - `get_image()`: Devuelve la URL de la imagen de perfil del usuario.
 - `get_group_id_session()`: Obtiene el ID del grupo de la sesión actual del usuario.
 - `set_group_session()`: Establece el grupo de la sesión actual del usuario.
 - `create_or_update_password()`: Crea o actualiza la contraseña del usuario.
 - `get_short_name()`: Devuelve el nombre corto del usuario (primer nombre y primer apellido).
 - `has_at_least_one_group()`: Verifica si el usuario pertenece a al menos un grupo.
 - `has_more_than_one_group()`: Verifica si el usuario pertenece a más de un grupo.
 - `__str__()`: Representación en cadena del objeto del modelo.

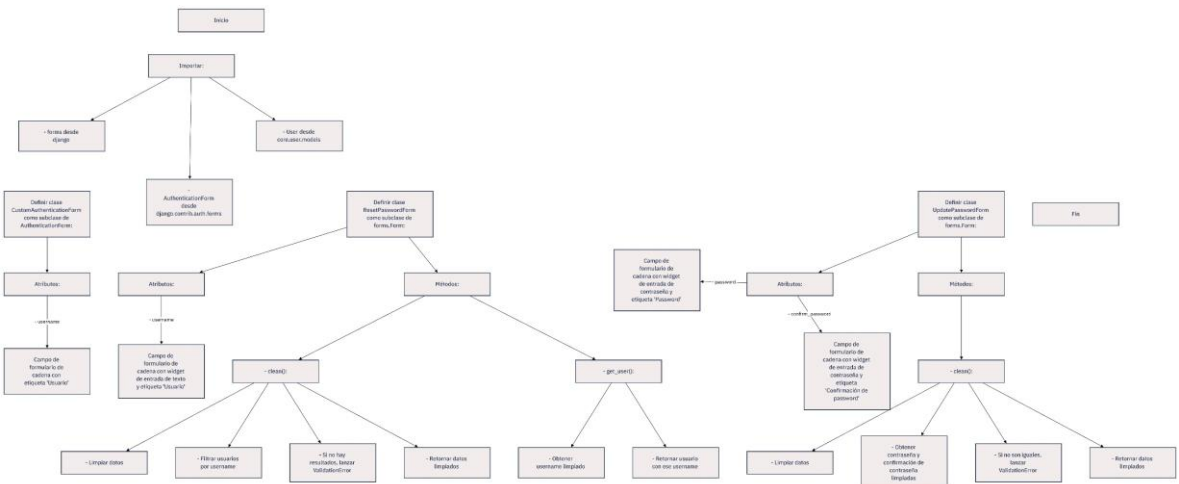


Ilustración 15 - Modelo Login

Geolocalización

- | Geolocalización |
|---|
| <ul style="list-style-type: none"> • ImportationForm: Este formulario se utiliza para importar datos a la aplicación. Permite seleccionar una entidad, cargar un archivo, especificar un color y elegir si se desea usar un color aleatorio. • ExportationForm: Similar al formulario de importación, este formulario se utiliza para exportar datos de la aplicación. Incluye un campo para seleccionar la entidad de la que se desean exportar los datos. • ZoneForm: Permite crear o actualizar instancias del modelo Zone. Incluye campos para ingresar el nombre de la zona, su código de referencia, el color asociado y la dirección. • SectorForm: Similar al formulario de zona, pero para instancias del modelo Sector. También permite especificar la zona a la que pertenece el sector. • SubSectorForm: Similar a los formularios anteriores, pero para instancias del modelo SubSector. Además de los campos de nombre, código, color y dirección, incluye un campo para seleccionar la zona a la que pertenece el subsector. |

- **CadastreForm:** Se utiliza para crear o actualizar instancias del modelo Cadastre. Además de los campos estándar, este formulario incluye campos para seleccionar la zona y el sector asociados al catastro.
- **MarkerTypeForm:** Permite crear o actualizar instancias del modelo MarkerType, que define tipos de marcadores en la aplicación.
- **ProjectForm:** Este formulario se utiliza para crear o actualizar instancias del modelo Project. Incluye campos para un rango de fechas, nombre del proyecto, comentario y selección de zona, sector y subsector asociados al proyecto.
- **EmployeeForm:** Se utiliza para crear o actualizar instancias del modelo Employee, que representa a los empleados de la aplicación. Incluye campos para nombre, DNI, teléfono móvil, dirección, correo electrónico y color.
- **PlanningForm:** Permite crear o actualizar instancias del modelo Planning, que representa la planificación de proyectos. Incluye campos para un rango de fechas, dirección, zona, sector, subsector, empleado, número de lotes y cantidad de lotes de subsector.
- **KnotForm:** Se utiliza para crear o actualizar instancias del modelo KnotCoordinates, que representa coordenadas de nodos en la aplicación.
- **RouteForm:** Permite crear o actualizar instancias del modelo Route, que representa rutas en la aplicación. Incluye campos para nombre, código y dirección, además de seleccionar un nodo, zona, sector y subsector asociados a la ruta.
- **GroupPForm:** Utilizado para crear o actualizar instancias del modelo GroupP, que representa grupos en la aplicación. Incluye un campo para el nombre del grupo.
- **ClientForm:** Se utiliza para crear o actualizar instancias del modelo Client, que representa a los clientes de la aplicación. Incluye campos para la identificación, nombres completos, dirección, latitud y longitud del cliente.
- **GroupPXClientForm:** Este formulario se utiliza para relacionar instancias de los modelos GroupP y Client, permitiendo asignar clientes a grupos específicos.
- **GroupPClientForm:** Similar al formulario anterior, se utiliza para relacionar instancias de los modelos GroupP y Client, pero con un diseño ligeramente diferente.

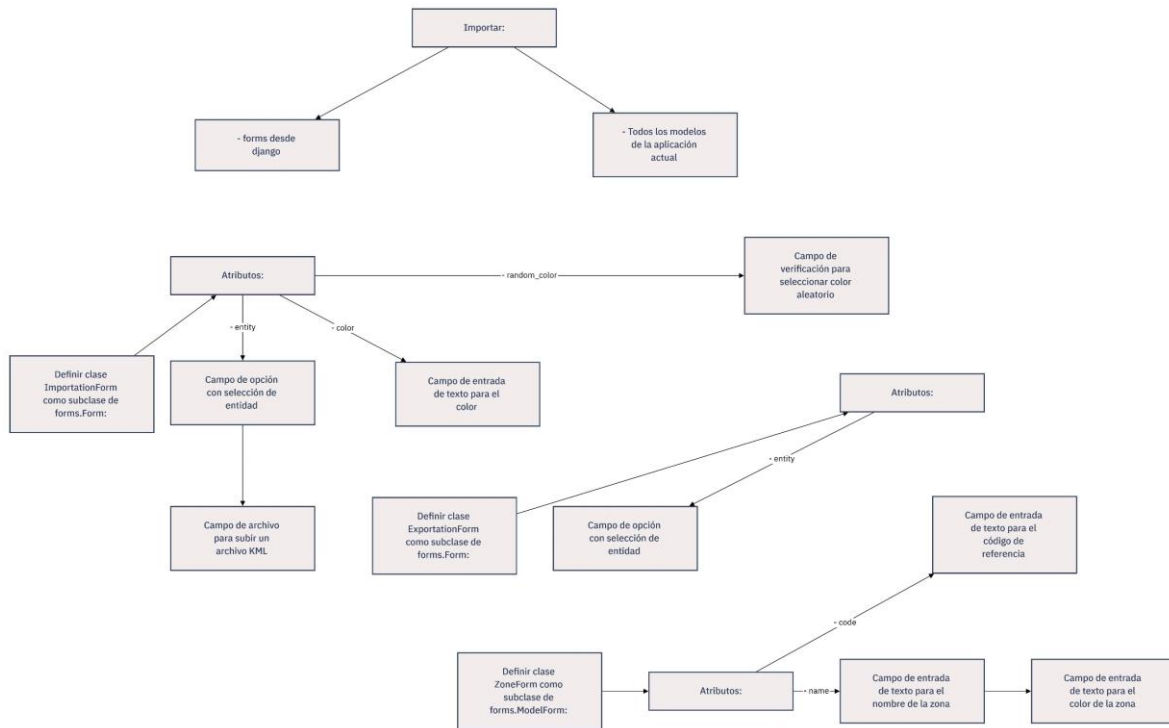


Ilustración 16 - Modelo GeoLocalización

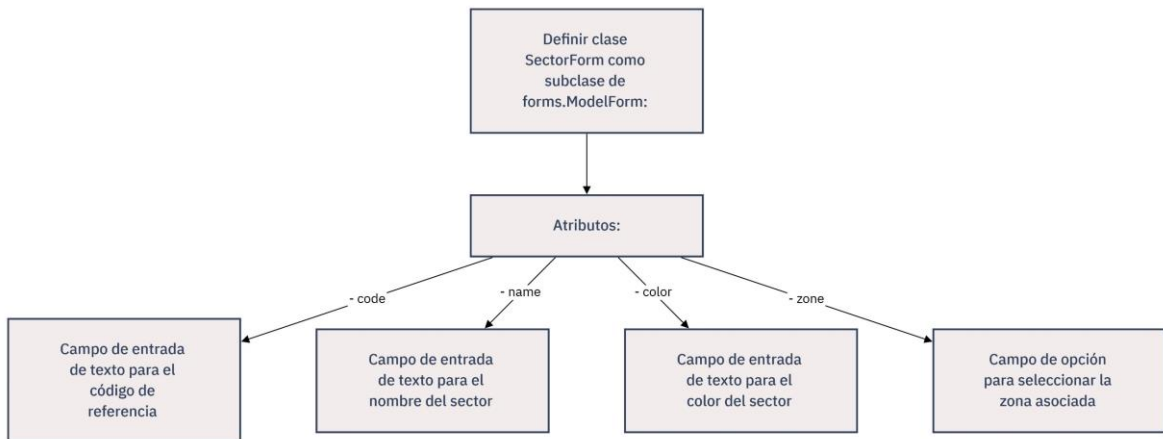


Ilustración 17 - Sectores

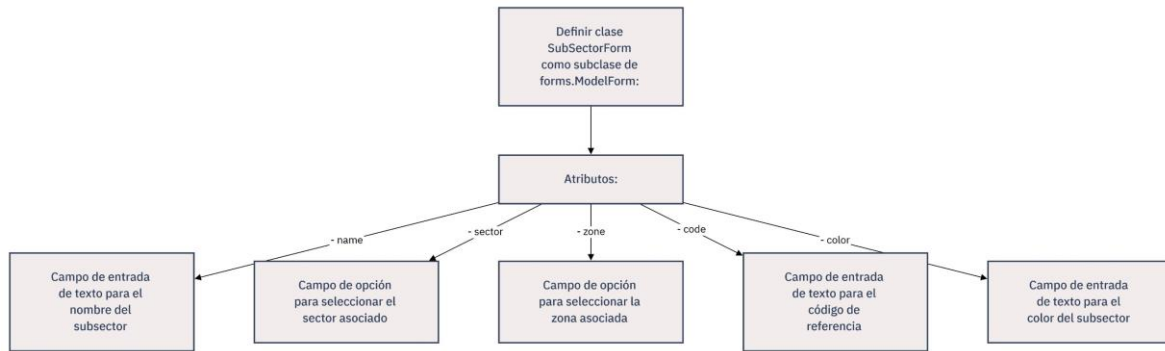


Ilustración 18 - Subsectores

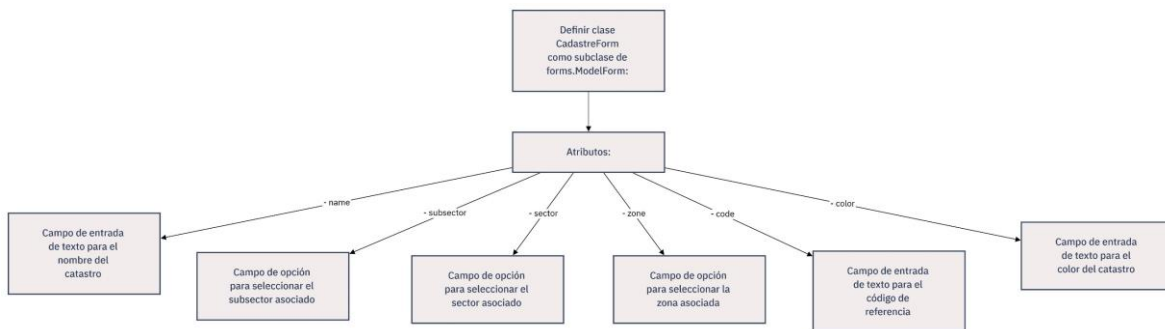


Ilustración 19 - Catastros

6. CONCLUSIONES

La aplicación móvil diseñada para la solicitud de servicios de gas en la ciudad de Cuenca representa una respuesta estratégica y necesaria para adaptarse a las modificaciones normativas y las demandas cambiantes del entorno urbano. La eliminación de notificaciones sonoras para la entrega de gas ha generado la urgencia de explorar soluciones innovadoras que garanticen la continuidad del servicio de manera eficiente.

El proyecto no solo busca cumplir con las nuevas regulaciones, sino que también se centra en mejorar la experiencia del usuario y optimizar la gestión de los distribuidores. La conveniencia, rapidez y accesibilidad ofrecidas por la aplicación contribuirán a mantener y mejorar la calidad del servicio de gas a domicilio.

La aplicación no solo se alinea con la tendencia general de utilizar la tecnología para simplificar procesos cotidianos, sino que también busca fomentar la adopción de tecnologías innovadoras en el sector de distribución de gas. La implementación de soluciones tecnológicas se presenta no solo como posible, sino también como beneficiosa para todas las partes involucradas.

La aplicación se enfoca en la creación dinámica de rutas entre puntos geográficos identificados por zonas. Con este precedente, la estructura del backend de la aplicación puede ser reutilizada para diversos proyectos de entregas, ya sea para generar rutas de taxis, busetas o para la distribución de productos a comercios.

7. RECOMENDACIONES

Durante el desarrollo del proyecto, hemos priorizado la implementación de métodos que no solo son empleados por nuestra aplicación, sino que también son susceptibles de ser reutilizados, reduciendo así la carga de ejecución de código. En este sentido, recomendamos la dockerización del servicio de OPEN ROUTE SERVICE. Esta medida permitirá independizar la ejecución del código de la latencia de conexión, ya que el servicio se convertirá en un microservicio alojado en el mismo backend.

Se recomienda continuar simplificando el proceso de solicitud de servicio de gas, manteniendo la fluidez y eliminando interacciones duplicadas para los usuarios. Es importante asegurar que el flujo de solicitud sea intuitivo y eficiente, evitando la repetición de acciones como realizar múltiples llamadas telefónicas o buscar distribuidores alternativos.

Ampliar el uso del backend de la aplicación móvil con distintos propósitos tales como la distribución de basura, el uso de transporte escolar o empresarial.

Utilizar las técnicas de IOT que permitan el acceso a la aplicación o mas que a una aplicación al servicio expuesto, por ejemplo la utilización de tecnologías o servicio como interactuar con el usuario mediante APIS y bots de whatsapp y sms o internet.

Establecer un proceso de monitoreo continuo del sistema para identificar áreas de mejora y oportunidades de optimización. Esto podría incluir la recopilación de comentarios de los usuarios, el análisis de métricas de rendimiento y la realización de pruebas de usabilidad periódicas. Mantener un enfoque proactivo en la mejora constante del sistema garantizará su eficacia y adaptabilidad a las necesidades cambiantes de los usuarios y del entorno operativo.

8. REFERENCIAS BIBLIOGRÁFICAS

Smith, J., & Johnson, M. (2019). Impacto de dispositivos móviles en la eficiencia y accesibilidad de los servicios a domicilio. *Journal of Mobile Technology*, 8(2), 45-56.

Muñoz, A. (2023). Aplicaciones móviles y sus características específicas. *Revista de Tecnología Móvil*, 15(4), 123-136.

Gupta, R., & Kumar, S. (2018). Georreferenciación y asignación de distribuidores: un enfoque eficiente. *International Journal of Geospatial Research*, 22(3), 87-102.

Han, J., & Kamber, M. (2006). QGIS: Herramienta de generación de geocercas para asignación eficiente de distribuidores. *International Journal of GIS Applications*, 14(1), 34-48.

Owens, L., et al. (2010). SQLite como base de datos relacional en sistemas de gestión. *Journal of Database Systems*, 18(2), 201-215.

Chiu, W., & Chen, H. (2018). Almacenamiento en la nube con Firebase para datos flotantes. *Cloud Computing Journal*, 25(4), 167-180.

Lankford, S. (2018). Insomnia: Herramienta esencial para el control de APIs en aplicaciones móviles. *API Management Review*, 30(1), 56-68.

Kemp, D., & Dale, R. (2015). PyCharm: Entorno de desarrollo integrado para programación eficiente. *Journal of Python Development*, 12(3), 112-125.

Flanagan, D. (2011). JavaScript y su papel en la generación dinámica de polígonos. *Web Development Quarterly*, 18(4), 89-102.

Bodin, E. (2017). React Native: Desarrollo de aplicaciones móviles eficientes y multiplataforma. *Mobile App Development Journal*, 24(5), 201-215.

Zipf, A., et al. (2020). OpenRouteService: Generación de rutas precisas entre dos puntos en el mapa. *Journal of Geospatial Routing*, 28(2), 145-158.

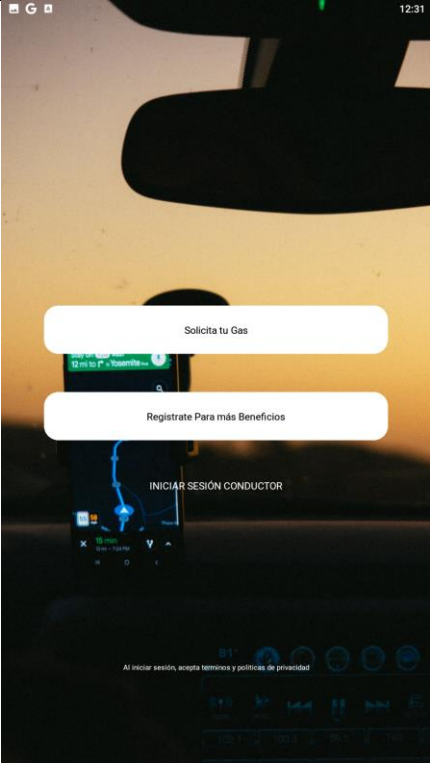
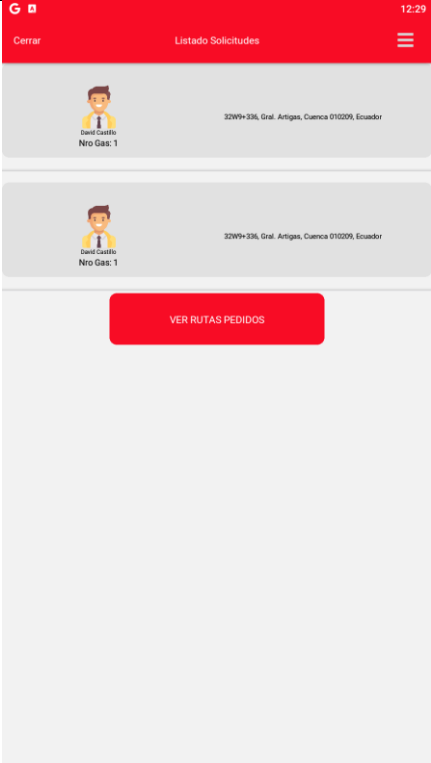
Jones, A., & Smith, B. (2020). Backend en Python: Columna vertebral para la gestión de información en aplicaciones móviles. *Python Backend Development*, 15(1), 78-91.

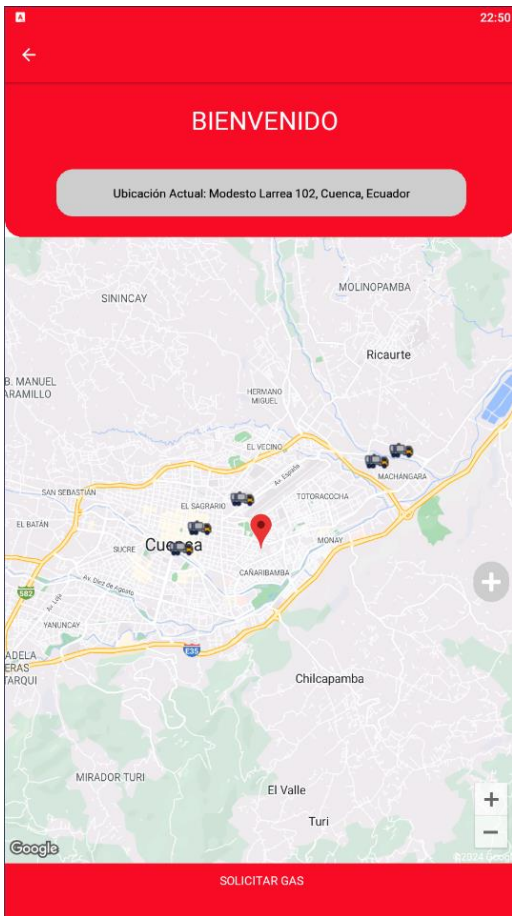
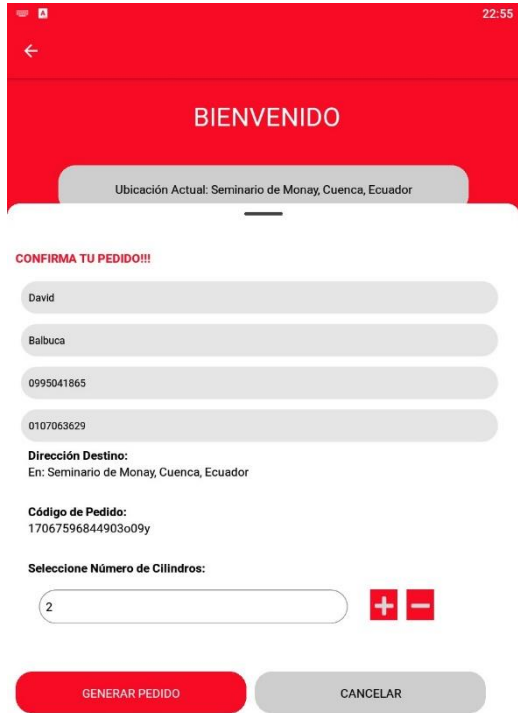
Clark, R., & Hall, S. (2018). Tecnologías móviles y enrutamiento eficiente en aplicaciones de solicitud de servicios. *Mobile Routing Trends*, 22(3), 102-115.

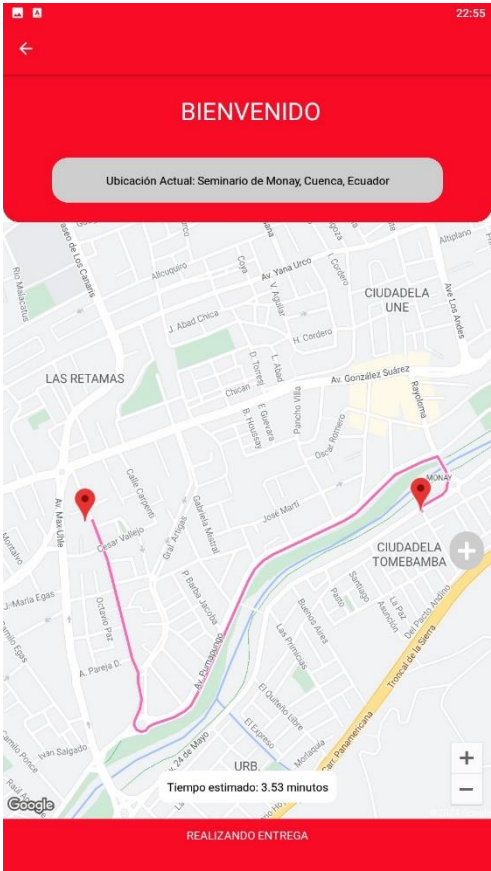
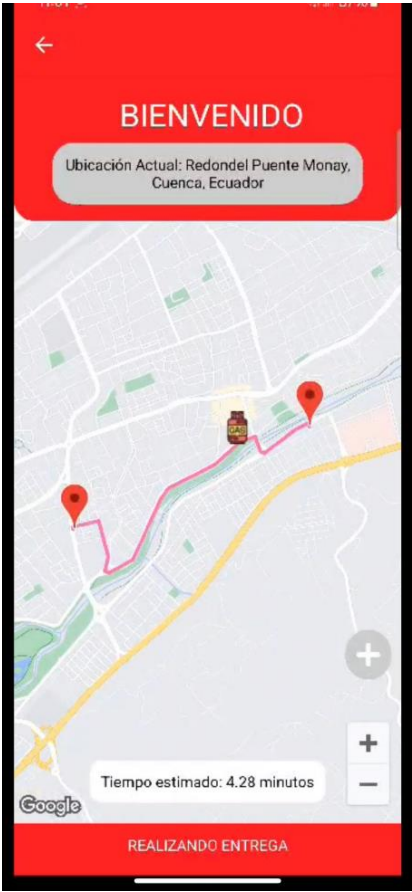
Miller, C., et al. (2017). Firebase como solución en la nube para almacenamiento de datos flotantes. *Cloud Storage Journal*, 32(4), 189-202.

9. ANEXOS

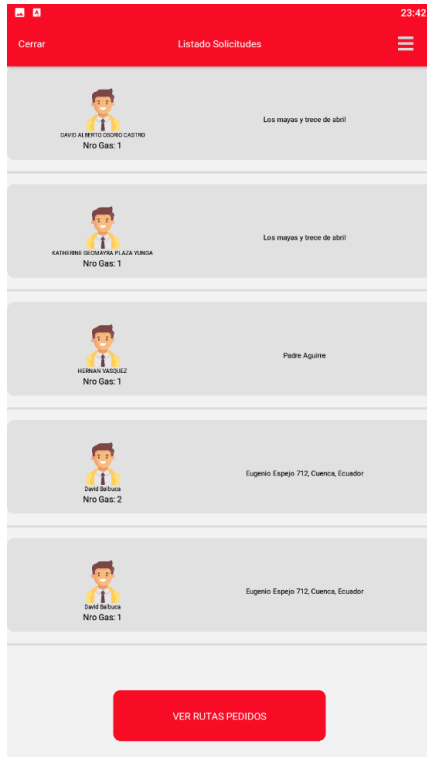
9.1. Anexo 1. Funcionamiento aplicación móvil:

Login	Dashboard Conductor
	
<i>Ilustración 20 - Pantalla Principal</i> Fuente: Elaboración propia.	<i>Ilustración 21 - Pedidos distribuidor</i> Fuente: Elaboración propia.

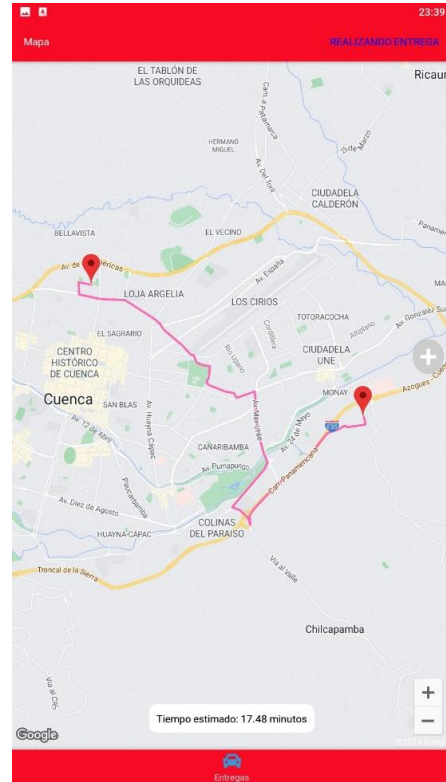
Dashboard cliente	Pedido Cliente
	
<i>Ilustración 22 - pantalla solicitud cliente</i> Fuente: Elaboración propia.	<i>Ilustración 23 - Pantalla Pedido cliente</i> Fuente: Elaboración propia.

Envío Pedido	Pedido en camino
	
<i>Ilustración 24 - Pantalla envío pedido cliente</i> Fuente: Elaboración propia.	<i>Ilustración 25 - Pantalla pedido en ruta</i> Fuente: Elaboración propia.

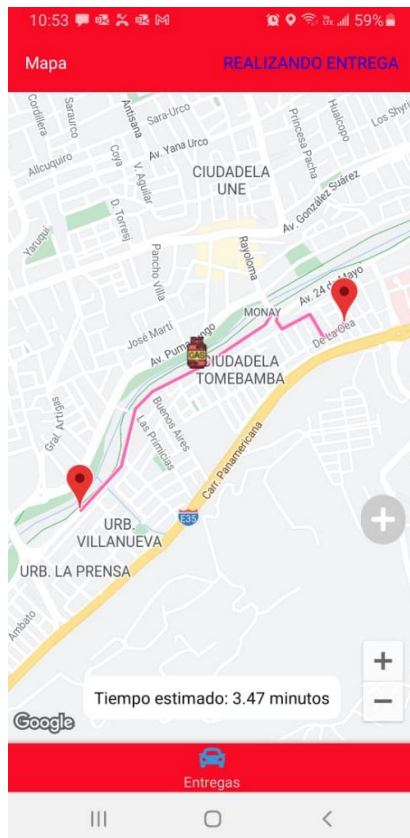
Pedidos Distribuidor	Generación de Ruta en Distribuidor
----------------------	------------------------------------



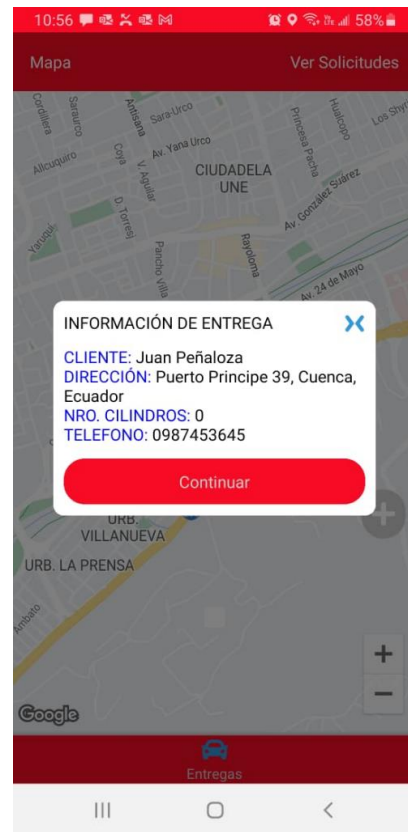
Aquí podemos evidenciar que los pedidos se encolan con el uso de Open Route Service.



Detalle de generación de ruta inicial y final de recorrido del distribuidor.

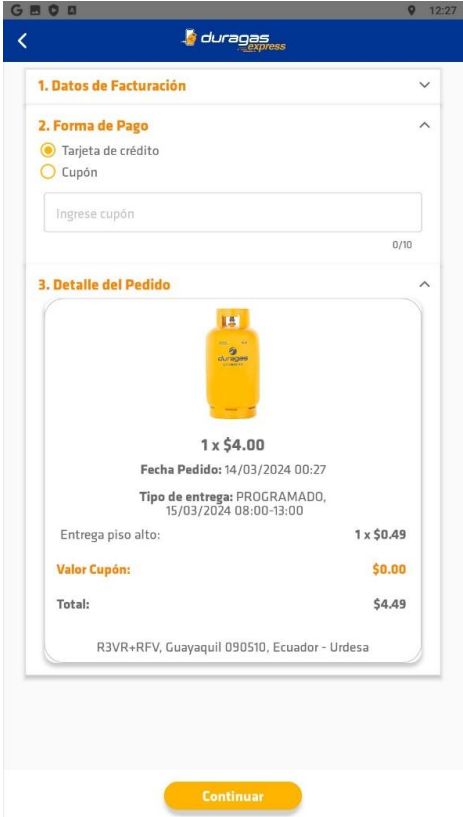
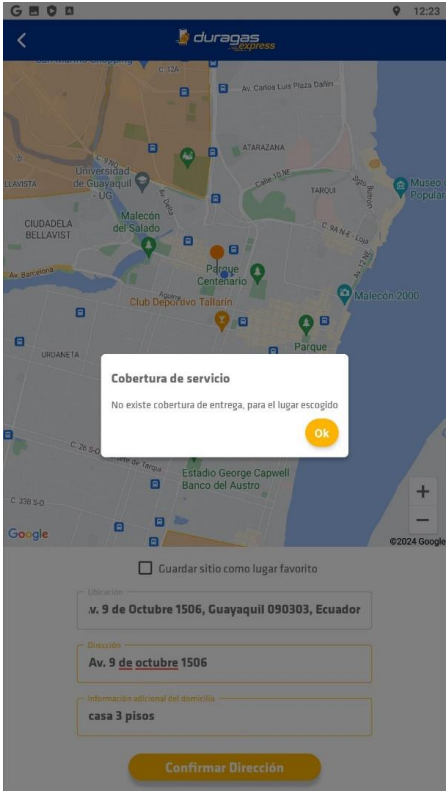


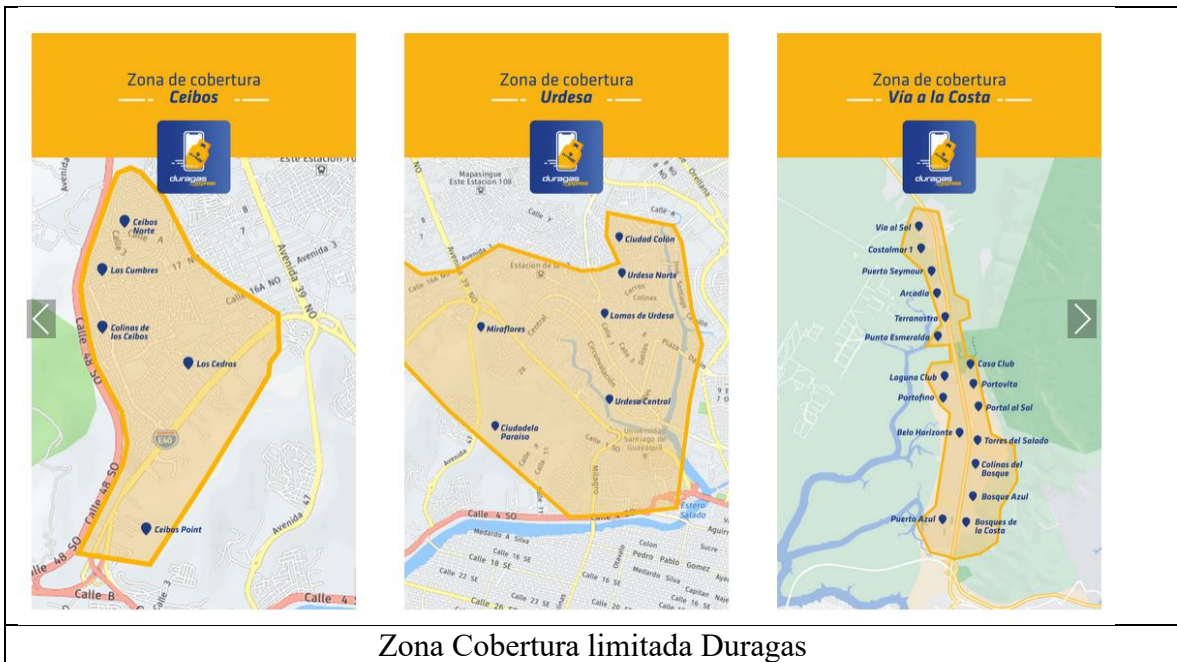
Detalle de Dashboard en tiempo real.



Detalle de pedido entregado en tiempo real.

9.2.Anexo 2. A continuación, se detalla el uso comparativo de las aplicaciones analizadas.

	
Análisis Pago aplicación Duragas	Aplicación Duragas, baja cobertura



Zona Cobertura limitada Duragas