

Optimización de la Logística en el Transporte de Carga Pesada a través de una Plataforma Móvil

Deivyd Dario Parra Perilla y Jose Issac Leal Manosalva

Trabajo de grado para optar por el título de ingeniero de sistemas

Director:

John William Vásquez Capacho, Ph.D en ingeniería, Escuela de ingeniería de sistemas e informática.

Codirector:

Jathinson Meneses Mendoza, MSc. Ingeniería de sistemas, Escuela de ingeniería de sistemas e informática.

Universidad Industrial de Santander

Facultad de ingenierías de ingenierías Físico-Mecánicas

Escuela de ingeniería de sistemas

Bucaramanga

2025

Agradecimientos

Queremos expresar nuestro agradecimiento al profesor John William Vásquez Capacho y al profesor Jathinson Meneses Mendoza, apreciamos de corazón su gran dedicación como docentes, su gran ambición por mejorar el sistema educativo en la universidad. Apreciamos su compromiso, conocimientos y confianza depositada en nosotros y su interés en nuestra idea. También queremos expresar nuestra gratitud a nuestras familias, quienes han brindado un apoyo incondicional y han sido nuestro pilar para poder continuar con nuestro desarrollo.

Tabla de contenido

Introducción..... 13

1. Planteamiento y justificación del problema.....15

2. Objetivos..... 17

 2.1 Objetivo general..... 17

 2.2 Objetivos específicos..... 17

3. Marco de referencia..... 18

 3.1 Estado del arte..... 18

 3.1.1 Loggi..... 18

 3.1.2 Truckstop..... 18

 3.1.3 CargaYa..... 19

 3.2 Marco teórico..... 19

 3.2.1 Sistema de gestión y asignación de cargas..... 19

 3.2.2 Extensibilidad..... 20

 3.2.3 Microservicios..... 20

 3.2.4 Alta escalabilidad..... 21

 3.2.5 Alta disponibilidad..... 21

 3.3 Marco tecnológico..... 21

 3.3.1 Typescript..... 21

 3.3.2 Nest.js..... 22

 3.3.3 Dart..... 22

 3.3.4 Flutter..... 22

 3.3.5 Android studio..... 24

 3.3.6 Xcode..... 24

 3.3.7 GIT..... 24

 3.3.8 MySQL..... 25

 3.3.9 Terraform..... 25

 3.3.10 TablePlus..... 25

 3.3.11 Docker..... 26

 3.3.12 AWS..... 26

 3.3.13 Firebase..... 27

 3.4 Marco tecnológico..... 28

4. Metodología..... 29

 4.1 Marco de trabajo..... 29

4.2 Gestión de versiones y control de código fuente con Git..... 30

5. Resultados..... 32

5.1 Análisis del sistema..... 32

5.1.1 Análisis del servicio actual..... 32

5.1.2 Propósito del sistema..... 32

5.1.3 Alcance del sistema..... 32

5.1.4 Criterios de éxito del proyecto..... 33

5.1.5 Características de los usuarios..... 34

5.1.6 Requerimientos funcionales y no funcionales..... 34

5.1.6.1 Requerimientos funcionales..... 35

5.1.6.2 Requerimientos no funcionales..... 43

5.1.7 Casos de uso..... 43

5.1.8 Diagramas de casos de uso..... 46

5.2 Diseño del sistema..... 47

5.2.1 Arquitectura del sistema..... 47

5.2.2 Modelado de procesos del sistema..... 49

5.2.3 Diseño de prototipos para la interfaz de usuario..... 51

5.2.4 Modelo de base de datos..... 53

5.3 Implementación..... 54

5.3.1 Autenticación de los usuarios..... 54

5.3.2 Envío de correos..... 55

5.3.2.1 Tecnologías Utilizadas..... 55

5.3.2.2 Proveedor de Servicio de Correo..... 55

5.3.2.3 Plantillas de Correos Electrónicos..... 56

5.3.3 Registrarse..... 59

5.3.4 Crear oferta de viaje..... 64

5.3.5 Menú de ofertas..... 66

5.3.6 Detalles de oferta de viaje..... 67

5.4 Pruebas..... 68

5.4.1 Pruebas unitarias..... 68

5.4.2 Automatización de Pruebas Funcionales..... 68

5.4.3 Ejecución de Pruebas en Simultáneo..... 69

5.4.4 Generación de Informes Detallados..... 69

5.5 Documentación..... 69

5.5.1 Documentación de herramientas..... 69

5.5.2 Documentación de código.....	70
5.6 Despliegue.....	71
5.6.1 Creación de imágenes Docker.....	73
5.6.2 Despliegue de infraestructura y sistema en EC2 utilizando Terraform.....	74
5.6.3 Configuración de red y almacenamiento.....	76
5.6.4 Instalación de Docker y despliegue de contenedores.....	77
5.6.5 Construcción, carga y ejecución de contenedores Docker desde Amazon ECR.....	77
5.6.6 Verificación del sistema.....	79
5.6.7 Costos de operación en la nube.....	79
6. Conclusiones.....	80
7. Recomendaciones.....	81
7.1 Validación por Especialistas en UX.....	81
7.2 Documentación y Guías de Estilo.....	81
7.3 Monitoreo Continuo.....	81
7.4 Evolución de diseño visual.....	82
Referencias Bibliográficas.....	87
Apéndices.....	91
Links Informativos de Referencia.....	131
Youtube.....	131
GitHub.....	131

Lista de Tablas

Tabla 1 Requerimientos funcionales de la plataforma.....	35
Tabla 2 Requerimientos funcionales de la plataforma.....	36
Tabla 3 Requerimientos funcionales de la plataforma.....	38
Tabla 4 Requerimientos funcionales de la plataforma.....	39
Tabla 5 Requerimientos funcionales de la plataforma.....	41
Tabla 6 Requerimientos NO funcionales de la plataforma.....	43

Lista de Figuras

Figura 1 Modelo sobre la Metodología Ágil Kanban..... 30

Figura 2 Diagrama trunk based development..... 31

Figura 3 Diagrama de casos de uso de conductor.....46

Figura 4 Diagrama de casos de uso empresa..... 47

Figura 5 Modelo de objetos del sistema..... 50

Figura 6 Diagrama de actividades.....51

Figura 7 Modelo de base de datos del sistema.....53

Figura 8 Olvidaste tu contraseña.....57

Figura 9 Restablecer tu contraseña..... 58

Figura 10 Opciones de registro..... 60

Figura 11 Registrarse como conductor..... 61

Figura 12 Formulario de empresa..... 62

Figura 13 Perfil de usuario..... 63

Figura 14 Formulario de oferta..... 65

Figura 15 ofertas de viaje..... 66

Figura 16 Detalles de la oferta..... 67

Figura 17 Infraestructura.....73

Figura 18 Costos Operacionales..... 80

Figura 19 Prototipo Login Original..... 83

Figura 20 Implementación Login final..... 83

Figura 21 Prototipo Registro Original..... 84

Figura 22 Implementación registro final.....84

Figura 23 Prototipo restablecer contraseña..... 85

Figura 24 Implementación restablecer contraseña..... 85

Figura 25 Prototipo publicar oferta..... 86

Figura 26 Implementación publicar oferta..... 86

Figura 27 Prototipo para pantalla de ingreso..... 91

Figura 28 Prototipo pantalla de ingreso con mensaje de error.....92

Figura 29 Prototipo para pantalla de opción de registro..... 92

Figura 30 Prototipo para pantalla de registro empresa..... 93

Figura 31 Prototipo para pantalla de registro transportista..... 94

Figura 32 Prototipo registro de vehículo..... 95

Figura 33 Prototipo perfil de usuario..... 96

Figura 34 Prototipo recuperar cuenta.....96

Figura 35 Prototipo restablecer contraseña..... 97

Figura 36 Prototipo publicar oferta.....	98
Figura 37 Prototipo ofertas de viajes.....	99
Figura 38 Prototipo detalles de oferta.....	99
Figura 39 Login.....	100
Figura 40 Login con manejo de errores.....	100
Figura 41 Olvidaste tu Contraseña.....	101
Figura 42 Restablecer Contraseña.....	101
Figura 43 Registrarse como.....	102
Figura 44 Registrarse como interacción.....	102
Figura 45 Registrar Conductor.....	103
Figura 46 Registrar Conductor manejo de errores.....	103
Figura 47 Registrar Empresa.....	104
Figura 48 Registro empresa manejo de errores.....	104
Figura 49 Home Empresa.....	105
Figura 50 Home Conductor.....	105
Figura 51 Perfil Empresa.....	106
Figura 52 Perfil Conductor.....	106
Figura 53 Publicar Carga como Empresa.....	107
Figura 54 Publicar Vehículo como Conductor.....	107
Figura 55 Buscar Conductor como Empresa.....	108
Figura 56 Buscar Viaje como Conductor.....	108
Figura 57 Mis cargas Empresa.....	109
Figura 58 Detalles del viaje Empresa.....	109
Figura 59 Mis viajes Conductor.....	110
Figura 60 Detalles del viaje Conductor.....	110
Figura 61 Chat Conductor.....	111
Figura 62 Perfil Público Empresa Chat Conductor.....	111
Figura 63 Lista de Interesados.....	112
Figura 64 Chat Empresa.....	112
Figura 65 Perfil Público Conductor Chat Empresa.....	113
Figura 66 Calificación Empresa.....	113
Figura 67 Mapa de navegación.....	114
Figura 68 Sección de Noticias.....	114
Figura 69 Centro de ayuda.....	115
Figura 70 Centro de ayuda de guías.....	115

Figura 71 Respuestas pregunta 01.....	116
Figura 72 Respuesta pregunta 02.....	117
Figura 73 Respuesta pregunta 03.....	117
Figura 74 Respuesta pregunta 04.....	118
Figura 75 Respuesta pregunta 05.....	118
Figura 76 Respuesta pregunta 06.....	119
Figura 77 Respuesta pregunta 07.....	120
Figura 78 Respuesta pregunta 08.....	120
Figura 79 Respuesta pregunta 09.....	121
Figura 80 Respuesta pregunta 10.....	121
Figura 81 Respuesta pregunta 11.....	122
Figura 82 Respuesta pregunta 12.....	122
Figura 83 Respuesta pregunta 13.....	123
Figura 84 Respuesta pregunta 14.....	123
Figura 85 Respuesta pregunta 15.....	124
Figura 86 Respuesta pregunta 16.....	124
Figura 87 Respuesta pregunta 17.....	125
Figura 88 Respuesta pregunta 18.....	125
Figura 89 Respuesta pregunta 19.....	126
Figura 90 Respuesta pregunta 20.....	126
Figura 91 Respuesta pregunta 21.....	127
Figura 92 Respuesta pregunta 22.....	127
Figura 93 Respuesta pregunta 23.....	128
Figura 94 Respuesta pregunta 24.....	128
Figura 95 Respuesta pregunta 25.....	129
Figura 96 Respuesta pregunta 26.....	129
Figura 97 Respuesta pregunta 27.....	130
Figura 98 Respuesta pregunta 28.....	130

Lista de apéndices

Apéndice A Mockups.....	91
Apéndice B Pantallas finales.....	100
Apéndice C Encuestas.....	116

Resumen

Título: Optimización de la Logística en el Transporte de Carga Pesada a través de una Plataforma Móvil.*

Autores: Deivyd Dario Parra Perilla, Jose Isaac Leal Manosalva**

Palabras Clave: Prototipo, Sistema de gestión, Solicitudes de viajes de carga, Plataforma móvil.

Descripción: El transporte de carga es crucial para la economía del país, pero enfrenta problemas como la falta de información en tiempo real, la ineficiencia en la asignación de recursos y la dependencia de métodos tradicionales como el "voz a voz" para encontrar cargas. Para abordar estos desafíos, se propuso una plataforma móvil que conecte a propietarios de camiones, conductores y empresas, mejorando la eficiencia operativa y reduciendo costos mediante una mejor coordinación, sistemas de calificación y notificaciones entre usuarios. Además, el proyecto tiene como objetivo proporcionar una herramienta tecnológica que mejore la competitividad del sector, que sea extensible y escalable para adaptarse a las necesidades futuras. Se exploran tecnologías como microservicios, Flutter, Android Studio, Nest.js, TypeScript, Firebase y Git, entre otros, para el desarrollo y gestión de la aplicación.

* Trabajo de grado.

** Facultad de Ingenierías Físico-Mecánica. Escuela de Ingeniería de Sistemas e Informática.

Director: Ph.D John William Vásquez Capacho.

Codirector: MSc. Jathinson Meneses Mendoza.

Abstract

Title: Optimization of Heavy Cargo Transportation Logistics through a Mobile Platform**

Author(s): Deivyd Dario Parra Perilla, Jose Isaac Leal Manosalva**

Key Words: Prototype, Management system, Cargo transport requests, Mobile platform.

Description: Freight transport is crucial for the country's economy, but it faces issues such as the lack of real-time information, inefficient resource allocation, and reliance on traditional methods like "word of mouth" to find loads. To address these challenges, a mobile platform was proposed to connect truck owners, drivers, and companies, improving operational efficiency and reducing costs through better coordination, rating systems, and user notifications. Additionally, the project aims to provide a technological tool that enhances the sector's competitiveness, making it extensible and scalable to adapt to future needs. Technologies such as microservices, Flutter, Android Studio, Nest.js, TypeScript, Firebase, and Git, among others, are explored for the development and management of the application.

* Degree work.

** School of Physical-Mechanical Engineering. Department of Systems Engineering and Informatics.

Director: PhD John William Vásquez Capacho.

Codirector: MSc. Jathinson Meneses Mendoza

Introducción

El transporte de carga pesada es un pilar fundamental para el desarrollo económico de todo país, debido a que facilita el flujo de mercancías a lo largo y ancho del país logrando conectar industrias y mercados. Sin embargo, a pesar de la importancia de este sector, enfrenta varios desafíos que limitan su eficiencia. Entre estos desafíos se encuentran la falta de integración de tecnologías modernas para la gestión logística, la asignación ineficiente de recursos y también sobresalen los problemas estructurales debido a la mala gestión por parte del gobierno, es por todo esto que existe la dependencia de métodos obsoletos para la comunicación entre transportistas y clientes para conseguir carga o gestionar viajes. Este panorama genera un nivel de ineficiencia significativo, lo que conlleva a aumentar los costos operativos y limitando la competitividad del sector.

Con el auge de las tecnologías móviles y la creciente digitalización de los procesos industriales, el sector del transporte de carga ha empezado a demandar soluciones más innovadoras que permitan optimizar la logística y la gestión de flotas. La implementación de plataformas móviles ha demostrado ser una alternativa viable para la solución de estos problemas, permitiendo la comunicación en tiempo real entre los diferentes actores involucrados: propietarios de vehículos, conductores y empresas contratantes. Estas plataformas permiten una mejor coordinación de los viajes de carga, la optimización de rutas, la asignación dinámica de recursos, así como la reducción de tiempos muertos y de costos asociados a la operación logística.

En este contexto, el presente proyecto propone el desarrollo de un prototipo de aplicación móvil que busca optimizar la logística del transporte de carga pesada. Esta plataforma quiere

permitir que los usuarios realicen solicitudes de viajes de forma rápida y sencilla, además de contar con un sistema de calificación para asegurar la calidad y confiabilidad del servicio. La aplicación contempla el uso de notificaciones en tiempo real para mejorar la interacción y coordinación entre los involucrados. La flexibilidad y escalabilidad del sistema aseguran que pueda adaptarse a las necesidades cambiantes del sector y facilitar su integración con tecnologías futuras. Para el desarrollo de la aplicación, se exploran tecnologías como Flutter para el desarrollo móvil, Nest.js y TypeScript para la lógica de negocio, y Firebase para la gestión de datos en la nube, etc.

Este prototipo no solo pretende resolver problemas actuales de eficiencia operativa en el transporte de carga, sino también brindar una solución extensible que potencie la competitividad del sector. Con ello, se busca contribuir al desarrollo económico y tecnológico de un sector clave en la economía nacional.

1. Planteamiento y justificación del problema

En la industria del transporte de carga pesada, la eficiencia y la competitividad son aspectos cruciales para el éxito de las operaciones. Sin embargo, los propietarios de camiones, conductores y empresas en el sector de carga pesada se enfrentan a la dificultad para encontrar oportunidades de carga y transporte de manera eficiente. Donde el mayor problema es la falta de información en tiempo real sobre la disponibilidad de carga y vehículos, así como la ineficiencia en la asignación de recursos, dificulta la planificación de rutas y la maximización de la capacidad de transporte entre otros problemas mencionados anteriormente.

Es por este motivo, que para abordar esta problemática, se propone el desarrollo de una aplicación móvil especializada en la gestión de carga pesada. Esta aplicación permitirá a los propietarios de camiones y empresas de transporte acceder a mayor información en tiempo real sobre oportunidades de carga y vehículos disponibles en diferentes ubicaciones geográficas. Además, pretende facilitar la comunicación y la coordinación entre propietarios de camiones, conductores y empresas de transporte, proporcionando de este modo la eficiencia operativa y reduciendo los costos asociados a las empresas y a los propietarios o conductores les brindaría mayor acceso a ofertas de negocio.

Se espera que esta aplicación contribuya a optimizar el proceso de búsqueda, gestión y asignación de cargas y viajes en el sector de carga pesada, mejorando así la competitividad de la industria y promoviendo un uso más eficiente de los recursos disponibles.

El desarrollo de una aplicación móvil especializada en la gestión de carga pesada es una respuesta necesaria a los desafíos actuales que enfrenta la industria del transporte de carga pesada, y tiene el potencial de mejorar significativamente la eficiencia y la competitividad en este sector

Para proporcionar un servicio de calidad, se llevará a cabo un análisis exhaustivo de las necesidades y expectativas de los usuarios (propietarios de camiones, conductores y empresas de transporte), con el objetivo de definir los requisitos funcionales y no funcionales del sistema. Además, se desarrollará un diseño de interfaz de usuario y se implementará un sistema de reseñas, calificaciones y validación para garantizar que los usuarios siempre encuentren cargas y viajes de calidad. También se integrará un sistema de chat entre propietarios de camiones y empresas de transporte para facilitar la comunicación y evitar la necesidad de compartir información personal innecesaria.

Pregunta de investigación: ¿Cómo podemos mejorar la eficacia de un sistema de gestión de carga pesada y establecer una relación de confianza entre propietarios de camiones, conductores y empresas de transporte a través de una aplicación móvil especializada?

2. Objetivos

2.1 Objetivo general

El objetivo general de este trabajo de grado es desarrollar un prototipo móvil que facilite la logística y operación en la industria de transporte de carga para propietarios de camiones, conductores y empresas en el sector de carga pesada, además de ello poder gestionar de manera eficiente las oportunidades de carga y transporte disponibles en diferentes ubicaciones geográficas en Colombia.

2.2 Objetivos específicos

- Definir los requerimientos funcionales según las necesidades y desafíos actuales en el sector de carga pesada en cuanto a la búsqueda y asignación de cargas y viajes.
- Establecer en el prototipo de la aplicación sistema de calificación y retroalimentación para mejorar la confiabilidad y la calidad del servicio ofrecido a través de la aplicación.
- Diseñar y desarrollar un prototipo de aplicación con una interfaz de usuario intuitiva y funcional, utilizando tecnologías de desarrollo multiplataforma, e integrar un sistema de notificaciones para mejorar la comunicación entre propietarios de camiones y empresas de transporte.

3. Marco de referencia

3.1 Estado del arte

3.1.1 Loggi

Loggi es una plataforma digital que conecta a transportistas, empresas y personas para la gestión de envíos y logística en Latinoamérica. Permite a los transportistas buscar y reservar entregas de diversos tipos, desde paquetes pequeños hasta mudanzas completas, con tarifas competitivas y en tiempo real. La plataforma ofrece herramientas para la gestión de las entregas como lo son el seguimiento en tiempo real, comunicación entre transportista y cliente, calificación del servicio y sistema de pagos seguro (Nowports, s. f.; Tiinside, 2025).

3.1.2 Truckstop

Truckstop es una plataforma digital que conecta a propietarios de camiones, conductores y empresas de transporte para la gestión de carga y viajes en Estados Unidos. Permite que los transportistas busquen cargas disponibles por ubicación, tipo de carga, dimensiones y peso. Ofrece filtros para la búsqueda y encontrar las que se ajusten mejor a sus necesidades. También permite reservar carga, enviar solicitudes, recibir información a detalle de cada carga y hacer la gestión de viajes con herramientas como seguimiento de ubicación, la comunicación entre transportista y dueño de la carga, envío de documentos, fotos y calificación del servicio. Truckstop ofrece un sistema de pagos seguro y transparente, donde los transportistas pueden recibir el pago de sus viajes directamente en la plataforma (Truckstop, s. f.).

3.1.3 CargaYa

CargaYa es una plataforma digital que conecta empresas con transportistas de carga independiente, facilitando la cotización de fletes para diferentes tamaños de carga y vehículos, y procurando un precio justo. (CargaYa, s. f.) .

En la actualidad, las plataformas analizadas confirman que existe un mercado creciente en digitalizar la gestión del transporte de carga. Sin embargo, estas aplicaciones presentan ciertas limitaciones:

- Contexto geográfico y regulatorio: Loggi opera principalmente en Brasil, CargaYa opera en Chile y Truckstop en Estados Unidos. Si bien estas soluciones ya han sido exitosas en este sector, no están adaptadas a las particularidades regulatorias del país ni al contexto colombiano.
- Barreras en la adopción local: La baja o nula cobertura regional y de integración con las necesidades del transporte colombiano (Informalidad, dependencia alta de métodos tradicionales, grupos o chats en aplicaciones que no están diseñadas para este mercado particular) dificulta que se adopten estas plataformas al país.

3.2 Marco teórico

3.2.1 Sistema de gestión y asignación de cargas

Este sistema hace referencia a una plataforma digital diseñada para facilitar la gestión, búsqueda y asignación de cargas para el sector de transporte de carga pesada. El objetivo principal es mejorar el proceso en la logística y transporte al proporcionar a los usuarios una herramienta centralizada para encontrar y gestionar cargas con una mejor eficiencia. Permitiendo

a los usuarios buscar y publicar cargas disponibles, establecer comunicación directa entre remitentes y transportistas, gestionar rutas y horarios de entrega, y realizar un seguimiento en de las operaciones de transporte. Incluso, integrar funcionalidades como sistemas que permitan la calificación y retroalimentación que garantice la calidad y seguridad del servicio ofrecido. Este sistema juega un papel fundamental en la mejora de la eficiencia operativa, la reducción de costos y la optimización de los recursos en la industria del transporte de carga pesada.

3.2.2 Extensibilidad

La extensibilidad en el desarrollo de software hace referencia a la capacidad que tienen los sistemas para adaptarse y crecer de manera eficiente en el tiempo. En el caso de una aplicación móvil que gestiona carga pesada, la extensibilidad es un punto muy importante que permitirá la incorporación de nuevas funcionalidades y la escalabilidad del sistema a medida que evolucionan las necesidades que se presentan en el mercado y los usuarios. Para lograr la extensibilidad, es fundamental seguir prácticas de diseño modular y arquitecturas flexibles que faciliten la incorporación de nuevos componentes, de tal modo que se evite la reescritura de código y modificar lo existente sin afectar el funcionamiento del sistema en su conjunto.

3.2.3 Microservicios

Los microservicios son una arquitectura en el desarrollo de software, la cual consiste en dividir una aplicación en pequeños servicios independientes, cada uno con un enfoque a una función específica y que se comunican entre sí a través de reglas claras. En el caso de una aplicación móvil para la gestión de carga pesada, hace uso de una arquitectura que se base en microservicios puede proporcionar ventajas como la flexibilidad, la escalabilidad y el mantenimiento simplificado. El uso de microservicios puede facilitar el desarrollo, el despliegue

y la actualización de manera independiente, permitiendo un trabajo más ágil en el desarrollo y que además se adapta a cambios en el negocio o en los requisitos del usuario.

3.2.4 Alta escalabilidad

Es la capacidad de un sistema para ajustarse y adaptarse al crecimiento en la carga de trabajo o demanda sin que su rendimiento o funcionalidad se vean comprometidos a bajos rendimientos. Un sistemas que se considera escalable es capaz de manejar un mayor volumen de usuarios o datos sin experimentar un desempeño más bajo o tiempos de respuesta más lentos. Esto se logra al implementar arquitecturas y tecnologías que permitan una distribución de la carga de trabajo de manera equitativa, así como la adopción de prácticas de diseño y desarrollo que minimicen los cuellos de botella.

3.2.5 Alta disponibilidad

Es la capacidad que tiene un sistema para estar disponible y operando durante periodos prolongados de tiempo, sin presentar caídas o tiempos de inactividad no planeados. Con ellos, garantizar a los usuarios poder acceder y usar los servicios que el sistema ofrece. Esto se logra al tener servidores, redes y bases de datos confiables, además de mecanismos de monitoreo, detección y recuperación de fallos que puedan permitir identificar los problemas para luego ser resueltos de manera rápida y eficiente.

3.3 Marco tecnológico

3.3.1 Typescript

TypeScript es un lenguaje de programación de código abierto desarrollado por Microsoft. Es una extensión tipada de JavaScript que permite a los desarrolladores crear aplicaciones escalables y robustas mediante la adición de tipos estáticos. Gracias a su fuerte tipado,

TypeScript facilita la detección temprana de errores, mejorando la productividad de los desarrolladores tanto en front-end como en back-end. La versatilidad de TypeScript lo convierte en una opción popular entre desarrolladores de pila completa.

3.3.2 Nest.js

NestJS, es un marco de desarrollo de back-end progresivo y eficiente para construir aplicaciones Node.js escalables y mantenibles. Inspirado en frameworks como Angular, NestJS aprovecha el poder de TypeScript para ofrecer una arquitectura modular que facilita el desarrollo de aplicaciones con una clara separación de responsabilidades. Además, NestJS se integra fácilmente con bibliotecas de terceros y ofrece soporte nativo para paradigmas de programación orientada a eventos y microservicios, lo que lo convierte en una opción ideal para aplicaciones modernas y complejas.

3.3.3 Dart

Dart es un lenguaje open source desarrollado en Google con el objetivo de permitir a los desarrolladores utilizar un lenguaje orientado a objetos y con análisis estático de tipo. Dart también se puede emplear para el desarrollo de aplicaciones web, dispositivos móviles y para servidores. Además, es un lenguaje que se basa en la sintaxis de C y Java. Posee una amplia cantidad de funcionalidades presentes en lenguajes modernos, como tipado fuerte, inferencia de tipo de datos, funciones de soporte para desarrollo asíncrono(Dart, s. f.).

3.3.4 Flutter

Flutter es un framework de código abierto desarrollado y compatible con Google. Los desarrolladores de front-end y pila completa utilizan Flutter para crear una interfaz de usuario (IU) de aplicación para varias plataformas con un único código base. Flutter utiliza el lenguaje

de programación de código abierto Dart, que también desarrolló Google. Dart está optimizado para la creación de interfaces de usuario, y muchos de los puntos fuertes de Dart se utilizan en Flutter. Además incluye un motor de renderizado propietario que permite a Flutter producir interfaces de usuario rápidamente y de manera eficiente (Wikipedia, s. f.).

La elección de Flutter como framework de desarrollo multiplataforma está justificada por los siguientes motivos:

1. Multiplataforma: Permite desarrollar con un solo código fuente aplicaciones tanto para Android como para iOS, lo que ahorra tiempos de desarrollo y mantenimiento del proyecto a futuro.
2. Mejor rendimiento: En comparación con React Native, flutter compila directamente el código nativo (ARM o x86) a través de su motor gráfico (Skia), lo que asegura mayor fluidez en interfaces complejas con menos consumos de recursos.
3. UI dinámica y personalizable: Este framework tiene un sistema de Widgets propios, lo que facilita diseños consistentes, garantizando sostenibilidad tecnológica.
4. Ecosistema confiable: Flutter está respaldado como framework de desarrollo por parte de Google, además cuenta con una comunidad en crecimiento quienes aportan widgets y mejoras al propio ecosistema.
5. Productividad en el desarrollo: Cuenta con una característica llamada “Hot reload” que permite ver en tiempo de desarrollo los cambios hechos en el código fuente, lo cual reduce tiempos muertos en la fase de desarrollo.

3.3.5 Android studio

Android Studio es el entorno de desarrollo integrado oficial (IDE) para crear aplicaciones Android. Desarrollado por Google, te ofrece un conjunto completo de herramientas y funcionalidades que te permiten dar vida a tus ideas y convertirlas en aplicaciones móviles de alta calidad (Android Developers, s. f.). Android Studio incluye un conjunto de herramientas para ayudar a los desarrolladores a programar y probar aplicaciones móviles para Android. Incluye un editor de código, un depurador, un diseñador de interfaz de usuario y un emulador de dispositivos móviles. También incluye integraciones con otras herramientas y servicios de Google, como Google Play Console y Firebase.

3.3.6 Xcode

Xcode es un entorno de desarrollo integrado (IDE) ofrecido por Apple, que proporciona un conjunto de herramientas que incluyen un editor de código, depurador, simuladores, y recursos como documentación, guías oficiales y soporte comunitario. Además, Apple ofrece foros de desarrolladores y recursos online que facilitan la colaboración y el aprendizaje dentro de la comunidad (Apple, s. f.; Apple Developer, s. f.).

3.3.7 GIT

Git es un sistema de control de versiones de código fuente. Es una herramienta muy útil para los desarrolladores de software, ya que permite llevar un registro de todos los cambios realizados en el código fuente de un proyecto y facilitar la colaboración entre múltiples desarrolladores (Git, s. f.). Git permite a los desarrolladores crear “ramas” (versiones) de un proyecto y trabajar en ellas de manera independiente, permitiendo que varios desarrolladores trabajen en diferentes partes del proyecto al mismo tiempo sin pisarse mutuamente. Al final, las

ramas pueden ser fusionadas para generar una versión final del proyecto. Git es una herramienta muy utilizada en el mundo del desarrollo de software y es esencial para trabajar de manera eficiente en proyectos de código abierto (Git, s. f.; Atlassian, s. f.).

3.3.8 MySQL

Es un sistema de gestión de bases de datos relacionales (SGBD) de código abierto, desarrollado y soportado por Oracle Corporation. Se basa en el lenguaje de consulta SQL (Structured Query Language) y se utiliza para almacenar y gestionar datos de forma organizada y eficiente (Oracle, s. f.; MySQL, s. f.).

3.3.9 Terraform

Terraform es una herramienta de código abierto desarrollada por HashiCorp que permite gestionar la infraestructura como código (IaC). Esto significa que puedes definir la infraestructura que deseas crear (servidores, redes, almacenamiento, etc.) en un archivo de configuración, y Terraform se encargará de provisionarla y administrarla de forma automática (HashiCorp, s. f.; IBM, 2025).

3.3.10 TablePlus

TablePlus es una aplicación multiplataforma (Windows, macOS, Linux e iOS) que permite gestionar una amplia variedad de bases de datos relacionales como MySQL, PostgreSQL, SQLite y Microsoft SQL Server, entre otras (TablePlus, s. f.). Gracias a su interfaz intuitiva y moderna, facilita tareas como la visualización, edición de datos, ejecución de consultas y filtrado, siendo útil para desarrolladores, administradores de bases de datos o cualquier usuario que necesite manipular información de forma organizada (TablePlus, s. f.; TablePlus Blog, 2018).

3.3.11 Docker

Docker es una plataforma de código abierto para desarrollo, envío (“shipping”) y ejecución de aplicaciones mediante contenedores. Un contenedor consiste en una unidad estandarizada de software que agrupa una aplicación junto con todas sus dependencias, para que se ejecute de forma consistente en distintos entornos (Docker, s. f.). Docker permite a desarrolladores y operadores crear, desplegar y gestionar estas aplicaciones de manera rápida, confiable y reproducible (IBM, s. f.).

3.3.12 AWS

Amazon Web Services (AWS) es una plataforma de computación en la nube que ofrece una amplia gama de servicios globales de computación, almacenamiento, bases de datos, análisis, redes, inteligencia artificial y aprendizaje automático, entre otros. Permite a las empresas escalar sus recursos de TI de forma rápida y flexible, sin necesidad de invertir en hardware o software propio (AWS, 2024; AWS, s. f.)

La elección de Amazon Web Services (AWS) como proveedor de infraestructura está fundamentada en las siguientes características:

1. **Liderazgo en el mercado:** AWS es el proveedor de nube con mayor solidez a nivel mundial, con la mayor cuota de mercado y un portafolio de servicios muy amplio que se adapta a proyectos de diferentes tamaños.
2. **Ecosistema integral:** Ofrece servicios que cubren todo el ciclo de vida de la aplicación:
 - a. **Amazon EC2** para cómputo escalable.
 - b. **Amazon RDS** para bases de datos gestionadas.

- c. **Amazon ECR** para contenedores
 - d. **Amazon SES** para el envío de correos a gran escala.
3. **Precios competitivos y flexibilidad:** El modelo “pay as you go” permite optimizar costos de operación, algo clave para proyectos en fase de prototipo que no requieren grandes cantidades de capital inicial.
 4. **Escalabilidad y alta disponibilidad:** Su red global de centros de datos garantiza disponibilidad de hasta un 99.99% y la capacidad de escalar según la demanda.

3.3.13 Firebase

Firestore es una plataforma de desarrollo de aplicaciones móviles y web, creada y respaldada por Google. Firestore ofrece una variedad de servicios backend como bases de datos en tiempo real, autenticación, almacenamiento y notificaciones push, todo a través de una única interfaz. Su base de datos en tiempo real, basada en NoSQL, permite a los desarrolladores sincronizar y almacenar datos entre usuarios de manera rápida y eficiente. Firestore también integra herramientas de análisis y seguimiento de usuarios, facilitando la monitorización del rendimiento y la toma de decisiones en tiempo real.

Uno de los principales puntos fuertes de Firestore es su capacidad para acelerar el desarrollo de aplicaciones, eliminando la necesidad de gestionar infraestructura backend compleja. Su integración nativa con Google Cloud y el soporte para diversas plataformas, como Android, iOS y web, permite a los desarrolladores crear aplicaciones multiplataforma con un único backend centralizado. Además, su arquitectura serverless permite a los desarrolladores enfocarse en escribir código sin preocuparse por la administración de servidores o escalado manual.

3.4 Marco tecnológico

Con el fin de garantizar la calidad, seguridad y usabilidad del prototipo desarrollado, se vió la necesidad de tener en cuenta diferentes normas y estándares internacionales que guían tanto el desarrollo de software como las prácticas del sector transporte:

- ISO/IEC 25010:2011 – Define un modelo de calidad para software y sistemas que abarca atributos como funcionalidad, fiabilidad, usabilidad, eficiencia, mantenibilidad, portabilidad y seguridad. Estos atributos sirvieron de referencia al evaluar el prototipo en aspectos como la experiencia de usuario, la seguridad de la información y la escalabilidad del sistema.
- ISO 9241-210:2019 – Estándar orientado al diseño centrado en el usuario, que recomienda iteraciones continuas y retroalimentación de usuarios para garantizar que la interfaz cumpla con las expectativas de usabilidad. Esta norma sustenta la validación realizada con especialistas en UX.
- ISO/IEC 27001:2013 – Estándar internacional para la gestión de seguridad de la información, que orienta la protección de datos sensibles mediante prácticas de cifrado, control de accesos y auditoría.
- ISO/IEC 27018:2019 – Enfocado en la protección de datos personales en entornos de computación en la nube. Es especialmente relevante dado que el sistema almacena datos de usuarios en AWS y Firebase.
- ISO 39001:2012 – Norma de gestión de seguridad vial, la cual puede ser de interés para el sector transporte en el que se enmarca este proyecto, con el fin de reducir riesgos asociados al tránsito de carga pesada.

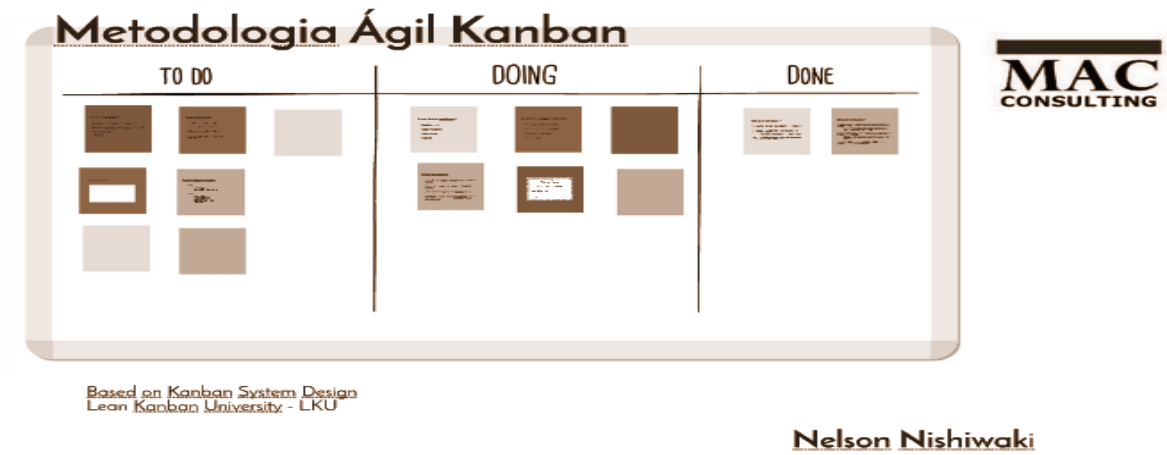
La integración de estas normas refuerza el alineamiento del proyecto con buenas prácticas internacionales, asegurando que el prototipo no solo cumpla con objetivos funcionales inmediatos, sino que también esté preparado para evolucionar bajo estándares de calidad reconocidos en la industria.

4. Metodología

4.1 Marco de trabajo

El desarrollo de este proyecto se basará en la metodología Kanban, una estrategia ágil que se caracteriza por su enfoque flexible, visual y orientado a la mejora continua. Con esta metodología, se pueden gestionar eficientemente las tareas y el flujo de trabajo, permitiendo una rápida adaptación a los cambios y una mayor transparencia en el proceso. Kanban se fundamenta en el uso de tableros visuales que representan el flujo de trabajo y las tareas pendientes, en curso y completadas. Cada tarea se representa como una tarjeta que se mueve a lo largo de las distintas etapas del proceso, desde la solicitud hasta la finalización. Esto facilita la identificación de cuellos de botella y la optimización del flujo de trabajo. En Kanban se trabaja de manera continua, priorizando las tareas según su importancia y completando a medida que surgen. El enfoque colaborativo y la retroalimentación constante permiten una mayor agilidad y eficiencia en la entrega de resultados.

Figura 1 Modelo sobre la Metodología Ágil Kanban



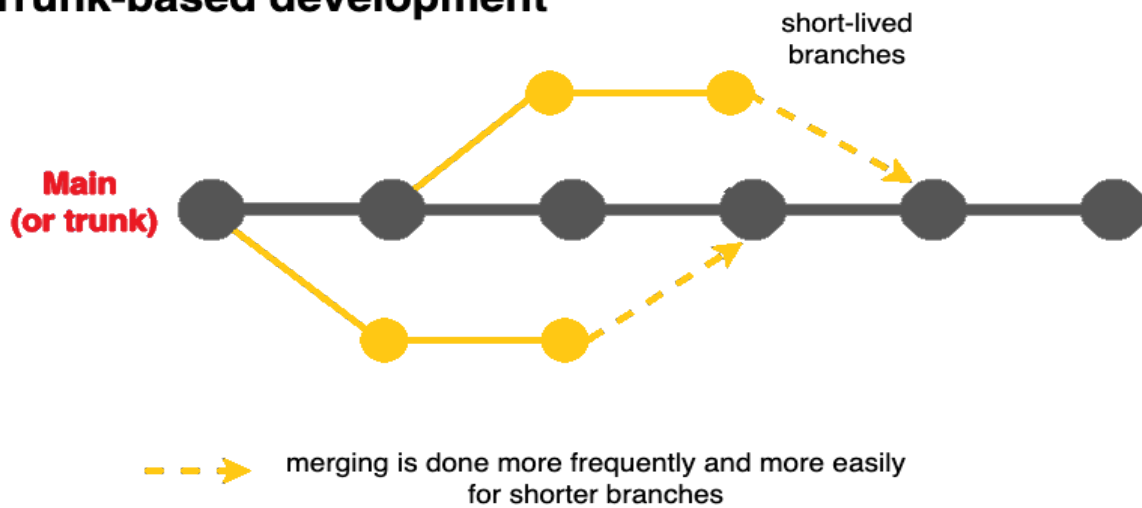
Tomado de [Metodología Ágil Kanban by Nelson Nishiwaki on Prezi](#)

4.2 Gestión de versiones y control de código fuente con Git.

Este proyecto de software estaba compuesto por dos personas, por lo cual la metodología Trunk Based Development se eligió como la más adecuada, ya que ofrece varios beneficios clave en este tipo de desarrollos. En primer lugar, la simplicidad de esta metodología facilita la colaboración y la integración continua sin la necesidad de gestionar múltiples ramas complejas. Cabe destacar que, al integrar todos los cambios en una rama principal, se fomenta la integración continua de la aplicación, permitiendo que se detecten y resuelvan conflictos de manera temprana. Esto permite una entrega más rápida de nuevas funcionalidades y correcciones de errores, lo que la convierte en una metodología ideal para proyectos que requieren iteraciones rápidas. Además, al reducir las ramas y simplificar el flujo de trabajo, los desarrolladores pueden centrarse más en el código que en la gestión del repositorio.

Figura 2 Diagrama trunk based development

Trunk-based development



Tomado de [Beginners Guide to Trunk-Based Development \(TBD\) - Sta](#)

5. Resultados

5.1 Análisis del sistema

5.1.1 Análisis del servicio actual.

El servicio actual relacionado con el **transporte de carga pesada** en Colombia presenta una serie de limitaciones que afectan la eficiencia operativa y la competitividad del sector. Esta falta de visibilidad genera ineficiencias en la **asignación de recursos**, ocasionando viajes vacíos, tiempos de espera prolongados y costos adicionales tanto para los conductores, los propietarios de camiones y para las empresas de transporte. Además, el sector carece de una herramienta tecnológica integral que permita gestionar de manera centralizada la disponibilidad de cargas, la comunicación entre actores y el seguimiento de los viajes.

5.1.2 Propósito del sistema

El propósito del sistema es desarrollar una plataforma móvil innovadora que permita optimizar la logística en el transporte de carga pesada. Esta solución busca conectar a propietarios de camiones, empresas y conductores en tiempo real, facilitando la coordinación y asignación de cargas de manera eficiente, mejorando el aprovechamiento de los recursos y reduciendo tiempos muertos. El sistema tiene como objetivo sustituir los métodos tradicionales y poco eficientes, por una herramienta tecnológica avanzada que ofrezca información precisa y actualizada sobre la disponibilidad de camiones, rutas optimizadas, y la situación de los viajes, contribuyendo a una gestión logística más eficiente y rentable.

5.1.3 Alcance del sistema

El alcance del sistema incluye la creación de una aplicación móvil que se enfocará en gestionar los siguientes aspectos del proceso logístico de transporte de carga pesada:

- **Registro de usuarios:** El sistema permitirá a los conductores, propietarios de camiones y empresas crear cuentas personalizadas.
- **Asignación de cargas:** Las empresas podrán publicar cargas disponibles, mientras que los conductores podrán postularse para esos trabajos, optimizando así la conexión entre oferta y demanda de transporte.
- **Notificaciones y comunicación en tiempo real:** Se enviarán notificaciones automáticas a los usuarios sobre el estado de las cargas, viajes en curso, disponibilidad de camiones, etc.
- **Sistemas de calificación:** Los usuarios podrán calificar tanto a empresas como a conductores, creando así un sistema de confianza basado en experiencias anteriores.

El sistema no contempla inicialmente la implementación de tecnologías avanzadas como inteligencia artificial o análisis predictivo, pero se diseñará para ser extensible a futuro, permitiendo la incorporación de estas funcionalidades conforme crezcan las necesidades del sector, con el fin de ser competitivo respecto a lo que el medio demande y mejorar la calidad del servicio al cliente.

5.1.4 Criterios de éxito del proyecto

Este proyecto se considera finalizado, una vez se cumplan los siguientes aspectos:

- Se ha logrado el desarrollo del prototipo basados en los requerimientos funcionales y no funcionales surgidos de las encuestas y análisis del sistema de transporte de carga pesada.
- Se han desarrollado las pruebas que den validez al prototipo.

- Se ha realizado la documentación adecuada sobre el desarrollo llevado a cabo.
- El sistema cumple con los objetivos y criterios establecidos.

5.1.5 Características de los usuarios

- A continuación, se presentan las características de los usuarios que interactúan con el sistema. Es importante aclarar que las empresas y los conductores en el ámbito de carga pesada pueden registrarse en la plataforma y ejercer roles específicos para facilitar la interacción y la consecución de carga pesada:
- Empresa: Como empresa, un usuario puede buscar conductores para el transporte de carga pesada, ver las calificaciones y reseñas de los conductores, solicitar servicios de transporte, comunicarse con los conductores a través del módulo de chat y, finalmente, calificar el servicio una vez finalizado.
- Conductor: Como conductor, un usuario puede ofrecer servicios de transporte de carga pesada, definir sus horarios de disponibilidad y aceptar o rechazar las solicitudes de servicio de las empresas.
- Administrador: Como administrador, el usuario en este prototipo puede bloquear y desbloquear el acceso de un usuario a la plataforma.

5.1.6 Requerimientos funcionales y no funcionales

A partir del análisis realizado sobre el sistema actual en el sector del transporte de carga pesada, se identificaron y definieron los siguientes requerimientos funcionales y no funcionales con el objetivo de cubrir todas las necesidades detectadas y asegurar un sistema eficiente, escalable y adaptable a las demandas del mercado.

5.1.6.1 Requerimientos funcionales. Estos requerimientos describen las funciones esenciales que el sistema debe cumplir para lograr los objetivos planteados. A continuación, se detallan los principales requerimientos funcionales identificados:

Tabla 1 Requerimientos funcionales de la plataforma

ID	RFI	Fuente	Product Owner	
Nombre		<i>Registro/Inicio de sesión de usuarios</i>		
Complejidad		<i>Baja</i>	Prioridad	<i>5</i>
Tipo		<i>Necesario</i>	Requerimiento que lo utiliza o especializa	<i>No aplica</i>
Crítico		<i>Sí, es indispensable para que transportistas, empresas y administradores tengan acceso a la plataforma</i>		
Documentos de visualización asociados		<i>No se requiere</i>		
Usuarios		<i>Conductor, empresas y administradores.</i>		
Entrada		<i>Datos del futuro usuario como:</i> <ul style="list-style-type: none"> ● <i>Nombre de usuario</i> ● <i>Contraseña encriptada</i> 	Salida	<i>Registro o inicio de sesión exitoso</i>
Descripción				
<i>El sistema permitirá a las personas registrarse o iniciar sesión como usuarios mediante credenciales requeridas: nombre de usuario y contraseña encriptada, para llevar a cabo el registro o inicio de sesión exitosos dentro del sistema, permitiendo a su vez, que los usuarios tengan acceso a su información relevante.</i>				
Precondición				
<i>El usuario deberá llenar los campos de registro/ingreso de manera correcta.</i>				

Postcondición
<i>El usuario quedará registrado en la base de datos y su inicio de sesión se validará mediante un token de acceso.</i>
Consideraciones
<ol style="list-style-type: none"> 1. <i>Los datos no fueron diligenciados en su totalidad o existen campos vacíos (mensaje que indique que faltan datos por ser diligenciados).</i> 2. <i>Un usuario intenta registrarse con un nombre de usuario que ya está en uso (mensaje que indique que usuario ya existe)</i>
Criterios de aceptación
<i>Tras efectuar el registro o inicio de sesión, los datos serán aprobados por el sistema y el registro o inicio de sesión será exitoso.</i>

Tabla 2 Requerimientos funcionales de la plataforma

ID	<i>RF2</i>	Fuente	<i>Product Owner</i>	
Nombre	<i>Crear solicitudes de transporte</i>			
Complejidad	<i>Baja</i>	Prioridad	<i>5</i>	
Tipo	<i>Necesario</i>	Requerimiento que lo utiliza o especializa	<i>No aplica</i>	
Crítico	<i>Sí, es indispensable para que operadores de transporte, clientes puedan gestionar las solicitudes de transporte de carga de manera eficiente.</i>			
Documentos de visualización asociados	<i>No se requiere</i>			
Usuarios	<i>Clientes, operadores de transporte .</i>			
Entrada	<i>Datos de la solicitud de transporte, tales como:</i> <ul style="list-style-type: none"> ● <i>Nombre del cliente</i> 	Salida	<i>Confirmación de la solicitud de transporte.</i> <i>Número de referencia de la solicitud.</i>	

	<ul style="list-style-type: none"> ● <i>Detalles de la carga (peso, dimensiones)</i> ● <i>Punto de recogida y destino</i> ● <i>Fecha y hora del transporte</i> ● <i>Tipo de transporte requerido (grúa, camión, etc.)</i> 		
Descripción			
<p>El sistema permitirá a los clientes crear solicitudes de transporte de carga pesada mediante un formulario donde se deben ingresar los datos de la carga y los detalles del transporte requerido. El sistema mostrará una lista de conductores con vehículos disponibles que cumplen con las condiciones necesarias para la solicitud, como el tipo de vehículo, capacidad de carga, y disponibilidad. Los conductores podrán aceptar o rechazar la solicitud una vez revisen los detalles.</p>			
Precondición			
<p><i>El cliente deberá ingresar correctamente toda la información relacionada con la carga y el transporte.</i></p>			
Postcondición			
<p><i>La solicitud de transporte será registrada en la base de datos y se presentará una lista de conductores con vehículos compatibles para la solicitud. Una vez aceptada por un conductor, se confirmará el transporte al cliente.</i></p>			
Consideraciones			
<ol style="list-style-type: none"> 1. <i>Si los datos ingresados no son completos, el sistema mostrará un mensaje indicando qué información falta.</i> 2. <i>Si el peso o dimensiones de la carga exceden los límites permitidos para los vehículos disponibles, el sistema mostrará un mensaje de advertencia.</i> 			
Criterios de aceptación			
<p><i>Tras efectuar la solicitud, los datos serán validados por el sistema y se generará una lista de conductores con vehículos compatibles para la solicitud.</i></p> <p><i>El cliente recibirá una confirmación una vez que un conductor acepte la solicitud de transporte.</i></p>			

Tabla 3 Requerimientos funcionales de la plataforma

ID	<i>RF3</i>	Fuente	<i>Product Owner</i>	
Nombre	<i>cancelar solicitudes de transporte</i>			
Complejidad	<i>Baja</i>	Prioridad	<i>4</i>	
Tipo	<i>Necesario</i>	Requerimiento que lo utiliza o especializa	<i>No aplica</i>	
Crítico	<i>No, pero es importante para mantener flexibilidad en la gestión de transporte.</i>			
Documentos de visualización asociados	<i>No se requiere por el momento.</i>			
Usuarios	<i>Clientes, operadores de transporte .</i>			
Entrada	<p><i>Datos de la solicitud de transporte que se desea cancelar, tales como:</i></p> <ul style="list-style-type: none"> <i>Número de referencia de la solicitud.</i> <i>Motivo de la cancelación (opcional).</i> 	Salida	<ul style="list-style-type: none"> <i>Confirmación de la cancelación.</i> <i>Notificación al conductor asignado (si ya había aceptado la solicitud).</i> 	
Descripción				
<p>El sistema permitirá a los clientes cancelar una solicitud de transporte de carga pesada antes de que el transportador la cargue. Para ello, se deberá proporcionar el número de referencia de la solicitud y, opcionalmente, un motivo de cancelación. Si la solicitud ya ha sido aceptada por un conductor, este será notificado de la cancelación de manera inmediata. El sistema también actualizará el estado de la solicitud como "Cancelada" y se reflejará en el historial del cliente.</p>				
Precondición				

<i>El cliente deberá proporcionar el número de referencia de la solicitud que desea cancelar.</i>
Postcondición
<i>La solicitud será marcada como cancelada en la base de datos, se notificará al conductor (si corresponde), y el cliente recibirá una confirmación de la cancelación.</i>
Consideraciones
<ol style="list-style-type: none"> 1. <i>Si el número de referencia ingresado es incorrecto o no existe, el sistema mostrará un mensaje de error.</i> 2. <i>Si la solicitud ya está en curso (por ejemplo, si el transporte ya ha comenzado), la cancelación no será posible y se notificará al cliente.</i> 3. <i>El sistema debe actualizar el inventario de vehículos y conductores disponibles tras la cancelación de la solicitud.</i>
Criterios de aceptación
<p><i>Tras cancelar la solicitud, el sistema debe actualizar el estado de la misma a "Cancelada" y notificar al conductor asignado si había aceptado la solicitud.</i></p> <p><i>El cliente o administrador debe recibir una confirmación de que la solicitud ha sido cancelada correctamente..</i></p>

Tabla 4 Requerimientos funcionales de la plataforma

ID	<i>RF4</i>	Fuente	<i>Product Owner</i>	
Nombre	<i>Finalizar solicitudes de transporte de carga pesada (Entrega de carga)</i>			
Complejidad	<i>Media</i>	Prioridad	<i>5</i>	
Tipo	<i>Necesario</i>	Requerimiento que lo utiliza o especializa	<i>No aplica</i>	
Crítico	<i>Sí, es esencial para completar el ciclo de transporte y asegurar la entrega exitosa de la carga.</i>			
Documentos de visualización asociados	<i>No se requiere por el momento.</i>			

Usuarios	<i>Conductores, clientes.</i>		
Entrada	<p><i>Datos de la solicitud finalizada, tales como:</i></p> <ul style="list-style-type: none"> ● <i>Número de referencia de la solicitud.</i> ● <i>Confirmación de entrega por parte del conductor (puede incluir firma digital o foto).</i> ● <i>Confirmación de recepción por parte del cliente (opcional).</i> ● <i>Fecha y hora de la entrega.</i> 	Salida	<ul style="list-style-type: none"> ● <i>Confirmación de finalización de la solicitud.</i> ● <i>Registro de entrega exitosa en la base de datos.</i> ● <i>Notificación al cliente sobre la entrega completada.</i>
Descripción			
<p>El sistema permitirá a los conductores finalizar una solicitud de transporte de carga pesada una vez la carga ha sido entregada en su destino. Los conductores podrán marcar la solicitud como "Completada" proporcionando la confirmación de entrega, que puede incluir una firma digital o una foto. El sistema registrará la fecha y hora de la entrega, actualizando el estado de la solicitud en la base de datos y notificando al cliente sobre la entrega exitosa.</p>			
Precondición			
<p><i>El conductor debe haber completado el transporte y entregado la carga en el lugar de destino especificado.</i></p>			
Postcondición			
<p><i>La solicitud será registrada como finalizada en la base de datos y el sistema enviará una notificación al cliente, indicando que la entrega se ha completado.</i></p>			

Consideraciones
<ol style="list-style-type: none"> 1. Si el conductor no ha llegado al punto de destino, el sistema no permitirá marcar la solicitud como completada. 2. Si el cliente reporta que la carga no ha sido entregada en condiciones adecuadas, el sistema permitirá que se levante una disputa. 3. El sistema debe registrar evidencia de la entrega (firma o foto) para evitar reclamaciones posteriores.
Criterios de aceptación
<p>Tras finalizar la solicitud, el estado de la misma debe cambiar a "Completada" en el sistema. El cliente deben ser notificados automáticamente de que la entrega ha sido realizada con éxito.</p> <p>La confirmación de entrega, con la firma digital o foto, debe quedar registrada en el sistema como evidencia.</p>

Tabla 5 Requerimientos funcionales de la plataforma

ID	<i>RF5</i>	Fuente	<i>Product Owner</i>	
Nombre	<i>Calificación del servicio de transporte de carga pesada</i>			
Complejidad	<i>Baja</i>	Prioridad	<i>3</i>	
Tipo	<i>Deseable</i>	Requerimiento que lo utiliza o especializa	<i>No aplica</i>	
Crítico	<i>No, pero es importante para mejorar la calidad del servicio mediante la retroalimentación de los usuarios.</i>			
Documentos de visualización asociados	<i>No se requiere por el momento.</i>			
Usuarios	<i>Clientes</i>			

<p>Entrada</p>	<p><i>Datos de la calificación del servicio, tales como:</i></p> <ul style="list-style-type: none"> ● <i>Número de referencia de la solicitud de transporte.</i> ● <i>Puntuación del servicio (en una escala del 1 al 5).</i> ● <i>Comentarios adicionales (opcional).</i> 	<p>Salida</p>	<ul style="list-style-type: none"> ● <i>Confirmación de que la calificación ha sido registrada.</i> ● <i>Visualización de la puntuación promedio para el conductor o servicio de transporte.</i>
<p>Descripción</p>			
<p>El sistema permitirá a los clientes calificar el servicio de transporte una vez la solicitud haya sido finalizada. Los clientes podrán asignar una puntuación basada en su satisfacción con el servicio, y opcionalmente, proporcionar comentarios adicionales. La puntuación promedio del conductor será actualizada en función de las calificaciones recibidas. Esta información permitirá a la plataforma evaluar el desempeño de los conductores y mejorar la calidad del servicio.</p>			
<p>Precondición</p>			
<p><i>El servicio de transporte debe haber sido completado, y la solicitud debe estar marcada como "Completada" en el sistema.</i></p>			
<p>Postcondición</p>			
<p><i>La calificación se registrará en el sistema y se actualizará la puntuación promedio del conductor o servicio en cuestión.</i></p>			
<p>Consideraciones</p>			
<ol style="list-style-type: none"> 1. <i>Si el cliente no proporciona una puntuación válida (fuera del rango de 1 a 5), el sistema solicitará que ingrese una puntuación correcta.</i> 2. <i>Los comentarios adicionales son opcionales, pero se incentivará al cliente a proporcionar retroalimentación para mejorar el servicio.</i> 3. <i>Si el cliente no desea calificar el servicio, no se le obligará a hacerlo, pero se le enviarán recordatorios automáticos para que complete la evaluación.</i> 			

Criterios de aceptación
<i>Tras ingresar la puntuación y los comentarios, el sistema debe confirmar que la calificación ha sido registrada correctamente.</i>
<i>La puntuación promedio del conductor o servicio debe actualizarse de acuerdo con las calificaciones recibidas.</i>
<i>Los administradores podrán acceder a las calificaciones y comentarios para supervisar la calidad del servicio.</i>

5.1.6.2 Requerimientos no funcionales. Un requerimiento no funcional es aquel que mide y describe las características y restricciones del sistema que no está directamente relacionado a su comportamiento con su comportamiento específico o funcionalidad, en cambio se centra en los aspectos como el rendimiento, la seguridad, la usabilidad, la escalabilidad y otros atributos de calidad en el servicio. Estos requerimientos pretenden definir cómo debe comportarse el sistema en términos de su operación y mantenimiento, y son cruciales para asegurar que el sistema sea robusto, eficiente, escalable y fácil de usar.

5.1.7 Casos de uso

Las siguientes tabla ilustran las condiciones que se deben cumplir para que un requerimiento no funcional se cumpla.

Tabla 6 Requerimientos NO funcionales de la plataforma

Nombre del requerimiento	Involucrados	Descripción	Condiciones	Flujo
Rendimiento	Equipo de desarrollo, QA.	El sistema debe responder a las solicitudes del usuario dentro de un tiempo específico y soportar un número determinado de usuarios concurrentes.	- Tiempo de respuesta ≤ 4 segundos - Soporte para 5,000 usuarios simultáneos	1. Usuario realiza una solicitud. 2. El sistema procesa la solicitud. 3. El sistema responde en al menos de 4 segundos.
Seguridad	Equipo de	El sistema debe	- Uso de HTTPS	1. Usuario

	desarrollo, QA.	únicamente permitir que los usuarios autorizados puedan acceder a ciertas funcionalidades y proteger los datos sensibles.	para todas las comunicaciones. - Encriptación de datos en tránsito y en reposo. - Tokenización para las sesiones iniciadas. - Posible cierre de sesión forzada desde un perfil administrativo.	intenta acceder a una funcionalidad. 2. El sistema verifica la autorización. 3. El sistema encripta los datos sensibles.
Usabilidad	Equipo de desarrollo, UX/UI, QA	El sistema debe tener una interfaz intuitiva y fácil de usar, y ser accesible para usuarios con discapacidades.	- Interfaz intuitiva - Compatibilidad con lectores de pantalla	1. Usuario accede a la interfaz. 2. Usuario navega y realiza tareas sin dificultad.
Escalabilidad	Equipo de desarrollo, Arquitectos de software, DevOps	El sistema debe poder aumentar su capacidad añadiendo más servidores o nodos y manejar un aumento en la carga de trabajo sin degradar el rendimiento.	- Escalabilidad horizontal y vertical - Soporte para aumento del 50% en el número de usuarios - Añadir nuevos servidores sin tiempo de inactividad	1. Aumento en el número de usuarios. 2. Sistema distribuye la carga entre servidores. 3. El sistema añade nuevos servidores si es necesario.
Mantenibilidad	Equipo de desarrollo, DevOps, QA	El sistema debe tener documentación clara y completa y estar diseñado de manera modular para facilitar las actualizaciones y	- Documentación completa - Diseño modular - Pruebas automatizadas cubriendo al menos el 80% del código	1. Identificación de un problema. 2. Desarrollador consulta la documentación. 3. Desarrollador realiza la corrección y ejecuta pruebas

		correcciones.		automatizadas.
Disponibilidad	Equipo de desarrollo, Soporte técnico	El sistema debe estar disponible el 99.9% del tiempo y poder recuperarse de fallos en un tiempo específico.	<ul style="list-style-type: none"> - Tiempo de actividad del 99.99% - Plan de recuperación ante desastres - Recuperación completa en menos de 4 horas 	<ol style="list-style-type: none"> 1. El sistema experimenta un fallo. 2. El sistema activa el plan de recuperación. 3. El sistema se recupera completamente en menos de 4 horas.
Compatibilidad	Equipo de desarrollo, UX/UI, QA	El sistema debe ser compatible con otros sistemas y plataformas y funcionar correctamente en los navegadores web más comunes.	<ul style="list-style-type: none"> - Interoperabilidad con otros sistemas - Soporte para navegadores como Chrome, Firefox, Safari - - Compatibilidad con dispositivos móviles 	<ol style="list-style-type: none"> 1. Usuario accede al sistema desde diferentes plataformas. 2. El sistema funciona correctamente en todos los navegadores y dispositivos.
Portabilidad	Equipo de desarrollo, DevOps	El sistema debe poder ser desplegado en diferentes entornos y adaptarse a diferentes configuraciones de hardware y software.	<ul style="list-style-type: none"> - Despliegue en la nube y on-premises - Adaptabilidad a diferentes configuraciones de hardware y software 	<ol style="list-style-type: none"> 1. El Sistema se despliega en un nuevo entorno. 2. El Sistema se adapta a la configuración del nuevo entorno. 3. El sistema funciona correctamente en el nuevo entorno.

5.1.8 Diagramas de casos de uso

Los siguientes diagramas de casos de uso del sistema de asignación de transporte de carga pesada.

Figura 3 Diagrama de casos de uso de conductor

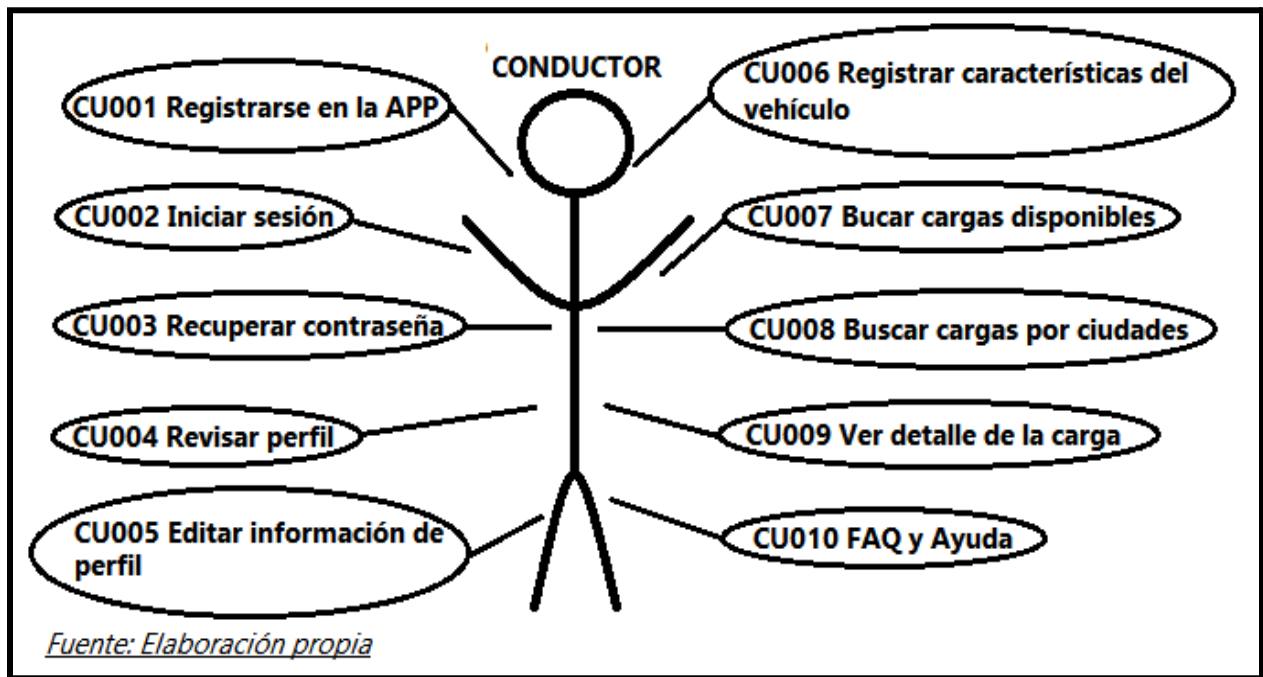
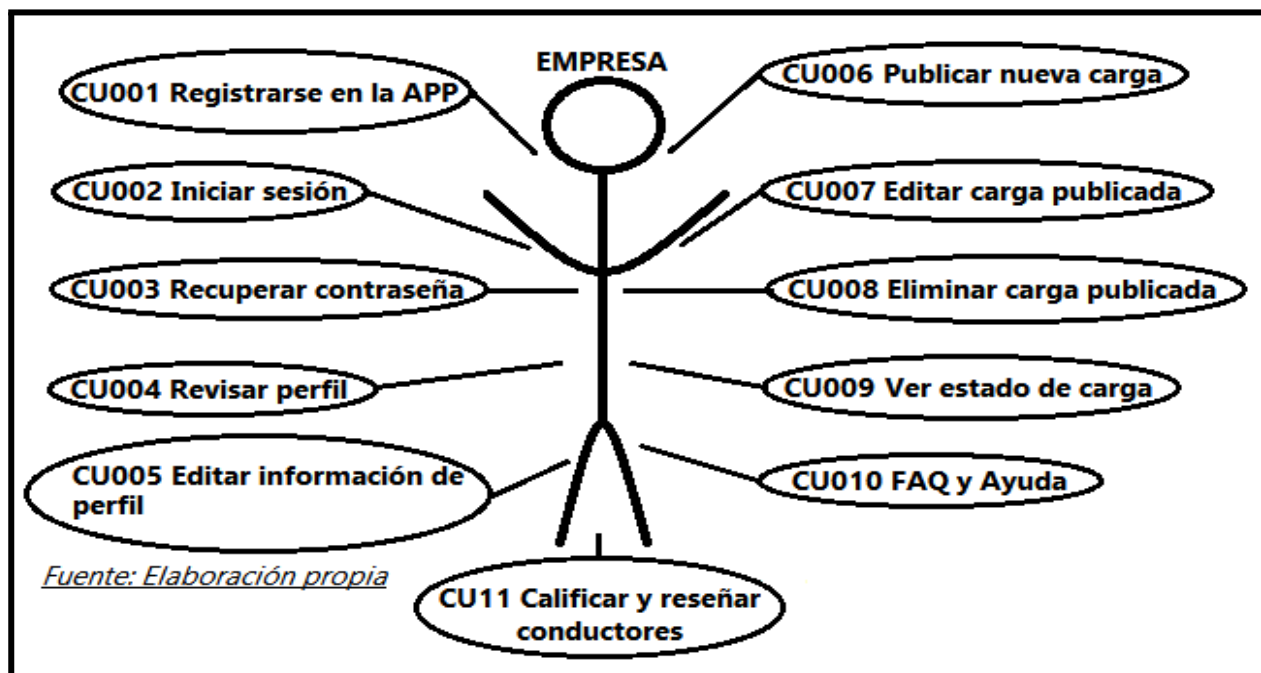


Figura 4 Diagrama de casos de uso empresa



5.2 Diseño del sistema

El diseño del sistema se estructuró de tal modo que el objetivo principal de lograr una arquitectura que sea tanto escalable como fácilmente extensible. Mediante este enfoque se asegura que el sistema se adapte a futuras demandas y crecimientos que posiblemente puedan no haber sido medidas en los análisis, sin necesidad de reestructuraciones significativas..

5.2.1 Arquitectura del sistema

La arquitectura del sistema está diseñada bajo un enfoque de microservicios, promoviendo la modularidad, escalabilidad y facilidad de mantenimiento. El backend ha sido desarrollado utilizando el framework NestJS, el cual permite una estructura clara y robusta basada en controladores, servicios e inyección de dependencias. Los distintos

microservicios se comunican mediante API REST, sin la incorporación de un sistema de mensajería o encolamiento, lo que simplifica la infraestructura y facilita el flujo sincrónico de la información.

En el frontend, se ha utilizado Flutter, lo que permite desarrollar una única base de código para aplicaciones móviles multiplataforma, manteniendo una experiencia de usuario consistente en dispositivos Android e iOS.

Cada componente del sistema está contenedorizado usando Docker, lo que garantiza la portabilidad, consistencia y facilidad en los despliegues. Las imágenes Docker de cada microservicio son almacenadas en Amazon Elastic Container Registry (ECR), mientras que la ejecución de los contenedores se realiza en instancias EC2 de AWS, proporcionando un entorno flexible y escalable para el procesamiento del backend.

La persistencia de datos está gestionada mediante MySQL, alojado en Amazon RDS (Relational Database Service). Esta configuración garantiza alta disponibilidad, respaldo automático y seguridad administrada, reduciendo la carga operativa y mejorando la confiabilidad del sistema.

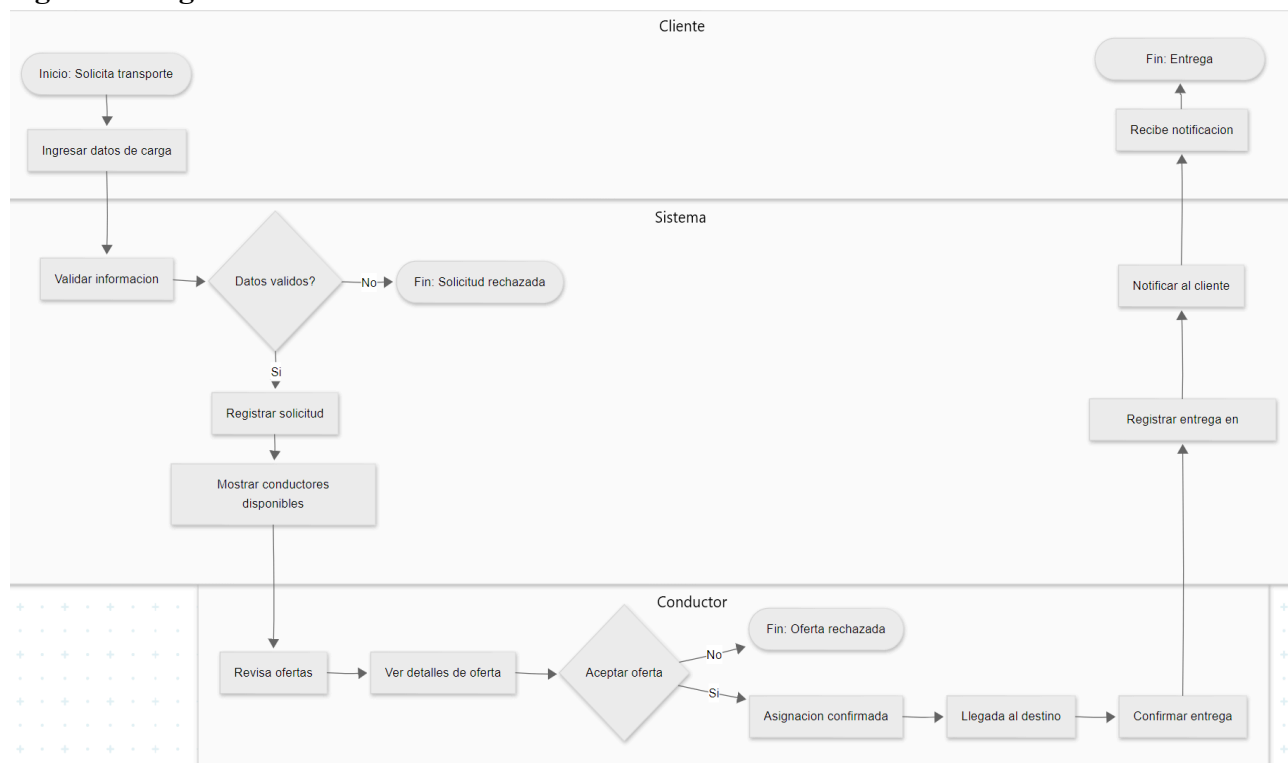
En conjunto, esta arquitectura aprovecha servicios clave de AWS para desplegar una solución robusta, escalable y fácilmente mantenible, sin la complejidad añadida de sistemas de mensajería distribuidos.

Para el desarrollo de la aplicación móvil, se ha utilizado el framework Flutter. La elección de Flutter se basa en su naturaleza multiplataforma, lo que permite crear una única base de código que puede ser utilizada tanto en dispositivos Android como iOS. Esta capacidad de desarrollo multiplataforma ahorra tiempo y recursos al equipo de desarrollo.

5.2.2 Modelado de procesos del sistema.

Durante el desarrollo del proyecto, la creación de estos diagramas fue una pieza fundamental para modelar los procesos y flujos de la plataforma, permitiendo visualizar de manera clara cómo interactúan los diferentes componentes y cómo se llevan a cabo las acciones dentro del sistema, lo que facilitó la comprensión y el desarrollo del mismo. Estos diagramas se encuentran en la sección de apéndice del proyecto, tales diagramas brindan una perspectiva más precisa de la estructura y funcionamiento de la plataforma.

Figura 6 Diagrama de actividades



5.2.3 Diseño de prototipos para la interfaz de usuario

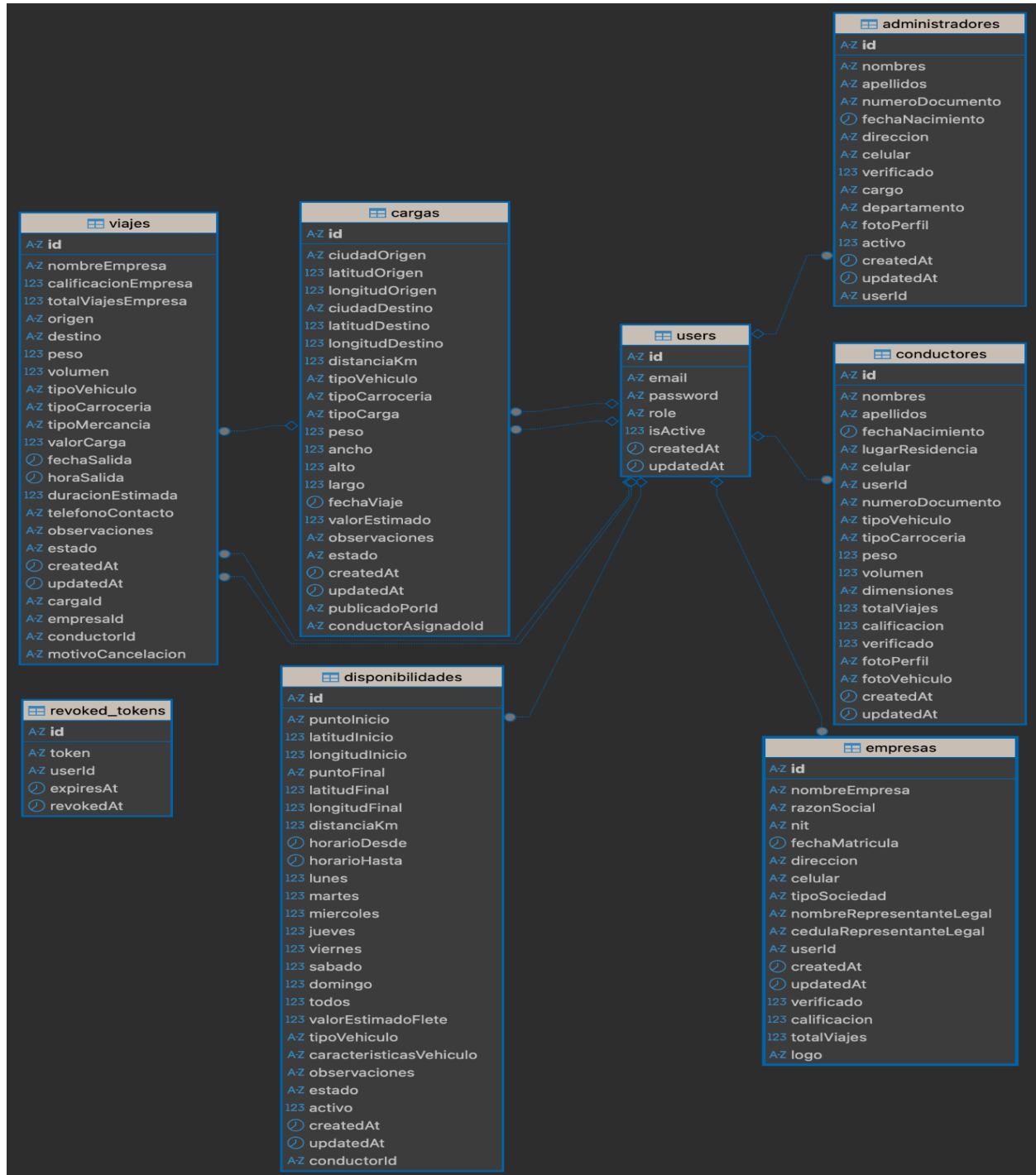
Se desarrollaron múltiples prototipos visuales utilizando Figma, esta herramienta permite colaboración en tiempo real y agilizar el diseño. Esta característica facilitó el proceso creativo y también agiliza la creación de mockups para la interfaz de usuario frontend, permitiendo visualizar el funcionamiento del sistema de una manera más clara tanto para los desarrolladores como para los interesados en el proyecto.

Los diseños se elaboraron con especial atención en la experiencia de usuario, buscando maximizar una experiencia placentera para los usuarios. Sin embargo, cabe resaltar que, debido a la limitada experiencia en diseño UX/UI, no se puede garantizar el cumplimiento de este aspecto.

Todos los prototipos mencionados estarán disponibles en la sección de apéndices del presente documento.

5.2.4 Modelo de base de datos

Figura 7 Modelo de base de datos del sistema



5.3 Implementación

En esta sección, se detallan los procesos y metodologías empleadas para el desarrollo de cada módulo del proyecto de gestión de carga pesada. Cada módulo fue diseñado y desarrollado con el objetivo de cumplir con los requisitos específicos del sistema que fueron planteados en la definición de los requerimientos funcionales y no funcionales.

5.3.1 Autenticación de los usuarios

El módulo de autenticación es uno de los componentes más importantes del sistema, ya que permite que los usuarios accedan y hagan uso de sus funcionalidades de forma segura. Este módulo se encarga de gestionar tanto la autenticación como la autorización, garantizando que solo los usuarios autorizados puedan acceder a la información y recursos del sistema.

La lógica de autenticación está integrada dentro de los servicios backend desarrollados con NestJS. Se utiliza Json Web Tokens (JWT) para la autenticación y autorización, con una política de expiración del token para limitar el tiempo de sesión activa y reducir riesgos de seguridad. El algoritmo bcrypt se emplea para el hash y la comparación segura de contraseñas, y las solicitudes a las API están protegidas mediante Bearer Tokens, asegurando que únicamente usuarios autenticados puedan acceder a los recursos expuestos.

Se implementaron AuthGuards y decoradores personalizados utilizando @nestjs/passport para verificar la autenticación y los permisos de acceso en rutas específicas del sistema. Desde la aplicación móvil, desarrollada en Flutter, se emplean paquetes como http para las peticiones a los servicios backend y flutter_secure_storage

para almacenar de forma segura el token de sesión y otros datos críticos de la sesión del usuario.

Además, el sistema cuenta con una funcionalidad de recuperación de contraseña por correo electrónico, que permite a los usuarios restablecer su contraseña mediante un enlace temporal y seguro. Esta funcionalidad se apoya en el paquete `@nestjs-modules/mailer` para el envío de correos, facilitando una experiencia segura y confiable para el manejo de credenciales olvidadas.

5.3.2 Envío de correos

5.3.2.1 Tecnologías Utilizadas. Para el desarrollo del microservicio, se utilizó el framework Nest.js debido a que es también la herramienta que se usó para el backend, este proporciona una estructura sólida y modular para la creación de aplicaciones Node.js. Además, se empleó el paquete NodeMailer con el protocolo SMTP para manejar el envío de correos electrónicos.

5.3.2.2 Proveedor de Servicio de Correo. Para el proveedor de servicio se consideraron algunas opciones como mailChimp, Brevo, entre otras. Pero finalmente se decidió utilizar Amazon Simple Email Service (SES) debido a su escalabilidad, familiaridad con el ambiente Amazon y la facilidad de integración con otros servicios en la nube de Amazon ya implementados. Amazon SES ofrece una infraestructura confiable y escalable que permite enviar correos electrónicos de manera eficiente y segura.

5.3.2.3 Plantillas de Correos Electrónicos. Además de la creación del microservicio de envío de correos, fue necesario desarrollar plantillas para los correos electrónicos. Estas plantillas están diseñadas con la extensión HBS (Handlebars) y contienen variables que se reemplazan en tiempo de ejecución con información específica del usuario o del contenido del correo. Las plantillas permiten personalizar los correos electrónicos según las necesidades del usuario, mejorando la experiencia de comunicación.

Figura 8 Olvidaste tu contraseña

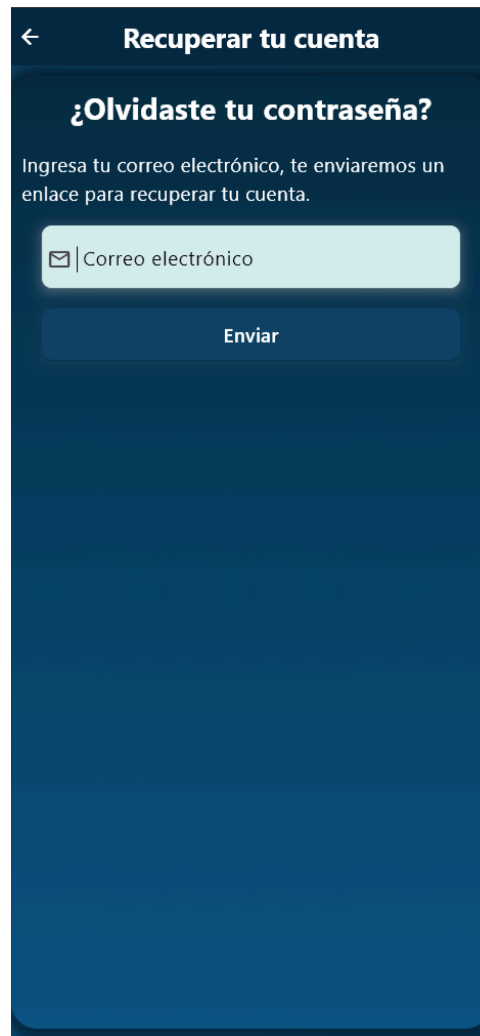


Figura 9 Restablecer tu contraseña

The image shows a mobile application interface for resetting a password. At the top left, there is a close button (an 'X' icon). The title of the screen is "Restablecer contraseña". Below the title, a light blue banner contains the instruction "Por favor digite la nueva contraseña". The form consists of two input fields: the first is labeled "Nueva contraseña" and contains the placeholder text "Digita tu nueva contraseña"; the second is labeled "Confirma nueva contraseña" and contains the placeholder text "Repite la nueva contraseña". At the bottom of the form is a dark blue button with the text "Guardar".

5.3.3 Registrarse

El módulo de registro ofrece dos opciones principales de creación de cuenta. La primera permite registrarse como conductor o empresa, requiriendo información básica pero esencial para:

- Verificación de identidad: Garantizar que todos los usuarios registrados sean personas reales o empresas registradas ante la DIAN, creando un entorno confiable.
- Seguridad comunitaria: Mitigar riesgos de malas prácticas o usos malintencionados dentro de la plataforma.
- Transparencia operativa: Brindar confianza a empresarios y otros conductores mediante perfiles validados.

Los datos solicitados siguen el principio de minimización, recolectando lo estrictamente necesario para estos propósitos, siempre cumpliendo con las normativas de protección de datos.

Figura 10 Opciones de registro



Al pulsar sobre esta alguna de las opciones opción, será redirigido a una ventana donde se le solicitarán los datos básicos para la creación del perfil ya sea de conductor o de empresa.

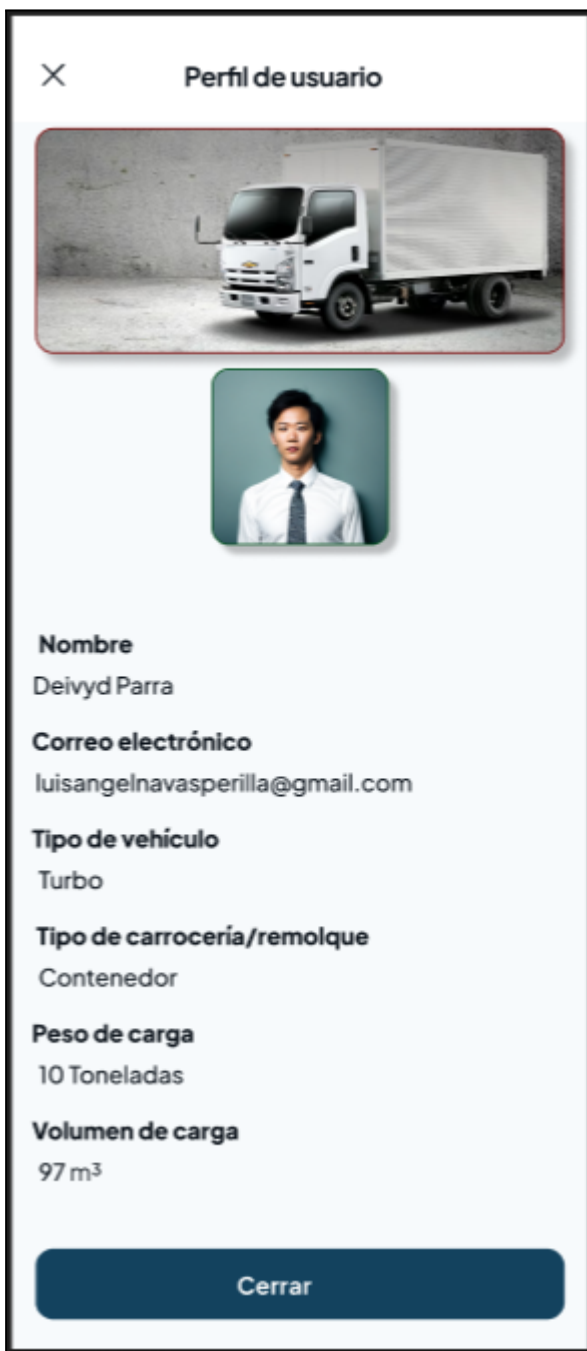
Figura 11 Registrarse como conductor

The image shows a mobile application registration screen with a dark blue background. At the top left is a back arrow icon, and at the top center is the title 'Registro'. Below the title are ten light blue input fields, each with a small icon on the left and a label on the right. The fields are: 'Nombres' (person icon), 'Apellidos' (person icon), 'Documento de identidad' (ID card icon), 'Fecha de nacimiento' (calendar icon), 'Correo electrónico' (envelope icon), 'Lugar de residencia' (location pin icon), 'Celular' (mobile phone icon), 'Contraseña' (lock icon), 'Confirmar Contraseña' (lock icon), and a 'Continuar' button. Below the button is a separator line with a circle in the middle, followed by the text '¿Ya tienes una cuenta?' and a button labeled 'Inicia sesión'.

Figura 12 Formulario de empresa

The image shows a mobile application interface for a company registration form. At the top, there is a back arrow on the left and the title "Empresa" in the center. Below the title, there are eight light blue rounded rectangular input fields stacked vertically. The fields are labeled: "Razon social", "NIT", "Fecha de matrícula", "Correo Electrónico", "Dirección", "Celular", "Tipo de sociedad", and "Contraseña". The "Contraseña" field has a small eye icon on the right side, indicating a toggle for password visibility. At the bottom of the form is a dark blue rounded rectangular button with the text "Continuar" in white.

Figura 13 Perfil de usuario



5.3.4 Crear oferta de viaje

Los perfiles de empresa tendrán la opción de crear "Nueva Oferta", en la cual deberán completar un formulario detallando las especificaciones de la carga, como el destino, peso, volumen, tipo de mercancía, fecha de entrega y valor de la carga, entre otros datos relevantes. Además, la aplicación realiza validaciones para asegurar que los datos ingresados sean correctos y completos antes de enviarlos al backend. Una vez enviada, la oferta se publica en la plataforma y se notifica a los conductores interesados, facilitando la conexión entre empresas y transportistas.

Figura 14 Formulario de oferta

← **Publicar oferta**

Lugar de origen
 Selecciona la ciudad origen

Lugar de destino
 Selecciona la ciudad destino

Tipo de vehículo
 Selecciona el tipo de vehículo

Tipo de carrocería/remolque
 Selecciona el tipo de carrocería

Peso de carga
 Indique el peso de la carga

Tipo de carga
 Indique el tipo de la carga

Volumen de carga
 Ingresa el volumen del vehículo

Fecha de viaje
 Ingresa la fecha del viaje

Valor de viaje
 Ingresa el valor del viaje

Observaciones
 Agrega cualquier detalle extra

Publicar oferta

5.3.5 Menú de ofertas

Desde el menú principal, el transportista podrá aplicar filtros sobre las diferentes características de carga, permitiendo que la búsqueda se adapte a sus necesidades y preferencias. Esta funcionalidad facilita la identificación rápida y eficiente de las ofertas de carga que mejor se ajusten a su disponibilidad y capacidad, optimizando así el proceso de selección y aceptación de viajes.

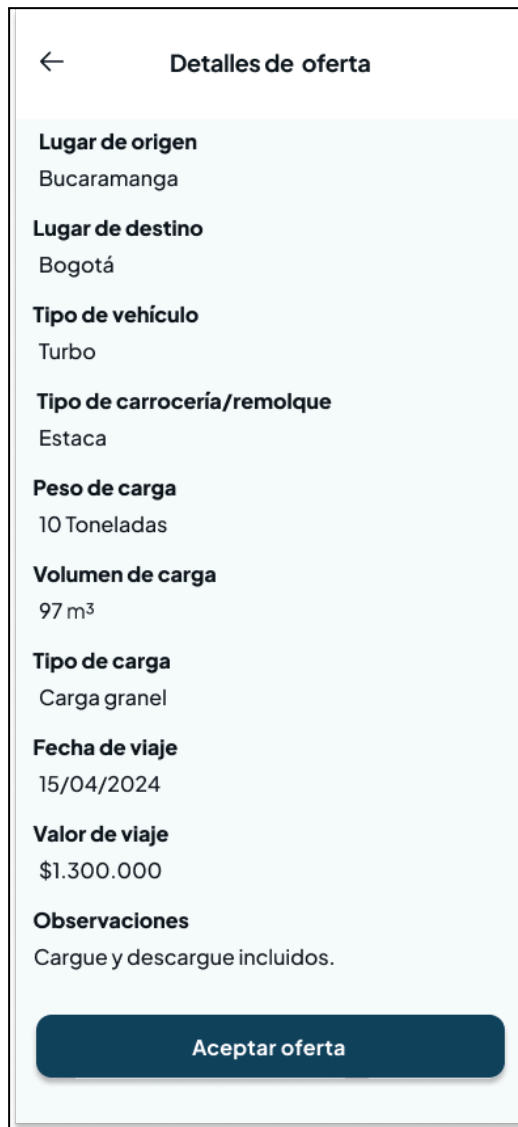
Figura 15 ofertas de viaje



5.3.6 Detalles de oferta de viaje

Los perfiles de transportista tendrán la opción de crear "ver oferta", en el menú principal, el cual permitirá ver con más detalle toda la información suministrada por parte de la empresa, de tal modo que con base en esta, pueda elegir si acepta o no la oferta en cuestión.

Figura 16 Detalles de la oferta



5.4 Pruebas

5.4.1 Pruebas unitarias

Para el desarrollo de las pruebas funcionales en la aplicación de carga pesada, se utilizó Playwright, una herramienta de pruebas de JavaScript que facilita la creación y ejecución de pruebas automatizadas. Playwright ofrece una amplia gama de funcionalidades que permiten realizar pruebas de manera eficiente y precisa, asegurando el correcto funcionamiento de la aplicación.

Las pruebas de calidad realizadas al prototipo se diseñaron en coherencia con la norma ISO/IEC 25010, la cual establece un modelo de calidad de software que incluye atributos como funcionalidad, fiabilidad, usabilidad, eficiencia, seguridad, mantenibilidad y portabilidad. Bajo este marco, se estructuraron tanto las pruebas unitarias como las pruebas funcionales automatizadas, con el objetivo de garantizar que el prototipo cumpla los estándares mínimos de calidad definidos internacionalmente. A continuación, se detallan las capacidades y características clave de Playwright que se aprovecharon en el proyecto:

5.4.2 Automatización de Pruebas Funcionales

Playwright se utilizó como herramienta gratuita para automatizar pruebas funcionales que verifican/simulan el comportamiento de un usuario en la aplicación desde la perspectiva del usuario final. Esto incluye la navegación por la interfaz de usuario, la interacción con los formularios y la validación de flujos de trabajo completos. Con Playwright, se pueden simular acciones del usuario como clics, ingreso de texto, desplazamiento y más, lo que permite verificar que la aplicación responde correctamente a estas interacciones.

5.4.3 Ejecución de Pruebas en Simultáneo

Playwright permite la ejecución de pruebas en simultáneo, lo que acelera significativamente el proceso de pruebas. Esta capacidad es especialmente útil en este proyecto, ya que permite analizar el desempeño de la aplicación simulando un gran tráfico de usuarios. La ejecución paralela de pruebas no solo ahorra tiempo, sino que también mejora la eficiencia del equipo de desarrollo al permitir la evaluación rápida y precisa del comportamiento del sistema bajo condiciones de alta demanda.

5.4.4 Generación de Informes Detallados

Playwright permite la generación de informes detallados sobre el estado de las pruebas y sus posibles fallos. Estos informes pueden incluir capturas de pantalla, registros de errores y detalles sobre las acciones realizadas durante la prueba. Los informes son esenciales para el proceso de análisis de errores y la resolución rápida de problemas, permitiendo a los desarrolladores identificar rápidamente las áreas problemáticas y tomar medidas correctivas.

5.5 Documentación

5.5.1 Documentación de herramientas

Este proyecto ha sido posible gracias al sólido respaldo documental de las tecnologías empleadas, las cuales no solo ofrecen guías claras, sino también un ecosistema bien estructurado para el desarrollo de software robusto y escalable.

- Flutter, con su documentación excepcionalmente detallada, permitió implementar interfaces multiplataforma de manera eficiente, reduciendo la curva de aprendizaje y facilitando la integración de widgets personalizados. Su enfoque en ejemplos prácticos

y actualizaciones constantes aseguró que cada componente UI cumpliera con los estándares de rendimiento y diseño.

- NestJS, por su parte, demostró ser un pilar fundamental para el backend gracias a su documentación modular y bien organizada. Desde la configuración de controladores hasta la implementación de microservicios, cada funcionalidad estuvo respaldada por guías paso a paso, haciendo que la creación de APIs RESTful y WebSockets fuera intuitiva y consistente.
- TypeScript elevó la calidad del código al proporcionar tipado estático y documentación clara sobre patrones avanzados. Su sinergia con NestJS y Flutter (a través de FFI) fue fluida, gracias a ejemplos concretos y buenas prácticas bien documentadas que evitaron errores comunes en tiempo de compilación.
- Terraform completó este ecosistema con una documentación que simplificó la infraestructura como código. Sus ejemplos para despliegues en la nube y la gestión de recursos fueron determinantes para garantizar entornos reproducibles y seguros, minimizando riesgos operativos.

La coherencia entre estas tecnologías —respaldada por documentación accesible y comunidades activas— no solo aceleró el desarrollo, sino que también aseguró que cada capa del sistema (frontend, backend e infraestructura) pudiera evolucionar manteniendo altos estándares de mantenibilidad.

5.5.2 Documentación de código

Así como la documentación exhaustiva del código fuente constituye un elemento esencial para cumplir el requerimiento no funcional de mantenibilidad, de modo que se asegura

que las funcionalidades del sistema puedan ser entendidas, corregidas y ampliadas por nuevos desarrolladores sin pérdida de tiempo ni ambigüedad. Por ejemplo, la documentación del módulo de registro e inicio de sesión de usuarios (RF1) y de la gestión de solicitudes de transporte (RF2, RF3, RF4) permite garantizar que dichas funcionalidades puedan evolucionar sin comprometer la estabilidad del sistema.

Este esfuerzo documental contribuye al requerimiento no funcional de portabilidad, dado que permite reproducir y adaptar el sistema en diferentes entornos de ejecución con instrucciones claras. En términos de seguridad, contar con lineamientos bien descritos facilita implementar controles coherentes en la autenticación, cifrado de datos y manejo de tokens (vinculado con el RNF de Seguridad). Finalmente, la documentación también apoya el cumplimiento del RNF de compatibilidad, pues describe cómo integrar los módulos internos con servicios externos (como AWS o Firebase) y cómo mantener interoperabilidad con navegadores y dispositivos móviles.

5.6 Despliegue

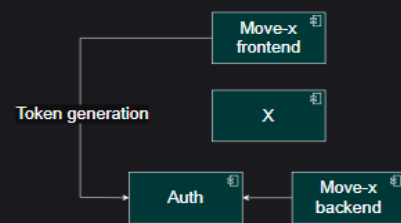
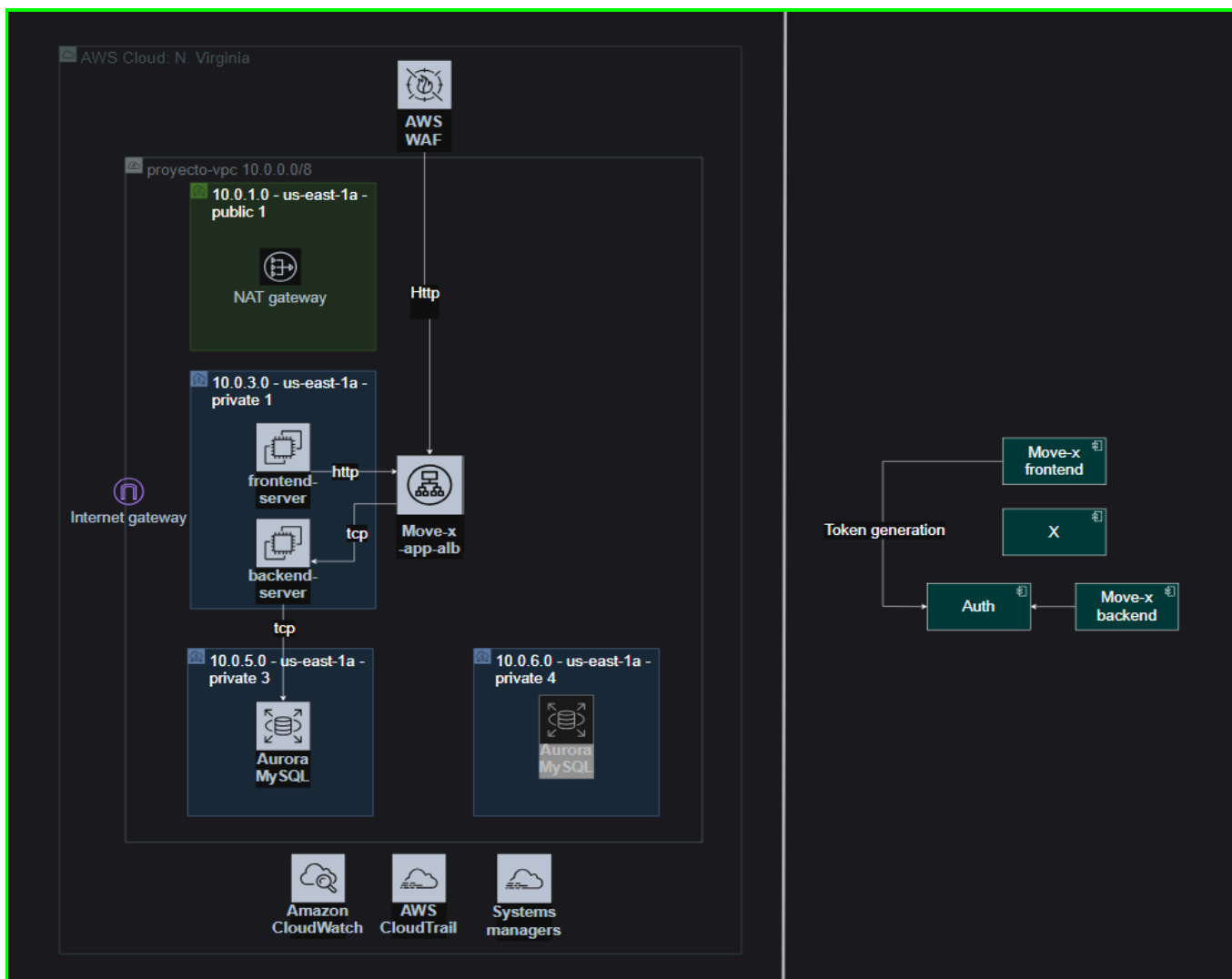
El despliegue del sistema mediante contenedores Docker, alojados en Amazon Web Services (AWS), responde de manera directa a múltiples requerimientos no funcionales:

- **Escalabilidad (RNF):** al usar Docker, los microservicios del backend (NestJS) y frontend móvil (Flutter) pueden replicarse en instancias adicionales cuando aumenta la demanda, soportando un incremento de usuarios simultáneos (5,000 o más) sin degradar el rendimiento.

- **Disponibilidad (RNF):** el despliegue en EC2, junto con RDS para la base de datos, garantiza una disponibilidad del 99.9% y mecanismos de recuperación ante fallos, lo cual asegura la continuidad de funciones críticas como la **publicación de ofertas de transporte (RF2)** y la **finalización de entregas (RF4)**.
- **Rendimiento (RNF):** Docker aísla las dependencias y evita conflictos de librerías, lo que permite respuestas rápidas y consistentes en procesos sensibles como la **autenticación (RF1)** y el **registro de calificaciones (RF5)**.
- **Seguridad (RNF):** AWS ofrece políticas de IAM que aseguran que solo usuarios autorizados accedan a las funcionalidades, alineándose con la necesidad de proteger los datos sensibles de clientes y transportistas.
- **Compatibilidad y Portabilidad (RNF):** los contenedores Docker garantizan que el sistema pueda ejecutarse en entornos distintos (nube, local, híbrido) con el mismo comportamiento. Esto resulta clave para que el prototipo pueda ser migrado o ampliado en otras plataformas sin reescribir código.

De este modo, el despliegue en AWS con Docker no solo asegura robustez técnica, sino que también brinda soporte integral a los requerimientos funcionales (RF1–RF5), garantizando que cada operación crítica esté respaldada por infraestructura confiable y escalable.

Figura 17 Infraestructura



5.6.1 Creación de imágenes Docker

Para contenerizar los diferentes componentes del sistema, se creó una imagen Docker para cada uno de los servicios que conforman el backend y frontend de la aplicación. Estas imágenes permiten encapsular todas las dependencias necesarias para su ejecución, garantizando portabilidad y facilidad de despliegue en el entorno de producción.

El backend fue desarrollado utilizando Node.js, por lo que se utilizó una imagen base oficial de Node en el Dockerfile. En este archivo se definieron instrucciones como la copia del código fuente, instalación de dependencias mediante npm install, exposición del puerto correspondiente y ejecución del servidor con el comando adecuado.

Por su parte, el frontend fue desarrollado utilizando Flutter, y se creó una imagen Docker que contiene el entorno necesario para compilar y servir la aplicación como una web app. El proceso incluía la compilación del proyecto con flutter build web y la posterior copia de los archivos generados a un servidor web ligero como nginx, desde donde se serviría la interfaz al usuario.

Una vez generadas las imágenes, se etiquetaron con su respectiva versión y se subieron a un repositorio privado de Amazon ECR (Elastic Container Registry).

5.6.2 Despliegue de infraestructura y sistema en EC2 utilizando Terraform

Para el despliegue del sistema se utilizó el servicio EC2 de Amazon Web Services (AWS), aprovechando su flexibilidad y escalabilidad. Toda la infraestructura, excepto la creación del repositorio en ECR, fue aprovisionada utilizando Terraform, lo cual permitió mantener una configuración reproducible y automatizada.

La adopción de Terraform como herramienta de infraestructura como código refuerza varios aspectos fundamentales de los requerimientos no funcionales definidos:

- **Disponibilidad (RNF):** permite la creación automatizada de entornos redundantes y balanceados, asegurando que los servicios como la **autenticación (RF1)** o la **gestión**

de solicitudes (RF2, RF3, RF4) no se vean interrumpidos por fallos de infraestructura.

- **Escalabilidad (RNF):** Terraform facilita la provisión dinámica de recursos adicionales (instancias EC2, nodos de base de datos RDS) cuando la demanda de usuarios aumenta, alineándose con el objetivo de soportar incrementos del 50% en la carga sin pérdida de rendimiento.
- **Seguridad (RNF):** al versionar y auditar la infraestructura, se asegura trazabilidad en la creación de redes, políticas de acceso y cifrado en reposo y tránsito, lo cual es indispensable para proteger datos de usuarios y cargas transportadas.
- **Mantenibilidad (RNF):** la configuración declarativa de Terraform facilita replicar la infraestructura en distintos ambientes (desarrollo, pruebas y producción), reduciendo errores humanos y asegurando consistencia en la operación.
- **Portabilidad (RNF):** al describir la infraestructura como código, se elimina la dependencia exclusiva de un entorno, permitiendo que la solución pueda ser migrada a otras nubes públicas o entornos on-premises si el proyecto lo requiere.

De esta manera, la infraestructura como código asegura que los **procesos funcionales principales (RF1–RF5)** estén sostenidos por una base técnica confiable, escalable y fácilmente auditable. Esto proporciona un mayor respaldo a las decisiones arquitectónicas del proyecto y asegura que las funcionalidades críticas se ejecuten en un entorno de calidad alineado con los atributos de rendimiento, seguridad y disponibilidad definidos previamente.

Como parte de la arquitectura se creó una VPC personalizada, dentro de la cual se configuraron subredes públicas y privadas, un Internet Gateway y un NAT Gateway. Esta configuración permite segmentar y controlar el tráfico entre los diferentes recursos de la infraestructura.

Se utilizaron dos instancias EC2 tipo t2.small, así como una base de datos en Amazon RDS para almacenar la información persistente del sistema. Para distribuir el tráfico entrante hacia las aplicaciones backend y frontend, se implementó un balanceador de carga de aplicación (ALB). Esta configuración asegura una distribución eficiente del tráfico y mejora la disponibilidad del sistema.

Una vez desplegadas las instancias, se optó por utilizar AWS Systems Manager para acceder a ellas de forma segura, eliminando la necesidad de manejar archivos .pem. Esta alternativa facilita la administración remota y mejora las prácticas de seguridad.

Las imágenes de Docker necesarias para el despliegue fueron descargadas desde un repositorio privado en Amazon ECR, el cual fue configurado previamente. Cabe destacar que la creación de este repositorio fue el único paso realizado manualmente desde la consola de AWS; el resto del proceso fue automatizado mediante Terraform.

5.6.3 Configuración de red y almacenamiento

Para asegurar la conectividad de los servicios desplegados, se configuró un grupo de seguridad que permite el tráfico entrante a través de puertos TCP del 8000 al 9000, asegurando el acceso a los servicios expuestos por los contenedores.

En cuanto al almacenamiento, se decidió aumentar la capacidad predeterminada de las instancias EC2 de 8 GB a 16 GB, con el fin de disponer del espacio suficiente para almacenar imágenes Docker y otros recursos necesarios durante la ejecución del sistema.

5.6.4 Instalación de Docker y despliegue de contenedores

Una vez que las instancias estuvieron operativas, se procedió con la instalación de Docker y Docker Compose. Esta herramienta facilitó la orquestación de los distintos servicios del sistema. Se utilizó un archivo `docker-compose.yml` modificado (sin la instrucción `build`) para levantar los contenedores directamente desde las imágenes descargadas de ECR.

5.6.5 Construcción, carga y ejecución de contenedores Docker desde Amazon ECR

Para desplegar los servicios del sistema, se utilizaron contenedores Docker previamente contruidos y almacenados en Amazon Elastic Container Registry (ECR). El proceso general consistió en construir la imagen localmente, etiquetarla adecuadamente, subirla al repositorio en ECR y luego desplegarla en las instancias EC2.

1. Autenticación en el repositorio ECR

Primero, se debe autenticar el cliente Docker con ECR utilizando el siguiente comando:

- a. `bash`
- b. Copiar
- c. Editar

d. `aws ecr get-login-password --region <region> | docker login --username AWS --password-stdin <aws_account_id>.dkr.ecr.<region>.amazonaws.com`

Reemplazar <region> y <aws_account_id> por los valores correspondientes a tu cuenta y región.

2. Construcción y publicación de la imagen Docker

Para construir la imagen y subirla al repositorio de ECR:

a. `bash`

b. Copiar

c. Editar

d. `docker build -t <image-name>:<version> . docker tag <image-name>:<version>`

`<aws_account_id>.dkr.ecr.<region>.amazonaws.com/<repository-name>:<version>`

e. `docker push`

`<aws_account_id>.dkr.ecr.<region>.amazonaws.com/<repository-name>:<version>`

Asegúrate de que el repositorio <repository-name> ya haya sido creado previamente en ECR.

3. Descarga y ejecución de la imagen en EC2

Una vez dentro de la instancia EC2, se vuelve a autenticar con ECR, se descarga la imagen y se ejecuta el contenedor:

a. `bash`

b. Copiar

- c. Editar
- d. `aws ecr get-login-password --region <region> | docker login --username AWS --password-stdin <aws_account_id>.dkr.ecr.<region>.amazonaws.com`
- e. `docker pull <aws_account_id>.dkr.ecr.<region>.amazonaws.com/<repository-name>:<version>`
- f. `docker run -d -p <host-port>:<container-port> --name <container-name> <aws_account_id>.dkr.ecr.<region>.amazonaws.com/<repository-name>:<version>`

Los puertos y nombres de los contenedores deben adaptarse a las necesidades de cada servicio.

5.6.6 Verificación del sistema

Para comprobar que los servicios se encuentran correctamente desplegados y accesibles, se utilizaron herramientas como Postman, realizando peticiones a la dirección DNS asignado al balanceador de carga. Esto permitió validar el correcto funcionamiento tanto del backend como del frontend en un entorno simulado de producción.

5.6.7 Costos de operación en la nube

A continuación, se presenta un desglose estimado de los costos y servicios necesarios para el funcionamiento de la aplicación en un entorno de producción utilizando los servicios de **Amazon Web Services (AWS)**. La siguiente imagen refleja la composición del sistema y la inversión:

Figura 18 Costos Operacionales

Componente	Tipo / Tamaño	Precio por hora (USD)	Precio mensual (USD)	Descripción
EC2 – Frontend	t2.small	\$0.023	\$ 20	Instancia dedicada al frontend del sistema.
EC2 – Backend	t2.small	\$0.023	\$ 20	Instancia dedicada al backend del sistema.
Aurora MySQL	db.t3.small	\$0.05	\$36.5	Base de datos administrada con alta disponibilidad.
Application Load Balancer (ALB)	-	\$0.0252/h + LCUs	\$ 25	Balancea el tráfico HTTP hacia frontend/backend.
NAT Gateway	-	\$0.045/h + \$0.045/GB	\$37-40	Permite que instancias privadas tengan salida a internet.
Almacenamiento EC2 (EBS)	50 GB gp3	\$0.10/GB	\$ 5	Discos de las instancias EC2.
Almacenamiento Aurora	100 GB	\$0.10/GB	\$ 10	Espacio para la base de datos.
Backups Aurora	100 GB	\$0.021/GB	\$ 2	Respaldos automáticos de la base de datos.
Monitoreo y Logs (CloudWatch, CloudTrail, SSM)	-	-	\$10-20	Logs, métricas y auditoría básica.
Total aproximado: \$155 – \$170 USD / mes				

6. Conclusiones

Se logró el objetivo de diseñar e implementar un prototipo de sistema de software para la gestión de carga pesada, siguiendo una arquitectura que podrá adaptarse a los cambios y escalabilidad. Este desarrollo tuvo como base el análisis llevado a cabo de los datos obtenidos en las encuestas y también en base a la información proporcionada por una persona que cuenta con una larga experiencia en el rubro, lo que permitió crear una solución adaptada para las necesidades específicas del sector. El resultado es un prototipo que facilita significativamente la comunicación y coordinación entre empresas y conductores, optimizando con ello la eficiencia operativa y ofreciendo un recurso logístico de gran valor.

Con estos resultados, el prototipo no solo cumple con los objetivos iniciales, sino que también establece una base para que empresas y conductores de transporte de carga pesada mejoren y expandan la comunicación y contratación de servicios. Esto posiciona a la herramienta como un recurso indispensable para el sector automovilístico.

7. Recomendaciones

Es importante tener en cuenta que los diseños de interfaz de usuario (UI) implementados en los prototipos son provisionales y están sujetos a cambios. Estos diseños deben ser validados por un especialista en experiencia de usuario (UX) para asegurar que cumplan con los estándares de usabilidad y accesibilidad. A continuación, se detallan las recomendaciones para mejorar y finalizar el diseño de la interfaz de usuario:

7.1 Validación por Especialistas en UX

La validación por parte de un especialista en UX es crucial para identificar áreas de mejora en el diseño de la interfaz. Este experto puede proporcionar una evaluación profesional sobre la facilidad de uso, la navegación intuitiva y la coherencia visual del prototipo. Su feedback es esencial para asegurar que la interfaz sea amigable y eficiente para los usuarios finales.

7.2 Documentación y Guías de Estilo

Es importante mantener una documentación detallada del diseño de la interfaz de usuario, incluyendo guías de estilo, patrones de diseño y directrices de uso. Esta documentación sirve como referencia para el equipo de desarrollo y asegura la coherencia en futuras actualizaciones y mejoras del sistema.

7.3 Monitoreo Continuo

Una vez que el sistema esté en producción, es importante monitorear continuamente la interfaz de usuario para identificar áreas de mejora. Herramientas de análisis y feedback de usuarios pueden proporcionar información valiosa sobre el rendimiento de la interfaz y ayudar a identificar problemas emergentes.

7.4 Evolución de diseño visual

A continuación se presentan unos ejemplos que pretenden dar una comparativa de la evolución que tuvo el diseño del Frontend desde sus primeras versiones hasta el diseño final implementado. En un inicio, el diseño fue concebido con un enfoque básico y funcional, orientado a cubrir las necesidades esenciales de navegación y visualización de información.

El diseño inicial, en comparación con el diseño final estuvo adaptado en función de la retroalimentación de usuarios, este nuevo diseño integra un enfoque más moderno y adaptable, incorporando principios de diseño responsivo que permiten su uso en distintos dispositivos móviles. Además, se añadieron elementos visuales como íconos representativos, colores contrastantes y una mejor distribución de los espacios, lo cual aporta claridad y dinamismo a la interacción.

Figura 19 Prototipo Login Original



Figura 20 Implementación Login final

Para esta parte inicial del desarrollo se hicieron mínimos ajustes relacionados al logo de la app y el nombre, también se añadieron íconos a los campos input con el fin de complementar la descripción de cada campo, además se añadió una opción al campo de contraseñas que permite que el usuario cambie entre un texto oculto o uno legible.

Figura 21 Prototipo Registro Original

Este prototipo muestra una pantalla de registro con el título 'Empresa'. Incluye los siguientes campos de texto: Razon social, NIT, Fecha de matrícula, Correo Electrónico, Dirección, Celular, Tipo de sociedad, y Contraseña (con un ícono de ojo para alternar visibilidad). Un botón 'Continuar' está ubicado al final de la lista de campos.

Esta implementación final muestra una pantalla de registro con el título 'Registro'. Incluye los siguientes campos de texto: Nombres, Apellidos, Fecha de nacimiento, Correo electrónico, Lugar de residencia, Celular, Contraseña (con un ícono de ojo para alternar visibilidad), y Confirmar Contraseña (con un ícono de ojo para alternar visibilidad). Un botón 'Continuar' está ubicado al final de la lista de campos. Debajo del botón, hay un enlace '¿Ya tienes una cuenta? Inicia sesión'.

Figura 22 Implementación registro final

En esta pantalla se ajustaron los colores para darle una identidad, además de añadir también las mejoras visuales a los campos como los íconos, manejo de contraseñas de forma visual y una opción clara de volver al Login en caso de así ser necesario.

Figura 23 Prototipo restablecer contraseña

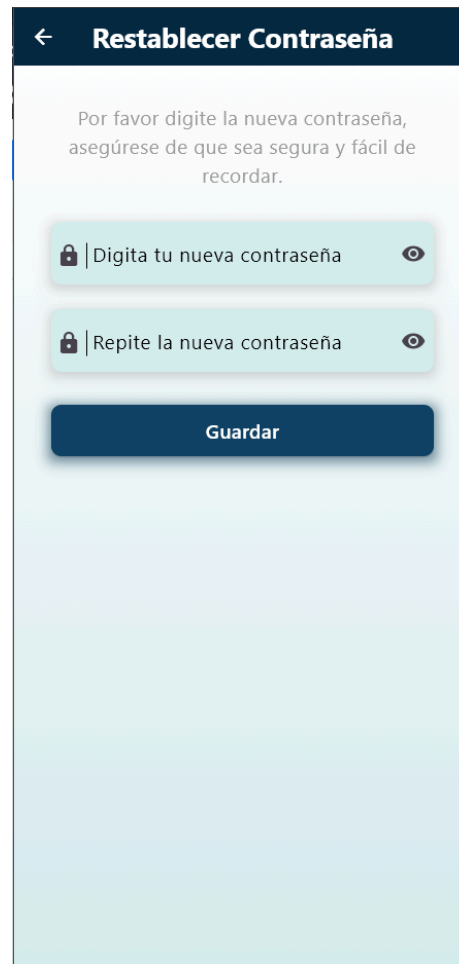
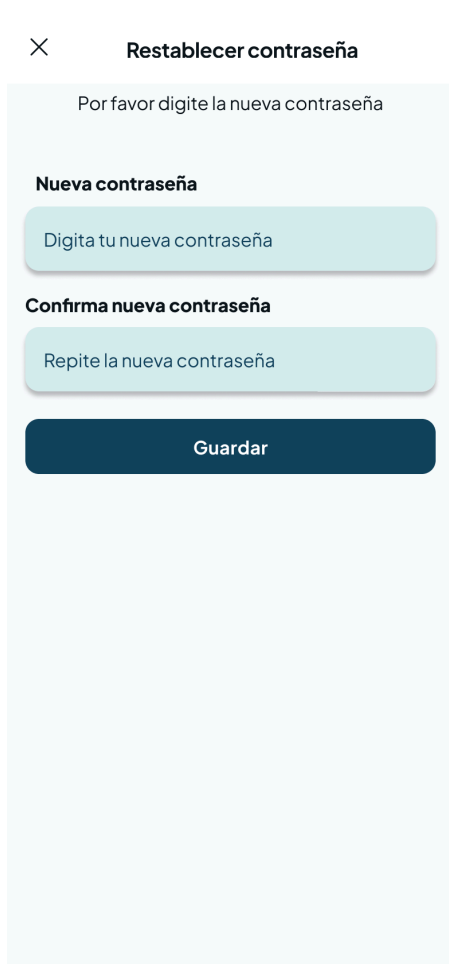


Figura 24 Implementación restablecer contraseña

El diseño de esta pantalla tuvo pocos cambios; el más notorio fue la modificación del color en la barra superior, con el fin de mantener coherencia en la paleta de colores que es identidad de la aplicación. Adicionalmente, se realizaron pequeños ajustes en los campos de texto, añadiendo íconos y un mensaje superior que recuerda a los usuarios la importancia de utilizar una contraseña segura y fácil de memorizar. Estos cambios responden a las características del público objetivo, el cual en gran parte está compuesto por personas de edad avanzada, para quienes la gestión de contraseñas puede representar una dificultad.

Figura 25 Prototipo publicar oferta

The wireframe shows a form titled "Publicar oferta" with a back arrow. It contains the following sections and input fields:

- Lugar de origen:** "Selecciona la ciudad origen"
- Lugar de destino:** "Selecciona la ciudad destino"
- Tipo de vehículo:** "Selecciona el tipo de vehículo"
- Tipo de carrocería/remolque:** "Selecciona el tipo de carrocería"
- Peso de carga:** "Indique el peso de la carga"
- Tipo de carga:** "Indique el tipo de la carga"
- Volumen de carga:** "Ingresa el volumen del vehículo"
- Fecha de viaje:** "Ingresa la fecha del viaje"
- Valor de viaje:** "Ingresa el valor del viaje"
- Observaciones:** "Agrega cualquier detalle extra"

A "Publicar oferta" button is located at the bottom.

The implementation shows a form titled "Publicar Carga" with a back arrow. It contains the following sections and input fields:

- Ciudad de Origen:** Dropdown menu with "Bogotá" selected.
- Ciudad de Destino:** Dropdown menu.
- Tipo de Vehículo:** Dropdown menu with "NPR" selected.
- Tipo de Carrocería:** Dropdown menu with "Furgón Refrigerado" selected.
- Tipo de Carga:** Dropdown menu.
- Peso de la Carga (kg):** Input field with a scale icon.
- Dimensiones del Volumen (metros):** Three buttons: "Ancho", "Alto", "Largo".
- Fecha del Viaje:** Input field with a calendar icon.
- Valor del Viaje (COP):** Input field with a dollar sign icon.
- Observaciones:** Input field with a document icon.

A "Publicar Oferta" button is located at the bottom.

Figura 26 Implementación publicar oferta

Referencias Bibliográficas

- Nowports. (s. f.). *Las mejores apps de logística en Latinoamérica*. Recuperado de <https://blog.nowports.com/es/las-mejores-apps-de-logistica-en-latinoamerica>
- Truckstop. (s. f.). *Load Board and Freight Management Software*. Recuperado de <https://truckstop.com/product/load-board/> Truckstop
- CargaYa, (s. f.) <https://www.cargaya.cl>
- GeeksforGeeks. (2025). *Introduction to Dart Programming Language*. Recuperado de <https://www.geeksforgeeks.org/introduction-to-dart-programming-language/>
- Wikipedia. (s. f.). *Flutter (software)*. Recuperado de https://en.wikipedia.org/wiki/Flutter_%28software%29
- Android Developers. (s. f.). *Meet Android Studio*. Recuperado de <https://developer.android.com/studio/intro>
- Apple Developer. (s. f.). *Xcode Resources*. Recuperado de <https://developer.apple.com/xcode/resources/>
- Git. (s. f.). *About Version Control* [documentación oficial]. Recuperado de <https://git-scm.com/book/en/v2/Getting-Started-About-Version-Control>
- Atlassian. (s. f.). *Git merge* [tutorial]. Recuperado de <https://www.atlassian.com/git/tutorials/using-branches/git-merge>
- GeeksforGeeks. (s. f.). *What Is Git Version Control?* Recuperado de <https://www.geeksforgeeks.org/what-is-git-version-control/>
- Oracle. (s. f.). *What is MySQL?*. Oracle. Recuperado de <https://www.mysql.com/what-is-mysql/>
- IBM. (2025). *What Is Terraform?* Recuperado de <https://www.ibm.com/think/topics/terraform>
- HashiCorp. (s. f.). *What is Terraform?* Recuperado de <https://developer.hashicorp.com/terraform>

- TablePlus. (s. f.). *About TablePlus / Supported Databases*. Recuperado de <https://docs.tableplus.com/overview>
- TablePlus. (2018). *TablePlus: A MySQL Client GUI that runs fast on Mac and Windows*. Recuperado de <https://tableplus.com/blog/2018/09/mysql-client-gui-for-mac-windows.html>
- Docker. *What is Docker?* Recuperado de la documentación oficial: <https://docs.docker.com/get-started/docker-overview/>
- Amazon Web Services. (2024, 27 de agosto). *Overview of Amazon Web Services*. Recuperado de <https://docs.aws.amazon.com/whitepapers/latest/aws-overview/introduction.html>
- Amazon Web Services. (s. f.). *What is AWS — About AWS*. Recuperado de <https://aws.amazon.com/about-aws/whats-new/application-integration/>
- B, G., & B, G. (2023). Qué es GitHub y cómo empezar a usarlo. *Tutoriales Hostinger*. <https://www.hostinger.es/tutoriales/que-es-github>
- Benedet, M. (2019). ¿Qué son los microservicios? Definición, características y retos. #ADN CLOUD. <https://blog.mdcloud.es/que-son-los-microservicios-definicion-caracteristicas-y-retos/>
- Bigelow, S. J., & Courtemanche, M. (2021). Docker. *ComputerWeekly.es*. <https://www.computerweekly.com/es/definicion/Docker>
- Chacón, J. L. (2022). TypeScript: qué es, diferencias con JavaScript y por qué aprenderlo. *Profile Software Services*. <https://profile.es/blog/que-es-typescript-vs-javascript/>
- Clean Code: código limpio, ¿qué es?* (n.d.). <https://www.hostgator.mx/blog/clean-code-codigolimpio/>
- Documentation | NestJS - A progressive Node.js framework*. (n.d.). Documentation | NestJS - a Progressive Node.js Framework. <https://docs.nestjs.com/openapi/introduction>

Epitech Spain. (2023). NestJS: ¿Qué es y por qué es interesante este framework Node.js? *Epitech Spain*. <https://www.epitech-it.es/nestjs-que-es/>

Introducción a Android Studio. (n.d.). *Android Developers*.
<https://developer.android.com/studio/intro?hl=es-419>

JavaScript With Syntax For Types. (n.d.). <https://www.typescriptlang.org/>

Kinsta. (2022, July 25). *¿Qué es Nest.js? Un Vistazo al Framework Ligerito de JavaScript*. Kinsta®. <https://kinsta.com/es/base-de-conocimiento/nestjs/>

NestJS - A progressive Node.js framework. (n.d.). NestJS - a Progressive Node.js Framework. <https://nestjs.com/>

Presta, M. (2022). Flutter vs Dart: ¿cuál es mejor? *Back4App Blog*.
<https://blog.back4app.com/es/flutter-vs-dart-cual-es-mejor/>

¿Qué es Flutter? - Explicación de la aplicación Flutter - AWS. (n.d.). Amazon Web Services, Inc. <https://aws.amazon.com/es/what-is/flutter/>

¿Qué es la computación en la nube? | Google Cloud | Google Cloud. (n.d.). Google Cloud. <https://cloud.google.com/learn/what-is-cloud-computing?hl=es-419>

Simões, C. (2021). ¿Qué es TypeScript, y por qué utilizarlo? *Blog ITDO - Agencia De Desarrollo Web, APPs Y Marketing En Barcelona*.
<https://www.itdo.com/blog/que-estypescript-y-por-que-utilizarlo/>

TablePlus. (2021, August 24). Capterra.
<https://www.capterra.es/software/170642/tableplus>

Trustpilot. (2023, April 27). *Wyzant Opiniones*.
<https://es.trustpilot.com/review/wyzant.com?languages=all>

What is Amazon EC2? - Amazon Elastic Compute Cloud. (n.d.).
<https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/concepts.html>

Systems and software engineering — Systems and software Quality Requirements and Evaluation (SQuaRE) — Product quality model (Modelo de calidad del producto para software/Sistemas de TIC) <https://www.iso.org/standard/78176.html>

Systems and software engineering — Systems and software Quality Requirements and Evaluation (SQuaRE) — System and software quality models (la versión anterior de 25010) <https://www.iso.org/standard/35733.html>

Ergonomics of human-system interaction — Part 210: Human-centred design for interactive systems (Diseño centrado en el usuario)
<https://www.iso.org/standard/77520.html>

Guidelines for protection of personally identifiable information (PII) in public clouds acting as PII processors (Privacidad de la información personal en la nube)
<https://www.iso.org/standard/27018.html>

- [Google Codelabs](#)
- <https://firebase.google.com/>
- [Documentation | NestJS - A progressive Node.js framework](#)
- [TypeScript: JavaScript With Syntax For Types. \(typescriptlang.org\)](#)

Apéndices

Apéndice A Mockups

A continuación, se presentan las diferentes vistas desarrolladas inicialmente como parte del proceso de planeación y maquetación del proyecto. Estos mockups sirven como guía visual para el diseño de la interfaz de usuario y ayudan a definir la estructura y funcionalidad de cada pantalla de la aplicación.

Figura 27 Prototipo para pantalla de ingreso



Figura 28 Prototipo pantalla de ingreso con mensaje de error



Figura 29 Prototipo para pantalla de opción de registro



Figura 30 Prototipo para pantalla de registro empresa

← Empresa

Razon social

NIT


Fecha de matrícula

Correo Electrónico

Dirección

Celular

Tipo de sociedad

Contraseña 

Continuar

Detailed description: This is a mobile application prototype for a company registration screen. The screen has a light blue background. At the top, there is a back arrow on the left and the title 'Empresa' in the center. Below the title, there are eight rounded rectangular input fields, each with a light blue background and a thin border. The fields are labeled: 'Razon social', 'NIT', 'Fecha de matrícula', 'Correo Electrónico', 'Dirección', 'Celular', 'Tipo de sociedad', and 'Contraseña'. The 'Contraseña' field includes a small eye icon on the right side to toggle visibility. At the bottom of the form is a dark blue button with the text 'Continuar' in white.

Figura 31 Prototipo para pantalla de registro transportista

Registro

←

👤 | Nombres

👤 | Apellidos

🪪 | Documento de identidad

📅 | Fecha de nacimiento

✉️ | Correo electrónico

📍 | Lugar de residencia

📱 | Celular

🔒 | Contraseña

🔒 | Confirmar Contraseña

Continuar

— ○ —

¿Ya tienes una cuenta?

Inicia sesión

Detailed description: This is a mobile app prototype for a transporter registration screen. The screen has a dark blue background. At the top, there is a back arrow and the title 'Registro'. Below the title are ten light blue input fields, each with a small icon and a label: 'Nombres' (person icon), 'Apellidos' (person icon), 'Documento de identidad' (ID card icon), 'Fecha de nacimiento' (calendar icon), 'Correo electrónico' (envelope icon), 'Lugar de residencia' (location pin icon), 'Celular' (mobile phone icon), 'Contraseña' (lock icon), 'Confirmar Contraseña' (lock icon), and a 'Continuar' button. At the bottom, there is a radio button, the text '¿Ya tienes una cuenta?', and a link 'Inicia sesión'.

Figura 32 Prototipo registro de vehículo

←
Registra tu vehículo

Placa del vehículo

Ingresa la placa del vehículo

Marca del vehículo

Selecciona la marca del vehículo

Modelo del vehículo

Selecciona el modelo del vehículo

Color del vehículo

Selecciona el color del vehículo

Tipo de vehículo

Selecciona el tipo de vehículo

Tipo de remolque

Selecciona el tipo de remolque

Capacidad de carga

Ingresa la capacidad del vehículo

Volumen del Vehículo

Ingresa el volumen del vehículo

Foto del vehículo

Toma o sube una foto clara

Continuar

Figura 33 Prototipo perfil de usuario

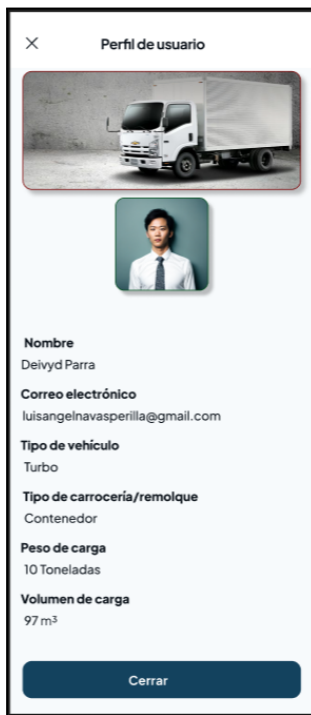


Figura 34 Prototipo recuperar cuenta



Figura 35 Prototipo restablecer contraseña

✕ **Restablecer contraseña**

Por favor digite la nueva contraseña

Nueva contraseña

Digita tu nueva contraseña

Confirma nueva contraseña

Repite la nueva contraseña

Guardar

Figura 36 Prototipo publicar oferta

← **Publicar oferta**

Lugar de origen
 Seleciona la ciudad origen

Lugar de destino
 Selecciona la ciudad destino

Tipo de vehículo
 Selecciona el tipo de vehículo

Tipo de carrocería/remolque
 Selecciona el tipo de carrocería

Peso de carga
 Indique el peso de la carga

Tipo de carga
 Indique el tipo de la carga

Volumen de carga
 Ingresa el volumen del vehículo

Fecha de viaje
 Ingresa la fecha del viaje

Valor de viaje
 Ingresa el valor del viaje

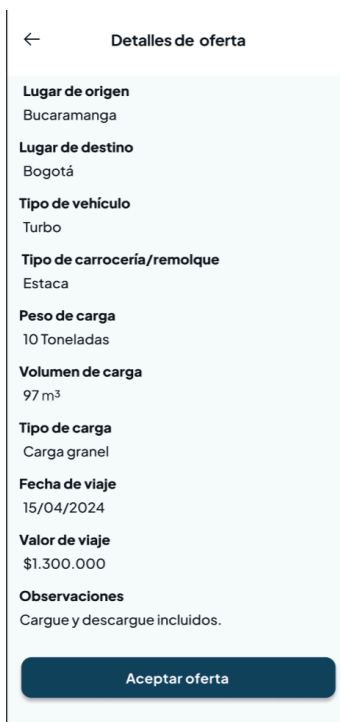
Observaciones
 Agrega cualquier detalle extra

Publicar oferta

Figura 37 Prototipo ofertas de viajes



Figura 38 Prototipo detalles de oferta



Apéndice B Pantallas finales

En este apéndice se muestra el diseño final de la maquetación de la aplicación. Estas interfaces representan la versión definitiva del proyecto y sirven como referencia visual para comprender la estructura, navegación y funcionalidad de cada pantalla.

Figura 39 Login

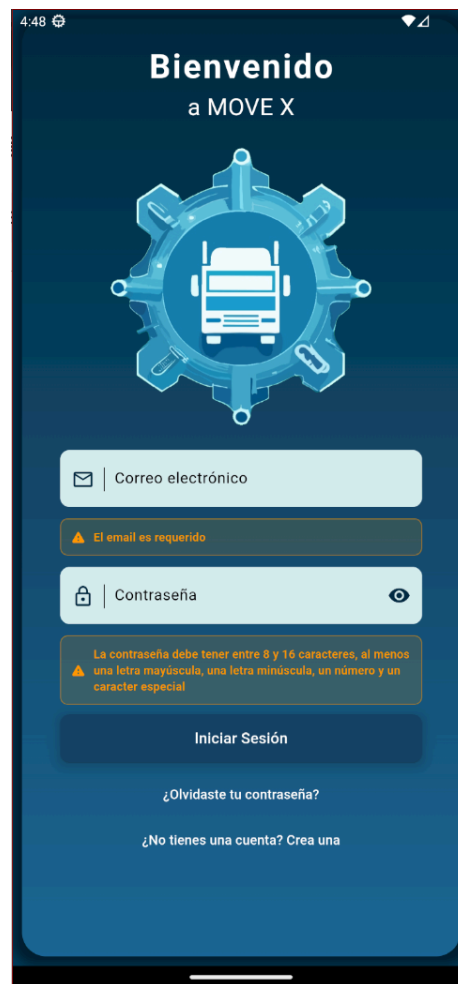


Figura 40 Login con manejo de errores

Figura 41 Olvidaste tu Contraseña

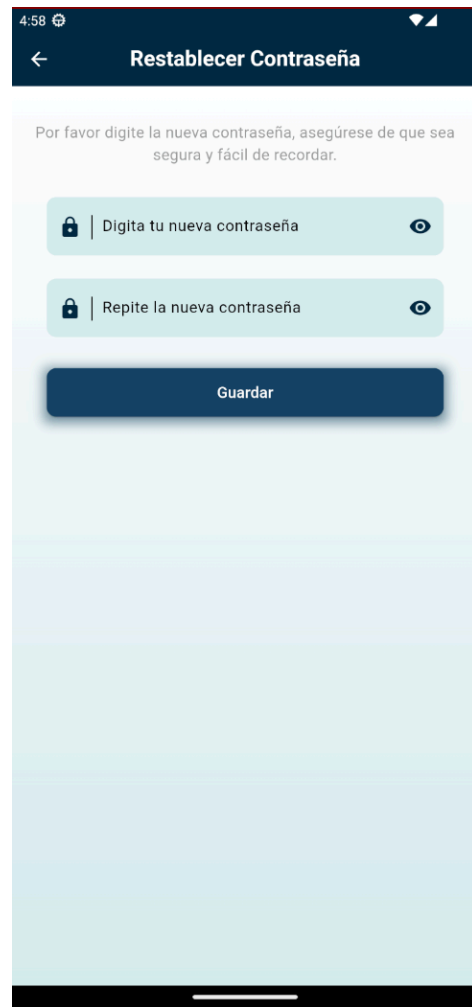
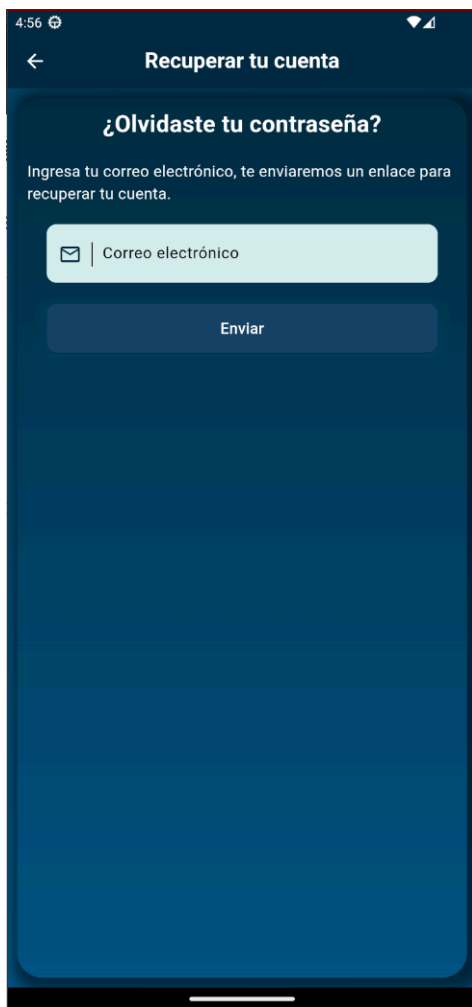


Figura 42 Restablecer Contraseña

Figura 43 Registrarse como

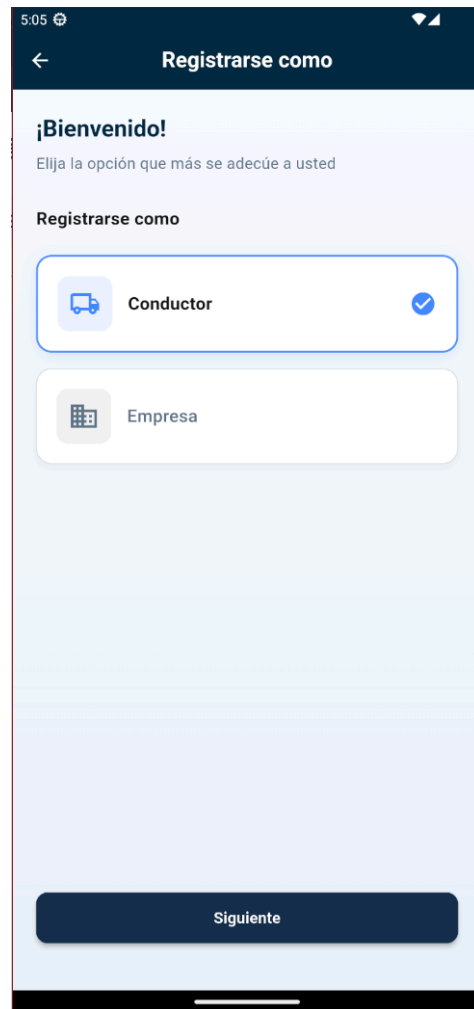
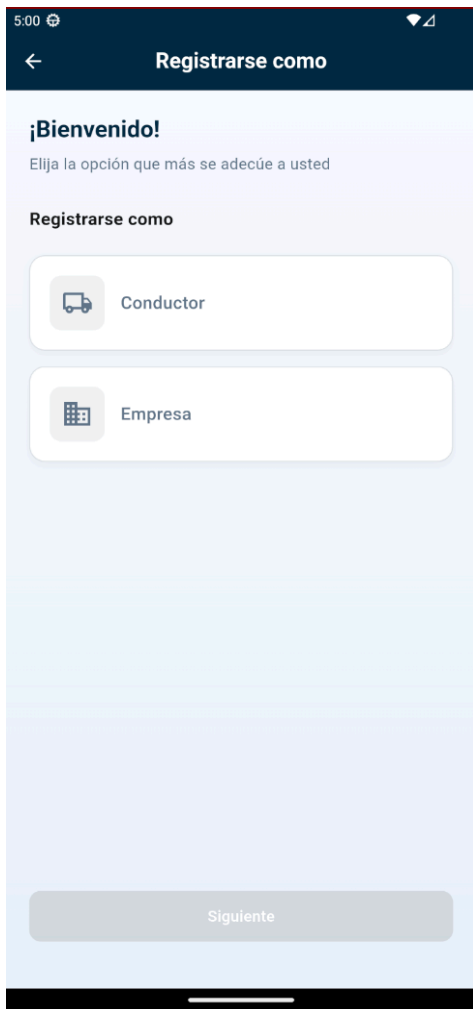


Figura 44 Registrarse como interacción

Figura 45 Registrar Conductor

The screenshot shows the registration form with the following fields:

- Nombres
- Apellidos
- Número de documento
- Fecha de nacimiento
- Correo electrónico
- Lugar de residencia
- Celular
- Contraseña
- Confirmar Contraseña

At the bottom, there is a 'Continuar' button and a link: '¿Ya tienes una cuenta?'.

The screenshot shows the registration form with the following data and error messages:

- Nombres: Juan Camilo
- Apellidos: Hernandez Perilla
- Número de documento: 1193034583
- Fecha de nacimiento: 30/11/2000
- Correo electrónico: juanchernandezp@dominio.com
- Lugar de residencia: (empty)
- Celular: (empty)
- Contraseña: (empty)
- Confirmar Contraseña: (empty)

Error messages:

- La contraseña es requerida
- La confirmación de contraseña es requerida

At the bottom, there is a 'Continuar' button.

Figura 46 Registrar Conductor manejo de errores

Figura 47 Registrar Empresa

5:26

← Registrar Compañía

Nombre de la empresa

Razón social

NIT

Fecha de matrícula

Correo electrónico

Dirección

Celular

Tipo de sociedad

Nombre del representante legal

Cédula del representante legal

Contraseña

Confirmar Contraseña

Continuar

— 0 —

¿Ya tienes una cuenta?

Inicia sesión

5:37

← Registrar Compañía

Nombre de la empresa

Sair

Razón social

Sair Envios

NIT

901939868

Fecha de matrícula

15/04/2025

Correo electrónico

contacto@sairenvios.com

Dirección

Celular

Sociedad por Acciones Simplificada (SAS)

Nombre del representante legal

Cédula del representante legal

Contraseña

.....

Confirmar Contraseña

La confirmación de contraseña es requerida

Continuar

— 0 —

¿Ya tienes una cuenta?

Inicia sesión

Figura 48 Registro empresa manejo de errores

Figura 49 Home Empresa



Figura 50 Home Conductor

Figura 51 Perfil Empresa

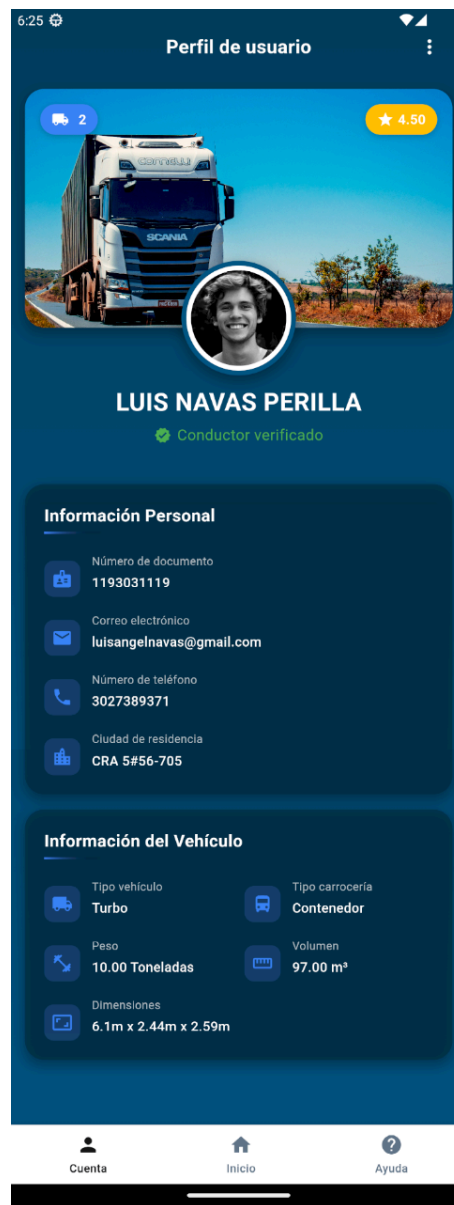
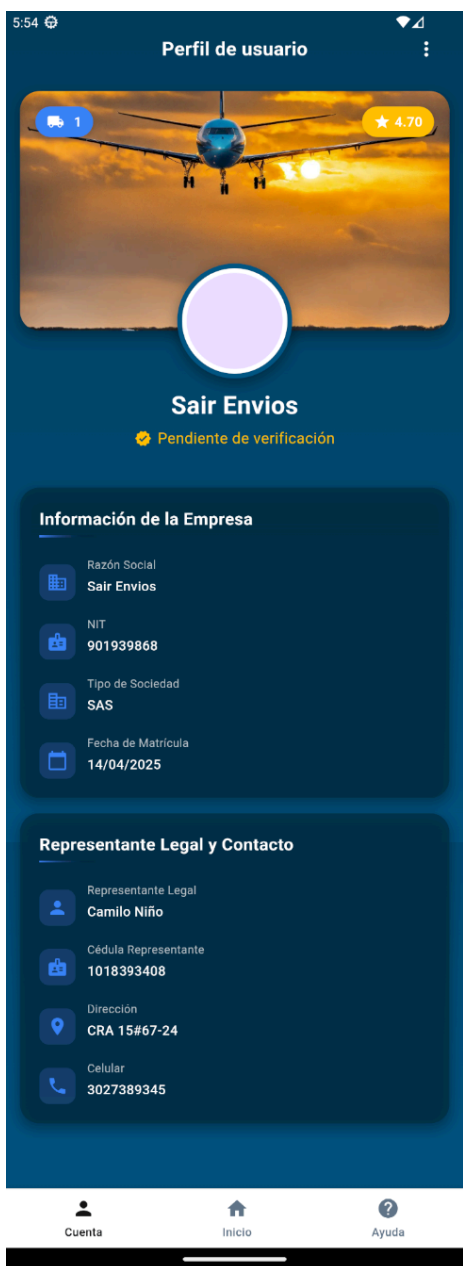


Figura 52 Perfil Conductor

Figura 53 Publicar Carga como Empresa

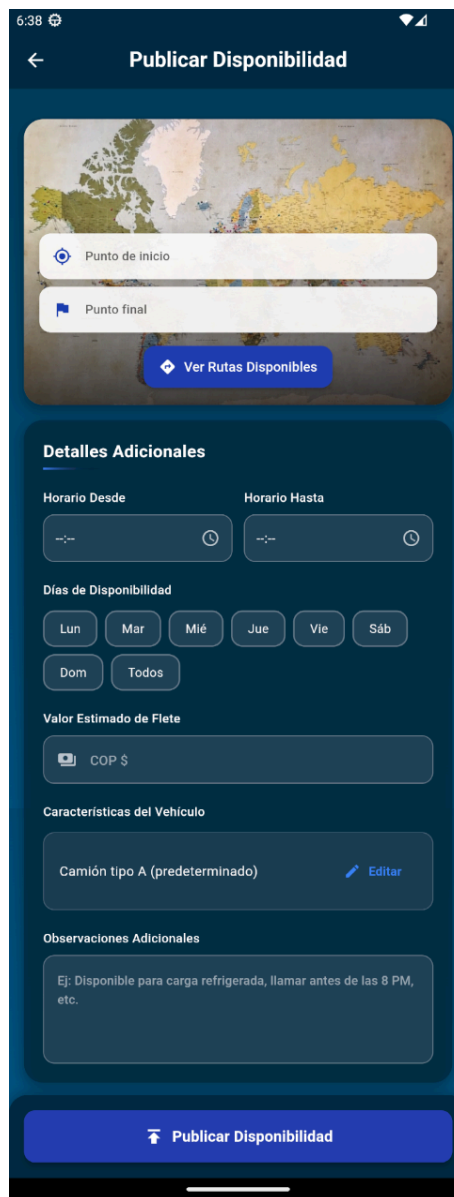
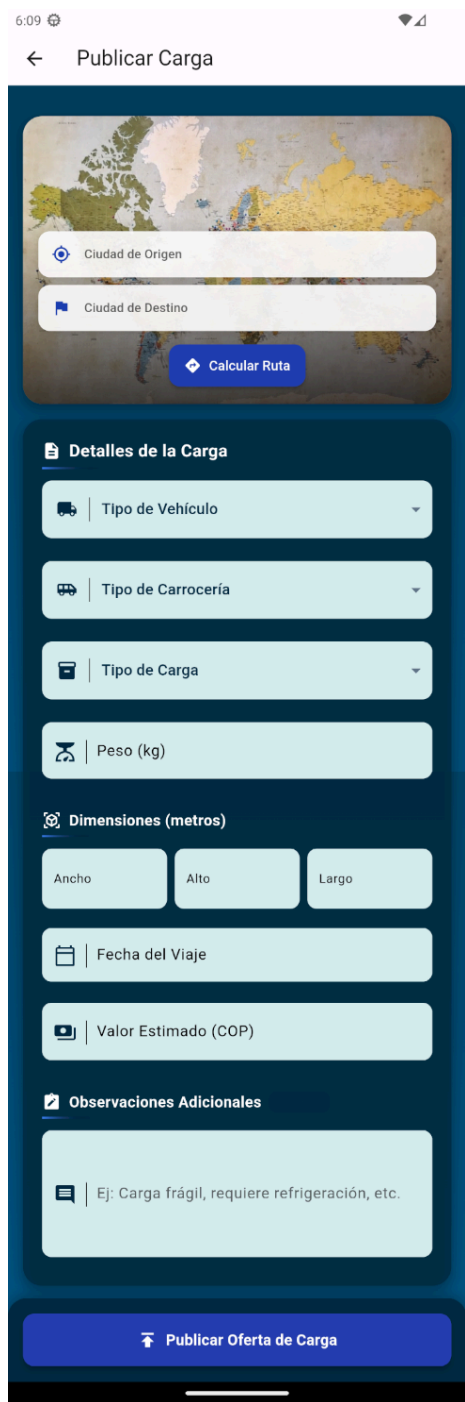


Figura 54 Publicar Vehículo como Conductor

Figura 55 Buscar Conductor como Empresa

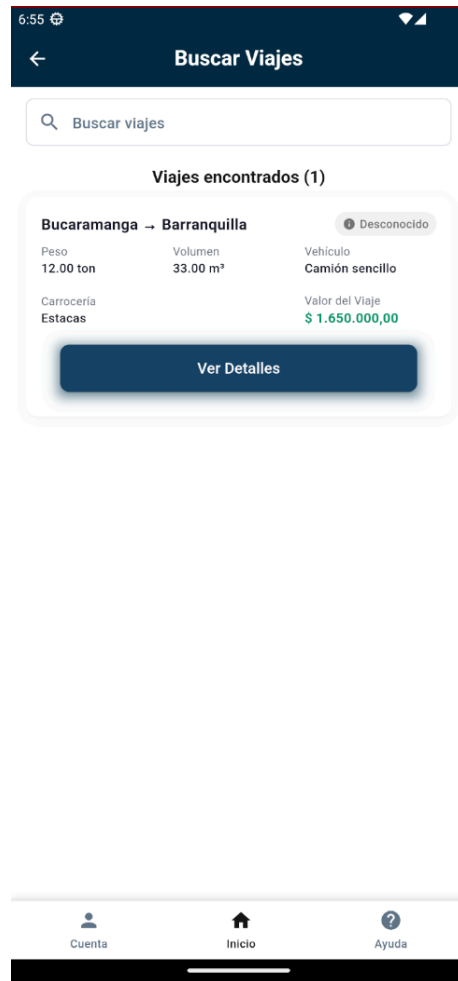
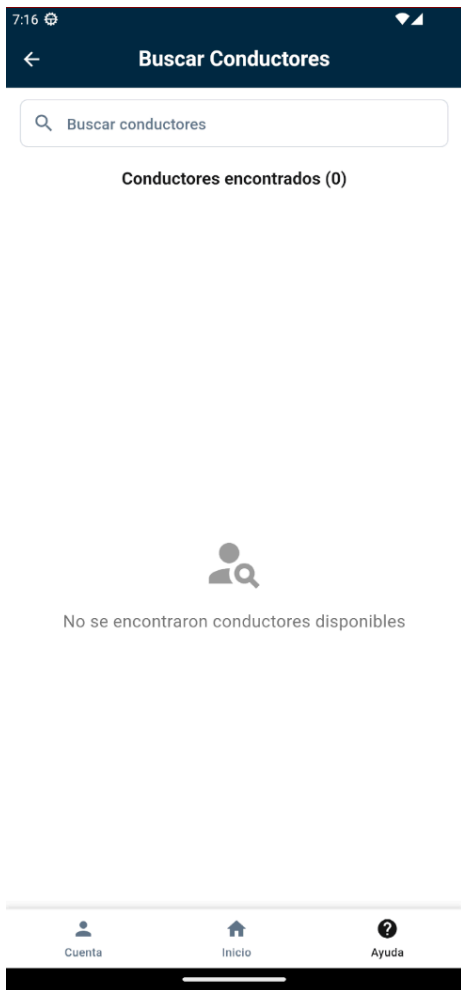


Figura 56 Buscar Viaje como Conductor

Figura 57 Mis cargas Empresa

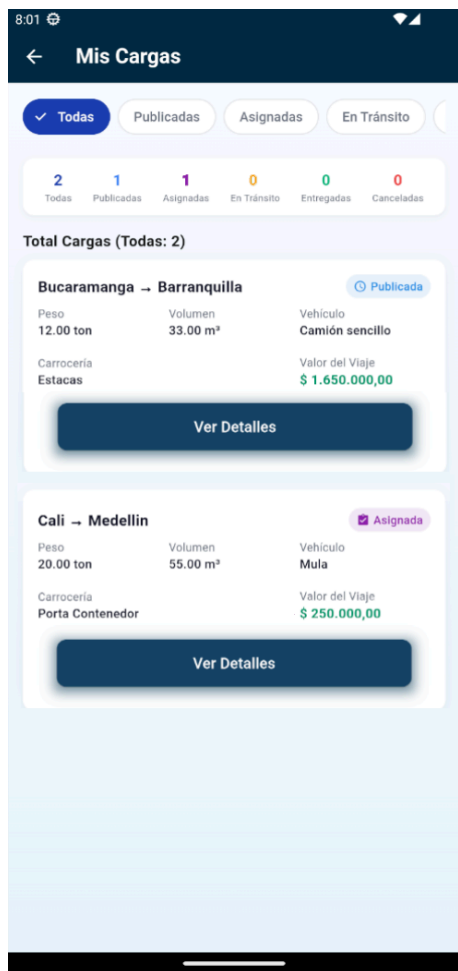


Figura 58 Detalles del viaje Empresa

Figura 59 Mis viajes Conductor

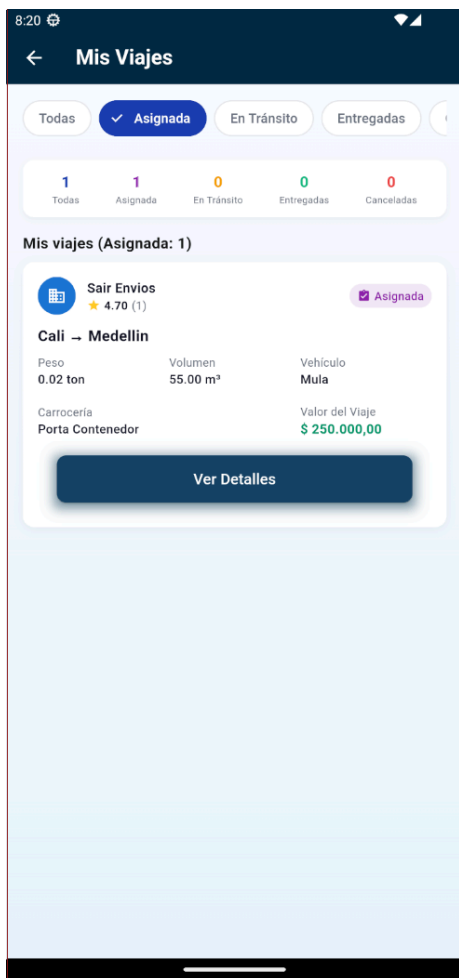


Figura 60 Detalles del viaje Conductor

Figura 61 Chat Conductor

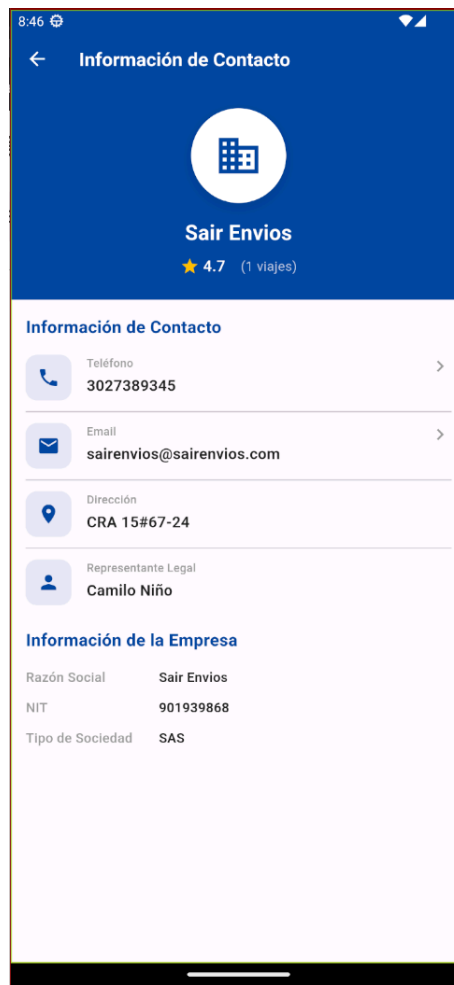
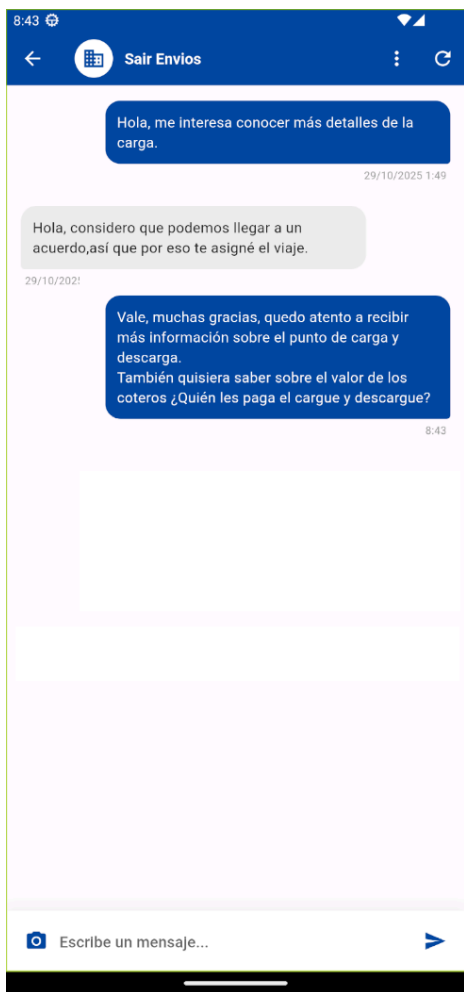


Figura 62 Perfil Público Empresa Chat Conductor

Figura 63 Lista de Interesados

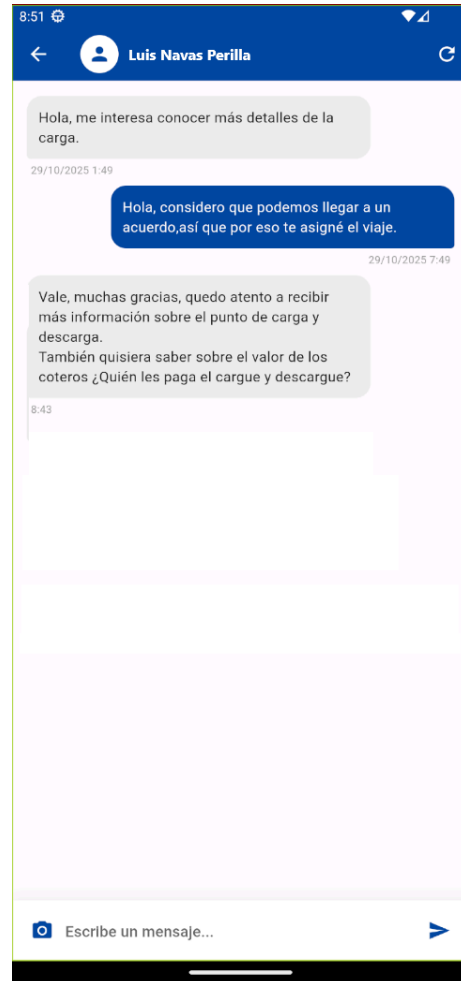
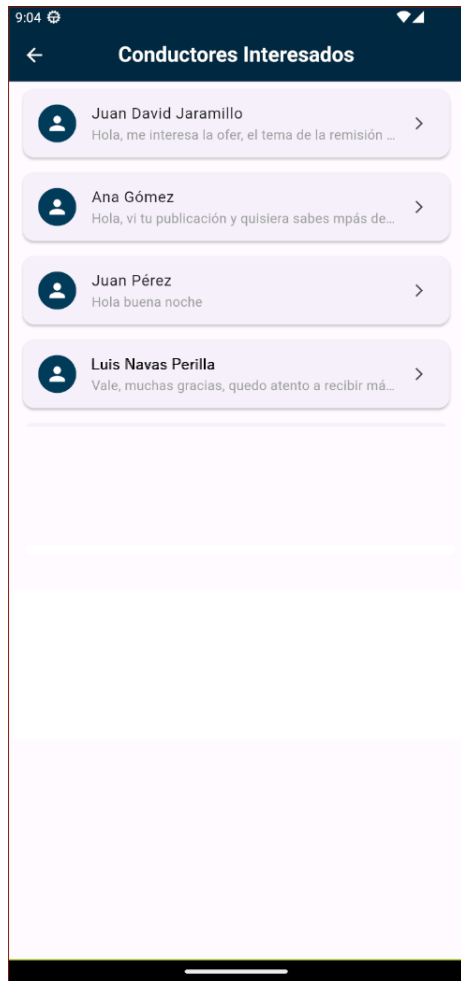


Figura 64 Chat Empresa

Figura 65 Perfil Público Conductor Chat Empresa

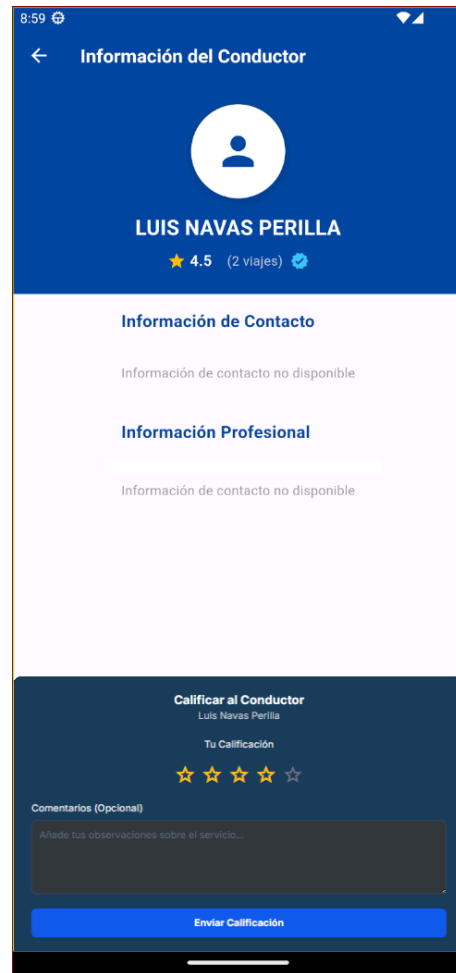
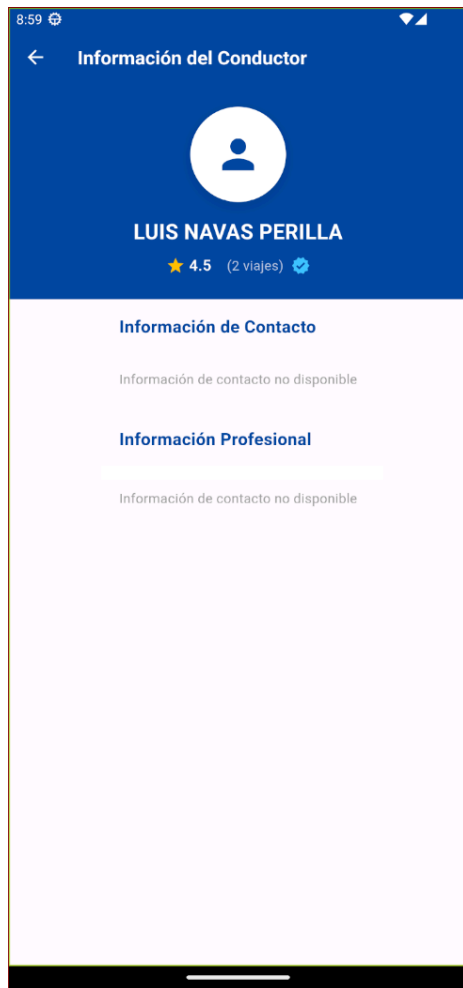


Figura 66 Calificación Empresa

Figura 67 Mapa de navegación

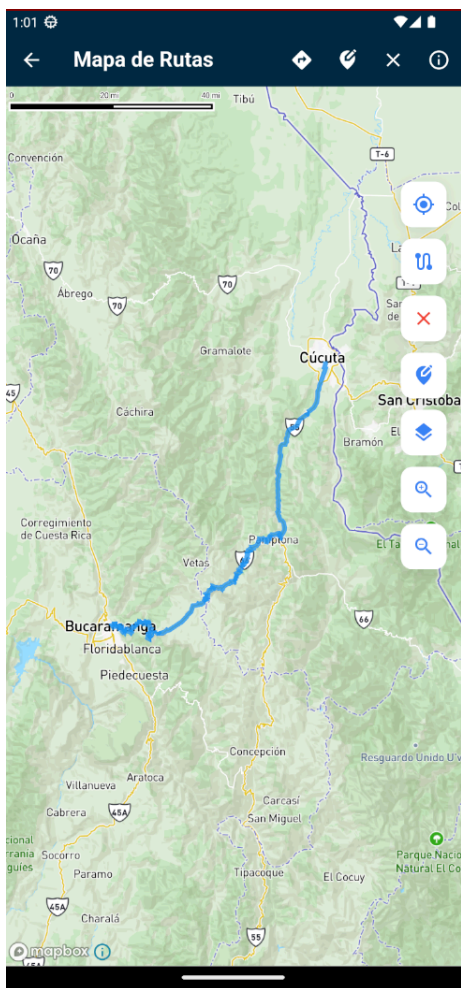


Figura 68 Sección de Noticias

Figura 69 Centro de ayuda



Figura 70 Centro de ayuda de guías

Apéndice C Encuestas

En la siguiente sección se presenta el análisis de resultados en las encuestas realizadas desde la cual se obtuvieron las opiniones de los estudiantes sobre los requerimientos de la aplicación. Estas encuestas han sido la base para determinar los requerimientos del proyecto, usando estos datos cuantitativos y cualitativos que han permitido ajustar los requerimientos y adaptar la solución a las necesidades de los usuarios.

Figura 71 Respuestas pregunta 01

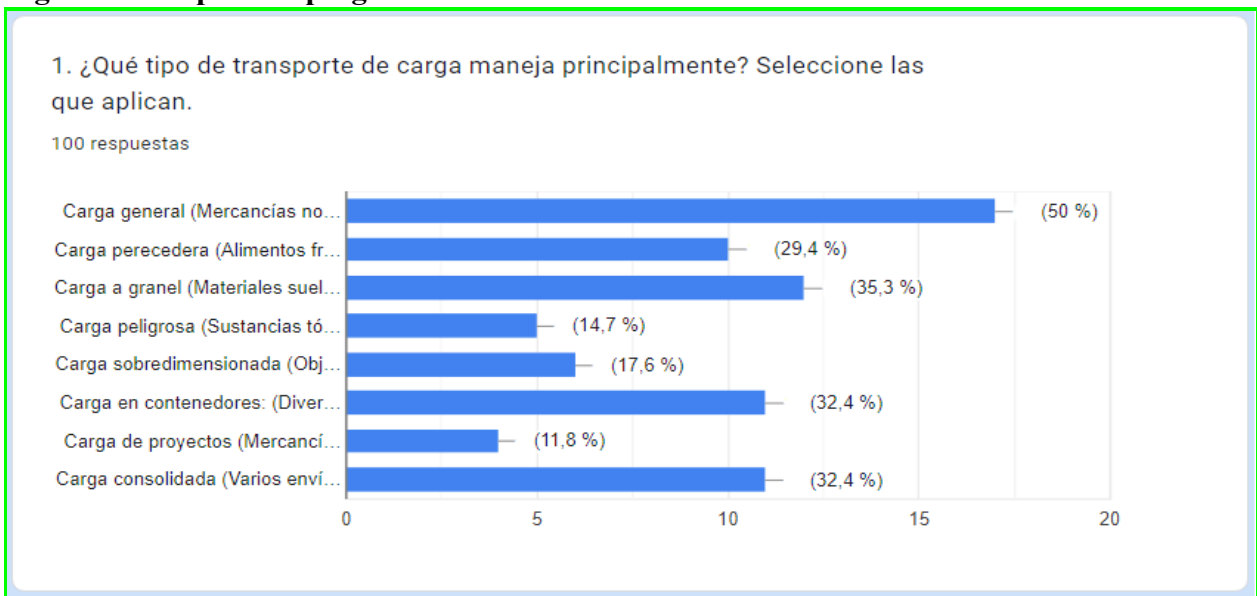


Figura 72 Respuesta pregunta 02

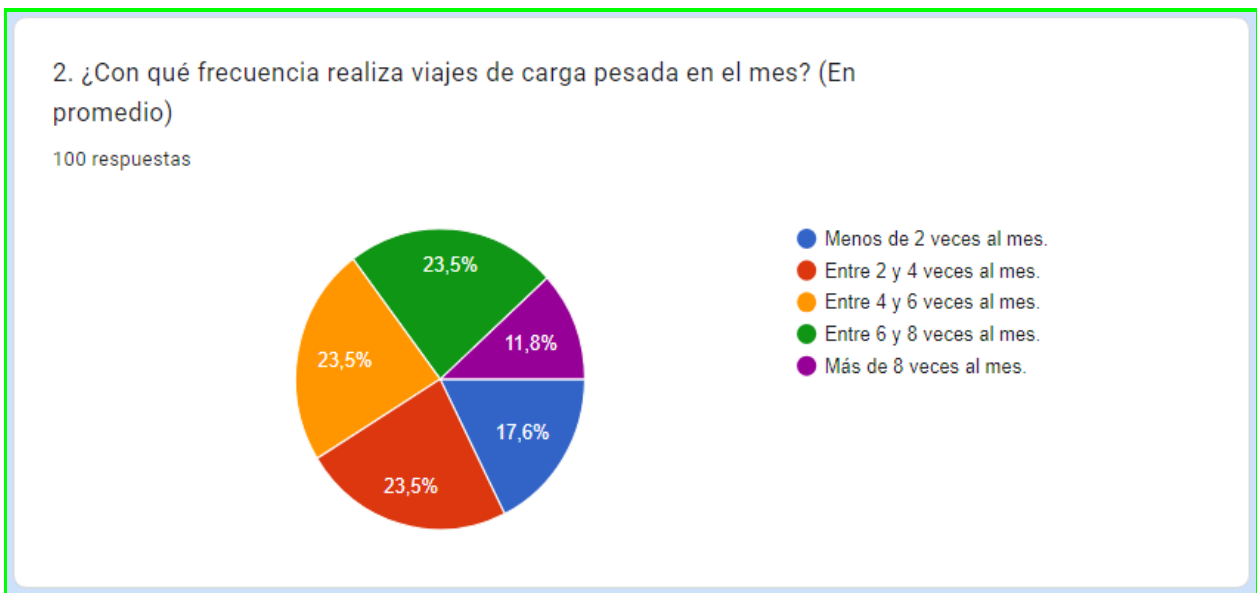


Figura 73 Respuesta pregunta 03

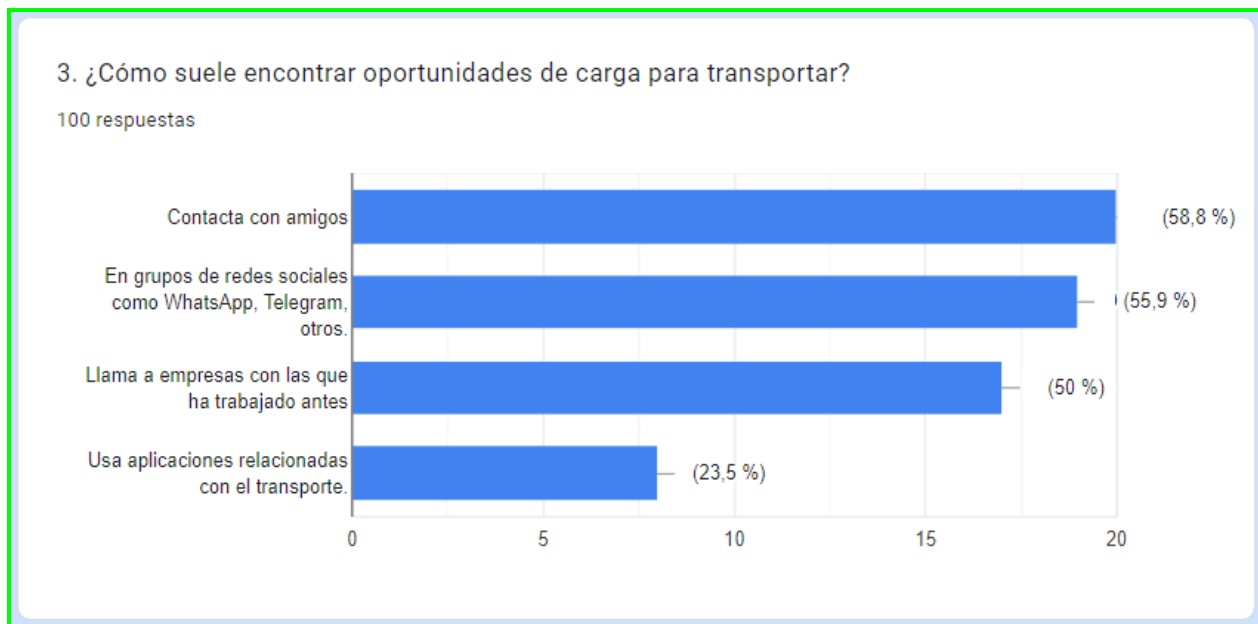


Figura 74 Respuesta pregunta 04

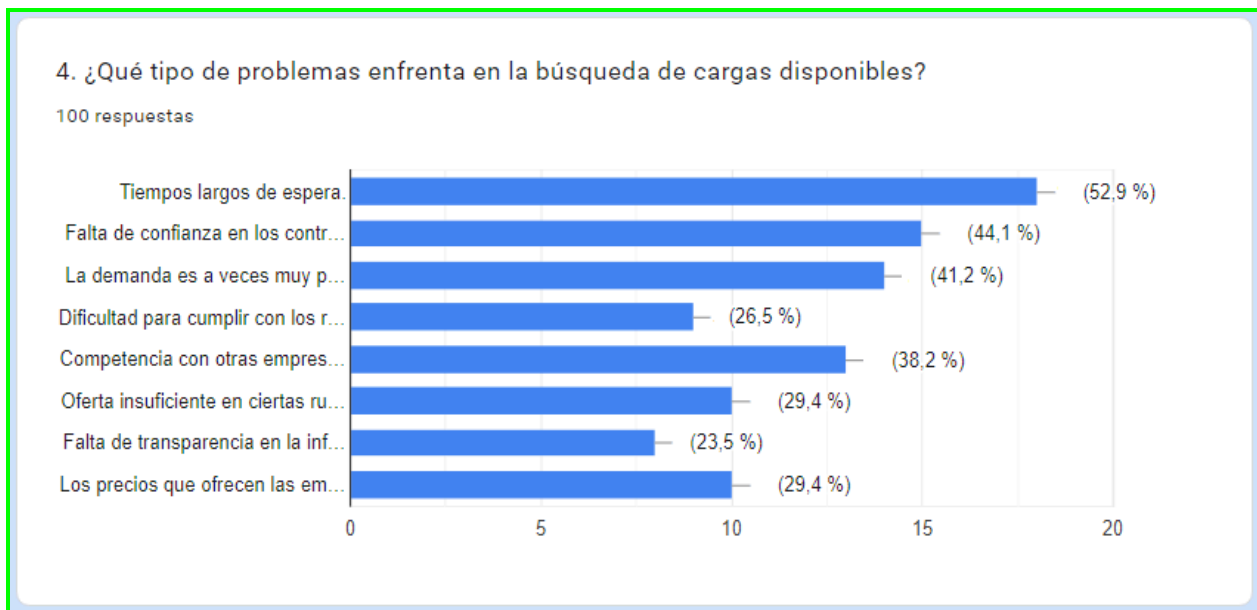


Figura 75 Respuesta pregunta 05

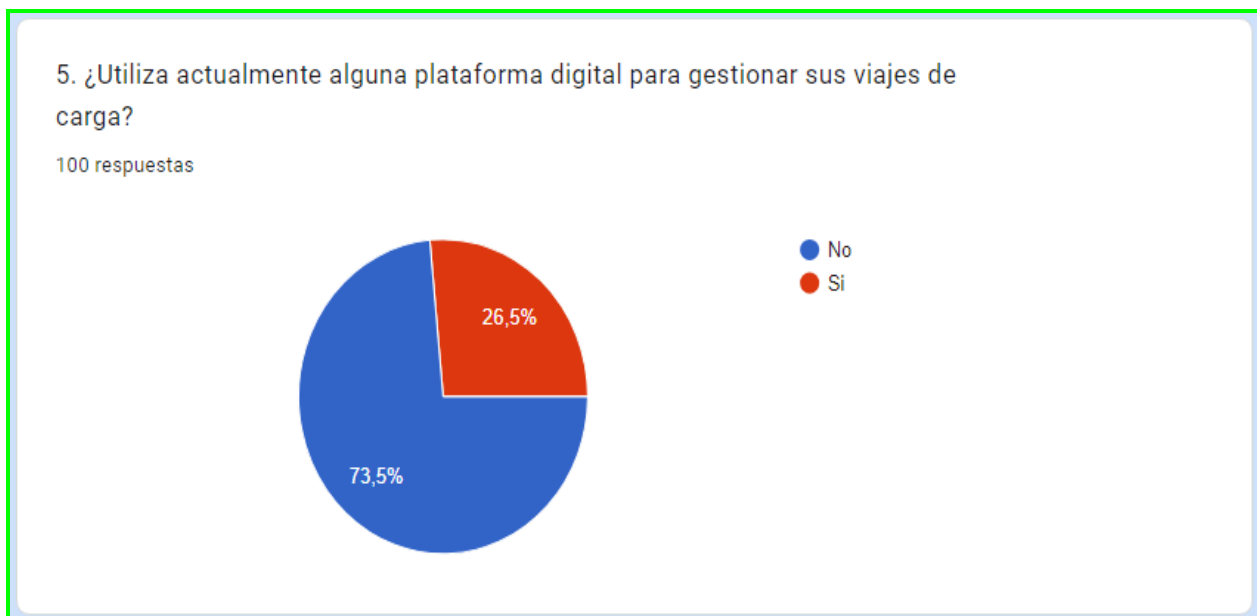


Figura 76 Respuesta pregunta 06



Figura 77 Respuesta pregunta 07

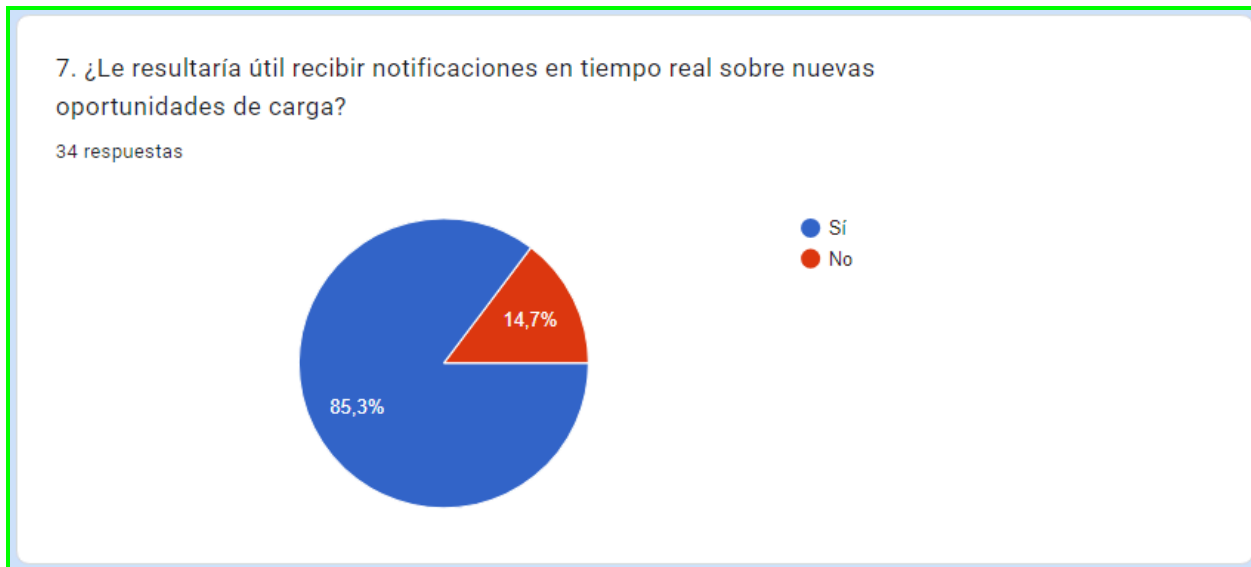


Figura 78 Respuesta pregunta 08



Figura 79 Respuesta pregunta 09

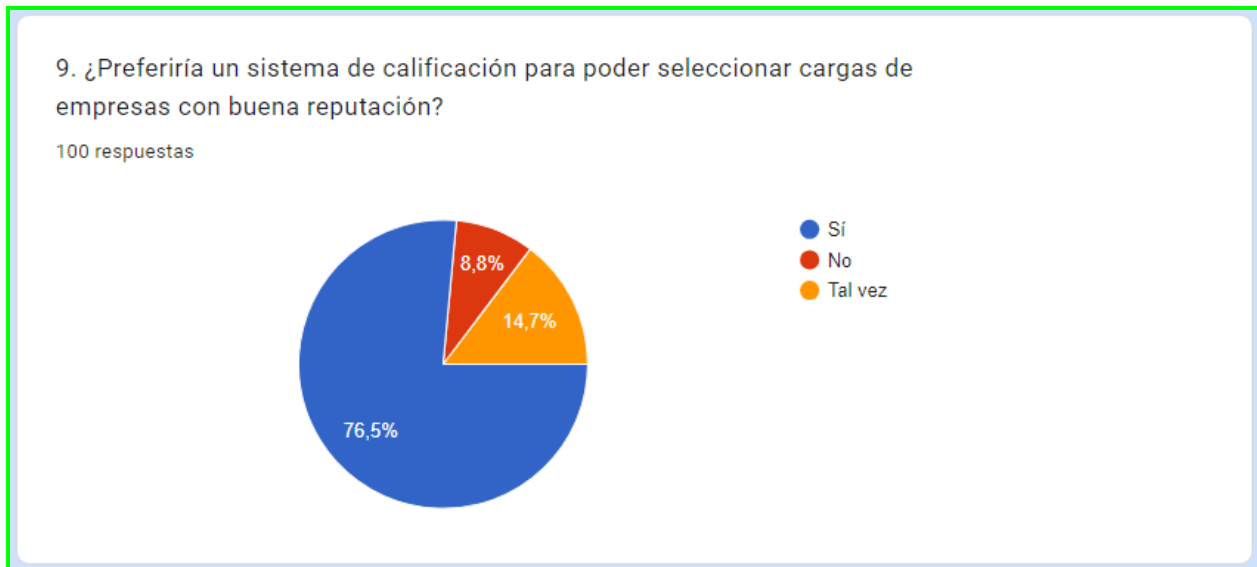


Figura 80 Respuesta pregunta 10

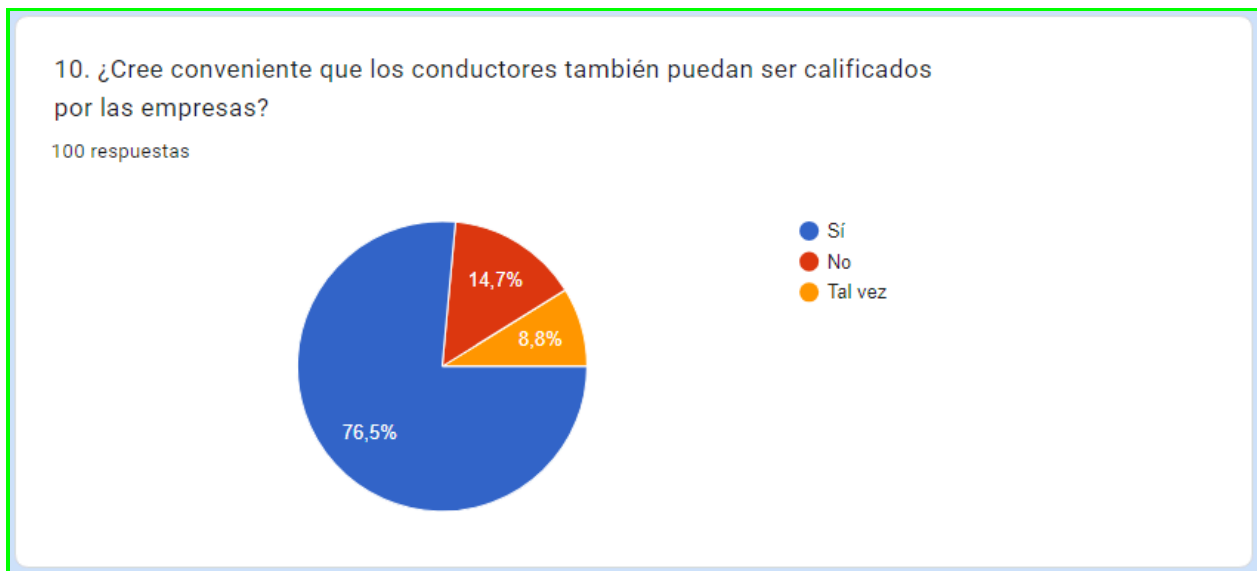


Figura 81 Respuesta pregunta 11

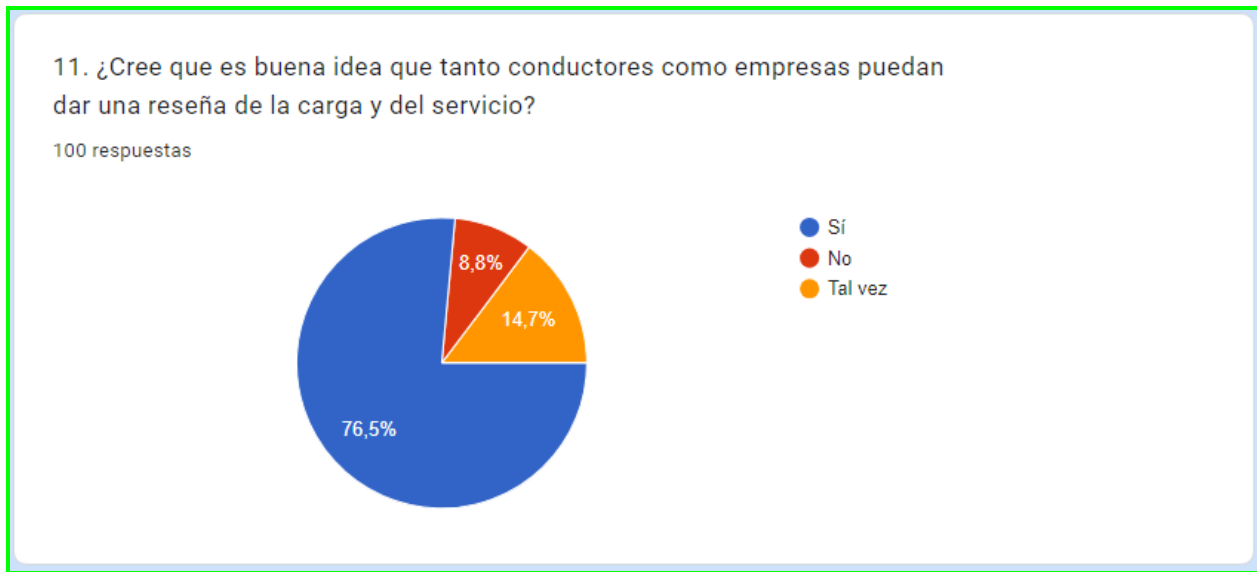


Figura 82 Respuesta pregunta 12

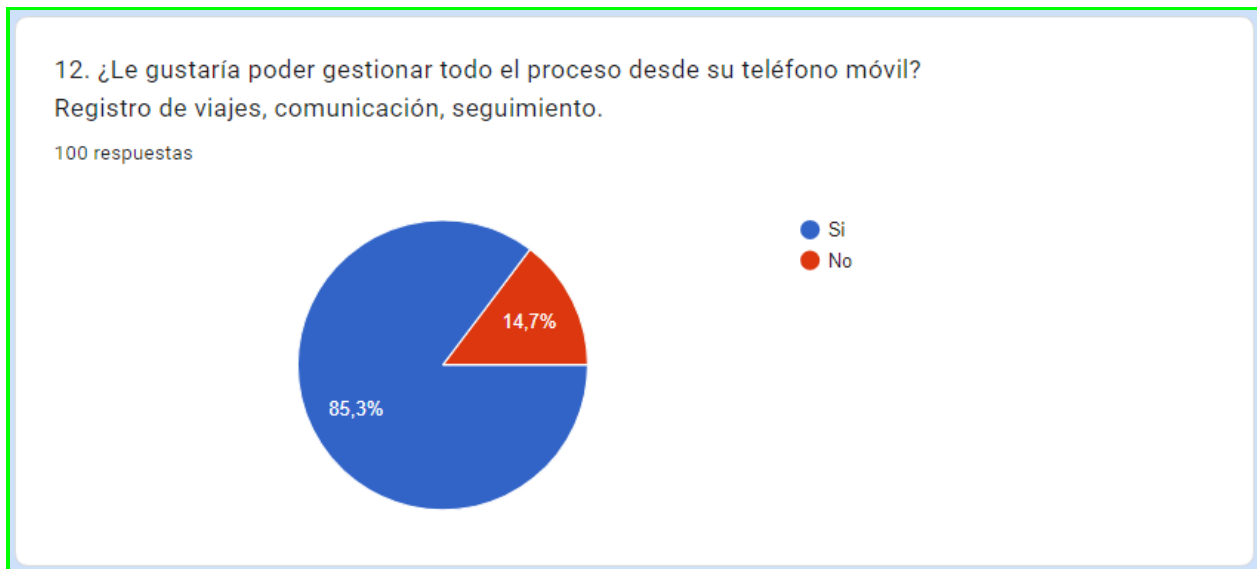


Figura 83 Respuesta pregunta 13

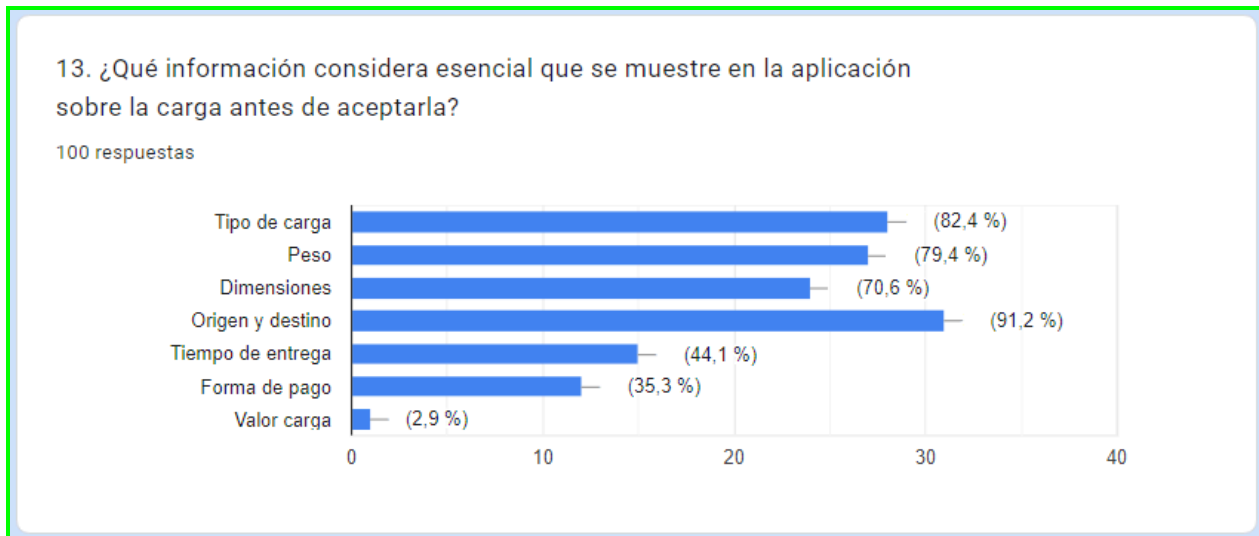


Figura 84 Respuesta pregunta 14



Figura 85 Respuesta pregunta 15

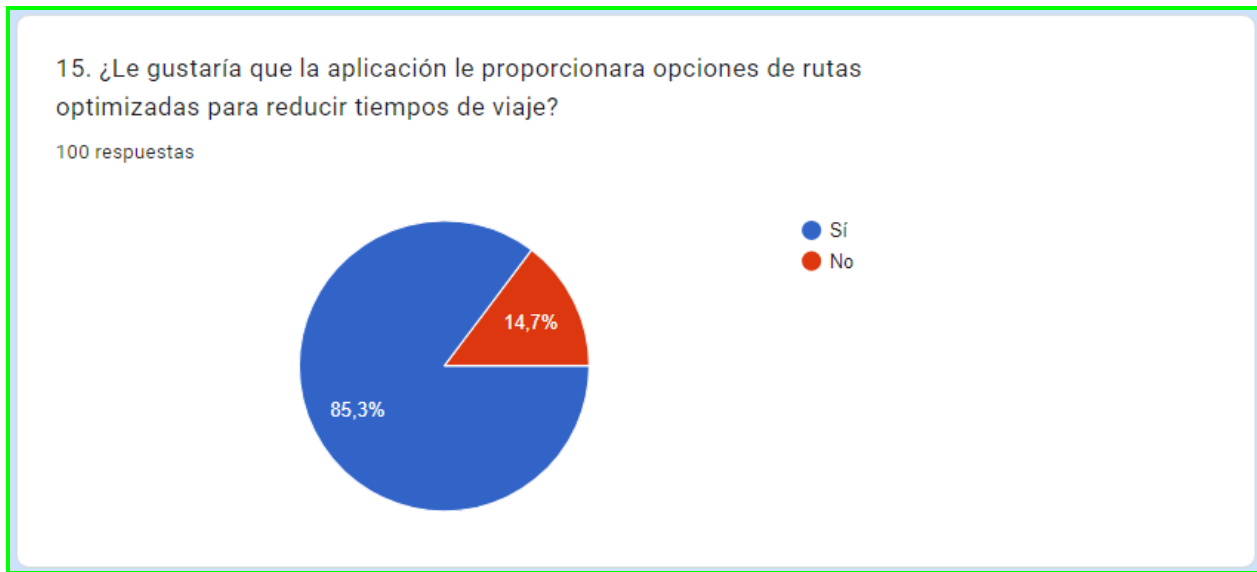


Figura 86 Respuesta pregunta 16

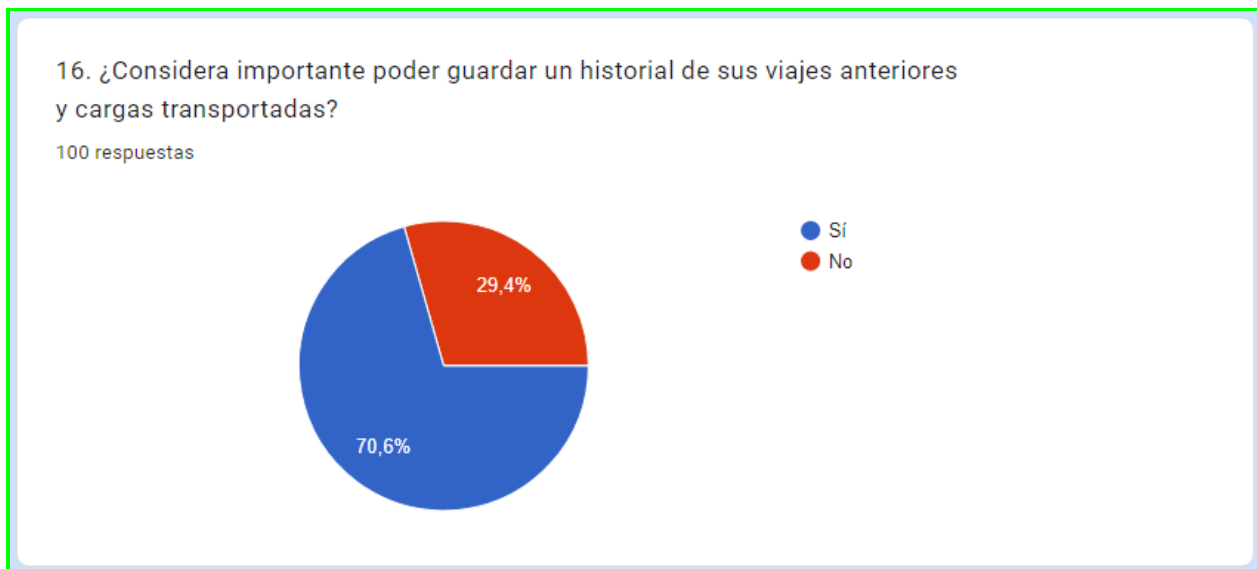


Figura 87 Respuesta pregunta 17

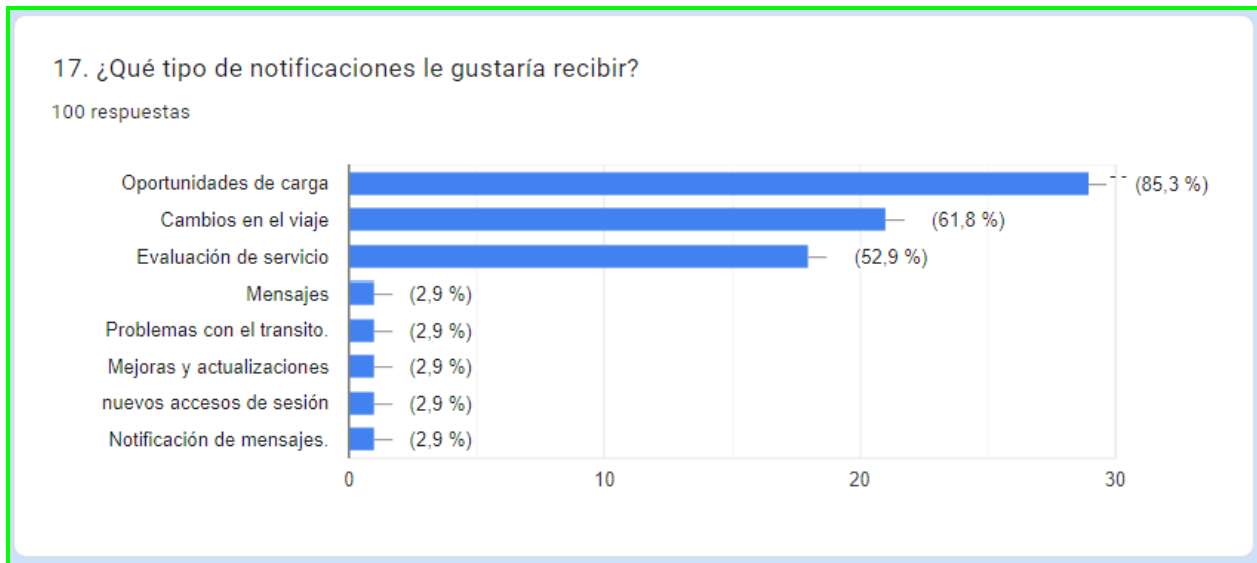


Figura 88 Respuesta pregunta 18

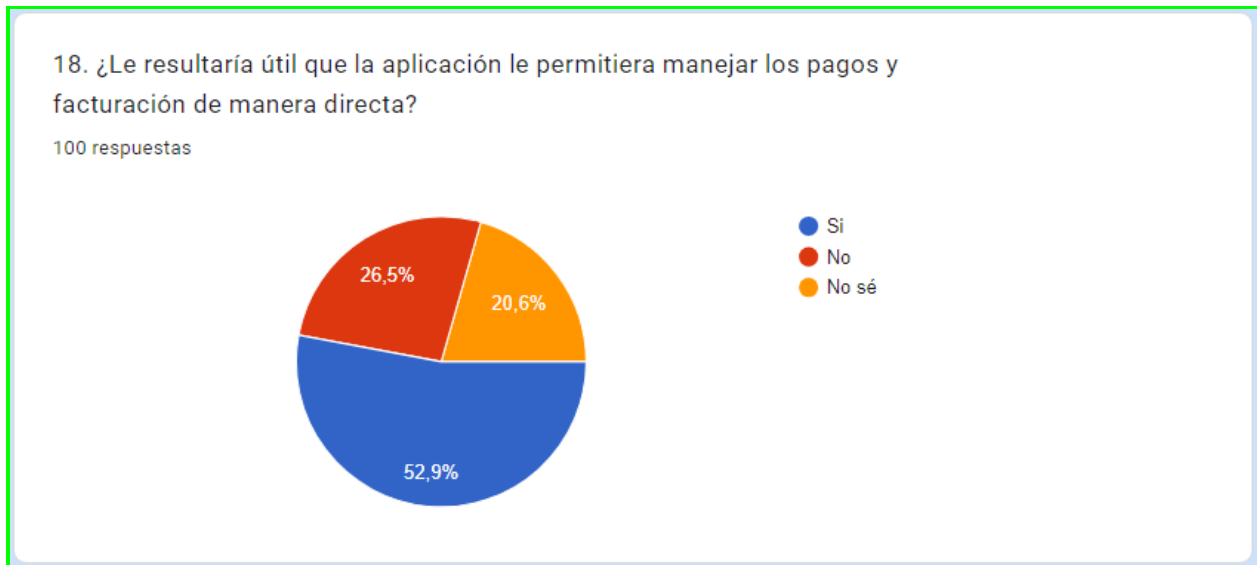


Figura 89 Respuesta pregunta 19

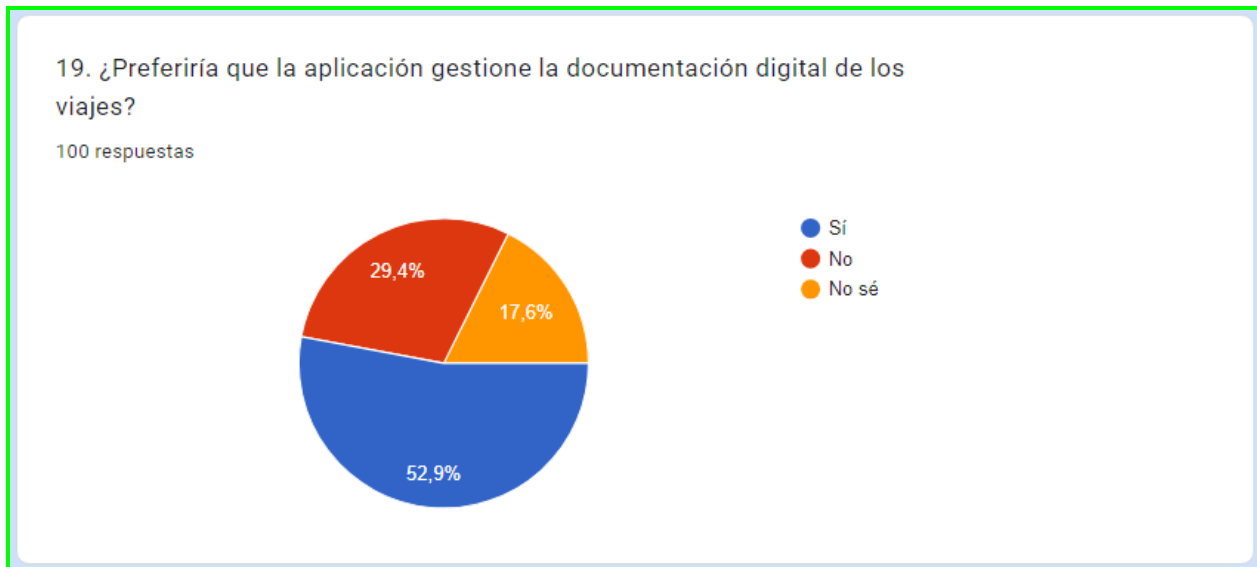


Figura 90 Respuesta pregunta 20

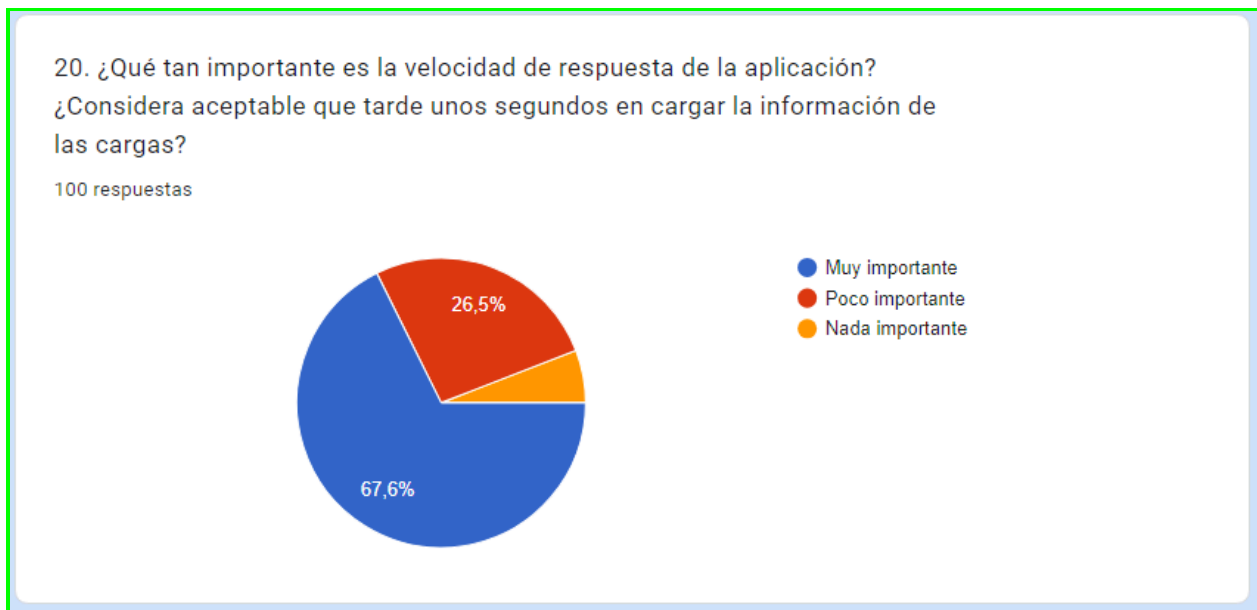


Figura 91 Respuesta pregunta 21

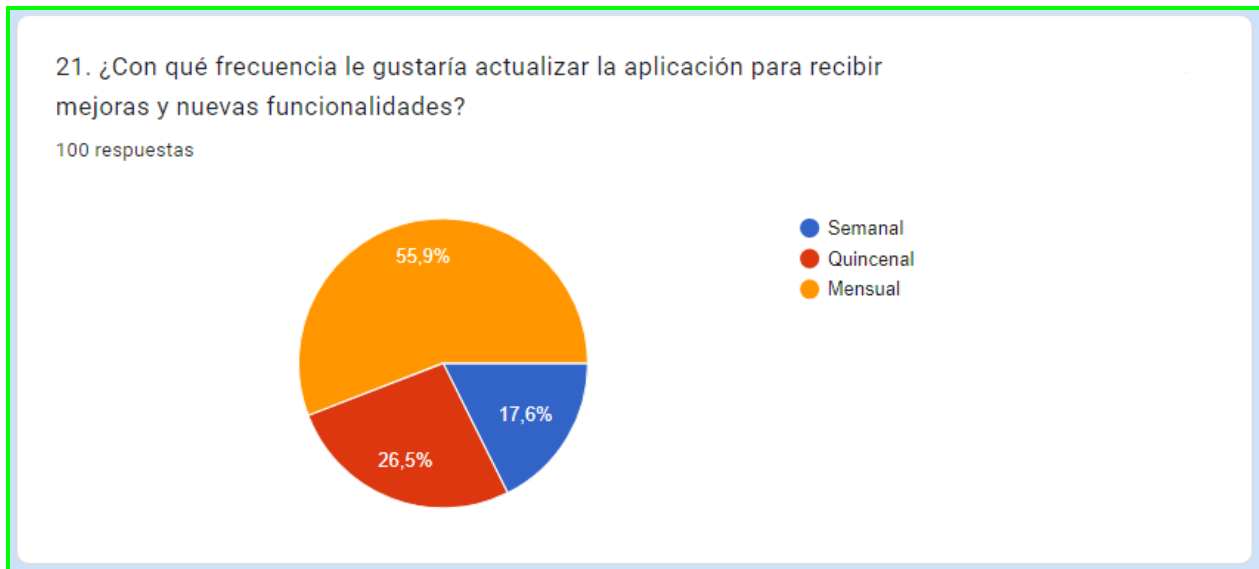


Figura 92 Respuesta pregunta 22

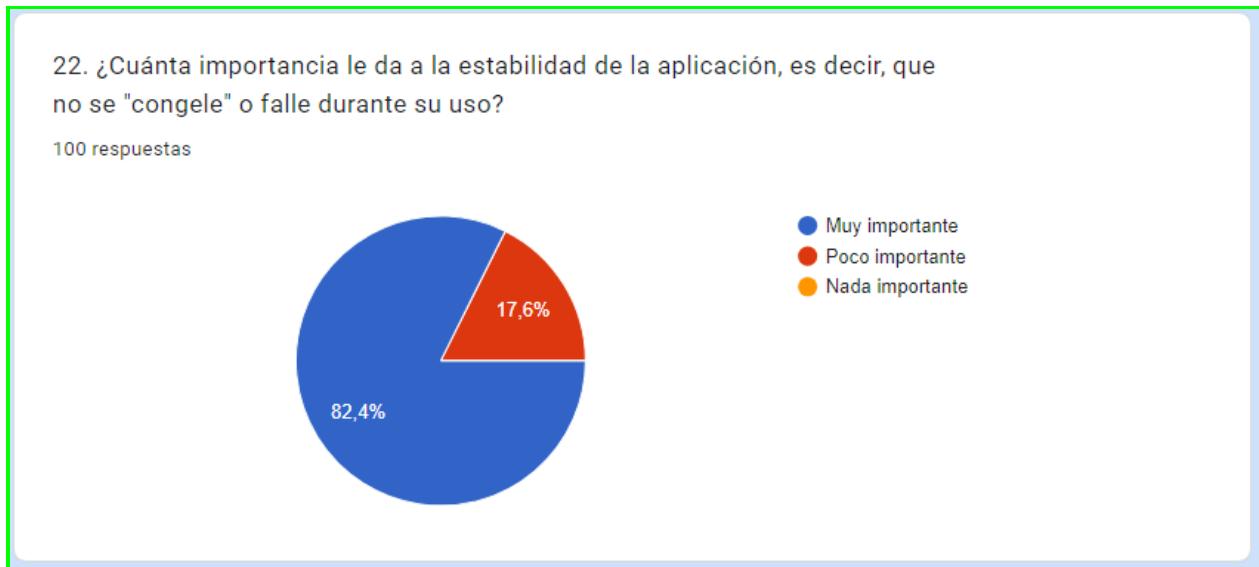


Figura 93 Respuesta pregunta 23

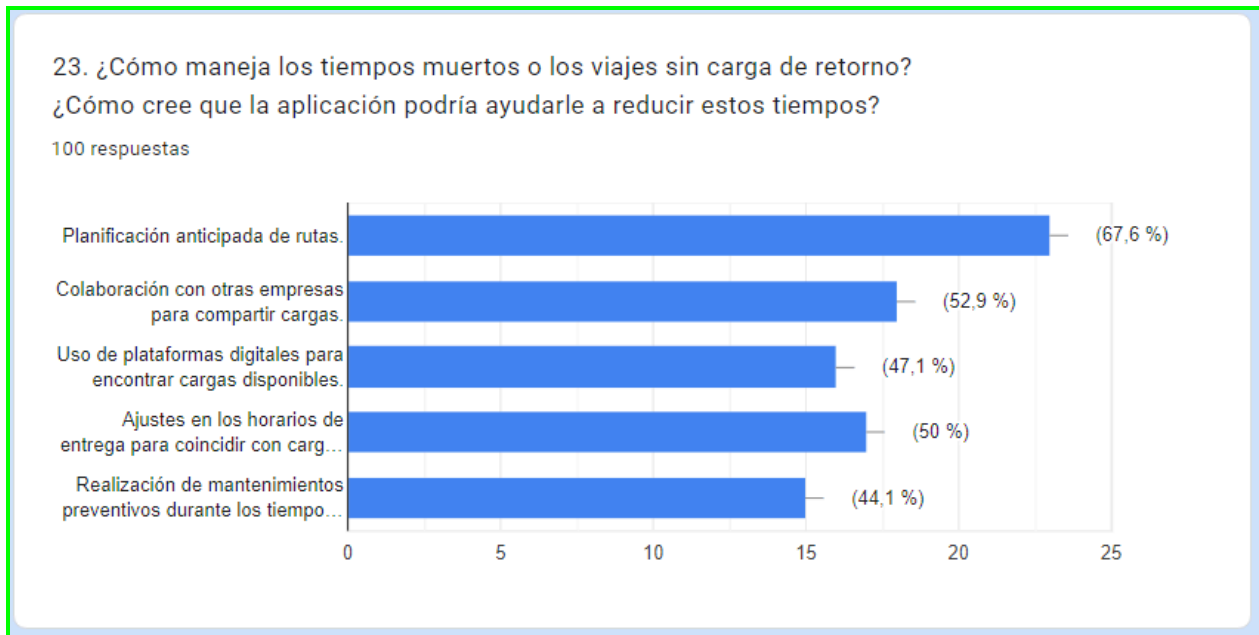


Figura 94 Respuesta pregunta 24

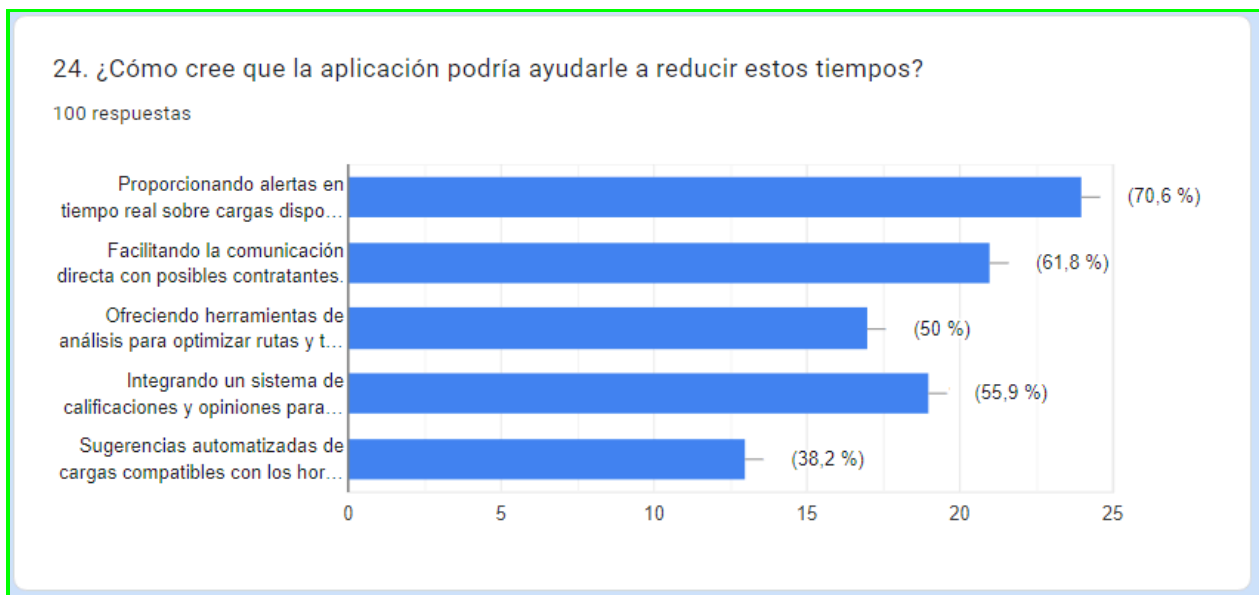


Figura 95 Respuesta pregunta 25



Figura 96 Respuesta pregunta 26

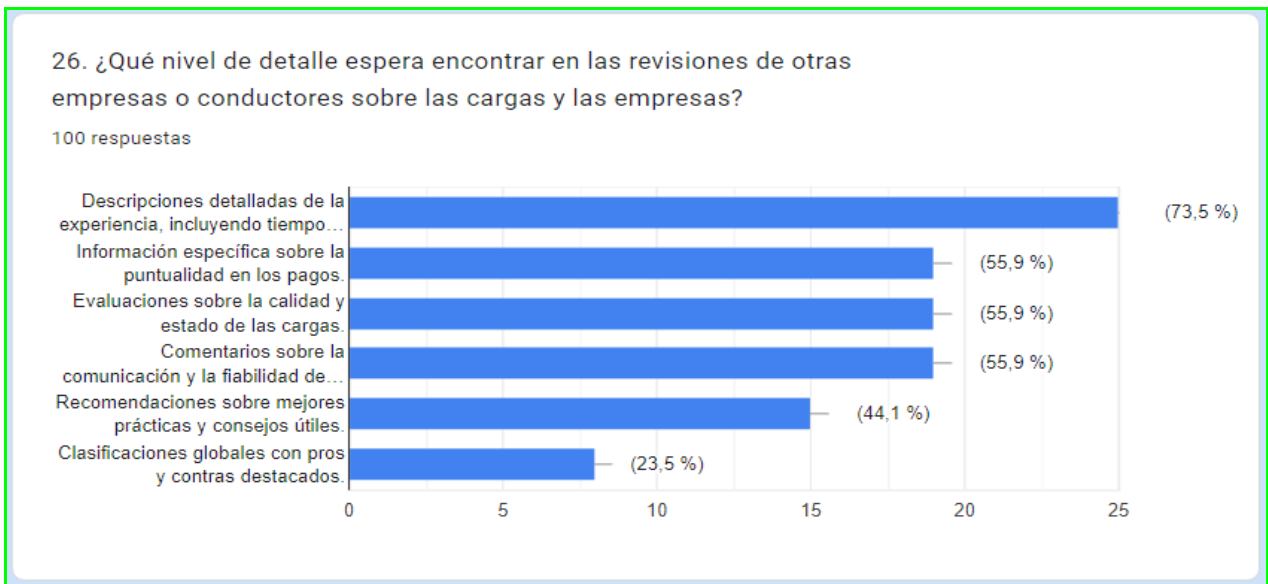


Figura 97 Respuesta pregunta 27

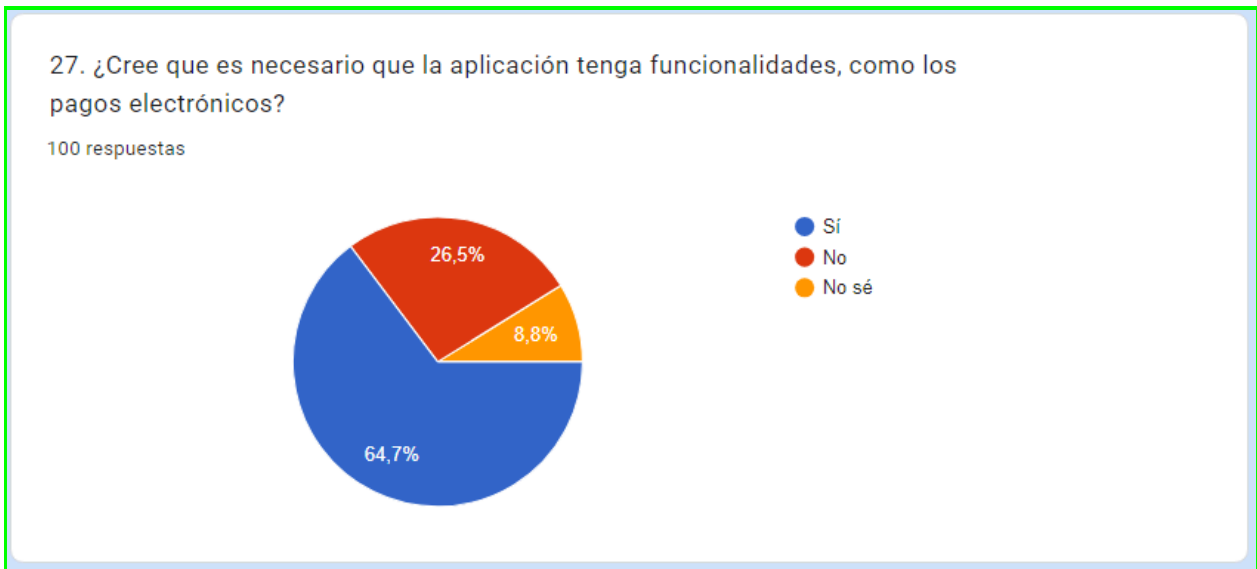
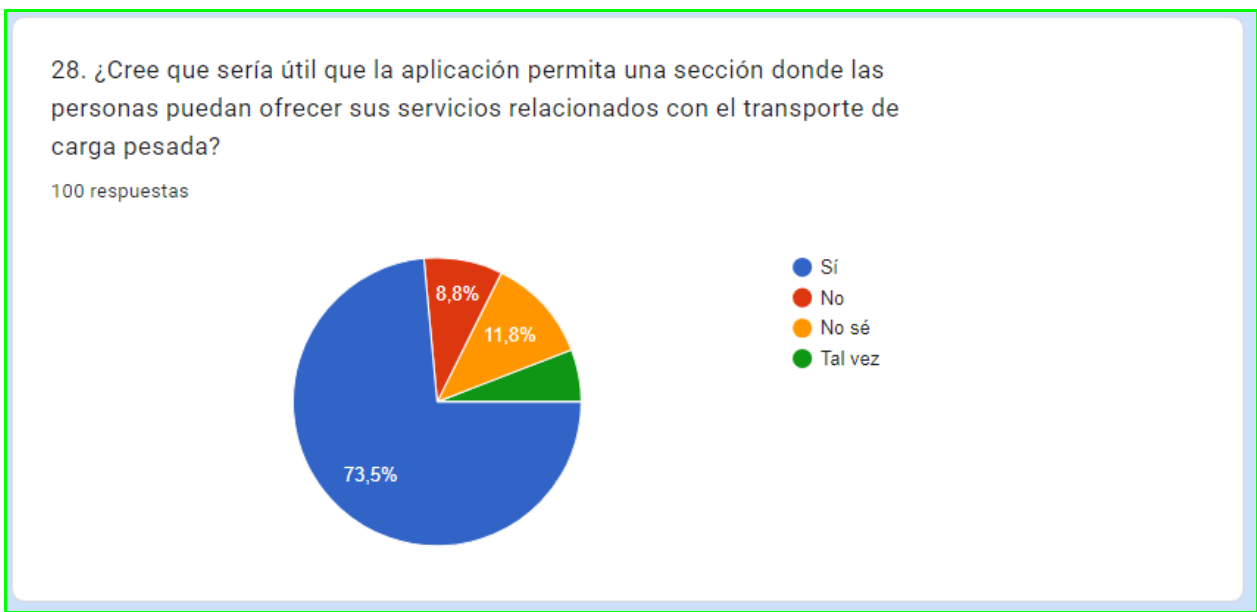


Figura 98 Respuesta pregunta 28



Links Informativos de Referencia

Youtube

Ruta Conductor 01

<https://youtu.be/bt5sUEDpDP0>

Ruta Conductor 02

<https://youtu.be/v2jM6AnDHTw?si=-lp-cF3DyzddhxRX>

Ruta Empresa 01

<https://youtu.be/dWhxy3atLs0>

Ruta Empresa 02

<https://youtu.be/-fF4S-U2jRI>

GitHub

Link git público

<https://github.com/isaac1803/move-x-tesis>