UNIVERSIDAD NACIONAL DE CHIMBORAZO
FACULTAD DE INGENIERÍA
CARRERA DE TECNOLOGÍAS DE LA INFORMACIÓN

Desarrollo de una Aplicación Móvil multiplataforma de Delivery para farmacias Cruz Azul utilizando Flutter y NestJS

Trabajo de Titulación para optar al título de Ingeniero en Tecnologías de la Información

Autor:
Farfan Miranda, Antonio Adolfo

Tutor:
Ing Jorge Edwin Delgado Altamirano. Msc

Riobamba, Ecuador. 2025

DECLARATORIA DE AUTORÍA

Yo, **Antonio Adolfo Farfan Miranda**, con cédula de ciudadanía **2350960924**, autor del trabajo de investigación titulado: **DESARROLLO DE UNA APLICACIÓN MÓVIL MULTIPLATAFORMA DE DELIVERY PARA FARMACIAS CRUZ AZUL UTILIZANDO FLUTTER Y NESTJS**, certifico que la producción, ideas, opiniones, criterios, contenidos y conclusiones expuestas son de mí exclusiva responsabilidad.

Asimismo, cedo a la Universidad Nacional de Chimborazo, en forma no exclusiva, los derechos para su uso, comunicación pública, distribución, divulgación y/o reproducción total o parcial, por medio físico o digital; en esta cesión se entiende que el cesionario no podrá obtener beneficios económicos. La posible reclamación de terceros respecto de los derechos de autor (a) de la obra referida, será de mi entera responsabilidad; librando a la Universidad Nacional de Chimborazo de posibles obligaciones.

En Riobamba, 10 de julio del 2025

Antonio Adolfo Farfan Miranda

C.I: 2350960924

DICTAMEN FAVORABLE DEL PROFESOR TUTOR

Quien suscribe, Ing. Jorge Delgado, Mgs catedrático adscrito a la Facultad de Ingeniería, por medio del presente documento certifico haber asesorado y revisado el desarrollo del Trabajo de Investigación titulado "Desarrollo de una Aplicación Móvil multiplataforma de Delivery para farmacias Cruz Azul utilizando Flutter y NestJS" bajo la autoría de Antonio Adolfo Farfan Miranda; por lo que se autoriza ejecutar los trámites legales para su sustentación.

Es todo cuanto informar en honor a la verdad; en Riobamba, a los 10 días del mes de julio de 2025

Ing. Jorge Delgado, Mgs

C.I: 0602759383

CERTIFICADO DE LOS MIEMBROS DEL TRIBUNAL

Quienes suscribimos, catedráticos designados Miembros del Tribunal de Grado para la evaluación del trabajo de investigación "Desarrollo de una Aplicación Móvil multiplataforma de Delivery para farmacias Cruz Azul utilizando Flutter y NestJS", presentado por Antonio Adolfo Farfan Miranda, con cédula de identidad número 2350960924, bajo la tutoría de Mgs. Jorge Delgado; certificamos que recomendamos la APROBACIÓN de este con fines de titulación. Previamente se ha evaluado el trabajo de investigación y escuchada la sustentación por parte de su autor; no teniendo más nada que observar.

De conformidad a la normativa aplicable firmamos, en Riobamba 9 de diciembre del 2025

Lady Espinoza Mgs.
PRESIDENTE DEL TRIBUNAL DE GRADO

Ana Congacha Mgs.
MIEMBRO DEL TRIBUNAL DE GRADO

Danny Velasco Mgs.
MIEMBRO DEL TRIBUNAL DE GRADO

C E R T I F I C A C I Ó N

Que, **ANTONIO ADOLFO FARFAN MIRANDA** con CC: **2350960924,** estudiante de la Carrera de Ingeniería en Tecnologías de la Información, Facultad de Ingeniería ha trabajado bajo mi tutoría el trabajo de investigación titulado "Desarrollo de una Aplicación Móvil multiplataforma de Delivery para farmacias Cruz Azul utilizando Flutter y NestJS" cumple con el 1 % de acuerdo al reporte del sistema Anti plagio Compilatio, porcentaje aceptado de acuerdo a la reglamentación institucional por consiguiente autorizo continuar con el proceso.

Riobomba, 15 de octubre de 2025.

Firmado electrónicamente por:
JORGE EDWIN DELGADO ALTAMIRANO
Validar únicamente con FirmaEC

Mg. Jorge Delgado A.
TUTOR(A)

DEDICATORIA

A Dios, fuente inagotable de fuerza, sabiduría y amor, quien ha sido mi guía en cada paso de este camino, dedico este proyecto con profundo agradecimiento.

A mis amados padres, por su amor incondicional, su apoyo constante y sus sacrificios, que me han impulsado a alcanzar esta meta. Ustedes, con su ejemplo y entrega, han sido mi mayor inspiración.

Este logro es tanto mío como suyo, pues sin su presencia y sus palabras de aliento, este sueño no habría sido posible.

Antonio Adolfo Farfan Miranda

AGRADECIMIENTO

Al culminar este proyecto, mi corazón se llena de gratitud hacia todas aquellas personas que, de una u otra manera, contribuyeron a que este logro fuera posible.

A mis padres, Gloria Miranda y Jhon Farfan quienes, con su amor incondicional, paciencia infinita y apoyo constante, han sido mi pilar en cada etapa de este camino. Su ejemplo de dedicación y esfuerzo ha sido mi mayor fuente de inspiración para no rendirme, incluso en los momentos más difíciles.

A mi familia, en especial a mi hermano José y cuñada Ivonne, por su apoyo inquebrantable, su comprensión y sus palabras de aliento. Cada gesto de cariño y motivación ha dejado una huella imborrable en mi corazón y me ha impulsado a seguir adelante.

A mi tutor, Jorge Delgado, cuya guía, paciencia y conocimiento fueron fundamentales para el desarrollo de este trabajo. Su orientación y correcciones me ayudaron a perfeccionar cada detalle y a superar los desafíos propios de este proceso. Aprecio profundamente su profesionalismo y dedicación, que me sirvieron como ejemplo durante esta etapa académica.

A mis amigos y compañeros de estudio, por estar a mi lado en este camino. Gracias, Karen, Ghandy y Kevin, por compartir conmigo momentos de alegría, por su apoyo en los días difíciles y por hacer que esta experiencia fuera más significativa y enriquecedora.

Y, finalmente, agradezco a Dios, por ser mi guía y fortaleza en cada paso de este proyecto. Su presencia en mi vida ha sido un faro de esperanza y una fuente constante de inspiración que me ha impulsado a alcanzar esta meta.

ÍNDICE GENERAL

ÍNDICE DE TABLAS

ÍNDICE DE FIGURAS

RESUMEN

El proyecto de investigación tuvo como objetivo desarrollar una aplicación móvil multiplataforma destinada a mejorar la experiencia de compra y entrega de productos farmacéuticos en las farmacias Cruz Azul. Esta aplicación permite a los usuarios consultar productos disponibles, realizar pedidos, gestionar pagos seguros y rastrear entregas en tiempo real, optimizando así la accesibilidad y eficiencia en la adquisición de medicamentos.

Para el desarrollo de la aplicación, se utilizó un entorno DevOps con herramientas como Git, GitHub, Jenkins, Docker y SonarQube, que facilitaron la integración continua, la automatización de procesos y la mejora constante del software. Asimismo, se emplearon tecnologías como JWT (JSON Web Token) para la autenticación segura de usuarios, Flutter para el desarrollo del frontend y NestJS para el backend, logrando una aplicación segura, escalable y con una interfaz intuitiva.

El proyecto se basó en la metodología ágil Mobile-D, diseñada específicamente para el desarrollo de aplicaciones móviles, lo que permitió una gestión eficiente de los requisitos y una rápida adaptación a los cambios. Además, se utilizó el modelo de calidad FURPS para evaluar aspectos clave del software, como rendimiento. Para esta evaluación, se realizaron pruebas de simulación en tiempo real utilizando Apache JMeter, evaluando indicadores como tiempo de respuesta, consumo de recursos y disponibilidad.

Los resultados obtenidos demostraron que la aplicación cumple con los más altos estándares de calidad, destacándose por su rendimiento, seguridad y experiencia de usuario. Esto la convierte en una solución tecnológica altamente eficiente para optimizar los procesos de delivery farmacéutico y mejorar la atención al cliente.

Palabras claves: Aplicación móvil, delivery, Flutter, NestJS, FURPS.

ABSTRACT

The research project aimed to develop a cross-platform mobile application to enhance the purchasing and delivery experience of pharmaceutical products for Cruz Azul pharmacies. This application allows users to browse available products, place orders, manage secure payments, and track deliveries in real-time, thereby refining accessibility and efficiency in acquiring medications.

For the application's development, a DevOps environment was employed, using tools such as Git, GitHub, Jenkins, Docker, and SonarQube, which eased continuous integration, process automation, and ongoing software improvement. Technologies such as JWT (JSON Web Token) were used for secure user authentication, Flutter for the frontend development, and NestJS for the backend, resulting in a secure, scalable application with an intuitive user interface.

The project adopted the Mobile-D agile method, specifically designed for mobile application development, enabling efficient requirements management and quick adaptability to changes. Additionally, the FURPS quality model was applied to evaluate key aspects of the software, including performance, usability, and reliability. Real-time simulation tests were conducted using Apache JMeter to assess indicators such as response time, resource consumption, and availability.

The results proved that the application meets the highest quality standards, excelling in performance, security, and user experience. This makes it a highly efficient technological solution for improving pharmaceutical delivery processes and improving customer service.

Reviewed by:

Mgtr. Mishell Salao Espinoza
ENGLISH PROFESSOR
C.C. 0650151566

CAPÍTULO I. INTRODUCCIÓN

Las aplicaciones móviles de delivery en el sector farmacéutico han demostrado ser fundamentales para mejorar la accesibilidad de los usuarios a productos médicos y medicamentos. Con el auge de la digitalización, especialmente tras la pandemia, las farmacias han tenido que adaptarse rápidamente a la demanda de servicios en línea. La venta de medicamentos a través de plataformas digitales ha crecido significativamente, con algunas farmacias experimentando aumentos notables en sus ventas online, reflejando una tendencia creciente hacia la comodidad de las compras digitales [1].

El desarrollo de aplicaciones móviles para la compra de productos médicos ha emergido como una herramienta valiosa para mejorar la accesibilidad y eficiencia en la adquisición de insumos y medicamentos. Las aplicaciones móviles, como parte de la transformación digital en el sector de la salud, permiten a los usuarios realizar pedidos de manera rápida y conveniente, desde cualquier lugar, asegurando una atención más eficiente y oportuna. Además, estas aplicaciones no solo facilitan la compra, sino también el acceso a información médica relevante, la gestión de recetas electrónicas, y la posibilidad de realizar pagos de manera segura desde dispositivos móviles [2].

El desarrollo de aplicaciones móviles multiplataforma se ha convertido en una estrategia clave para optimizar el acceso a productos y servicios, especialmente en sectores como la farmacia. Con herramientas como Flutter, las aplicaciones móviles ahora permiten crear experiencias de usuario coherentes y eficientes tanto para iOS como para Android desde una sola base de código. Esto no solo reduce los costos y el tiempo de desarrollo, sino que también ofrece la posibilidad de ofrecer un servicio más accesible y ágil, que es crucial en el contexto de las farmacias [3].

La aplicación móvil multiplataforma de delivery para Farmacias Cruz Azul, desarrollada utilizando Flutter y NestJS, tiene como objetivo optimizar la experiencia del usuario, facilitando la realización de pedidos de medicamentos y productos farmacéuticos. El sistema estará diseñado para permitir a los usuarios consultar productos disponibles, realizar compras de manera rápida y hacer pagos seguros, brindando así una experiencia más ágil y conveniente.

1.1 Planteamiento del problema

Las farmacias Cruz Azul enfrentan una necesidad creciente de ofrecer acceso rápido y confiable a productos farmacéuticos, especialmente para clientes que requieren medicamentos urgentes, tienen dificultades de movilidad o buscan reducir salidas para evitar exposición a enfermedades. Sin embargo, aunque la demanda de servicios de entrega de productos de farmacia ha aumentado, muchas farmacias no cuentan con sistemas especializados para gestionar eficientemente los pedidos, seguimiento de entregas y pagos en un entorno digital seguro. Esto genera desafíos tanto para las farmacias Cruz Azul, que enfrentan dificultades para coordinar sus entregas, como para los clientes, quienes pueden experimentar demoras, falta de información en tiempo real y escasa transparencia en el estado de sus pedidos.

Para solucionar estos problemas, se propone desarrollar una aplicación móvil multiplataforma que permita a los clientes de Cruz Azul acceder eficientemente a productos farmacéuticos. La aplicación, creada con Flutter para el frontend y NestJS para el backend, facilitará la compra y entrega de medicamentos, integrando gestión de pedidos, rastreo en tiempo real y opciones de pago seguro.

El uso del modelo de calidad FURPS permitirá realizar una evaluación exhaustiva de la aplicación, midiendo el rendimiento clave que garanticen el cumplimiento de los requisitos funcionales y no funcionales, y proporcionen una experiencia de usuario satisfactoria.

1.2 Formulación del Problema

¿Cómo el desarrollo de una aplicación móvil multiplataforma de delivery para farmacias Cruz Azul puede mejorar la eficiencia, en la gestión de la información de pedidos de productos farmacéuticos?

1.3 Objetivos

General

Implementar una aplicación móvil multiplataforma de delivery para farmacias Cruz Azul utilizando Flutter y NestJS

Específicos

- Investigar la metodología ágil Mobile-D y las herramientas tecnológicas propuestas (como Flutter y NestJS) para la elaboración de una aplicación móvil multiplataforma de delivery en farmacias Cruz Azul.
- Desarrollar la aplicación móvil multiplataforma para delivery en farmacias Cruz Azul utilizando Flutter y NestJS.
- Evaluar el rendimiento de la aplicación móvil multiplataforma para delivery en farmacias Cruz Azul, utilizando el modelo de calidad FURPS.

CAPÍTULO II. MARCO TEÓRICO

2.1 Flutter

Flutter es un framework gratuito y de código abierto creado por Google en mayo de 2017, diseñado para desarrollar aplicaciones móviles nativas utilizando una sola base de código. Esto significa que los desarrolladores pueden escribir un único conjunto de código y crear aplicaciones para dos plataformas diferentes, iOS y Android, lo que simplifica y acelera el proceso de desarrollo. La principal ventaja de Flutter radica en su capacidad para permitir el desarrollo multiplataforma eficiente, haciendo que sea una herramienta muy valiosa para la creación de aplicaciones móviles [4].

El desarrollo de aplicaciones multiplataforma permite a los desarrolladores utilizar un único lenguaje de programación y base de código para crear aplicaciones que funcionen en varias plataformas, lo que resulta más económico y rápido en comparación con el desarrollo de aplicaciones nativas. Este enfoque facilita una experiencia de usuario coherente a través de diferentes dispositivos. Aunque tiene ciertas limitaciones, como el acceso restringido a las funcionalidades nativas de los dispositivos, tecnologías como Flutter han mejorado la fluidez y el rendimiento en el desarrollo de aplicaciones multiplataforma, brindando una solución eficiente para crear aplicaciones de alto rendimiento [5].

2.2 NestJS

NestJS es un marco de trabajo para aplicaciones backend que utiliza Node.js y TypeScript, permitiendo la construcción de aplicaciones escalables y eficientes. Ofrece una estructura modular que facilita la organización del código y la implementación de patrones arquitectónicos como el de microservicios. Además, es compatible con herramientas populares como GraphQL, WebSockets y bases de datos, lo que lo convierte en una opción robusta para crear aplicaciones de gran escala, especialmente en entornos empresariales. [6]

2.3 PostgreSQL

PostgreSQL es una base de datos de código abierto reconocida por su fiabilidad, flexibilidad y el cumplimiento de estándares técnicos abiertos. Su capacidad para gestionar tanto datos relacionales como no relacionales la convierte en una de las bases de datos más completas y estables disponibles. Además, su diseño versátil y accesible la hace una opción ideal para muchas empresas que buscan soluciones rentables y eficientes para optimizar sus sistemas de gestión de bases de datos, convirtiéndola en una solución "todo en uno" que se adapta a diversas necesidades. [7]

2.4 App móvil multiplataforma

Las aplicaciones de software son programas desarrollados en diversos lenguajes de programación con el propósito de realizar tareas específicas en dispositivos tecnológicos

como smartphones, tabletas, relojes inteligentes y televisores. Estas aplicaciones se distinguen por ser prácticas, fáciles de usar e instalar, además de ser dinámicas y adaptarse a las necesidades del usuario en diferentes dispositivos[8].

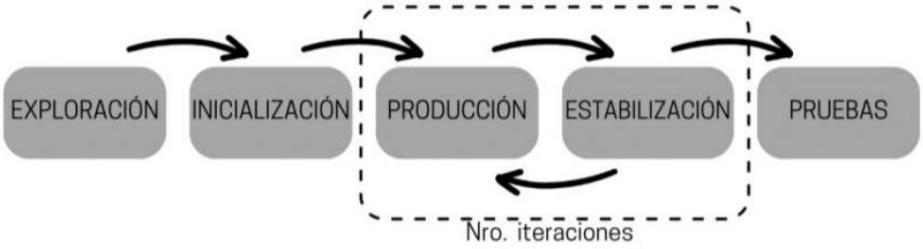
2.5 Metodologías Tradicionales y Ágiles

En el desarrollo de software, las metodologías tradicionales y las metodologías ágiles representan dos enfoques con diferencias significativas. Las metodologías ágiles se caracterizan por enfatizar la participación activa del cliente durante la planificación del proyecto y valoran la flexibilidad y la experiencia del equipo, que suele ser reducido en comparación con los equipos de enfoque tradicional[9]. Por el contrario, las metodologías tradicionales se centran en una planificación exhaustiva durante las etapas iniciales del proyecto. En este enfoque, la documentación es esencial, ya que en las primeras fases se definen los requisitos y los recursos necesarios, que se mantienen como base a lo largo de todo el desarrollo del software[10]

2.6 Mobile-D

Abrahamsson et al [12], citados por[13], desarrollaron la metodología Mobile-D, reconociendo que los procesos ágiles son los más adecuados para los entornos de desarrollo móvil. Mobile-D integra elementos de la metodología eXtreme Programming (XP), principios escalables de la familia de metodologías Crystal, y las etapas del ciclo de vida de Rational Unified Process (RUP). Esta metodología se estructura en cinco fases (ver Figura 2, proporcionando un enfoque ágil y adaptable para el desarrollo de aplicaciones móviles, optimizando los procesos y asegurando la calidad del producto final.[14]

Figura 1: Proceso de Mobile-D



Fuente: https://ieeexplore.ieee.org/document/9096751

2.7 Docker

Docker es una plataforma de software que facilita la creación, prueba y despliegue rápido de aplicaciones. Su enfoque se basa en empaquetar el software dentro de contenedores estandarizados, los cuales incluyen todo lo necesario para su funcionamiento, como

bibliotecas, herramientas del sistema, código y versiones ejecutables. Gracias a Docker, es posible implementar y escalar aplicaciones de manera eficiente en cualquier entorno, con la garantía de que el código se ejecutará de forma consistente[11].

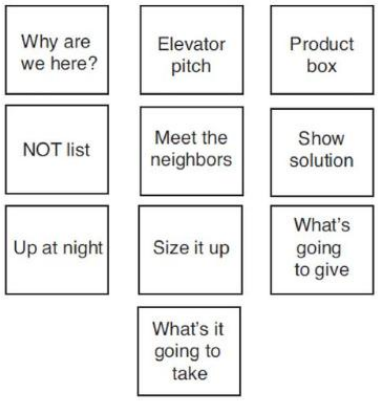
Figura 2: Docker

Fuente: https://www.vikingsoftware.com/services/technologies/what-is-docker/

2.8 Inception Deck

El Inception Deck es una herramienta fundamental en los proyectos ágiles, utilizada principalmente en la fase inicial de desarrollo. Su propósito es eliminar incertidumbres y proporcionar una visión clara del proyecto respondiendo a una serie de preguntas (ver Figura 3). Esto se logra a través de una serie de preguntas y ejercicios cuidadosamente seleccionados, diseñados para establecer expectativas claras y fomentar una comprensión compartida entre todos los involucrados, incluyendo el equipo de trabajo y las partes interesadas.[15]

Figura 3: El Inception Deck



Fuente: https://slideplayer.com/slide/12675391/

2.9 Fursp

De acuerdo con lo enunciado Yeomans y Rogers actualmente, existen diversos modelos utilizados para evaluar la calidad de las herramientas de software, desarrollados por distintas empresas y organizaciones con el propósito de verificar si un software cumple con los estándares de calidad a través del análisis de sus características. Uno de los modelos más destacados por su flexibilidad es el modelo FURPS, creado por Hewlett Packard. Este modelo permite evaluar diversos aspectos de calidad del software, abarcando desde la funcionalidad hasta la usabilidad y el rendimiento [16]. En la tabla se detalla los criterios de calidad que posee este modelo.

Tabla 1: Modelo de calidad FURPS

Atributos	Factores Contenidos	Descripción
Funcional	Factores Contenidos	El conjunto de características, capacidades y seguridad.
No funcional	Usabilidad	Puede incluir factores humanos, estética, coherencia en la interfaz de usuario, ayuda en línea y sensible al contexto, asistentes y agentes, documentación de usuario y materiales de formación
	Rendimiento	Condiciones sobre requisitos funcionales como velocidad, eficiencia, disponibilidad, precisión, rendimiento, tiempo de respuesta, tiempo de recuperación y uso de recursos
	Fiabilidad/ confiabilidad	Frecuencia y gravedad de los fallos, recuperabilidad, previsibilidad, precisión y tiempo medio entre fallos.
	Apoyabilidad/ capacidad de soporte	Testabilidad, extensibilidad, adaptabilidad, mantenibilidad, compatibilidad, configurabilidad, mantenibilidad, instalabilidad, localizabilidad

Fuente: https://ieeexplore.ieee.org/document/9096751

2.10 JMeter

Apache JMeter es una herramienta de código abierto desarrollada en Java, inicialmente creada en 1999 para realizar pruebas de carga en servidores HTTP, especialmente en el proyecto Apache Tomcat. Con el tiempo, JMeter se ha expandido para soportar pruebas en una variedad de protocolos de comunicación, convirtiéndose en una de las plataformas más populares a nivel mundial para evaluar el rendimiento de sitios web y aplicaciones. Usada ampliamente por ingenieros, desarrolladores y equipos de DevOps, JMeter ha sido adoptada por empresas destacadas como Google, Microsoft y Facebook, quienes la emplean para garantizar la eficiencia y el rendimiento de sus sistemas[17]

2.11 Google cloud

Google Cloud es una plataforma de servicios en la nube que proporciona herramientas y soluciones para la computación, almacenamiento, redes, y análisis de datos. Permite a las organizaciones ejecutar aplicaciones a gran escala utilizando la infraestructura global de Google. Además, ofrece avanzadas soluciones de inteligencia artificial, machine learning y almacenamiento de datos seguros. Su enfoque en escalabilidad y flexibilidad la convierte en una opción ideal para empresas de diferentes tamaños[18].

2.12 Api Google Geolocalización

La API de Google Geolocalización permite obtener la ubicación de los dispositivos móviles de manera precisa utilizando GPS, Wi-Fi y otras fuentes. Esta herramienta es fundamental para aplicaciones que requieren funcionalidades como geocodificación, seguimiento en tiempo real y análisis de distancias. Es ampliamente usada en aplicaciones de mapas, servicios de delivery y rastreo de ubicaciones, mejorando la experiencia del usuario al ofrecer servicios basados en su localización.[19]

2.13 Firebase

Firebase es una plataforma de desarrollo de aplicaciones diseñada por Google que ofrece un conjunto de servicios en la nube para facilitar la construcción, mejora y escalado de aplicaciones móviles y web.[20]

CAPÍTULO III. METODOLOGÍA

La investigación se enfoca en llevar a cabo un análisis cuantitativo con el objetivo de recopilar datos medibles sobre el desempeño de la aplicación móvil. Esto se logrará mediante la aplicación del modelo de calidad FURPS.

3.1 Tipo de investigación

Este estudio adopta un enfoque aplicado con diseño cuantitativo, enmarcado en un esquema cuasi-experimental. Se utilizarán técnicas estadísticas para medir la eficiencia, tiempo de respuesta y uso de recursos del sistema propuesto, minimizando posibles sesgos y garantizando la validez de los resultados obtenidos. El diseño cuasi-experimental permitirá evaluar el rendimiento del desarrollo de la aplicación en la gestión de pedidos y entregas de productos farmacéuticos.

3.2 Diseño de la Investigación

- Según el objeto de estudio

  Aplicada: La investigación tiene como propósito resolver un problema práctico específico para las farmacias Cruz Azul mediante el desarrollo de una solución tecnológica que optimice la gestión de pedidos, pagos y entregas

- Según la fuente de investigación.

  Bibliográfica: Durante la fase teórica, se recopilaron y analizaron datos provenientes de libros, artículos científicos, publicaciones académicas y documentación técnica sobre desarrollo de aplicaciones móviles y modelos de calidad.

3.3 Operacionalización de variables

En la Tabla 2, se proporciona una descripción de la operacionalización de variables del proyecto.

Tabla 2: Operacionalización de Variables

Problema	Tema	Objetivos	Variables	Indicadores
¿Cómo el desarrollo de una aplicación móvil multiplataforma de delivery para farmacias Cruz Azul puede mejorar la eficiencia, en la gestión de la información de pedidos de productos farmacéuticos?	Desarrollo de una Aplicación Móvil multiplataforma de Delivery para farmacias Cruz Azul utilizando Flutter y NestJS	**Objetivo general** • Implementar una aplicación móvil multiplataforma de delivery para farmacias cruz azul utilizando Flutter y NestJS **Objetivos específicos** • Investigar la metodología ágil Mobile-D y las herramientas tecnológicas propuestas (como Flutter y NestJS) para la elaboración de una aplicación móvil multiplataforma de delivery en farmacias Cruz Azul. • Desarrollar la aplicación móvil multiplataforma para delivery en farmacias Cruz Azul utilizando Flutter y NestJS. • Evaluar el rendimiento de la aplicación móvil multiplataforma para delivery en farmacias Cruz Azul, utilizando el modelo de calidad FURPS.	**Independiente:** Aplicación móvil multiplataforma delivery **Dependiente:** Evaluar el rendimiento de la aplicación móvil para gestión de compras y entrega de productos	**Independiente:** Número de módulos **Dependiente:** Criterios de rendimiento del modelo de calidad FURPS: Velocidad de Procesamiento. Tiempo de respuesta. % de consumo de recursos (RAM, procesador)

Fuente: El autor

3.4 Población y Muestra

Se trabajó teniendo en cuenta una población amplia correspondiente a los usuarios latentes del servicio de delivery de farmacias Cruz Azul. Esta decisión metodológica justifica la utilización de herramientas como JMeter para la recolección y simulación de datos, ya que permite emular cargas de trabajo simbólico sin depender de un universo finito o controlado. Al simular múltiples interacciones convergentes, se puede evaluar el rendimiento del sistema en condiciones similares a las que encararía en un entorno real de producción con alta demanda.

No aplica debido al tamaño de población que se usa en dicho caso de estudio

3.5 Identificación de variables

- **Variable independiente**

  Aplicación móvil multiplataforma delivery

- **Variable dependiente**

  Evaluar el rendimiento de la aplicación móvil para gestión de compras y entrega de productos

3.6 Técnicas e instrumentos de recolección de datos

Para la recolección de datos se empleó la técnica de herramienta informática, utilizando como instrumento el software Apache JMeter, el cual permitió realizar pruebas de carga y rendimiento sobre la aplicación móvil, obteniendo métricas objetivas que facilitaron la evaluación de su comportamiento en distintos escenarios de uso.

3.7 Técnicas de análisis e interpretación de la información

La herramienta JMeter fue empleada para evaluar el rendimiento de la aplicación móvil en diversas condiciones, considerando indicadores del modelo de calidad FURPS, tales como el número de solicitudes, el tiempo promedio de respuesta y el consumo de recursos.

3.8 Desarrollo de propuesta

El desarrollo de la aplicación móvil multiplataforma para farmacias Cruz Azul se llevó a cabo usando la metodología Mobile-D, la cual estructuró el proceso en cinco fases clave para garantizar un desarrollo. Este enfoque permitió optimizar cada fase del proyecto, desde la definición del alcance hasta la validación final, asegurando una solución tecnológica funcional y de calidad.

3.8.1 Exploración

En la etapa de exploración, se utilizó el Inception Deck como una ayuda para estructurar los requisitos iniciales y establecer una visión clara del producto. Este proceso se llevó a cabo de forma individual, respondiendo las preguntas clave del Inception Deck para establecer los objetivos del proyecto, las funcionalidades principales y los criterios de éxito, permitiendo una conceptualización precisa y organizada del desarrollo de la aplicación.

3.8.1.1 ¿Por qué estamos aquí? (Why are we here)

En respuesta a la primera pregunta del Inception Deck, "¿Por qué estamos aquí?", el objetivo principal de este proyecto es desarrollar una solución tecnológica que facilite y modernizar el acceso a medicamentos y productos farmacéuticos en las farmacias Cruz Azul. A través del desarrollo de una aplicación móvil multiplataforma, buscamos complacer la necesidad de los usuarios de adquirir estos productos de manera eficiente, segura y desde la comodidad de su hogar. Este esfuerzo también promueve la digitalización en el sector farmacéutico, ayudando a optimizar los procesos de compra y entrega para ofrecer una experiencia más ágil y confiable.

3.8.1.1.1 Requerimientos funcionales

Tabla 3: Requerimientos funcionales

Identificación	Requerimiento	Descripción	Prioridad
RF01	Registro de usuario	Los usuarios podrán registrarse proporcionando su nombre, correo electrónico y contraseña.	Alta
RF02	Inicio de sesión	Los usuarios registrados podrán iniciar sesión utilizando su correo electrónico y contraseña.	Alta
RF03	Catálogo de productos	Se mostrará un listado de productos farmacéuticos organizados por categorías y con información detallada.	Alta
RF04	Carrito de compras	Los usuarios podrán agregar productos a un carrito para gestionar sus pedidos antes de confirmar la compra.	Alta
RF05	Rastreo de pedidos	Los usuarios podrán rastrear el estado de sus pedidos en tiempo real desde la aplicación.	Alta
RF06	Gestión de ubicaciones	Permitir a los usuarios registrar y seleccionar direcciones para la entrega utilizando geolocalización.	Alta

Fuente: El autor

3.8.1.1.2 Requerimientos no funcionales

Tabla 4: Requerimientos no funcionales

Identificación	Descripción	Categoría
RF01	La aplicación móvil será compatible con las principales versiones de sistemas operativos móviles.	Compatibilidad

RF02	El diseño modular de la aplicación seguirá buenas prácticas de programación para facilitar su mantenimiento futuro.	Mantenibilidad
RF03	El sistema garantizará tiempos de respuesta menores a 2 segundos en las interacciones del usuario.	Rendimiento
RF04	La interfaz de usuario será intuitiva, amigable y coherente, ofreciendo una experiencia visual atractiva.	Interfaz
RF05	La aplicación garantizará seguridad en los datos, implementando cifrado y cumpliendo con regulaciones de privacidad.	Seguridad
RF06	El sistema será escalable para gestionar múltiples usuarios concurrentes sin afectar su rendimiento.	Escalabilidad

Fuente: El autor

3.8.1.2 Discurso de Ascensor (Elevator Pitch)

La siguiente técnica en el Inception Deck es el Elevator Pitch, que implica presentar de forma atractiva, breve y veloz la idea central del proyecto:

"Una aplicación móvil multiplataforma diseñada para las farmacias Cruz Azul, que permite a los clientes consultar un catálogo de medicamentos, realizar pedidos de manera eficaz y segura, efectuar pagos en línea y rastrear en tiempo real el estado de sus entregas. Con una interfaz amigable y un diseño centrado en la usabilidad, esta solución busca mejorar la accesibilidad y optimizar la experiencia de compra, fortaleciendo la transformación digital en el sector farmacéutico."

3.8.1.3 Caja del Producto (Product Box)

Esta técnica se usa con el propósito de generar de manera ágil una representación visual del proyecto. Entre los beneficios que proporciona al usuario se encuentran características específicas acompañadas de un eslogan claro y directo. A modo de ejemplo, se presenta el siguiente Product Box en la Figura 4:

Figura 4: Caja del Producto

Fuente: El autor

3.8.1.4 No enlistado (Not List)

Dentro del Inception Deck se incluye una sección denominada "Not List", donde se identifican las características y funcionalidades que no serán consideradas dentro del alcance del proyecto. Esta técnica asegura que en el futuro no se añadan elementos que no aporten valor al producto. A continuación, se presentan las características excluidas en la Tabla 6.

Tabla 5: Not List

LO QUE NO ESTÁ AL ALCANCE	SIN RESOLVER
Integración con redes sociales	La aplicación móvil será compatible con las principales versiones de sistemas operativos móviles.
Opciones avanzadas de pago, como PayPal o criptomonedas	Interfaz multilingüe
Soporte para pedidos internacionales	Implementación de autenticación biométrica

Fuente: El autor

3.8.1.5 Conoce a tus vecinos (Meet Your Neighbors)

En el punto "Meet Your Neighbors", se identifican a los actores o elementos clave que tendrán influencia directa o indirecta en el proyecto. Aunque este es un proyecto individual, este apartado te ayudará a mapear a los usuarios finales, los beneficiarios indirectos, las influencias externas y los recursos que pueden apoyar el desarrollo y éxito de tu aplicación. Esto asegura que mantengas un enfoque claro en las necesidades y expectativas del entorno que rodea tu proyecto.

Tabla 6: Conoce a tus vecinos

Categorías	Descripción
Usuarios finales	Clientes que necesitan una solución rápida, confiable y segura para adquirir medicamentos.
Stakeholders externos	Farmacias Cruz Azul, quienes se beneficiarán del aumento en ventas y eficiencia operativa
Recursos de apoyo	Comunidades técnicas, foros y documentación para herramientas de desarrollo como Flutter y NestJS

Fuente: El autor

3.8.2 Inicialización

El objetivo principal de esta fase dentro de la metodología Mobile-D es garantizar el éxito del proyecto a través de una planificación clara de los recursos necesarios y el tiempo estimado para su desarrollo. Siguiendo las directrices establecidas en la fase de exploración, esta etapa da inicio a la ejecución de las técnicas restantes del Inception Deck.

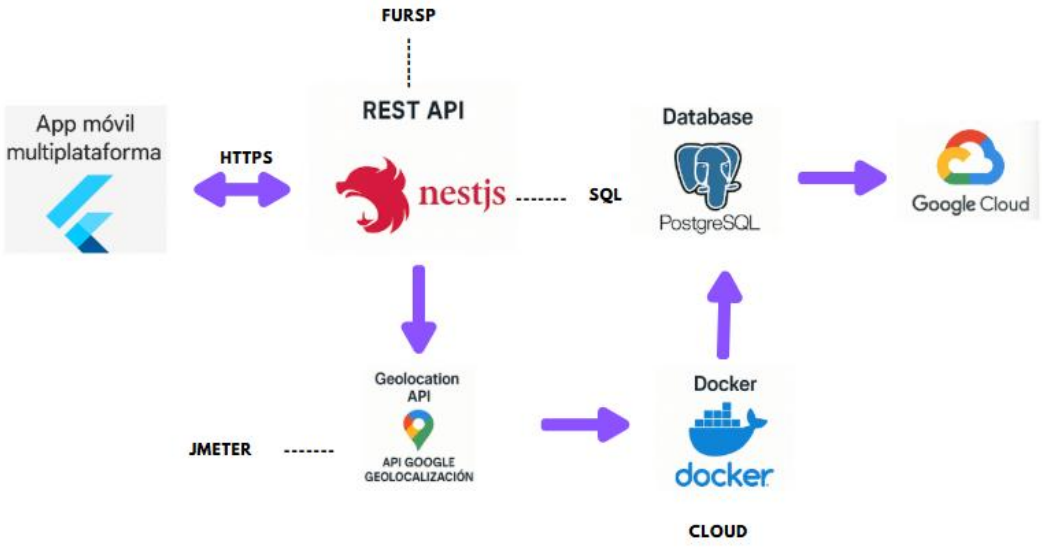
3.8.2.1 Mostrar la solución (Show Your Solution)

En esta técnica se presenta de manera clara y estructurada la solución al problema identificado, proporcionando una visión general de alto nivel sobre sus componentes principales, así como la arquitectura o diseño propuesto.

Arquitectura del Sistema

El proyecto implementa una arquitectura cliente-servidor (ver Figura 5) en la cual la aplicación móvil se comunica con un servidor mediante internet. Las solicitudes enviadas por los usuarios desde la app son procesadas por el servidor, que se encarga de gestionar los datos almacenados en una base de datos y responder con la información requerida. Para el correcto funcionamiento del sistema, es indispensable que los usuarios cuenten con acceso a internet.

Figura 5: Arquitectura del sistema

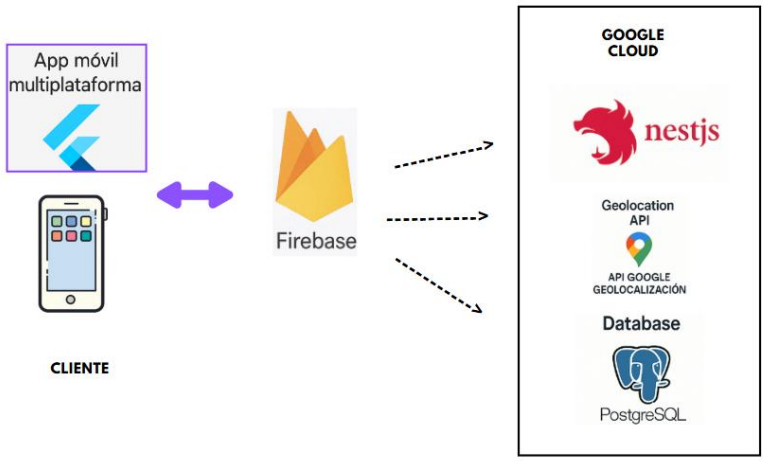


Fuente: El autor

Arquitectura de implementación

En el proyecto, los usuarios acceden a la aplicación móvil desde sus dispositivos, la cual se comunica con un servidor utilizando una API REST. Este servidor aloja la base de datos, donde se almacenan los datos esenciales para el correcto funcionamiento de la aplicación. La arquitectura incluye una comunicación eficiente entre la aplicación móvil, el servidor y la base de datos, como se detalla en la Figura correspondiente.

Figura 6: Arquitectura de implementación

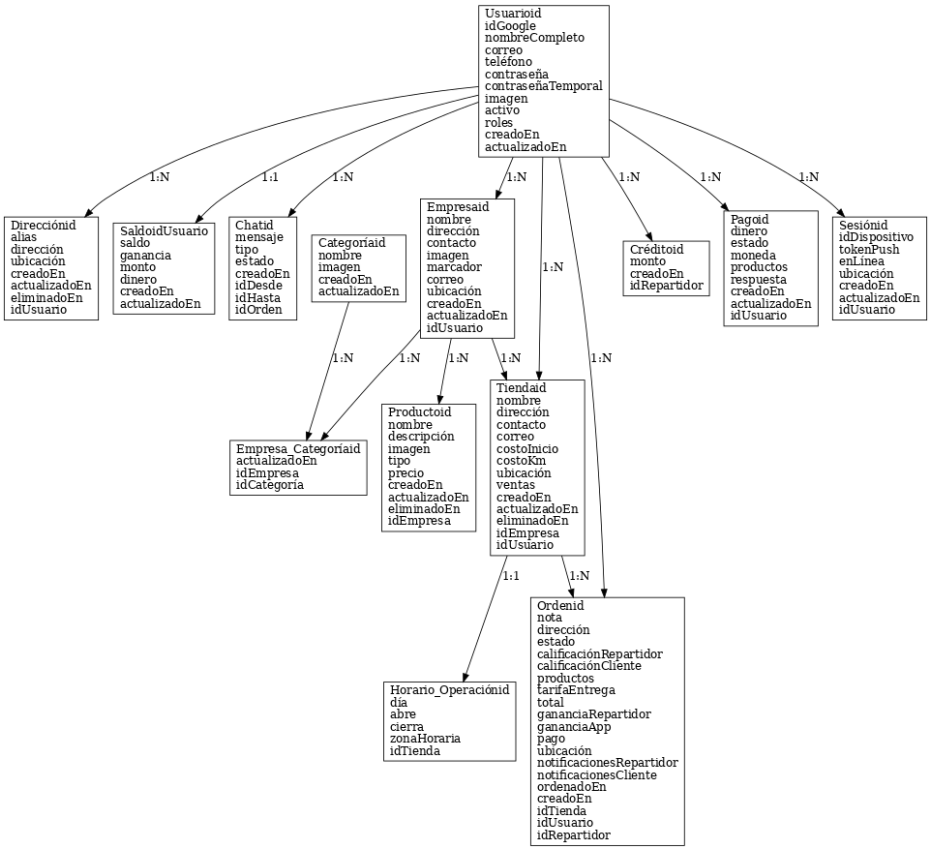


Fuente: El autor

Diagrama de Base de Datos

Se utilizó PostgreSQL como sistema de gestión de bases de datos relacional, en conjunto con TypeORM(ver Anexo 1), para modelar las relaciones entre las diferentes entidades de datos. Este enfoque permite un diseño de base de datos eficiente, con relaciones sólidas y una gestión optimizada del almacenamiento. El modelo resultante se presenta en la Figura correspondiente, mostrando la estructura organizada y las interconexiones de las tablas.

Figura 7: Diagrama de base de datos



Fuente: El autor

3.8.2.2 ¿Qué nos mantiene despiertos en la noche? (What Keeps us Up at Night?)

El propósito de esta técnica es identificar y analizar los riesgos potenciales que podrían impactar negativamente el proyecto, priorizando aquellos con mayor impacto. Se busca implementar medidas preventivas y estrategias de mitigación desde el inicio del proyecto para garantizar su éxito y minimizar posibles problemas, como se detalla en la Tabla correspondiente.

Tabla 7: ¿Qué nos mantiene despiertos en la noche?

Categorías	Riesgos
Riesgos de gestión	Mala planificación del proyecto.
	Falta de formación del desarrollador.
Riesgos tecnológicos	Fallas en la aplicación.
	Disponibilidad de los servicios de proveedores.
	Cambios en los servicios de proveedores.

Fuente: El autor

En el desarrollo de la aplicación se identificaron los siguientes riesgos:

- **Riesgo de mala planificación del proyecto:** Existe la posibilidad de que una planificación inadecuada cause retrasos en el cronograma y aumentos en los costos del desarrollo. Se recomienda un seguimiento continuo del plan y la ejecución de revisiones periódicas para ajustar las actividades según sea necesario.
- **Riesgo de falta de formación del desarrollador:** Si el desarrollador no cuenta con los conocimientos técnicos necesarios, podría haber retrasos o problemas en el proyecto. Se sugiere invertir en capacitación o acceder a recursos formativos relacionados con las tecnologías utilizadas.
- **Riesgo de fallas en la aplicación:** Las fallas o errores en la funcionalidad de la aplicación podrían impactar la experiencia del usuario. Es fundamental realizar pruebas exhaustivas durante todo el ciclo de desarrollo para minimizar estos problemas.
- **Riesgo de disponibilidad de servicios de proveedores:** Dependencia de terceros para servicios como pasarelas de pago o alojamiento. Es importante identificar alternativas viables y mantener redundancias para garantizar la continuidad del sistema.
- **Riesgo de cambios en los servicios de proveedores:** Los proveedores pueden modificar o descontinuar sus servicios sin previo aviso, lo que podría impactar la funcionalidad de la aplicación. Se recomienda monitorear activamente las políticas de los proveedores y mantener planes de contingencia actualizados.

3.8.2.3 Estimar el Tamaño Size It UP

Para este proyecto independiente, se establece un roadmap con las diferentes iteraciones (releases) que guían el proceso de desarrollo, priorizando la implementación gradual de funcionalidades clave. Cada release representa una entrega del software con incrementos funcionales completos, diseñados para asegurar el avance progresivo hacia los objetivos del proyecto.

Tabla 8: Roadmap de las iteraciones

Roadmap Viality			
Release 0	**Release 1**	**Release 2**	**Release 3**
RF01	RF02	RF03	• RF04

Fuente: El autor

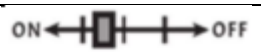
Objetivos de cada iteración:

- Release 0: Configuración del entorno de desarrollo, estableciendo las bases necesarias para comenzar con la implementación del proyecto (herramientas de desarrollo y estructura del código).
- Release 1: Desarrollo de la funcionalidad de registro e inicio de sesión de usuarios.
- Release 2: Implementación del catálogo de productos con búsqueda y selección interactiva.
- Release 3: Finalización de la funcionalidad de pedidos y rastreo en tiempo real para los usuarios.

3.8.2.4 ¿Qué vamos a ceder? (What's Going to give?)

El propósito de este punto es identificar y priorizar los factores clave que influyen en el proyecto, conocidos como los "Furious Four": presupuesto, alcance, tiempo y calidad. Estos factores se visualizan como un ecualizador, lo que permite ajustar cada uno de ellos en función de las necesidades y restricciones del proyecto

Tabla 9: Ecualizador de factores

Furious Four	
	Presupuesto
	Alcance
	Tiempo
	Calidad

Fuente: https://agilewarrior.wordpress.com/2010/11/06/the-agile-inception-deck/

Con lo expuesto en la tabla anterior, se puede expresar las siguientes conclusiones:

- **Presupuesto:** En un proyecto individual como este, el presupuesto es flexible debido al uso de herramientas gratuitas o de bajo costo como Flutter y NestJS. Esto permite reducir los gastos, aunque siempre es importante monitorear cualquier incremento inesperado.
- **Alcance:** El alcance puede ser modificado si se presentan cambios en los requisitos o necesidades adicionales. Sin embargo, es importante mantener el enfoque en los objetivos principales para evitar desviaciones significativas.

- **Tiempo:** Se establece una fecha límite para la entrega del proyecto, que debe ser respetada para cumplir con los plazos establecidos. Dado que es un proyecto independiente, gestionar el tiempo de manera eficiente es clave para evitar retrasos.
- **Calidad:** La calidad del software no debe comprometerse, aunque pueden realizarse ajustes en caso de limitaciones de tiempo o presupuesto. Esto asegura que el producto final sea funcional y cumpla con las expectativas.

Los valores de los factores están interconectados, lo que significa que cualquier modificación en uno de ellos impactará a los demás. Por ejemplo, si se amplía el alcance, puede ser necesario ajustar el presupuesto o el tiempo. Es fundamental equilibrar estos elementos para garantizar el éxito del proyecto

3.8.2.5 ¿Qué se necesita para lograrlo? (What's Going to Take?)

En este punto se realiza una estimación de los costos del proyecto considerando el tiempo, los recursos humanos y los servicios tecnológicos requeridos. Esto permite establecer un presupuesto general y planificar adecuadamente los recursos necesarios para la ejecución del proyecto (ver Tabla 11).

Tabla 10: Presupuesto

Cantidad	Concepto	Cobro Mensual	Total 4 Meses
1	Personal	0$	0$
2	Servidores	0$	0$
	Total	0$	0$

Fuente: El autor

Al finalizar las herramientas del Inception Deck, se alcanza la meta de la fase de Inicialización, que incluye:

- Comprender la infraestructura necesaria para el desarrollo del proyecto.
- Definir y mitigar los riesgos potenciales.
- Establecer un roadmap claro de iteraciones y funcionalidades.
- Determinar las prioridades de los factores críticos del proyecto.

Con esta base, se detalla el presupuesto con el que contará el proyecto, asegurando que los costos estén controlados y se ajusten a los recursos disponibles. En este caso, como se trata de un proyecto independiente y se utilizan herramientas gratuitas, los costos directos son mínimos o inexistentes.

3.8.2.6 Herramienta de desarrollo

En el desarrollo del proyecto se emplearon diversas herramientas para garantizar una implementación eficiente y funcional. Las herramientas utilizadas son expuestas en el marco teoría del documento.

3.8.3 Producción y Estabilización

El proceso de producción y estabilización para el desarrollo de la aplicación móvil para farmacias se llevó a cabo de manera estructurada, utilizando un servidor Ubuntu en Google Cloud como entorno base. En esta fase, se implementaron las configuraciones necesarias para garantizar el funcionamiento correcto del backend, el frontend y la comunicación entre ellos a través de un proxy inverso configurado en NGINX. Además, se usaron herramientas como Docker para manejar contenedores y Node.js para la ejecución del backend.

3.8.3.1 Configuración del Servidor

Se configuró un servidor Ubuntu proporcionado por Google Cloud, asegurándose de que contara con las dependencias necesarias, como Node.js y NGINX. A continuación, se realizó la instalación y configuración inicial de Docker para la gestión de contenedores.

Tabla 11: Implementación del Servidor en Google Cloud

Etapa	Descripción
Configuración inicial del servidor	Se aprovisionó un servidor Ubuntu en Google Cloud. Se instalaron y configuraron dependencias imprescindibles como Node.js, NGINX y Docker.
Instalación de Docker	Se instaló Docker para gestionar contenedores, simplificar la administración del entorno de backend y base de datos.
Contenerización de aplicaciones	Se empaquetaron las aplicaciones del backend (NestJS) y la base de datos PostgreSQL en contenedores individuales usando Docker Hub.

Fuente: El autor

Servidor Backend

Para la implementación del servidor backend, se utilizó NGINX como proxy inverso para

Tabla 12: Implementación del Servidor Backend

Componente	Descripción
Servidor Backend (NGINX + NestJS)	Se configuró NGINX como proxy inverso para redirigir las solicitudes entrantes al servidor NestJS en el puerto 3000. Se ajustaron parámetros como: • Límite de solicitud (12 MB) • Redirección de tráfico • Manejo de errores 404 y 500.

NGINX	Se definieron reglas en nginx.conf para manejar rutas HTTP eficientemente y garantizar un flujo ordenado hacia el backend.
Node.js (NestJS y PM2)	El backend se desarrolló en Node.js con NestJS. Se desplegó en Ubuntu y se utilizó PM2 como administrador de procesos para mantener el servicio activo en producción.
Base de datos PostgreSQL	PostgreSQL se desplegó en un contenedor Docker. Se crearon relaciones según el modelo de datos, asegurando integridad referencial.
Comunicación entre componentes	Se realizaron pruebas de comunicación entre frontend, backend y base de datos mediante rutas definidas en la API REST y herramientas de monitoreo.

Fuente: El autor

3.8.3.2 Desarrollo

Durante la fase de desarrollo, se realizaron diversas actividades clave para lograr la creación del producto. A continuación, se describen las tareas principales realizadas en esta etapa.

3.8.3.2.1 Diseño

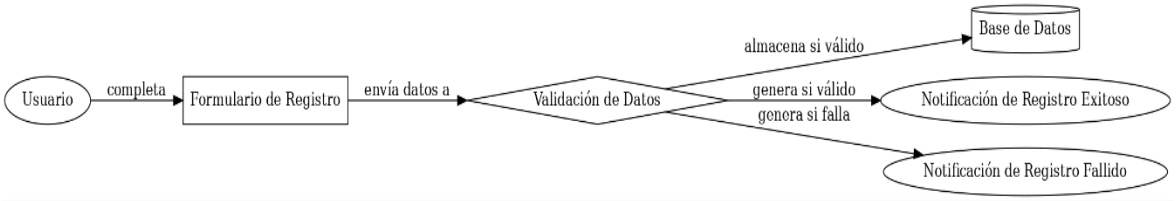
Como parte fundamental del proceso de diseño de la aplicación, se utilizaron diagramas y mapas mentales para conceptualizar las diferentes pantallas y flujos de interacción del sistema. En esta fase, se puso especial énfasis en el diseño de las pantallas para el registro de usuarios, la navegación dentro del sistema, y la experiencia del usuario en general.

Diseño Conceptual

El diseño conceptual de estas pantallas buscó ofrecer una representación clara de los elementos visuales y las interacciones que deben realizar los usuarios. Además, se priorizó la usabilidad y la seguridad, asegurando que cada proceso fuera eficiente y que los usuarios pudieran completar las tareas necesarias sin dificultades.

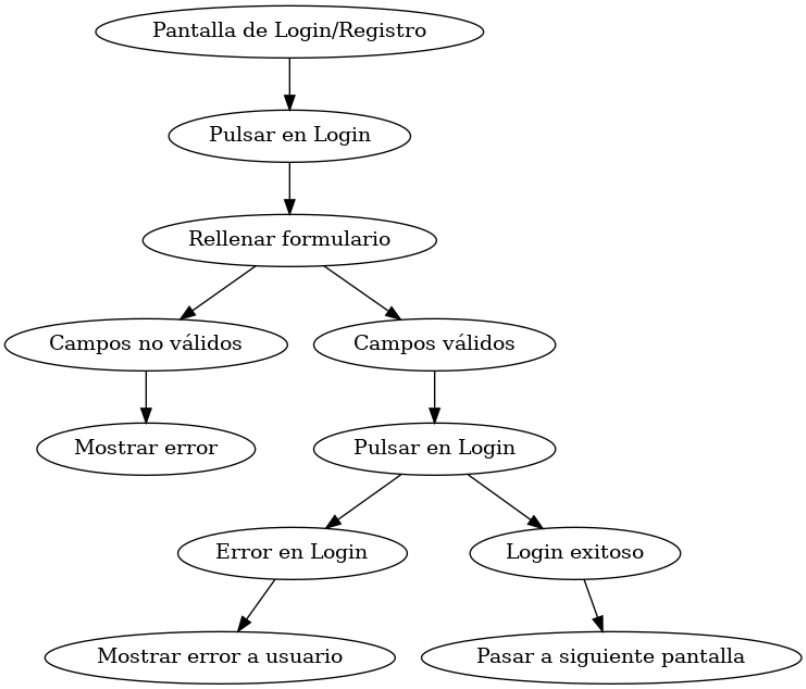
A continuación, se presenta un esquema conceptual de cómo se desarrolló el proceso de registro de usuarios

Figura 8: Registro de usuarios



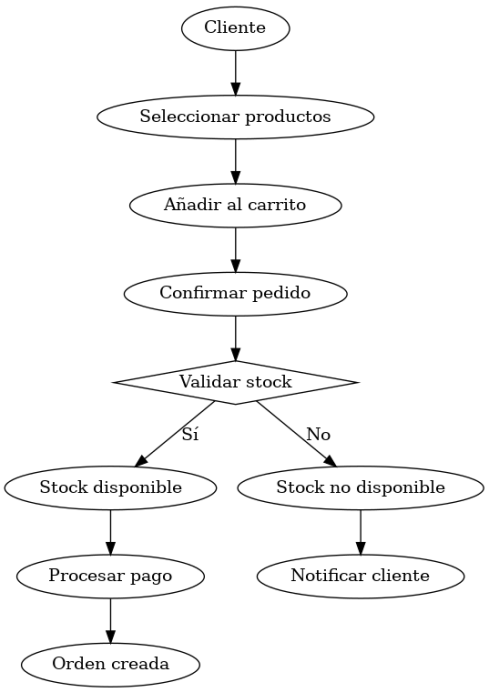
Fuente: El autor

Figura 9: Inicio de sesión



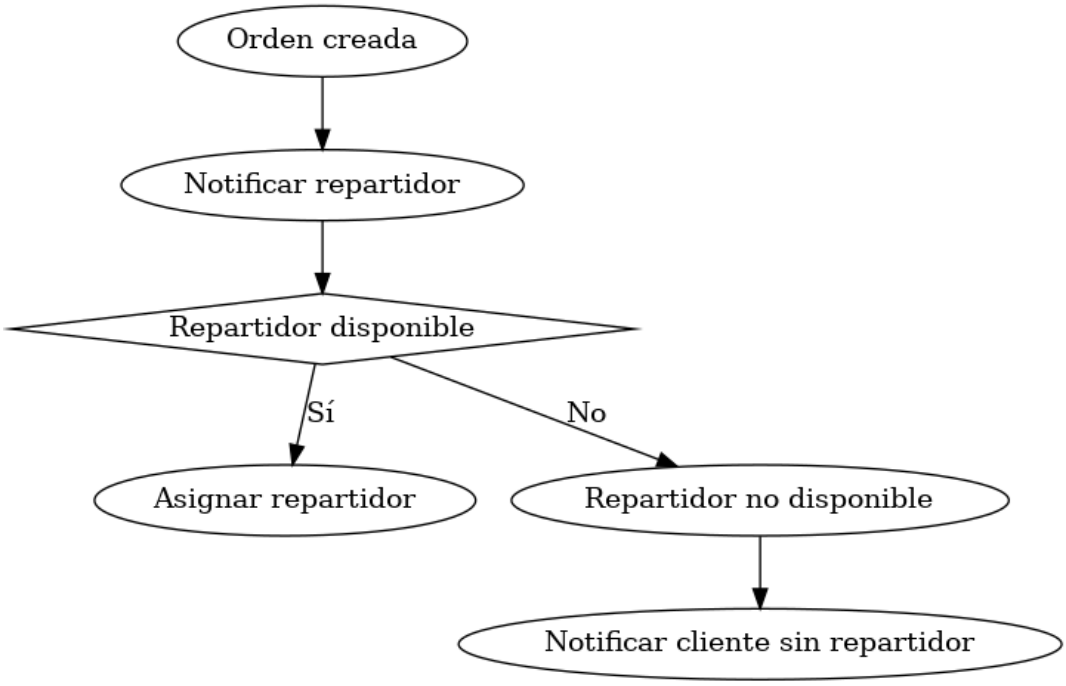
Fuente: El autor

Figura 10: Solicitud de pedido



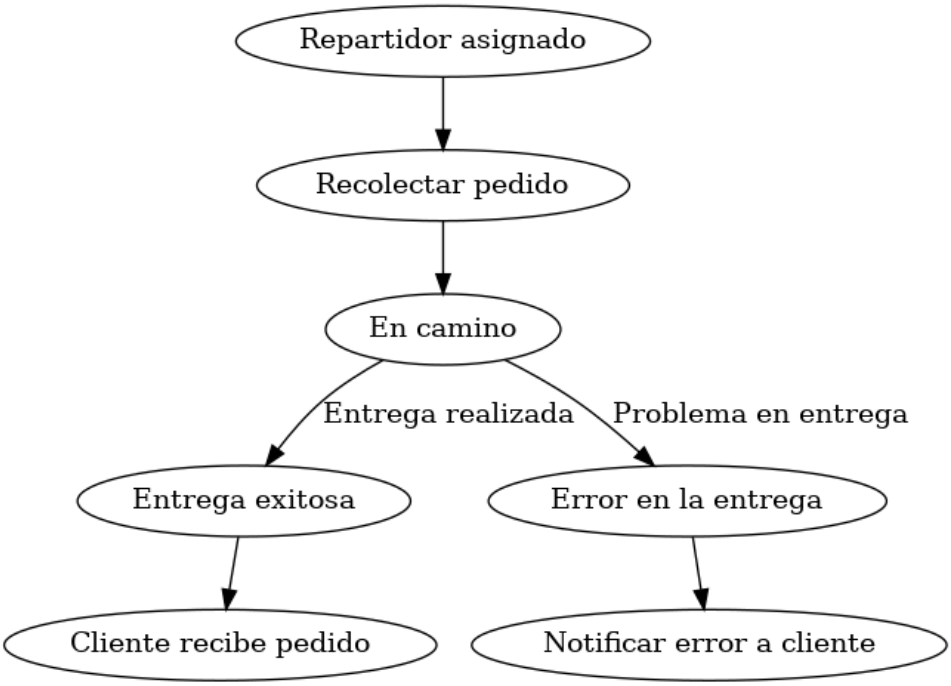
Fuente: El autor

Figura 11: Aceptación de pedido



Fuente: El autor

Figura 12: Entrega de pedido



Fuente: El autor

Diseño Interfaz

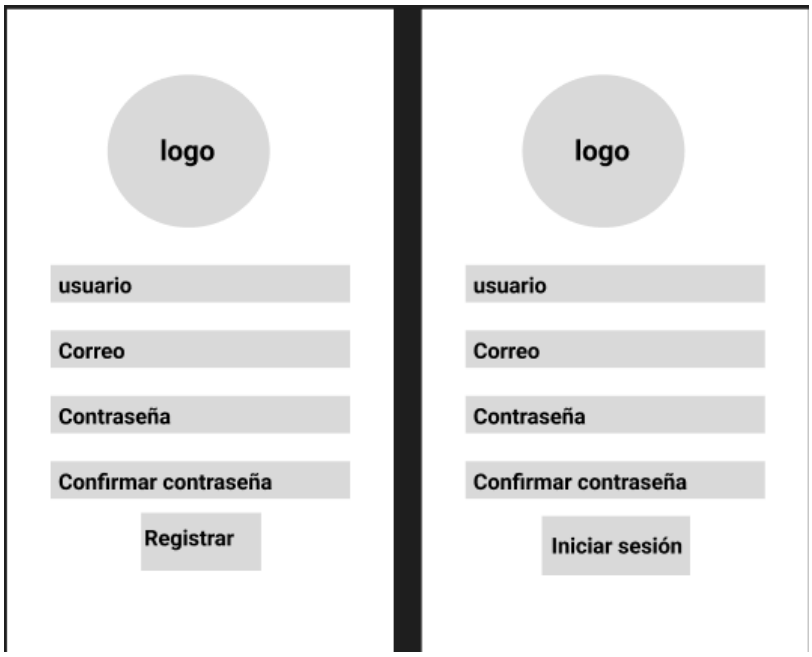
Con base en el diseño conceptual, se desarrollaron las interfaces para las diferentes pantallas de la aplicación de delivery de farmacias utilizando exclusivamente la herramienta Figma.

El objetivo principal fue crear diseños visualmente atractivos, intuitivos y bien organizados que permitan a los usuarios acceder fácilmente a las funcionalidades de la aplicación.

Para la interfaz relacionada con la gestión de pedidos, se diseñaron pantallas que priorizan la experiencia del usuario mediante una disposición clara de los elementos. Estas pantallas incluyen formularios sencillos para seleccionar productos, confirmar el pedido y elegir opciones de pago. Cada elemento fue optimizado para garantizar una navegación fluida y una interacción intuitiva.

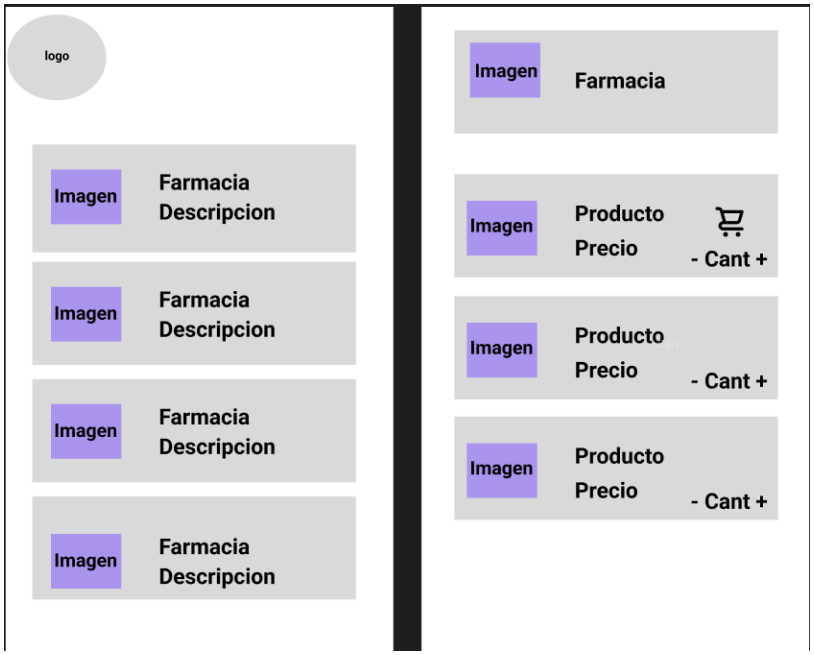
Además, el diseño de las pantallas para el seguimiento de pedidos se completó asegurando que toda la información necesaria estuviera claramente presentada. Esto permitió a los usuarios rastrear sus pedidos en tiempo real, recibir notificaciones sobre el estado del delivery y calificar el servicio, todo dentro de un entorno visual limpio y fácil de usar.

Figura 13: Diseño en Figma Registro Usuario e Inicio de Sesión



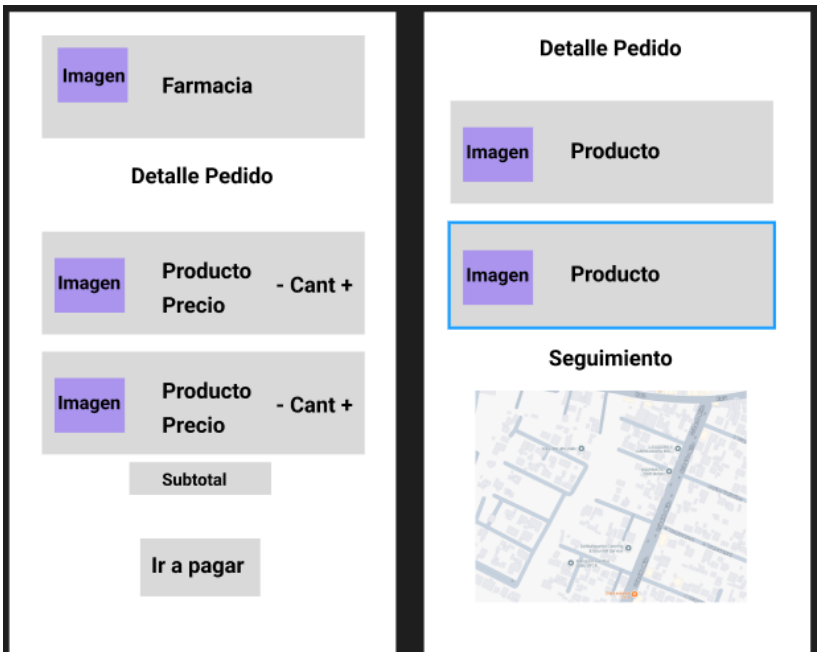
Fuente: El autor

Figura 14: Menú inicial y pantalla de farmacia.



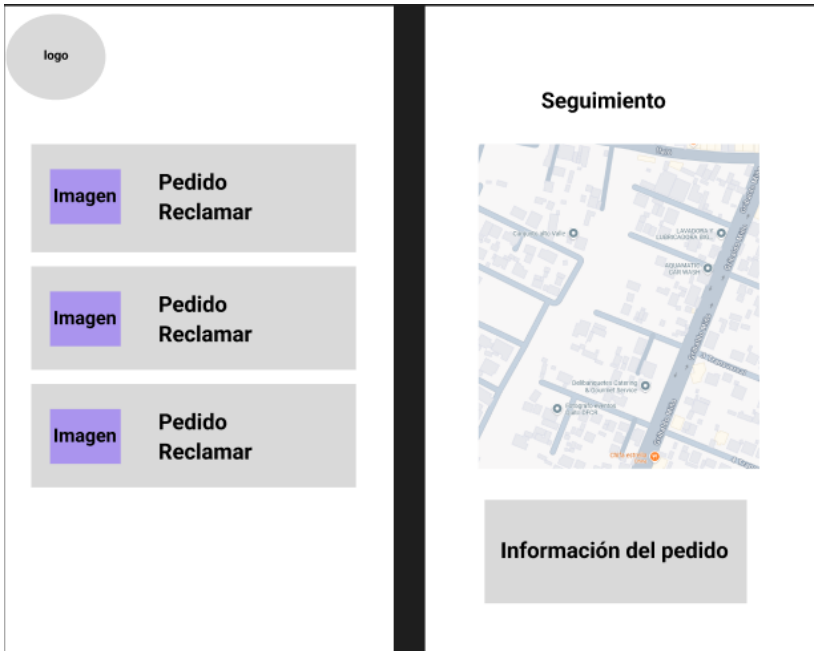
Fuente: El autor

Figura 15: Ir a pagar pedido y seguimiento de pedidos.



Fuente: El autor

Figura 16: Menú delivery y seguimiento.



Fuente: El autor

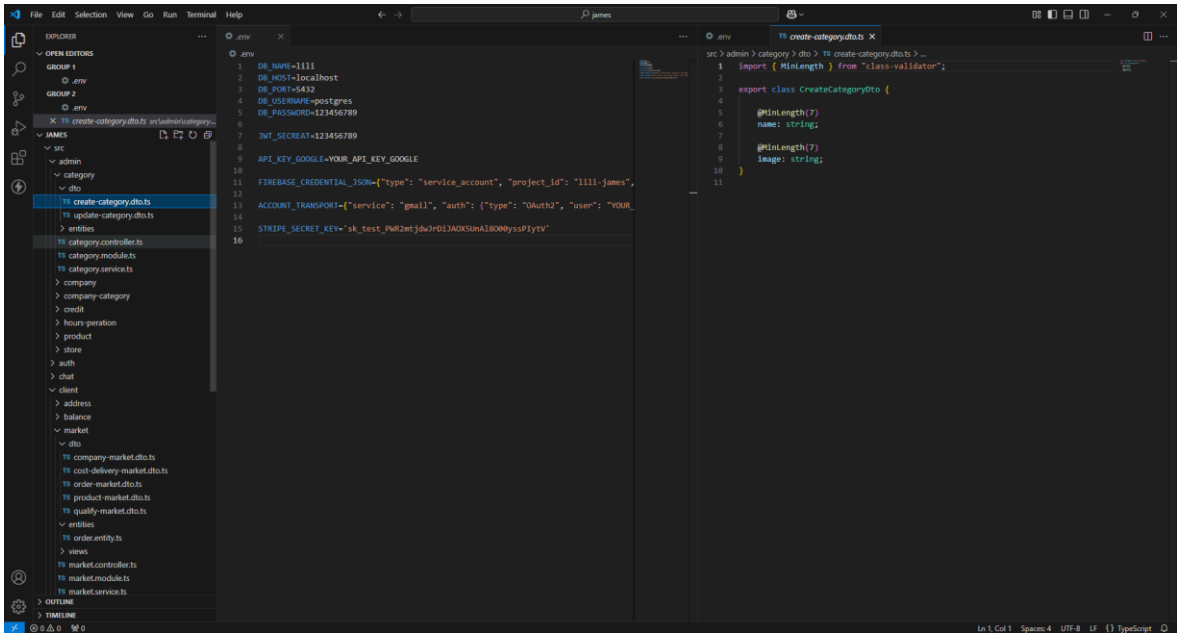
Figura 17: Calificación de delivery y cliente.

Fuente: El autor

Backend

La implementación del backend mostrada en las imágenes está desarrollada utilizando el framework NestJS, que proporciona una estructura modular para construir aplicaciones web robustas y escalables.

Figura 18: Código de implementación framework Nestjs.



Fuente: El autor

Lanzamiento

Tras completar el desarrollo de la aplicación móvil, se realizaron pruebas específicas enfocadas en el sistema operativo Android. Para ello, se generó un archivo APK que contenía todos los elementos esenciales para la instalación en dispositivos Android. Posteriormente, se llevó a cabo la instalación en un dispositivo Android mediante el archivo APK, validando el correcto funcionamiento y asegurando la compatibilidad de la aplicación con dicho entorno. Con estos pasos realizados, la aplicación quedó lista para su lanzamiento y disponible para los usuarios de dispositivos Android.

3.8.4 Pruebas

Se llevó a cabo un análisis inicial de los indicadores clave relacionados con el rendimiento de la aplicación, ya que estos aspectos son cruciales para garantizar la calidad de la experiencia del usuario. Posteriormente, se desarrolló una aplicación móvil que incluía funcionalidades esenciales, como autenticación, registro de usuarios, modificación de datos y visualización de información.

El rendimiento de la aplicación fue evaluado utilizando JMeter, simulando un total de 10 solicitudes con diferentes objetivos y un intervalo de 1 segundo entre cada una. Esta prueba permitió observar cómo respondía la aplicación bajo carga. La evaluación se realizó en un dispositivo Android con especificaciones particulares.

Finalmente, se recopilaron, tabularon y analizaron los datos obtenidos durante las pruebas. Los resultados fueron representados mediante gráficos estadísticos y comparados con los criterios definidos en el modelo de calidad FURPS (Rendimiento).

CAPÍTULO IV. RESULTADOS Y DISCUSIÓN

En el marco del presente proyecto, se ha culminado el desarrollo de una aplicación móvil diseñada específicamente para mejorar la gestión de la entrega de medicamentos y productos farmacéuticos a domicilio. Como parte de los objetivos principales, se planteó medir el rendimiento y la eficiencia de la aplicación a través de pruebas específicas que validaran su funcionalidad y desempeño.

Parámetros de evaluación

El análisis de indicadores clave resulta esencial para evaluar y optimizar la calidad y el desempeño de la aplicación. En la siguiente tabla se presentan los indicadores seleccionados para esta evaluación, los cuales desempeñan un papel crucial en garantizar un rendimiento óptimo y una experiencia de usuario satisfactoria.

Tabla 13: Parámetros de Evaluación

Tipo de Requerimiento	Indicadores
Rendimiento	Eficiencia Tiempo de respuesta Utilización de recursos: • Uso CPU • Uso de memoria RAM • Uso de almacenamiento

Fuente: El autor

Dispositivo utilizado para la evaluación

En la Tabla 15 se observan las características del dispositivo móvil utilizado.

Tabla 14: Características del dispositivo

Descripción	Dispositivo móvil Android
Marca	Pixel 9 Pro
Modelo	XL API 35
Procesador	Google Tensor G4
Versión sistema operativo	Android 14
RAM	8 GB
Almacenamiento	128GB

Fuente: El autor

4.1 Resultados

Los resultados de cada parámetro de evaluación fueron obtenidos utilizando Apache JMeter, una herramienta versátil diseñada para simular pruebas de carga (ver Anexo correspondiente). Durante estas pruebas, se llevaron a cabo un total de 10 solicitudes simuladas por segundo. Posteriormente, se calculó la media ponderada en porcentaje de cada indicador, siguiendo las directrices del modelo de calidad FURPS. Este enfoque permitió realizar una evaluación más precisa y detallada de los resultados obtenidos.

4.1.1 Simulación en el dispositivo móvil

Tabla 13: Parámetros de Evaluación

N	Tiempo de respuesta (s)	CPU%	RAM%	Almacenamiento%
1	0,92s	8	10	2
2	0,97s	9	9	1
3	0,98s	9	8	2
4	0,10s	8	10	3
5	0,11s	10	8	2
6	0,94s	10	8	1
7	0,10s	8	9	3
8	0,101s	9	10	2
9	0,11s	9	8	1
10	0,9s	8	11	3

Fuente: El autor

4.1.2 Análisis de los resultados

Una vez concluidas las pruebas, se obtuvieron datos cuantitativos que fueron analizados. La Tabla 16 presenta los promedios correspondientes a dichos resultados(, los cuales fueron obtenidos mediante las pruebas de carga realizadas con la herramienta Apache JMeter.

Tabla 15: Parámetros de Evaluación

Indicadores	Tiempo de respuesta (s)
Eficiencia	100%
Tiempo de respuesta	0.82s
CPU	8.6%
RAM	9.3%
Almacenamiento	2%

Fuente: El autor

Comparación de valora obtenidos y establecidos en el modelo FURPS

Es fundamental contrastar los resultados obtenidos en las pruebas con los valores definidos en el modelo de calidad FURPS, ya que esto permite verificar si la aplicación satisface los estándares de rendimiento esperados. La Tabla 17 ofrece una perspectiva clara sobre este aspecto. Cabe destacar que los valores presentados corresponden a pruebas realizadas bajo una carga simulada de 10 solicitudes por segundo.

Tabla 16: Comparación de valores obtenidos con el modelo FURPS

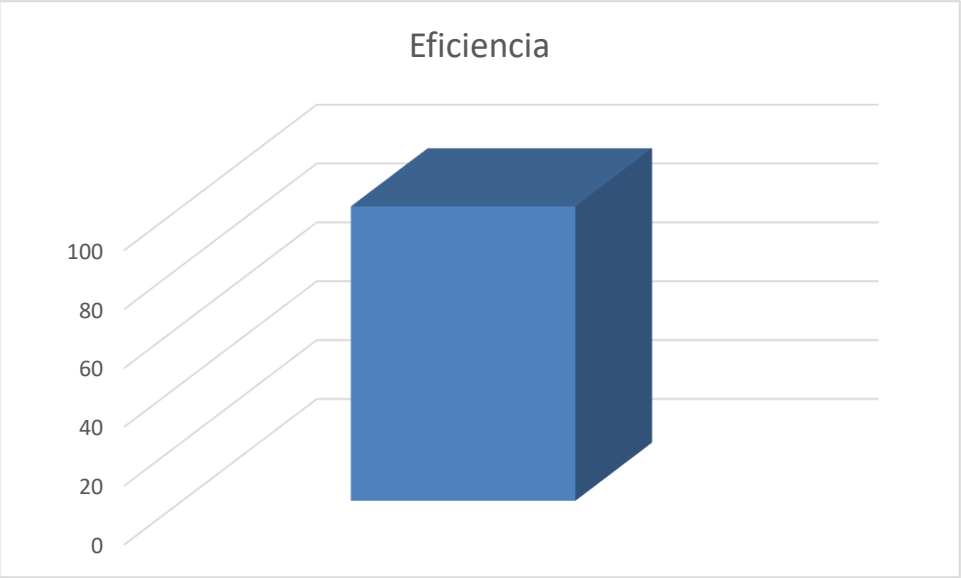
Indicador	FURPS	Valor obtenido
Eficiencia	95%	100%
Tiempo de respuesta	5s	0.82s
Utilización de recursos	25%	6.63%

Fuente: El autor

4.1.2.1 Eficiencia

La aplicación móvil evidenció un nivel óptimo de eficacia, alcanzando un valor del 100%. Este resultado puede visualizarse en la Figura 19, la cual ilustra el desempeño del indicador previamente mencionado.

Figura 19: Resultado de eficiencia.

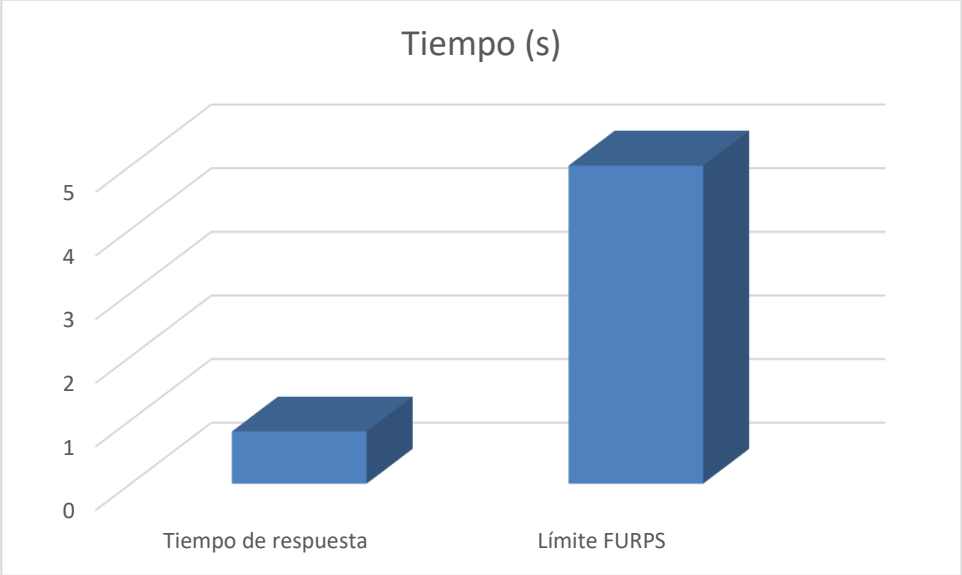


Fuente: El autor

4.1.2.2 Tiempo de respuesta

Tal como se muestra en la Figura 20, se alcanzó un resultado altamente favorable al registrarse un tiempo de respuesta de 0,82 segundos. Este valor se encuentra considerablemente por debajo del límite de 5 segundos definido como aceptable por el modelo de calidad FURPS, lo que evidencia que la aplicación es capaz de responder de forma ágil a las acciones del usuario.

Figura 20: Resultado de tiempo



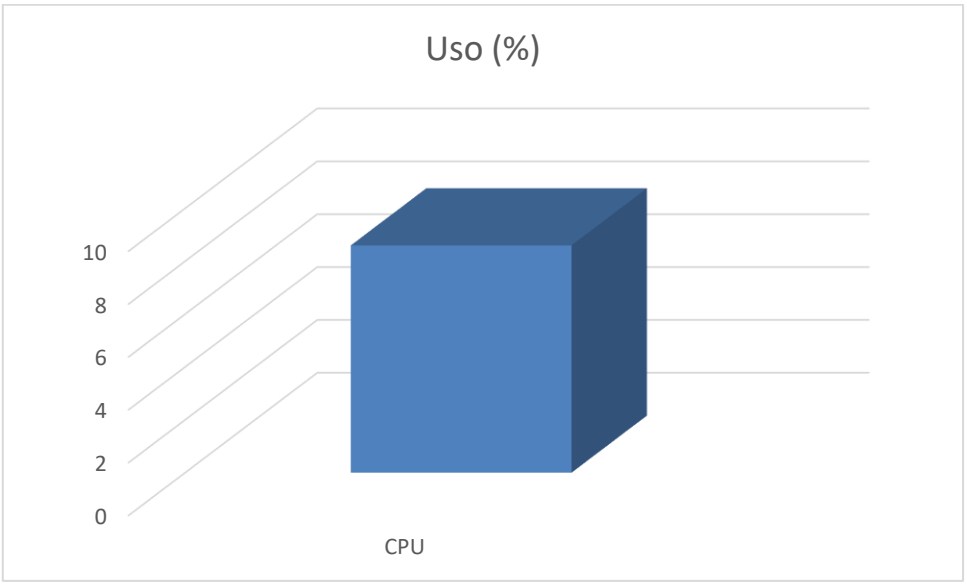
Fuente: El autor

4.1.2.3 Utilización de recursos

CPU

Con base en los datos mostrados en la Figura 21, se identificó que el consumo promedio de CPU durante las pruebas fue del 8,6%. Esto evidencia que la aplicación hace un uso eficiente del procesador, sin generar una sobrecarga, lo cual es indicativo de una adecuada optimización del rendimiento en esta área.

Figura 21: CPU.

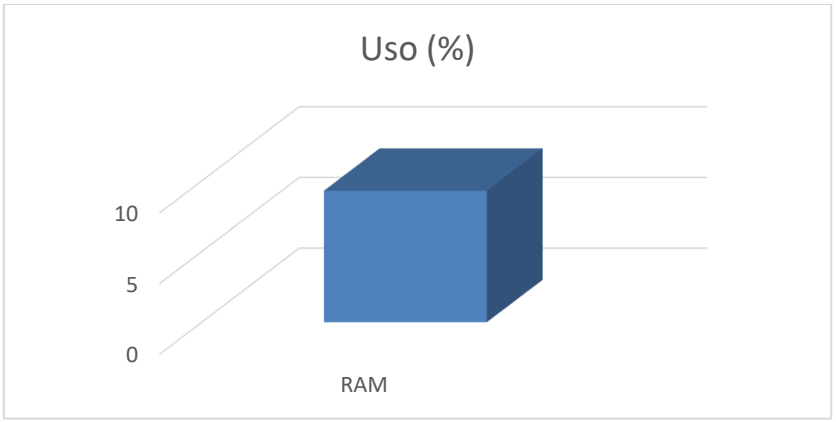


Fuente: El autor

RAM

Según lo representado en la Figura 22, el uso medio de memoria RAM por parte de la aplicación fue del 7.3%. Este resultado refleja una correcta administración de los recursos de memoria, evitando un consumo excesivo y contribuyendo así a un funcionamiento eficiente del sistema.

Figura 22: RAM.

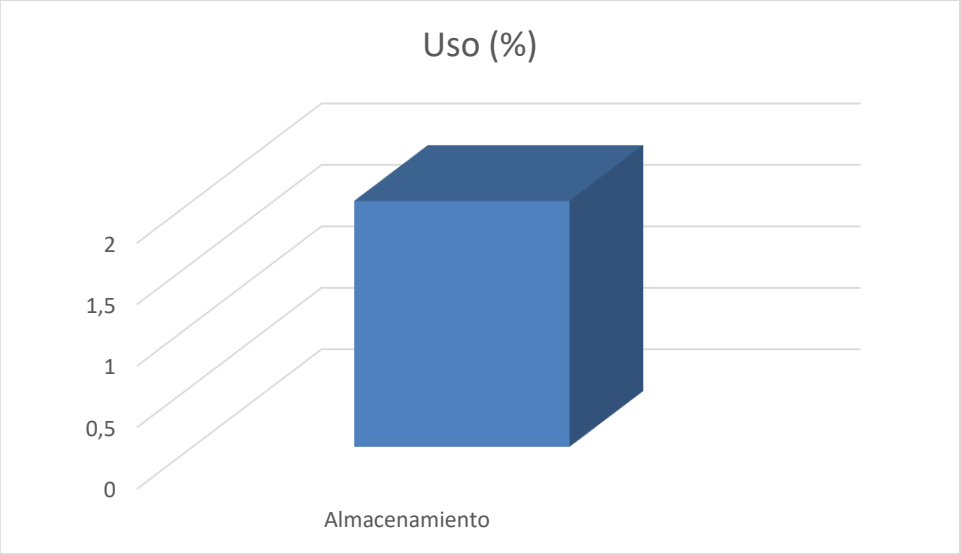


Fuente: El autor

Almacenamiento

Según los resultados reflejados en la Figura 23, la aplicación utilizó en promedio un 2% del almacenamiento del dispositivo. Este dato indica que la aplicación hace un uso eficiente del espacio disponible, evitando ocupar memoria de forma innecesaria y garantizando así una gestión adecuada del almacenamiento.

Figura 23: Almacenamiento.



Fuente: El autor

CAPÍTULO V. CONCLUSIONES y RECOMENDACIONES

5.1 Conclusiones

- Se identificó el enfoque, particularmente a través del uso de Mobile-D e Inception Deck, permitió una planificación clara y centrada en el usuario con entregas iterativas. Además, la selección de herramientas como Flutter, NestJS, Firebase, Docker y Nginx fue esencial para una arquitectura escalable y moderna. Estas tecnologías permitieron la automatización de procesos, el control de versiones y la integración continua elemento clave para un desarrollo ágil y sostenible.

- El desarrollo de la aplicación móvil multiplataforma con Flutter permitió utilizar una base de código unificado lo que facilitó la implementación de funciones, la corrección de errores y el desarrollo de productos con un esfuerzo de mantenimiento reducido, por otro lado, NestJS permitió una estructura de Backend robusta, lo cual favoreció al tener una lógica de negocio controlable y bien organizada.

- La evaluación según el modelo FURPS mostró que la aplicación que se cumplen los estándares de rendimiento, Firebase, permitió un rendimiento óptimo mediante la reducción de la carga del dispositivo y tiempos de respuesta mejorados. Flutter permitió una interfaz de usuario fluida y consistente en múltiples plataformas.

5.2 Recomendaciones

- Se sugiere profundizar en la automatización de procesos DevOps mediante la incorporación de herramientas como Ansible para la gestión de configuración o Kubernetes para la orquestación de contenedores. Esto permitirá mejorar la escalabilidad y robustez del sistema, especialmente en entornos de producción más exigentes.

- Es recomendable fortalecer la seguridad de la aplicación móvil durante el desarrollo y despliegue. Se puede integrar herramientas de análisis de vulnerabilidades como OWASP ZAP, además de implementar mecanismos de autenticación segura, control de acceso y cifrado de datos sensibles, asegurando así la protección de la información médica de los usuarios.

- Establecer un sistema de monitoreo continuo con herramientas como Prometheus o ELK Stack permitirá registrar métricas clave del sistema en tiempo real. Esto ayudará a identificar cuellos de botella, anticipar fallas, mejorar la experiencia del usuario y tomar decisiones informadas sobre futuras mejoras basadas en datos concretos.

BIBLIOGRÁFIA

[1] Reyes Erandira, “Las apps de delivery no convencen a las farmacias.” Accessed: Nov. 10, 2024. [Online]. Available: https://expansion.mx/tecnologia/2020/07/31/las-apps-de-delivery-no-convencen-a-las-farmacias

[2] A. E. Chabanovska Diana, “Healthcare Mobile App Development in 2024 [The Ultimate Guide].” Accessed: Nov. 10, 2024. [Online]. Available: https://www.cleveroad.com/blog/healthcare-app-development/

[3] Nubeser, “Apps para farmacias: Una apuesta de futuro.” Accessed: Nov. 10, 2024. [Online]. Available: https://nubeser.com/apps-para-farmacias-una-apuesta-de-futuro/

[4] Cristancho Felipe, “¿Qué es Flutter y para qué sirve? - Talently | Talently Blog.” Accessed: Nov. 10, 2024. [Online]. Available: https://talently.tech/blog/que-es-flutter/

[5] Inc. Amazon Web Services, “¿Qué es Flutter? - Explicación de la aplicación Flutter - AWS.” Accessed: Nov. 10, 2024. [Online]. Available: https://aws.amazon.com/es/what-is/flutter/

[6] NestJS Team, “NestJS, Documentation.” Accessed: Nov. 10, 2024. [Online]. Available: https://docs.nestjs.com/

[7] “¿Qué es PostgreSQL? | IBM.” Accessed: Nov. 10, 2024. [Online]. Available: https://www.ibm.com/mx-es/topics/postgresql

[8] GoDaddy, “¿Qué es una app? Explorando las aplicaciones móviles.” Accessed: Nov. 10, 2024. [Online]. Available: https://www.godaddy.com/resources/es/tecnologia/que-es-una-app-y-para-que-se-utiliza

[9] K. Buganová and J. Šimíčková, “Risk management in traditional and agile project management,” *Transportation Research Procedia*, vol. 40, pp. 986–993, Jan. 2019, doi: 10.1016/J.TRPRO.2019.07.138.

[10] S. , & A. S. Shaikh, “View of Comparison of Traditional & Agile Software Development Methodology: A Short Survey.” Accessed: Jan. 02, 2025. [Online]. Available: https://journal.ump.edu.my/ijsecs/article/view/2583/627

[11] “Contenedores de Docker | ¿Qué es Docker? | AWS.” Accessed: Jan. 04, 2025. [Online]. Available: https://aws.amazon.com/es/docker/

[12] P. Abrahamsson *et al.*, “Mobile-D: An agile approach for mobile application development,” *Proceedings of the Conference on Object-Oriented Programming Systems, Languages, and Applications, OOPSLA*, pp. 174–175, 2004, doi: 10.1145/1028664.1028736.

[13] B. Mathur and S. M. Satapathy, “An analytical comparison of mobile application development using agile methodologies,” *Proceedings of the International Conference on Trends in Electronics and Informatics, ICOEI 2019*, vol. 2019-April, pp. 1147–1152, Apr. 2019, doi: 10.1109/ICOEI.2019.8862532.

[14] A. Aljuhani and A. Alhubaishy, “Incorporating a Decision Support Approach within the Agile Mobile Application Development Process,” *ICCAIS 2020 - 3rd International Conference on Computer Applications and Information Security*, Mar. 2020, doi: 10.1109/ICCAIS48893.2020.9096751.

[15] J. Rasmusson, *The Agile Samurai: How Agile Masters Deliver Great Software*. 2010.

[16] P. R. Daniel C. Yeomans, “Project Management Made Simple and Effective - Daniel C. Yeomans, Peter Rogers - Google Libros.” Accessed: Nov. 10, 2024. [Online]. Available: https://books.google.es/books?hl=es&lr=&id=-hdgDgAAQBAJ&oi=fnd&pg=PR15&dq=D.+C.+Yeomans+y+P.+Rogers,+Project+Management+Made+Simple+and+Effective.+Dog+Ear+Publishing,+2017&ots=QYCLoWY4bP&sig=ah0nHOmu9FdYbBfxESzpCqYp5RU#v=onepage&q&f=false

[17] Satish Shethi, “Apache Jmeter: Todo lo que necesita saber - Geekflare Spain.” Accessed: Nov. 10, 2024. [Online]. Available: https://geekflare.com/es/apache-jmeter-guide/

[18] “Documentación de Google Maps Platform | Google for Developers.” Accessed: Nov. 10, 2024. [Online]. Available: https://developers.google.com/maps/documentation?hl=es-419

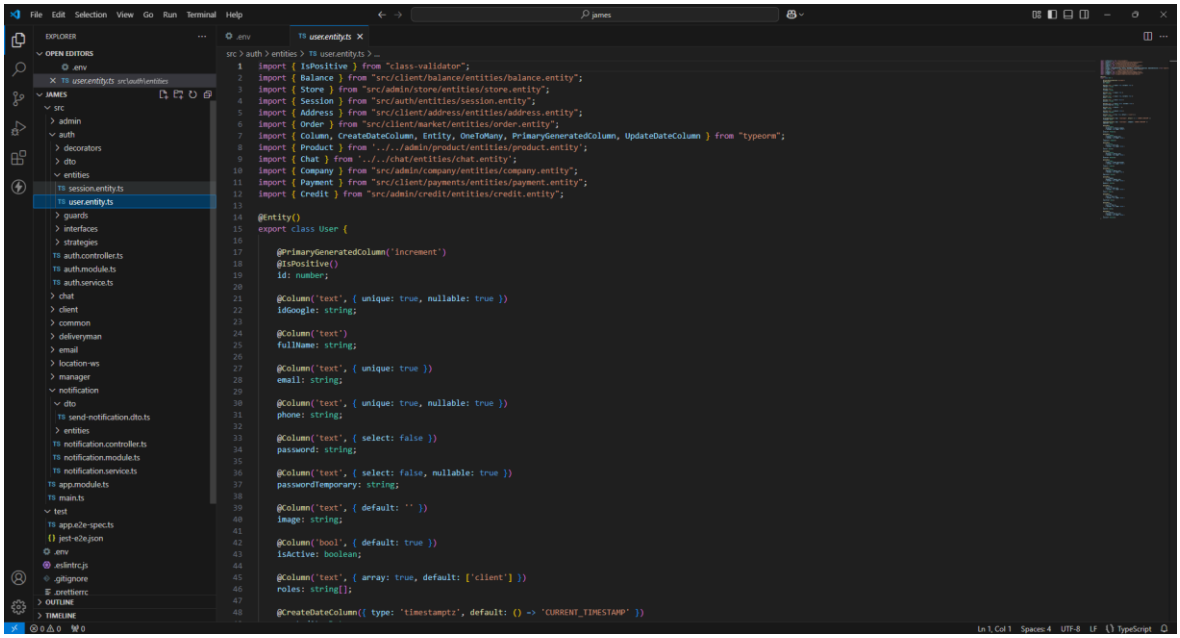
[19] Google Maps Platform Team, “Google Maps Platform, Geolocation API.” Accessed: Nov. 10, 2024. [Online]. Available: https://developers.google.com/maps/documentation?hl=es-419

[20] Google. (s. f.). *Firebase*. Google., “Google Maps Platform, Geolocation API.” Accessed: Dec. 11, 2023. [Online]. Available: https://firebase.google.com

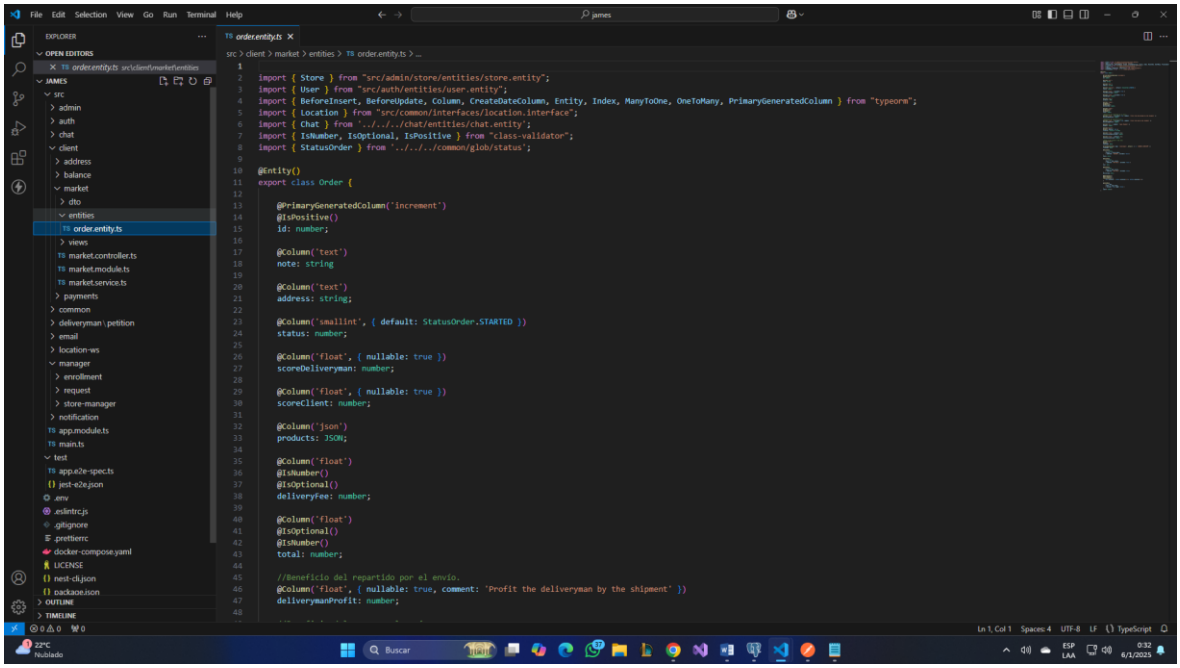
ANEXOS

ANEXO 1:

Decoradores para establecer relaciones entre entidades en TypeORM.

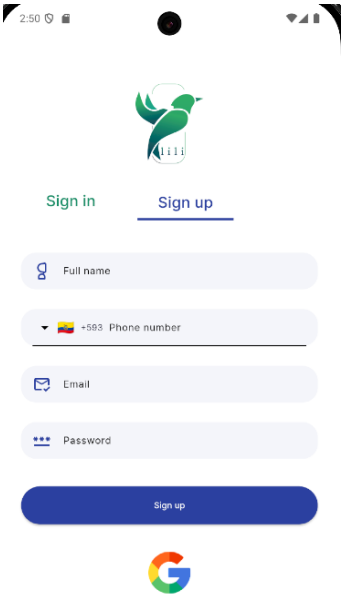


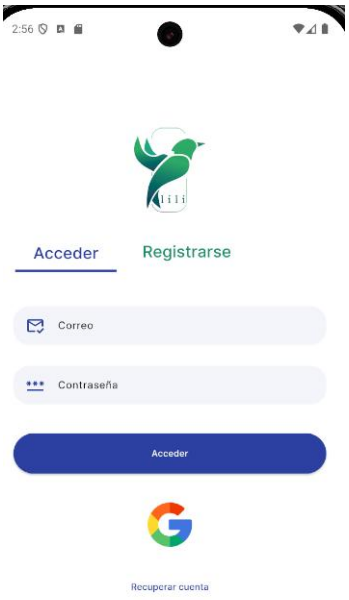
```ts
import { IsPositive } from "class-validator";
import { Balance } from "src/client/balance/entities/balance.entity";
import { Store } from "src/admin/store/entities/store.entity";
import { Session } from "src/auth/entities/session.entity";
import { Address } from "src/client/address/entities/address.entity";
import { Order } from "src/client/market/entities/order.entity";
import { Column, CreateDateColumn, Entity, OneToMany, PrimaryGeneratedColumn, UpdateDateColumn } from "typeorm";
import { Product } from '../../admin/product/entities/product.entity';
import { Chat } from '../../chat/entities/chat.entity';
import { Company } from "src/admin/company/entities/company.entity";
import { Payment } from "src/client/payments/entities/payment.entity";
import { Credit } from "src/admin/credit/entities/credit.entity";

@Entity()
export class User {

    @PrimaryGeneratedColumn('increment')
    @IsPositive()
    id: number;

    @Column('text', { unique: true, nullable: true })
    idGoogle: string;

    @Column('text')
    fullName: string;

    @Column('text', { unique: true })
    email: string;

    @Column('text', { unique: true, nullable: true })
    phone: string;

    @Column('text', { select: false })
    password: string;

    @Column('text', { select: false, nullable: true })
    passwordTemporary: string;

    @Column('text', { default: '' })
    image: string;

    @Column('bool', { default: true })
    isActive: boolean;

    @Column('text', { array: true, default: ['client'] })
    roles: string[];

    @CreateDateColumn({ type: 'timestamptz', default: () => 'CURRENT_TIMESTAMP' })
```

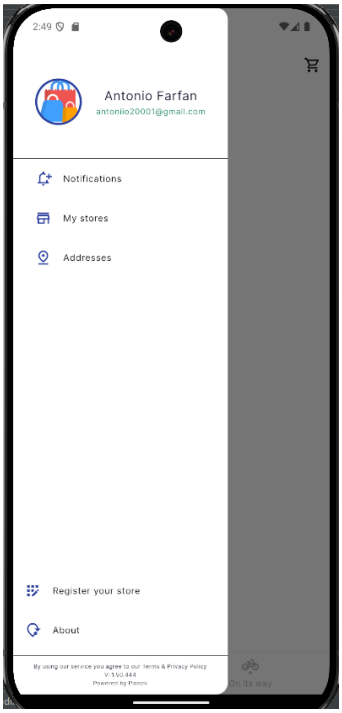


```ts
import { Store } from "src/admin/store/entities/store.entity";
import { User } from "src/auth/entities/user.entity";
import { BeforeInsert, BeforeUpdate, Column, CreateDateColumn, Entity, Index, ManyToOne, OneToMany, PrimaryGeneratedColumn } from "typeorm";
import { Location } from "src/common/interfaces/location.interface";
import { Chat } from '../../../chat/entities/chat.entity';
import { IsNumber, IsOptional, IsPositive } from "class-validator";
import { StatusOrder } from '../../../../common/glob/status';

@Entity()
export class Order {

    @PrimaryGeneratedColumn('increment')
    @IsPositive()
    id: number;

    @Column('text')
    note: string

    @Column('text')
    address: string;

    @Column('smallint', { default: StatusOrder.STARTED })
    status: number;

    @Column('float', { nullable: true })
    scoreDeliveryman: number;

    @Column('float', { nullable: true })
    scoreClient: number;

    @Column('json')
    products: JSON;

    @Column('float')
    @IsNumber()
    @IsOptional()
    deliveryFee: number;

    @Column('float')
    @IsOptional()
    @IsNumber()
    total: number;

    //Beneficio del repartido por el envio.
    @Column('float', { nullable: true, comment: 'Profit the deliveryman by the shipment' })
    deliverymanProfit: number;
```

ANEXO 2:
Despliegue de proyecto







ANEXO 3:

Pruebas