**UNIVERSIDAD TECNOLÓGICA**

**EMILIANO ZAPATA DEL ESTADO DE MORELOS**

**DIVISIÓN ACADÉMICA DE TECNOLOGÍAS DE LA INFORMACIÓN Y DISEÑO**

**MIGRACIÓN DEL PROYECTO TARITAXI**

**REPORTE DE ESTADÍA**

**QUE PARA OBTENER EL GRADO DE:**

**NOMBRE DEL PROGRAMA EDUCATIVO**

**ÁREA DEL PROGRAMA (SI APLICA)**

|  |  |  |
| --- | --- | --- |
| **PRESENTA:**  **Marcos Ariel Ordoñez Jiménez** | | |
| ASESOR EMPRESARIAL |  | ASESOR UNIVERSITARIO |
| **ING. UZZIEL GARCÍA MARTÍNEZ** |  | **DRA. VIRNA VIRIDIANA VELA RINCON** |

**EMILIANO ZAPATA, MOR., MAYO DE 2024**

|  |
| --- |
| **CONTENIDO** |

Índice de figuras

Índice de tablas

Agradecimientos

Resumen

Summary

[CAPÍTULO 1. DESCRIPCIÓN DEL PROYECTO 9](#_Toc161911064)

[1.1 Datos generales de la empresa 9](#_Toc161911065)

[1.2 Antecedentes del proyecto 10](#_Toc161911066)

[1.3 Objetivo general 11](#_Toc161911067)

[1.4 Objetivos específicos 12](#_Toc161911068)

[1.5 Justificación 12](#_Toc161911069)

[1.6 Alcances 13](#_Toc161911070)

[1.7 Restricciones 14](#_Toc161911071)

[CAPÍTULO 2. MARCO DE REFERENCIA 15](#_Toc161911072)

[2.1 Conceptos, metodologías y herramientas 15](#_Toc161911073)

[2.1.1 Flutter 15](#_Toc161911074)

[2.1.2 Ubicación en Flutter 16](#_Toc161911075)

[2.1.3 Notificaciones en Flutter 16](#_Toc161911076)

[2.1.4 Contactos de Flutter 17](#_Toc161911077)

[2.1.5 Productos Comerciales Relacionados: 17](#_Toc161911078)

[2.1.6 Herramientas Adicionales 18](#_Toc161911079)

[2.1.7 Endpoints en Azure 18](#_Toc161911080)

[2.1.8 Clean Architecture 19](#_Toc161911081)

[2.2 Propuesta de solución 20](#_Toc161911082)

[CAPÍTULO 3. DESARROLLO 22](#_Toc161911083)

[3.1 Inicio 22](#_Toc161911084)

[3.2 Planeación 23](#_Toc161911085)

[3.3 Ejecución 24](#_Toc161911086)

[3.4 Control 35](#_Toc161911087)

[3.5 Cierre 36](#_Toc161911088)

[CAPÍTULO 4. CONCLUSIONES 39](#_Toc161911089)

[4.1 Cumplimiento de objetivos 39](#_Toc161911090)

[4.2 Resultados 39](#_Toc161911091)

[4.3 Contribuciones 39](#_Toc161911092)

**REFERENCIAS**

**ANEXOS**

|  |
| --- |
| **ÍNDICE DE FIGURAS** |

[Figura 1.1Logotipo de la empresa 10](#_Toc164352956)

[Figura 3.3.3.1 Selectores de inicio de sesión pasajero 27](#_Toc164352957)

[Figura 3.3.2 Inicio de sesión pasajero 28](#_Toc164352958)

[Figura 3.3.3 Menú de aplicación pasajero 29](#_Toc164352959)

[Figura 3.3.4 Pantalla principal pasajero 30](#_Toc164352960)

[Figura 3.3.5 Pantalla de inicio de sesión conductor 31](#_Toc164352961)

[Figura 3.3.6 Pantalla del menú principal conductor 32](#_Toc164352962)

[Figura 3.3.7 Pantalla principal en espera de viaje conductor 33](#_Toc164352963)

[Figura 3.3.8 Pantalla de viajes concluidos conductor 34](#_Toc164352964)

[Figura 3.3.9 Pantalla de vehículos asignados 35](#_Toc164352965)

[Figura 3.3.10 Figma nuevos requerimientos 38](#_Toc164352966)

|  |
| --- |
| **ÍNDICE DE TABLAS** |

[Tabla 3.2.1 Pantallas TariTaxi Pasajero y Conductor 24](#_Toc164249545)

|  |
| --- |
| **AGRADECIMIENTOS** |

Quisiera expresar mi más sincero agradecimiento a todas las personas que han contribuido de alguna manera al desarrollo y finalización de este reporte,

En primer lugar cada uno de mis maestros que me orientaron y guiaron en cada paso, les doy las gracias por su orientación experta, paciencia y apoyo constante a lo largo de este proceso, sus conocimientos y consejos han sido fundamentales para dar forma a este trabajo y alcanzar mis objetivos, les agradezco por su valiosa retroalimentación y sugerencias durante las diferentes etapas de mis estudios, Sus comentarios han enriquecido considerablemente el contenido y la calidad de este reporte.

Agradezco a cada uno de mis compañeros de clase, especialmente a mis mejores amigos por su estímulo, comprensión y colaboración durante estos años de estudio. Su apoyo incondicional ha sido un motor importante para superar los desafíos que se presentaron en el camino.

Por último y lo más importante agradezco a mi familia, a mi hermano, a mi mamá y a mi tía, por su apoyo incondicional, por apoyarme en mis sueños y metas, por su amor, sacrificio y constante motivación, por estar siempre presentes y brindarme su apoyo en cada paso de este camino académico

|  |
| --- |
| **RESUMEN** |

Este informe detalla la migración del proyecto Taritaxi de Expo con React Native a Flutter. Esta transición implica una actualización tecnológica y una reingeniería profunda de los procesos internos. La adopción de Flutter no solo mejora la experiencia del usuario, sino que también fortalece la capacidad de medición y mejora continúa de los procesos internos. Además, impulsa el desarrollo de habilidades técnicas avanzadas en el equipo laboral. Esta migración no solo es tecnológica, sino también cultural y organizativa, promoviendo un mayor control y transparencia en la gestión interna. En resumen, la migración a Flutter marca un hito en la evolución tecnológica y refleja el compromiso con la excelencia y la adaptación a las últimas tecnologías.

Capítulo 1. Descripción del proyecto. Datos acerca de la empresa que necesita el proyecto, donde se encuentra ubicada, quién está a cargo, los objetivos generales y específicos además de las restricciones que se tienen al hacer el proyecto.

Capítulo 2. Marco de referencia. Se describen y analizan conceptos, metodologías y herramientas, a partir de las cuales se tiene una propuesta de solución que servirá como referencia de desarrollo.

Capítulo 3. Desarrollo. Se desarrollan procesos relacionados con actividades y documentos aplicados en la administración de proyectos, en especial durante la ejecución, donde se aplican las fases de construcción del producto.

Capítulo 4. Conclusiones. Se describe el cumplimiento de los objetivos del proyecto, los resultados y las contribuciones hacia la organización, y los beneficios en el desarrollo de este proyecto.

|  |
| --- |
| **SUMMARY** |

This report details the migration of the Taritaxi project from Expo with React Native to Flutter. This transition involves a technological upgrade and a deep reengineering of internal processes. The adoption of Flutter not only enhances the user experience but also strengthens the measurement capability and continuous improvement of internal processes. Additionally, it drives the development of advanced technical skills within the workforce. This migration is not only technological but also cultural and organizational, promoting greater control and transparency in internal management. In summary, the migration to Flutter marks a milestone in technological evolution and reflects a commitment to excellence and adaptation to the latest technologies.

Chapter 1. Project Description. Data about the company needing the project, its location, who is in charge, the general and specific objectives, as well as the constraints involved in the project.

Chapter 2. Framework. Concepts, methodologies, and tools are described and analyzed, providing a solution proposal that will serve as a development reference.

Chapter 3. Development. Processes related to activities and documents applied in project management are developed, especially during execution, where the phases of product construction are applied.

Chapter 4. Conclusions. The fulfillment of project objectives, results, contributions to the organization, and benefits in the development of this project are described.

|  |
| --- |
| DESCRIPCIÓN DEL PROYECTO |

## Datos generales de la empresa

GM Robotic SA de CV es una empresa líder especializada en el desarrollo de soluciones tecnológicas innovadoras, con un enfoque destacado en el diseño y creación de aplicaciones móviles y web. Nos encontramos ubicados en la calle Mirlos, Fraccionamiento Los Arcos N° 31, Colonia Centro, Temixco, Morelos, México, C.P. 62586.

Con un equipo altamente capacitado de desarrolladores y profesionales en tecnología, GM Robotic se especializa en la creación de aplicaciones móviles y web de última generación. Su compromiso con la excelencia y la satisfacción del cliente la ha convertido en referentes en el sector de desarrollo de software.

En la sucursal ubicada en Temixco nos encontramos únicamente el equipo de desarrollo, un equipo de alrededor de 8 personas especializadas en un aspecto especifico del desarrollo.

La misión de GM Robotic es impulsar el éxito de sus clientes a través de la entrega de aplicaciones móviles y web que superen las expectativas, proporcionando soluciones innovadoras y eficientes que impulsen el crecimiento y la competitividad de sus clientes.

En cuanto a su visión, GM Robotic se proyecta como líder en el desarrollo de aplicaciones tecnológicas, contribuyendo al avance digital de empresas y organizaciones tanto a nivel regional como internacional.

GM Robotic SA de CV se enorgullece de ser un socio tecnológico confiable, ofreciendo soluciones a medida que potencian la presencia en línea y la eficiencia operativa de nuestros clientes.

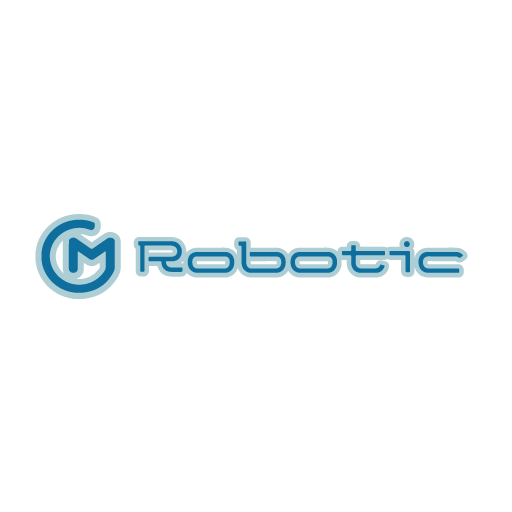


Figura 1.1Logotipo de la empresa

Bajo la dirección del Ingeniero Uzziel García Martínez, quien ostenta el título de Ingeniero en Control y Automatización y se desempeña actualmente como Director General, GM Robotic se enorgullece de ser un socio tecnológico confiable, ofreciendo soluciones a medida que potencian la presencia en línea y la eficiencia operativa de sus clientes.

## Antecedentes del proyecto

GM Robotic, en colaboración con un servicio de taxis ubicado en Playa del Carmen, ha enfrentado desafíos en el desarrollo de dos aplicaciones móviles críticas: una destinada a pasajeros y otra a conductores. Estas aplicaciones, originalmente construidas en React Native y gestionadas por Expo, han revelado limitaciones que han impulsado la búsqueda de alternativas más sólidas.

Un obstáculo significativo radica en la gestión de notificaciones, una funcionalidad esencial para la interacción en tiempo real entre pasajeros y conductores. A pesar de que Expo facilita el desarrollo en muchos aspectos, presenta restricciones al trabajar con notificaciones, lo que limita las opciones disponibles para ofrecer una experiencia de usuario óptima.

Además, al utilizar Expo, la selección de bibliotecas y módulos está limitada a aquellas compatibles con el framework. Esto ha llevado a la exclusión de ciertas librerías clave que podrían mejorar significativamente la funcionalidad y el rendimiento de las aplicaciones.

En respuesta a estos desafíos, se está considerando una migración hacia un enfoque que ofrezca mayor flexibilidad y acceso a un conjunto más amplio de recursos. La exploración de alternativas a Expo se ha vuelto imperativa para asegurar una implementación más personalizada y eficiente. Flutter, por ejemplo, se presenta como una solución prometedora, ya que proporciona una amplia gama de opciones y librerías para la gestión de notificaciones, permitiendo una implementación más robusta y eficiente en comparación con las restricciones presentes en Expo.

## Objetivo general

Mejorar significativamente la funcionalidad, rendimiento y experiencia del usuario en las aplicaciones móviles desarrolladas para el servicio de taxis en Playa del Carmen. La meta principal es superar las limitaciones identificadas en la gestión de notificaciones y la selección de librerías, mediante la migración hacia un entorno más flexible, como Flutter.

Este proceso de migración busca proporcionar a GM Robotic y al servicio de taxis una solución tecnológica más robusta y personalizada, permitiendo una implementación eficiente de notificaciones y la incorporación de librerías clave que optimicen la funcionalidad de las aplicaciones. Además, se persigue garantizar una experiencia de usuario más fluida y satisfactoria tanto para pasajeros como para conductores.

El éxito del objetivo general se medirá por la mejora palpable en la eficiencia del sistema, la capacidad de respuesta en tiempo real y la versatilidad en la selección de librerías, contribuyendo así a la competitividad y eficacia del servicio de taxis en Playa del Carmen a través de soluciones tecnológicas innovadoras.

## Objetivos específicos

1. **Diseñar la Versión para Pasajeros en Flutter:** Desarrollar la interfaz de usuario para la aplicación destinada a pasajeros utilizando Flutter, tomando como referencia la versión existente en Expo. Esto incluirá la adaptación de los elementos de diseño y la integración de funcionalidades previamente implementadas.
2. **Diseñar la Versión para Conductores en Flutter:** Crear la interfaz de usuario para la aplicación orientada a conductores utilizando Flutter, basándose en la versión ya existente en Expo. Este objetivo abarca la adecuación de la interfaz de usuario y la incorporación de características específicas destinadas a los conductores.
3. **Implementar los Endpoints Existentes en Expo y Adaptarlos a Flutter:** Migrar y adaptar los endpoints existentes en Expo para que sean compatibles con Flutter. Esto implica ajustar las solicitudes y respuestas de la API para garantizar la conectividad efectiva entre las aplicaciones móviles y el backend, manteniendo la coherencia en la transferencia de datos.

## Justificación

La migración de las aplicaciones móviles del servicio de taxis en Playa del Carmen de Expo a Flutter se fundamenta en diversas razones estratégicas y técnicas, respaldadas por los desafíos y limitaciones identificados en las versiones que se encuentran en producción gestionadas con expo. Las razones clave para justificar este proyecto son:

1. **Mejora en la Gestión de Notificaciones:** La limitación de Expo en la gestión de notificaciones ha impactado negativamente la interacción en tiempo real entre pasajeros y conductores. La migración a Flutter permitirá aprovechar las capacidades avanzadas de Flutter en el manejo de notificaciones, mejorando la comunicación y la experiencia del usuario.
2. **Flexibilidad en la Selección de Librerías:** La restricción de Expo en cuanto a la selección de librerías ha llevado a la exclusión de herramientas esenciales para mejorar la funcionalidad y el rendimiento de las aplicaciones. Con Flutter, se obtendrá una mayor libertad para elegir librerías específicas, adaptándose mejor a las necesidades del proyecto.
3. **Unificación del Código Base:** La migración a Flutter permitirá consolidar el código base para ambas versiones de la aplicación (pasajeros y conductores). Esto simplificará la gestión del proyecto, reducirá la duplicación de esfuerzos y facilitará futuras actualizaciones y mantenimiento.
4. **Eficiencia y Rendimiento Mejorado:** Flutter, al ser un framework altamente optimizado, promete un rendimiento superior en comparación con Expo. La migración tiene como objetivo mejorar la eficiencia y la velocidad de respuesta de las aplicaciones, contribuyendo a una experiencia de usuario más fluida.
5. **Capacidad de Adaptación a Futuras Tecnologías:** Al migrar a Flutter, se asegura una mayor adaptabilidad a las tecnologías emergentes y a las actualizaciones del ecosistema móvil. Flutter es conocido por su capacidad de integración con las últimas innovaciones, lo que asegura la relevancia a largo plazo de las aplicaciones.
6. **Competitividad y Diferenciación:** Adoptar tecnologías modernas y eficientes, como Flutter, no solo resuelve los problemas actuales, sino que también posiciona al servicio de taxis como líder en innovación, mejorando su competitividad en el mercado

## Alcances

* **Módulos de Gestión de Usuarios:** El sistema contará con un módulo integral de gestión de usuarios que permitirá la creación y modificación de perfiles. Además, se incluirán funcionalidades de inicio de sesión seguras.
* **Módulos de Gestión de Viajes:** Se implementará un módulo dedicado a la gestión de viajes, que abarcará la creación, cancelación de viajes. Este módulo también proporcionará información detallada sobre el historial de viajes y cualquier funcionalidad adicional relacionada con la gestión de viajes.
* **Ubicación:** El sistema integrará funciones de ubicación para rastrear en tiempo real la posición de los usuarios y/o vehículos asociados a los viajes. Esto incluirá la visualización de mapas y la actualización continua de la ubicación durante los viajes en curso.
* **Notificaciones Personalizadas:** Se desarrollará un sistema de notificaciones personalizadas para informar a los usuarios sobre eventos importantes, cambios en los viajes programados, actualizaciones de perfil, entre otros. Estas notificaciones serán configurables y se presentarán de manera clara y concisa.
* **Pagos por Tarjeta o en Efectivo:** Se incorporará un sistema de pagos que permitirá a los usuarios realizar transacciones tanto con tarjeta como en efectivo al finalizar los viajes. Esto incluirá la integración de pasarelas de pago para transacciones con tarjeta y un registro adecuado para los pagos en efectivo

## Restricciones

* **Problemas de Integración con Algunas Plataformas Específicas:** Algunas características específicas de las plataformas nativas pueden ser difíciles de implementar o pueden requerir soluciones específicas de plataforma. En estos casos, es posible que se necesite el uso de canales de plataforma o código nativo para la integración.
* **Compatibilidad con Algunas Características Nativas:** Aunque Flutter proporciona una amplia variedad de widgets y capacidades, es posible que algunas características específicas de una plataforma no estén completamente soportadas o puedan tener limitaciones.
* **Curva de Aprendizaje:**

Aunque Flutter es conocido por su facilidad de uso, puede haber una curva de aprendizaje para los desarrolladores que no estén familiarizados con Dart

|  |
| --- |
| MARCO DE REFERENCIA |

## Conceptos, metodologías y herramientas

### Flutter

Flutter es un framework de código abierto desarrollado por Google que se utiliza para crear aplicaciones móviles nativas tanto para Android como para iOS desde una sola base de código. Ofrece un amplio conjunto de widgets personalizables, un rendimiento excepcional y una rápida velocidad de desarrollo.

Flutter utiliza un modelo de desarrollo basado en el paradigma de "Widgets". En el contexto de Flutter, un widget es una clase que describe cómo se debe visualizar la interfaz de usuario en un determinado momento. Los widgets pueden ser simples, como un botón o un icono, o más complejos, como una lista desplazable o una pantalla completa de la aplicación.

El modelo de widgets en Flutter se basa en tres conceptos principales:

* **Inmutabilidad:** Los widgets en Flutter son inmutables, lo que significa que una vez que se crean, no se pueden cambiar. En lugar de modificar un widget existente, se crea un nuevo widget con las propiedades actualizadas.
* **Composición:** Flutter utiliza la composición de widgets para construir interfaces de usuario complejas a partir de widgets más simples. Los widgets se combinan en una jerarquía de árbol, donde cada widget puede contener otros widgets como hijos.
* **Reactividad:** Flutter es reactiva por naturaleza. Cuando cambia el estado de un widget, Flutter se encarga automáticamente de reconstruir la interfaz de usuario para reflejar esos cambios. Esto permite una actualización eficiente y fluida de la interfaz de usuario en respuesta a eventos del usuario o cambios en los datos.

Además, Flutter sigue el concepto de "Everything is a Widget" (Todo es un Widget), lo que significa que todo en la interfaz de usuario de una aplicación Flutter, desde el diseño general hasta los elementos individuales como texto, imágenes y botones, son widgets.

Este modelo de desarrollo basado en widgets proporciona una gran flexibilidad y potencia para crear interfaces de usuario altamente personalizadas y dinámicas en aplicaciones Flutter:

**Beneficios de Flutter:** La arquitectura de Flutter permite una rápida iteración, un rendimiento superior y una experiencia de usuario fluida en todas las plataformas móviles (GRUPO EBIM, 2023).

### Ubicación en Flutter

La integración de la ubicación es fundamental para las aplicaciones de servicios de taxis, permitiendo la localización precisa de conductores y pasajeros en tiempo real.

* **Google Maps en Flutter:** La integración de Google Maps proporciona capacidades avanzadas de visualización de mapas y navegación, permitiendo la ubicación en tiempo real y la planificación de rutas.
* **Geolocator:** La dependencia Geolocator en Flutter permite acceder a la ubicación del dispositivo del usuario, facilitando la determinación de la posición actual y la navegación.

(3Androides, 2023).

### Notificaciones en Flutter

Las notificaciones son elementos clave para mantener a los usuarios informados sobre eventos relevantes y actualizaciones en tiempo real en las aplicaciones móviles.

* **Flutter Local Notifications:** Esta dependencia proporciona una forma sencilla de enviar notificaciones locales dentro de la aplicación, permitiendo la personalización de contenido y horarios de entrega.
* **Firebase Cloud Messaging (FCM):** Integrar FCM con Flutter permite enviar notificaciones push a dispositivos móviles de forma eficiente, permitiendo la comunicación bidireccional entre la aplicación y el servidor.

(FIREBASE, 2023).

### Contactos de Flutter

La integración de contactos permite a los usuarios acceder a su lista de contactos y compartir información de manera rápida y sencilla.

* **Contacts Service:** Esta dependencia en Flutter facilita el acceso a los contactos del dispositivo, permitiendo la lectura, escritura y actualización de la libreta de direcciones del usuario.
* **Contact Picker:** La integración del selector de contactos permite al usuario seleccionar y compartir contactos específicos dentro de la aplicación, mejorando la experiencia de usuario y la interoperabilidad.

(Vaidya, 2024).

### Productos Comerciales Relacionados:

* **Openpay BBVA:** Esta herramienta proporciona soluciones de pago seguras y confiables, permitiendo la integración de métodos de pago electrónicos en las aplicaciones móviles desarrolladas con Flutter (BBVA, 2022).
* **Easy Localization:** Easy Localization simplifica la integración de múltiples idiomas y localizaciones en aplicaciones Flutter, facilitando la internacionalización y personalización del contenido para usuarios de diferentes regiones y culturas (RyseSoft, 2022).

### Herramientas Adicionales

En el desarrollo de aplicaciones móviles con Flutter, es fundamental contar con un conjunto de herramientas que faciliten el proceso de desarrollo y pruebas.

* **Visual Studio Code:** Un editor de código liviano y altamente personalizable que ofrece soporte completo para Flutter, incluyendo resaltado de sintaxis, depuración y extensiones útiles (Flores, 2022).
* **Android Studio:** Un entorno de desarrollo integrado (IDE) preferido por muchos desarrolladores de Android, que ofrece un conjunto completo de herramientas para el desarrollo de aplicaciones Flutter y Android (Santaella, 2022).
* **Emuladores de Android Studio:** Permiten simular dispositivos Android en una variedad de configuraciones, lo que facilita las pruebas y depuración de aplicaciones Flutter en diferentes dispositivos y versiones de Android (Ramírez, 2021).
* **Flutter Inspector:** Una herramienta integrada en Flutter que permite inspeccionar y diagnosticar la interfaz de usuario de la aplicación durante el desarrollo (Flutter, 2023).
* **Google Maps API:** Para integrar capacidades avanzadas de mapas y geolocalización en aplicaciones Flutter, permitiendo una experiencia de usuario mejorada y funcionalidades de ubicación avanzadas (GOOGLE, 2020).

### Endpoints en Azure

Para la gestión de servicios en la nube, los **endpoints** se encuentran alojados en **Azure**, una plataforma de computación en la **nube de Microsoft** que ofrece una amplia gama de servicios para el desarrollo, implementación y administración de aplicaciones y servicios (Fernández, 2023).

### Clean Architecture

La arquitectura limpia, también conocida como Clean Architecture, es un enfoque de diseño de software que se centra en la separación de responsabilidades y la independencia de capas dentro de una aplicación. Este enfoque promueve la creación de sistemas modulares y fácilmente mantenibles, donde cada capa tiene un propósito específico y claro.

**Capas de la arquitectura:**

* **Capa de Presentación (UI):** En Flutter, esta capa corresponde a la interfaz de usuario de la aplicación, donde se definen y gestionan los widgets visuales. Aquí se encuentra la lógica relacionada con la presentación de datos y la interacción del usuario.
* **Capa de Negocio (Domain):** Esta capa contiene la lógica de negocio de la aplicación, independiente de cualquier detalle de implementación. Aquí se definen los casos de uso y las reglas de negocio que guían el comportamiento de la aplicación.
* **Capa de Datos (Data):** En esta capa se maneja el acceso a los datos, ya sea desde una base de datos local, servicios web remotos u otras fuentes de datos. Se encarga de la obtención y manipulación de datos para satisfacer las necesidades de la capa de negocio.

(Sánchez, 2023).

**Principio de independencia de frameworks:**

Las capas internas no dependen de las capas externas. Por lo tanto, la capa de negocio no debe depender de detalles de implementación específicos de Flutter, como los widgets o las APIs de Flutter. Esto se logra definiendo interfaces limpias y abstracciones que separan las capas y promueven la modularidad y la reutilización del código.

Principio de inversión de dependencias (Dependency Inversion Principle, DIP):

En lugar de que las capas de nivel superior dependan directamente de las capas de nivel inferior, las dependencias se invierten para que las capas internas definan interfaces que las capas externas implementan. En Flutter, esto se puede lograr mediante la definición de clases abstractas o interfaces que representan contratos entre las diferentes capas de la aplicación.

Al aplicar los principios de Clean Architecture en Flutter, puedes crear aplicaciones móviles bien estructuradas, fáciles de mantener y probar. Esto permite una mayor flexibilidad y escalabilidad a medida que el proyecto crece y evoluciona con el tiempo. Además, facilita la integración de bibliotecas y herramientas de terceros, como las mencionadas en tu reporte, sin comprometer la integridad y la coherencia de la arquitectura de la aplicación.

## Propuesta de solución

El proyecto de migración de las aplicaciones Tari Taxi a Flutter surge como respuesta a las limitaciones y desafíos encontrados en las versiones actuales desarrolladas en React Native y gestionadas por Expo. Estas limitaciones abarcan desde restricciones en la personalización de notificaciones hasta problemas de rendimiento y acceso a herramientas específicas.

El objetivo principal de la migración a Flutter es mejorar la experiencia del usuario tanto para conductores como para pasajeros, al tiempo que se optimizan aspectos clave como la personalización de notificaciones, la eficiencia del rendimiento y la accesibilidad a herramientas avanzadas de desarrollo.

Obteniendo finalmente mejoras como las siguientes en comparación con lo que ya se tenía y se estaba manejando en expo.

* **Mejora en el Rendimiento:** Flutter ofrece un rendimiento superior gracias a su motor de renderizado personalizado, lo que se traduce en una experiencia de usuario más fluida y receptiva.
* **Personalización Avanzada de Notificaciones:** Con Flutter, se pueden implementar notificaciones altamente personalizadas y dinámicas, lo que permite una interacción más efectiva con los usuarios.
* **Acceso a Herramientas Específicas de Desarrollo:** Flutter proporciona un conjunto completo de herramientas y widgets que facilitan la creación de interfaces de usuario complejas y atractivas, sin comprometer la experiencia del usuario.
* **Compatibilidad Multiplataforma Mejorada:** Flutter permite desarrollar una sola base de código que se puede implementar en múltiples plataformas, lo que simplifica el mantenimiento y reduce los costos a largo plazo.

La migración de las aplicaciones TariTaxi a Flutter representa un paso importante hacia la mejora de la experiencia del usuario y la optimización del rendimiento. Al adoptar esta tecnología moderna y versátil.

La versión del pasajero incluirá selección de destino en el mapa, mensajería con el conductor y con servicio técnico, una opción para compartir la informacion del viaje con los contactos de confianza, notificaciones con sonido personalizado, por parte de la búsqueda de una ubicación se cuenta con 2 métodos , los recuperados por mapbox y los recuperados por google, esto dependiendo la cantidad de resultados mostradas en cada uno, utilizando como primer opción mapbox, se cuenta con 2 formas de pago, efectivo y pago con tarjeta

En ambas versiones se contara con una gestión de los datos de perfil, ubicación en tiempo real y por último en la versión de conductor se mostrará datos del gafete y del vehículo activo del conductor, se buscara disminuir el tiempo al momento de obtener datos del viaje y aceptar o rechazar el viaje asi como tambien optimizar la experiencia del usuario al realizar un viaje completo.

|  |
| --- |
| DESARROLLO |

## Inicio

En la primer reunión se me entrego el código fuente de ambas aplicaciones realizadas en expo, presentándome tambien los problemas a corregir, detalles estéticos, recepción de notificaciones, velocidad de los procesos y una falla muy específica en la que se perdía la informacion del viaje al aceptar el viaje por lo que no se podía avanzar o retroceder.

Durante el proceso de corrección de los errores y mejoras estéticas en las aplicaciones realizadas en Expo, surgieron varios desafíos debido a las limitaciones de las bibliotecas disponibles en Expo. Una de las principales áreas de mejora fue la recepción de notificaciones, ya que las opciones para personalizar las notificaciones eran limitadas en Expo. Esto dificultaba la implementación de funcionalidades específicas relacionadas con las notificaciones, lo que requería encontrar soluciones alternativas o adaptar las limitaciones existentes.

Otro desafío importante fue la velocidad de los procesos en las aplicaciones. Se identificó la necesidad de optimizar el rendimiento y la eficiencia de las aplicaciones para garantizar una experiencia fluida y rápida para los usuarios. Esto implicó la revisión del código existente y la implementación de mejoras en los algoritmos y estructuras de datos utilizados en las aplicaciones.

Además, se encontró una falla muy específica en la que se perdía la información del viaje al aceptarlo, lo que impedía avanzar o retroceder en el proceso. Esta falla causaba que el pasajero no pudiera realizar ningún otro viaje hasta que el servicio técnico lo restaurara. Solucionar esta falla se convirtió en una prioridad para garantizar el correcto funcionamiento de las aplicaciones y la satisfacción de los usuarios.

A medida que se abordaban estos problemas, también se tomó en cuenta la migración posterior de las aplicaciones a Flutter. Sin embargo, debido a las limitaciones de Expo, surgieron dificultades al intentar utilizar ciertas bibliotecas específicas de pago y otras bibliotecas de React Native que no eran compatibles con Expo. Esto requirió una cuidadosa evaluación de las alternativas disponibles y una planificación adecuada para garantizar una transición exitosa a Flutter en el futuro.

En resumen, la tarea inicialmente consistía en corregir errores y mejorar las aplicaciones existentes en Expo, centrándose en problemas como detalles estéticos, recepción de notificaciones, velocidad de los procesos y la falla específica que causaba la pérdida de información del viaje. Sin embargo, surgieron desafíos debido a las limitaciones de las bibliotecas en Expo, especialmente en relación con las notificaciones personalizadas, las bibliotecas de pago y las bibliotecas de React Native. A pesar de estos desafíos, se trabajó en encontrar soluciones alternativas y se planificó una migración futura a Flutter.

## Planeación

Para el proyecto se planeó realizar en pequeñas fases:

* Análisis del funcionamiento completo de ambas aplicaciones comenzando con la del pasajero,
* Curva de aprendizaje con flutter, viendo tutoriales y tomando cursos breves
* Comienzo del desarrollo utilizando cada uno de los endpoints
* Conversión del método de cifrado utilizado en la app de expo
* Diseño e implementación de las nuevas pantallas
* Desarrollo del ciclo de viaje para ambas aplicaciones
* Diseño de la nueva pasarela de pagos
* Pruebas de las aplicaciones en diversas situaciones

Las pantallas a desarrollar son las siguientes:

Tabla .2. Pantallas TariTaxi Pasajero y Conductor

|  |  |  |
| --- | --- | --- |
| **TARI TAXI** | | |
| **PASAJERO** | | **CONDUCTOR** |
| Splash | Agregar Método de pago | Splash |
| Selección | Políticas de privacidad | Log-In |
| Recuperación de contraseña | Términos y  condiciones | Mapa principal |
| Log-in | Ayuda | Perfil |
| Registro paso 1 | Seguridad | Viajes |
| Registro paso 2 | Agregar contacto de seguridad | Vehículo |
| Registro paso 3 | Búsqueda de destino | Políticas |
| Mapa Inicial | Selector de  conductor | Términos y condiciones |
| Perfil | Selector de origen en mapa | Ayuda |
| Viajes | Selector de destino en mapa | Viaje |
| Método de pago | Viaje |  |

Se estudiaron varios tutoriales y un curso de flutter para analizar las mejores opciones que resolvieran las problemáticas en el entorno anterior, una vez realizado ese análisis se procedió a calcular las fechas de desarrollo y prueba para agregarlos al calendario de actividades usado por el equipo.

## Ejecución

Para el tiempo proporcionado por este periodo de estadias en el que nos encontramos, el proyecto quedará desarrollado en un 80% al momento de entrega de este mismo reporte ya que la fecha de entrega se considera alrededor de agosto, teniendo en cuenta los demás proyectos en los que trabajo y que ya se encuentran en periodo de entrega y montaje a producción asi como la prioridad que estos mismos representan.

**Fase inicial: Recopilación y clasificación de endpoints**

Como paso inicial, se procedió a la recopilación y clasificación de todos los endpoints presentes en los archivos de los proyectos desarrollados en Expo ([Expo, n.d.]). Se eliminaron aquellos endpoints que ya no aportaban funcionalidad al sistema o que habían quedado obsoletos. Este proceso de clasificación se realizó de manera exhaustiva, realizando múltiples recorridos a través del ciclo completo que representa un viaje en la aplicación. El objetivo fue analizar exhaustivamente la navegación del sistema y poder identificar áreas de mejora y optimización en diversos procesos, considerando principios de arquitectura limpia ([Stack Overflow, n.d.]) para una mayor mantenibilidad y flexibilidad del código.

**Diseño de nuevas pantallas**

Se comenzó a diseñar cada nueva pantalla, comenzando con el selector de Login y registro, la pantalla de Login y la pantalla principal. Esta última incluye un mapa con la ubicación del usuario y los conductores cercanos. Teniendo en cuenta la experiencia de usuario (UX), se utilizaron herramientas como Visual Studio Code ([Visual Studio Code, n.d.]) para facilitar el desarrollo y la implementación de un diseño atractivo e intuitivo.

**Conversión de código JavaScript a Dart**

Se procedió a convertir cada método de JavaScript a Dart, probando la funcionalidad de cada uno y asegurando su correcto funcionamiento durante todo el proceso. Para facilitar la migración del código base de Expo a Flutter ([Flutter, n.d.]), se consideraron herramientas como los Unimodules de Flutter ([Expo, n.d.]).

**Fase de pruebas exhaustivas**

Después de la conversión, se realizó una fase de pruebas exhaustivas para garantizar que cada funcionalidad se comportara de manera adecuada en el nuevo entorno Dart. Esto implicó la simulación de diversos escenarios, como la autenticación de usuarios, la búsqueda y visualización de conductores cercanos, así como la actualización en tiempo real de la ubicación del usuario en el mapa. Se utilizaron herramientas como Easy Localization ([Easy Localization, n.d.]) para garantizar la correcta internacionalización y localización de la aplicación en diferentes mercados.

**Mejoras en la interfaz de usuario**

Se implementaron mejoras significativas en la interfaz de usuario para asegurar una experiencia fluida y satisfactoria para los usuarios finales. Esto incluyó la optimización de la velocidad de carga de las pantallas, la mejora de la legibilidad de la información presentada y la implementación de un diseño más atractivo y moderno. Se tuvieron en cuenta las directrices de diseño de Flutter para lograr una estética consistente y alineada con las mejores prácticas de la industria.

**Integración de servicios externos**

El proceso de desarrollo también involucró la integración de servicios externos, como APIs de mapas (por ejemplo, Google Maps API [Google Maps API, n.d.]), para enriquecer la funcionalidad de la aplicación y proporcionar una experiencia completa a los usuarios. Para la comunicación en tiempo real, se consideraron opciones como Firebase Cloud Messaging (FCM) [Firebase Cloud Messaging, n.d.] o soluciones similares ofrecidas por proveedores como Microsoft Azure ([Microsoft Azure, n.d.]). Los servicios de pago, si son necesarios, podrían integrarse con plataformas como Openpay BBVA ([Openpay BBVA, n.d.]).

La primera versión de las pantallas muestra una clara diferencia en la estética y la fluidez de la interfaz entre las desarrolladas en Expo y las nuevas creadas en Flutter. En el lado izquierdo, la pantalla en Expo exhibe un diseño más básico y una experiencia de usuario menos dinámica, mientras que en el lado derecho, la nueva pantalla en Flutter ofrece una apariencia más moderna y una navegación más fluida.

Ambas capturas fueron tomadas del mismo dispositivo, un Xiaomi Redmi con la versión MIUI 14.0.3, lo que garantiza una comparación justa en términos de rendimiento y visualización en el mismo entorno.

A continuación, se presentan algunas de las pantallas, comenzando por la aplicación del pasajero, donde se pueden apreciar las mejoras en diseño y experiencia de usuario que ofrece la transición a Flutter.

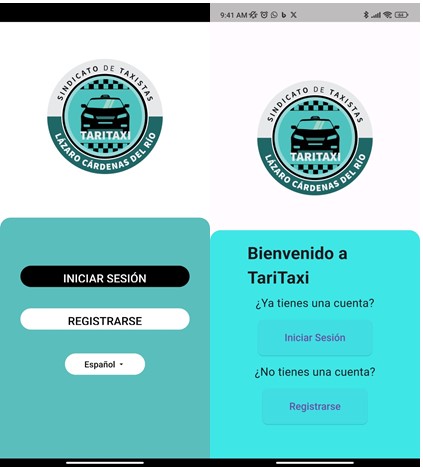


Figura .3.. Selectores de inicio de sesión pasajero

En la Figura 3.3.1 anterior, la pantalla inicial de la aplicación presenta un selector entre el registro y el inicio de sesión, eliminando el botón de selección de idioma y reemplazándolo automáticamente al detectar el idioma del sistema.

En la siguiente figura, se observa la pantalla de inicio de sesión, donde se ha intentado mantener el orden de los componentes, pero también se ha trabajado en mejorar el diseño para hacerlo más atractivo y optimizar la experiencia de usuario, así como la velocidad de navegación.

Una vez que se inicia sesión, el endpoint recibe un token, el cual se guarda mediante Shared Preference. Dado que este token se utiliza en cada uno de los procesos de la aplicación, se convierte en una parte esencial del método de autenticación bearer. Este enfoque garantiza la seguridad y la eficiencia en la autenticación de los usuarios en la aplicación.



Figura .. Inicio de sesión pasajero

La Figura 3.3.2 ilustra las pantallas de inicio de sesión, donde se introduce una funcionalidad adicional: si los permisos de notificación, acceso a la lista de contactos y ubicación no han sido aprobados previamente, se solicitan en este punto. Este ajuste representa una mejora significativa en la experiencia del usuario, asegurando un acceso completo a las funciones de la aplicación desde el inicio.

Un desafío destacado durante este proceso fue adaptar el método de encriptación de contraseñas, ya que las bibliotecas disponibles en Expo difieren considerablemente de las que se encuentran en Flutter. Esta discrepancia requirió un esfuerzo adicional para garantizar la coherencia y seguridad en la gestión de contraseñas.

Este ajuste representó uno de los desafíos más difíciles de abordar y consumió una cantidad considerable de tiempo después de la implementación de la lógica de viajes. Sin embargo, su resolución satisfactoria contribuyó significativamente a la robustez y la funcionalidad general de la aplicación.

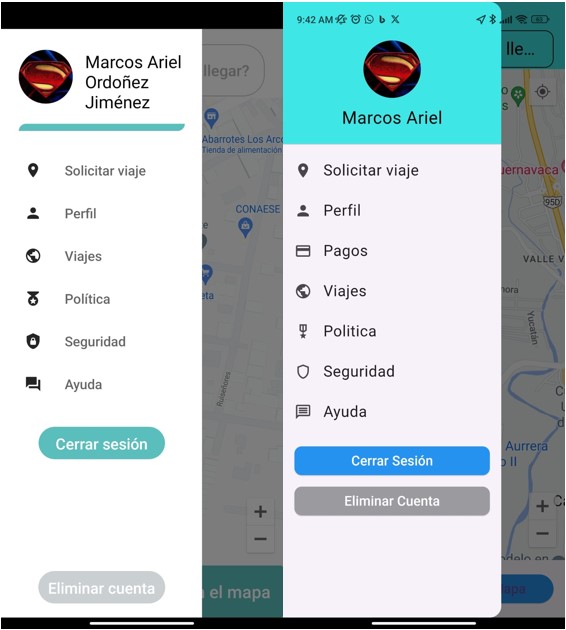


Figura .. Menú de aplicación pasajero

En la Figura 3.3.3 se presenta una comparativa del menú de la aplicación del pasajero. Se destaca la adición de una sección de pagos en la nueva versión, en contraste con la versión anterior donde esta característica estaba ausente. Esta diferencia se debe a la reciente deshabilitación de la funcionalidad de pagos, que ocurrió hace aproximadamente un mes.

Anteriormente, la aplicación utilizaba Openpay de BBVA para gestionar los pagos de los viajes. Sin embargo, debido a un problema con la cuenta receptora de los pagos, se tomó la decisión de cambiar a una nueva pasarela de pagos. Actualmente, se encuentra en curso un período de investigación para determinar cuál será la mejor opción a utilizar. Este proceso de evaluación y selección de una nueva pasarela de pagos es crucial para garantizar la eficiencia y la seguridad en las transacciones financieras dentro de la aplicación. Aunque aún no se ha definido cuál será la pasarela seleccionada, este paso refleja el compromiso continuo con la mejora y la optimización de la experiencia del usuario en la aplicación del pasajero.

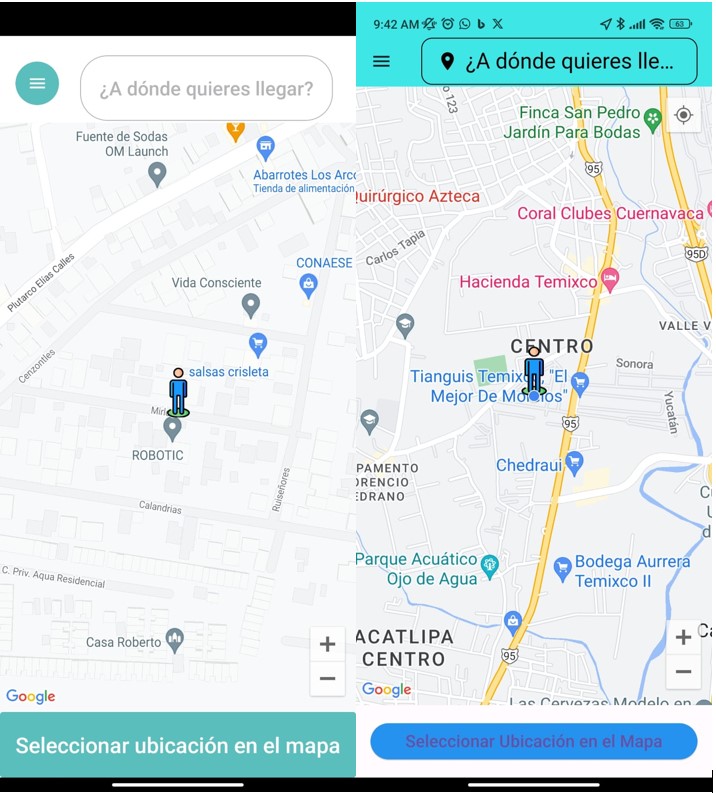


Figura .. Pantalla principal pasajero

En la figura 3.3.4, se presenta la pantalla principal del pasajero, donde se destacan cambios que van más allá del diseño superficial. En esta versión, se ha realizado una notable modificación en el sistema de notificaciones, otorgándole una funcionalidad ampliada y mejorada. Estos ajustes permiten una experiencia más fluida y eficiente para los usuarios al momento de recibir y gestionar alertas importantes durante su viaje.

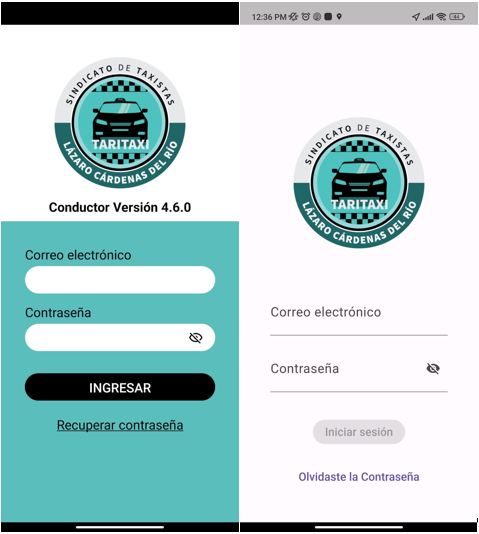


Figura .. Pantalla de inicio de sesión conductor

En la siguiente figura se comienza con las pantallas relacionadas con las pantallas del conductor comenzando por la pantalla de inicio de sesión donde se optó por un diseño más minimalista y con menos color para poder diferenciarla con la versión del pasajero

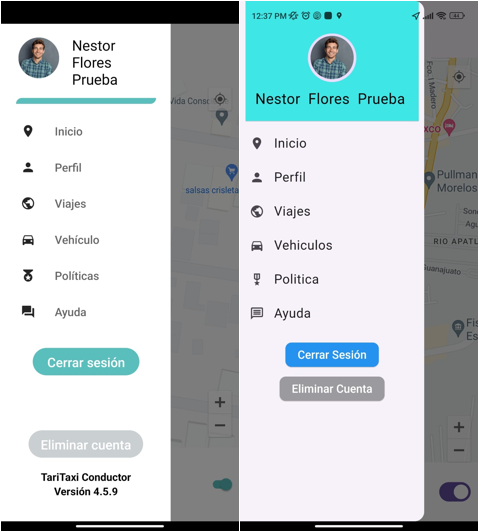


Figura .3. Pantalla del menú principal conductor

En la figura 3.3.6 se muestra el menú que observa el conductor, diferenciándose con el del pasajero en la sección de vehículos y viajes

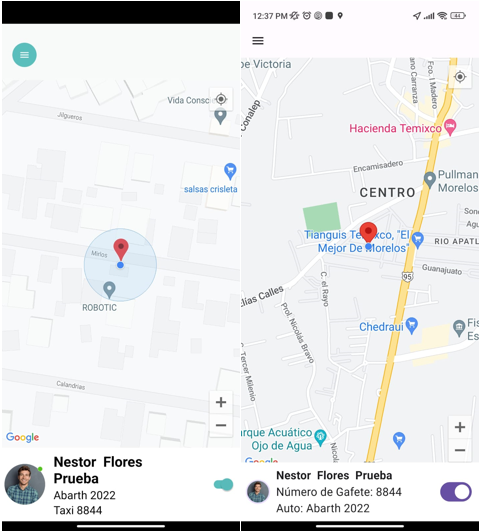


Figura .3. Pantalla principal en espera de viaje conductor

En la figura 3.3.7 se muestra la pantalla de espera de un viaje para el conductor donde la notificación de un nuevo viaje en la nueva versión rediecciona a una pantalla nueva con más informacion del viaje a realizar a diferencia de la anterior que se desplegaba como algún tipo de modal proporcionando muy poca informacion

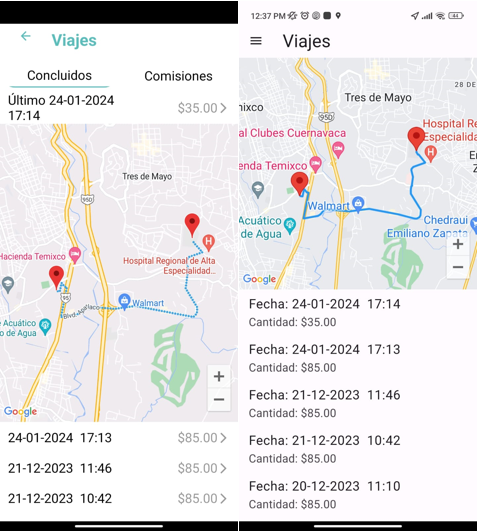


Figura .3. Pantalla de viajes concluidos conductor

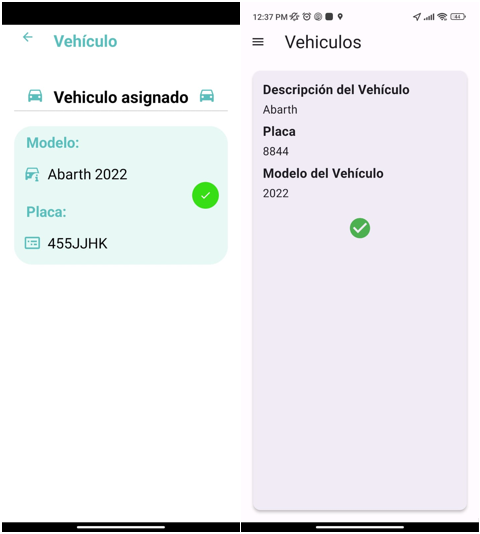


Figura .3. Pantalla de vehículos asignados

En la figura 3.3.9 se muestra la pantalla que informa de que vehículo ha sido asignado y si este se encuentra activo para su uso

## Control

Para garantizar el éxito del proyecto y mantener un seguimiento adecuado del avance, se implementaron varias prácticas de control. Estas acciones fueron fundamentales para monitorear el progreso, identificar posibles desviaciones y realizar ajustes en consecuencia. A continuación, se detallan las principales actividades de control realizadas durante el desarrollo del proyecto:

**Reuniones de seguimiento periódicas:** Se establecieron reuniones regulares, programadas semanalmente, para revisar el estado del proyecto, discutir cualquier problema o desafío que surgiera y definir las próximas tareas a realizar. Estas reuniones permitieron mantener a todo el equipo informado y alineado con los objetivos del proyecto.

**Seguimiento del avance:** Se llevó a cabo un seguimiento constante del progreso realizado en comparación con el plan inicial. Esto implicó la revisión periódica de las actividades planificadas y su correspondiente avance real. En caso de desviaciones significativas, se tomaron medidas correctivas para ajustar la planificación o asignar recursos adicionales según fuera necesario.

**Bitácora de cambios en el proyecto:** Se mantuvo una bitácora detallada de todos los cambios realizados durante el desarrollo del proyecto dentro de la aplicación asana que de igual manera funciona como calendario de actividades. Esto incluyó modificaciones en los requisitos, ajustes en el diseño, correcciones de errores y cualquier otra actualización relevante. Esta bitácora sirvió como referencia histórica y facilitó la comunicación entre los miembros del equipo.

**Seguimiento de riesgos:** Se identificaron y evaluaron los riesgos potenciales asociados con el proyecto, y se establecieron medidas para mitigarlos o gestionarlos adecuadamente. Esto implicó la identificación temprana de posibles obstáculos y la implementación de estrategias para minimizar su impacto en el desarrollo del proyecto.

Estas acciones de control fueron fundamentales para mantener el proyecto en curso y asegurar su éxito dentro de los plazos establecidos y con los estándares de calidad esperados. La combinación de reuniones periódicas, seguimiento del avance, registro de cambios y gestión de riesgos proporcionó un marco efectivo para la supervisión y el control del proyecto.

## Cierre

Tras realizar exhaustivas pruebas durante la entrega de este reporte, se ha constatado con satisfacción que la velocidad y eficiencia de los procesos han experimentado una mejora considerable con respecto al proyecto anterior desarrollado con React. Este progreso significativo en el rendimiento del sistema no solo cumple, sino que supera las expectativas iniciales del proyecto, representando un hito importante en el camino hacia la optimización continua y la excelencia operativa.

Durante la reunión de entrega de avances, se recibió con entusiasmo un nuevo diseño elaborado en Figma, el cual introduce una serie de nuevas pantallas y procesos. Este acontecimiento marca el inicio de una nueva fase de expansión y enriquecimiento para la aplicación. Entre las próximas funcionalidades a integrar se encuentran secciones dedicadas al turismo, entrega de comida y renta de autos. Estas adiciones no solo ampliarán el alcance y la utilidad de la aplicación, sino que también elevarán la experiencia del usuario a un nivel superior al proporcionarle una gama más amplia de opciones y servicios dentro de la plataforma.

Estas emocionantes novedades en desarrollo abren una ventana de oportunidad sin igual para potenciar aún más el valor y la versatilidad de nuestra aplicación. Este compromiso con la mejora continua evidencia nuestra decidida dedicación a satisfacer de manera integral y satisfactoria las necesidades y expectativas de nuestros usuarios. Ansiamos seguir avanzando en estas mejoras, con el objetivo de brindar una aplicación aún más completa y enriquecedora para nuestra creciente comunidad de usuarios.

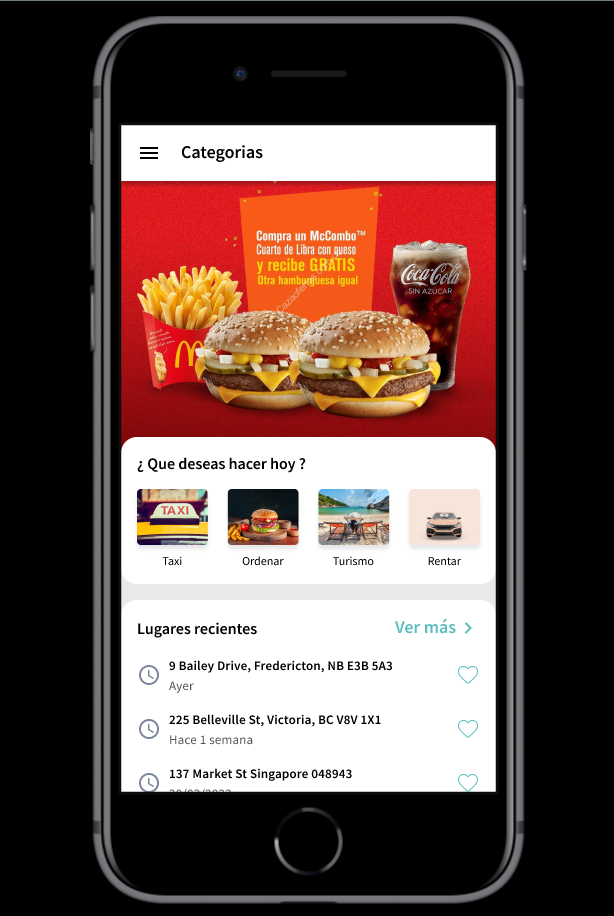


Figura .3. Figma nuevos requerimientos

|  |
| --- |
| CONCLUSIONES |

## Cumplimiento de objetivos

En el punto actual del desarrollo, el sistema no solo cumple con los objetivos establecidos, sino que también ha superado las expectativas en varios aspectos clave. Uno de los principales objetivos era mejorar la velocidad de procesamiento, y los resultados obtenidos hasta ahora indican un rendimiento significativamente más rápido en comparación con la versión anterior desarrollada en React. Además, se ha logrado un mayor rendimiento general en términos de estabilidad y eficiencia, lo que contribuye en gran medida a una experiencia de usuario más fluida y satisfactoria.

## Resultados

La entrega de las dos aplicaciones en formato APK marca un hito importante en el progreso del proyecto. La versión para pasajeros y la versión para conductores representan el resultado tangible del arduo trabajo del equipo de desarrollo. Estas versiones no solo permiten a los interesados probar y evaluar las funcionalidades implementadas hasta el momento, sino que también ofrecen una visión clara del potencial y la dirección futura de la aplicación.

## Contribuciones

Durante el desarrollo del proyecto, se han realizado diversas contribuciones significativas que han impactado positivamente tanto en la empresa como en la formación profesional del alumno involucrado. Una de las principales aportaciones a la empresa ha sido la implementación de una metodología de desarrollo ágil, que ha permitido una mayor flexibilidad y adaptabilidad a los cambios en los requisitos del proyecto. Esta metodología ha facilitado la entrega de avances de manera iterativa y ha mejorado la comunicación y colaboración entre los miembros del equipo.

Además, se han introducido nuevas tecnologías y herramientas en el proceso de desarrollo, lo que ha permitido una mayor eficiencia y calidad en la entrega de productos. Por ejemplo, la adopción de un nuevo marco de trabajo para el desarrollo de aplicaciones móviles ha mejorado la velocidad de desarrollo y la experiencia del usuario final.

En cuanto a la formación profesional del alumno, el proyecto ha proporcionado una valiosa experiencia práctica en el campo de la ingeniería de software y el desarrollo de aplicaciones móviles. La participación en el proyecto ha permitido al alumno aplicar los conocimientos teóricos adquiridos en el aula a situaciones reales, enfrentarse a desafíos técnicos y trabajar en equipo para alcanzar objetivos comunes. Además, ha tenido la oportunidad de aprender nuevas tecnologías y metodologías de desarrollo, lo que ha enriquecido su conjunto de habilidades y le ha preparado para enfrentarse a futuros desafíos profesionales en la industria.

|  |
| --- |
| **REFERENCIAS** |

3Androides. (2023). *FLUTTER Y GPS: ¿CÓMO SE ACTIVA?* Obtenido de https://www.3androides.com/actualidad/215-flutter-y-gps-como-se-activa

BBVA. (2022). *Link de pago Openpay*. Obtenido de https://www.bbva.mx/empresas/productos/medios-de-cobro/pago-electronico-Openpay.html

Fernández, D. A. (14 de Abril de 2023). *Errores al usar Private EndPoints*. Obtenido de https://ifgeekthen.nttdata.com/es/errores-al-usar-private-endpoints#:~:text=Empecemos%20por%20el%20principio%2C%20un,manual)%20de%20nuestra%20Virtual%20Network.

FIREBASE. (2023). *Recibe mensajes en una app de Flutter*. Obtenido de https://firebase.google.com/docs/cloud-messaging/flutter/receive?hl=es-419

Flores, F. (22 de Julio de 2022). *Qué es Visual Studio Code y qué ventajas ofrece*. Obtenido de https://openwebinars.net/blog/que-es-visual-studio-code-y-que-ventajas-ofrece/

Flutter. (2023). *Use the Flutter inspector*. Obtenido de https://docs.flutter.dev/tools/devtools/inspector

GOOGLE. (2020). *API de Maps Javascript*. Obtenido de https://developers.google.com/maps/documentation/javascript/overview?hl=es-419

GRUPO EBIM. (28 de Junio de 2023). *Conoce las ventajas y desventajas del desarrollo de aplicaciones con Flutter*. Obtenido de https://www.grupoebim.com/blog/ventajas-desventajas-flutter/#:~:text=El%20objetivo%20de%20Flutter%20es,videos%20y%20anuncios%20en%20l%C3%ADnea.

Ramírez, I. (10 de Junio de 2021). *Cómo instalar el emulador de Android Studio para usar Android en el PC*. Obtenido de https://www.xatakandroid.com/programacion-android/como-instalar-emulador-android-studio-para-usar-android-pc-1

RyseSoft. (19 de Agosto de 2022). *Easy Localization with Flutter*. Obtenido de https://medium.com/@rysesoft/easy-localization-with-flutter-d8ad10c5e4da

Sánchez, D. H. (21 de Enero de 2023). *Clean Architecture en Flutter*. Obtenido de https://medium.com/bancolombia-tech/clean-architecture-en-flutter-c%C3%B3mo-implementarla-en-el-2023-34f4d28d06b5

Santaella, J. (26 de Abril de 2022). *¿Qué es Android Studio?* Obtenido de https://talently.tech/blog/que-es-android-studio/

Vaidya, K. (16 de Enero de 2024). *(Flutter) futter\_native\_contact\_picker with Firebase firestore*. Obtenido de https://medium.com/@pro.vaidya4413/flutter-futter-native-contact-picker-with-firebase-firestore-851f1de945f0

|  |
| --- |
| **ANEXOS** |

**ANEXO A. NOMBRE DEL ANEXO**