



CARTA DE AUTORIZACIÓN

CÓDIGO

AP-BIB-FO-06

VERSIÓN

1

VIGENCIA

2014

PÁGINA

1 de 1

Neiva, 2024

Señores

CENTRO DE INFORMACIÓN Y DOCUMENTACIÓN

UNIVERSIDAD SURCOLOMBIANA

Ciudad

El (Los) suscrito(s):

Gabriela Florez Mora, con C.C. No. 1.007.681.768, Jairo Andrés Pastrana Hernández, con C.C. No. 1.075.302.877, Autor(es) de la tesis y/o trabajo de grado titulado FLOPA APLICACIÓN MÓVIL PARA EL ÁREA DE MANTENIMIENTO SECTOR HIDROCARBUROS MAXIM & FISHING SOLUTIONS PROVIDE S.A.S, presentado y aprobado en el año 2024 como requisito para optar al título de Ingenieros Electrónicos;

Autorizo (amos) al CENTRO DE INFORMACIÓN Y DOCUMENTACIÓN de la Universidad Surcolombiana para que, con fines académicos, muestre al país y el exterior la producción intelectual de la Universidad Surcolombiana, a través de la visibilidad de su contenido de la siguiente manera:

- Los usuarios puedan consultar el contenido de este trabajo de grado en los sitios web que administra la Universidad, en bases de datos, repositorio digital, catálogos y en otros sitios web, redes y sistemas de información nacionales e internacionales "open access" y en las redes de información con las cuales tenga convenio la Institución.
- Permita la consulta, la reproducción y préstamo a los usuarios interesados en el contenido de este trabajo, para todos los usos que tengan finalidad académica, ya sea en formato Cd-Rom o digital desde internet, intranet, etc., y en general para cualquier formato conocido o por conocer, dentro de los términos establecidos en la Ley 23 de 1982, Ley 44 de 1993, Decisión Andina 351 de 1993, Decreto 460 de 1995 y demás normas generales sobre la materia.
- Continúo conservando los correspondientes derechos sin modificación o restricción alguna; puesto que, de acuerdo con la legislación colombiana aplicable, el presente es un acuerdo jurídico que en ningún caso conlleva la enajenación del derecho de autor y sus conexos.

De conformidad con lo establecido en el artículo 30 de la Ley 23 de 1982 y el artículo 11 de la Decisión Andina 351 de 1993, "Los derechos morales sobre el trabajo son propiedad de los autores", los cuales son irrenunciables, imprescriptibles, inembargables e inalienables.

EL AUTOR/ESTUDIANTE:

Firma:

EL AUTOR/ESTUDIANTE:

Firma:

Vigilada Mineducación

La versión vigente y controlada de este documento, solo podrá ser consultada a través del sitio web Institucional www.usco.edu.co, link Sistema Gestión de Calidad. La copia o impresión diferente a la publicada, será considerada como documento no controlado y su uso indebido no es de responsabilidad de la Universidad Surcolombiana.



DESCRIPCIÓN DE LA TESIS Y/O TRABAJOS DE GRADO

CÓDIGO

AP-BIB-FO-07

VERSIÓN

1

VIGENCIA

2014

PÁGINA

1 de 3

TÍTULO COMPLETO DEL TRABAJO: FLOPA APLICACIÓN MÓVIL PARA EL ÁREA DE MANTENIMIENTO
SECTOR HIDROCARBUROS MAXIM & FISHING SOLUTIONS PROVIDE S.A.S

AUTOR O AUTORES:

Primero y Segundo Apellido	Primero y Segundo Nombre
FLOREZ MORA	GABRIELA
PASTRANA HERNANDEZ	JAIRO ANDRES

DIRECTOR Y CODIRECTOR TESIS:

Primero y Segundo Apellido	Primero y Segundo Nombre
MOLINA MOSQUERA	JOHAN JULIÁN

ASESOR (ES):

Primero y Segundo Apellido	Primero y Segundo Nombre
MARTINEZ BARRETO	GERMÁN EDUARDO
BRAVO OBANDO	MARTÍN DIOMEDES

PARA OPTAR AL TÍTULO DE: INGENIERO ELECTRÓNICO

FACULTAD: INGENIERIA

PROGRAMA O POSGRADO: INGENIERIA ELECTRÓNICA

CIUDAD: NEIVA

AÑO DE PRESENTACIÓN: 2024

NÚMERO DE PÁGINAS: 69



DESCRIPCIÓN DE LA TESIS Y/O TRABAJOS DE GRADO

CÓDIGO

AP-BIB-FO-07

VERSIÓN

1

VIGENCIA

2014



DEPARTAMENTO TECNICO SERVICIO OFICINA EXPRESA

PÁGINA 2 de 3

TIPO DE ILUSTRACIONES (Marcar con una X):

Diagramas Fotografías Grabaciones en discos Ilustraciones en general Grabados
Láminas Litografías Mapas Música impresa Planos Retratos Sin ilustraciones Tablas o Cuadros

SOFTWARE requerido y/o especializado para la lectura del documento:

MATERIAL ANEXO:

PREMIO O DISTINCIÓN (En caso de ser LAUREADAS o Meritoria):

PALABRAS CLAVES EN ESPAÑOL E INGLÉS:

Español	Inglés	Español	Inglés
1. <u>Interfaz</u>	<u>Frontend</u>	6. _____	_____
2. <u>Subordinado</u>	<u>Backend</u>	7. _____	_____
3. <u>Estructura</u>	<u>Framework</u>	8. _____	_____
4. <u>Artilugio</u>	<u>Widget</u>	9. _____	_____
5. _____	_____	10. _____	_____

RESUMEN DEL CONTENIDO: (Máximo 250 palabras)

El proyecto realizado tiene como objetivo mejorar la organización y garantizar el desarrollo de mantenimientos preventivos y correctivos realizados en la empresa Maxim Fishing SAS, para esto se realiza un prototipo de una aplicación móvil cuya funcionalidad rige el objetivo, este prototipo nos especifica mediante formularios los mantenimientos registrados, pendientes a ejecutar y los que se encuentran ejecutados, el desarrollo de esta app lo dividimos en dos (frontend – backend),

El frontend se encarga de escribir el código para hacer visible el sitio que permite la interacción con los usuarios y de la comunicación con el backend, su principal función es tomar los formatos de mantenimiento de los vehículos y pasarlo al backend, tomar los datos del backend y visualizarlos a los usuarios.



El backend se encarga de escribir el código que crea o utiliza los servicios que dan respuesta a los formatos de mantenimiento y de la comunicación con el frontend, su principal función es acceder a los datos y servicios para dar gestión al mantenimiento y así visualizarlos en el frontend

La unión de estas dos partes se crea nuestro prototipo generando una forma más fácil y organizada de realizar mantenimientos teniendo un respaldo de los formularios ya gestionados.

ABSTRACT: (Máximo 250 palabras)

The project aims to improve the organization and ensure the development of preventive and corrective maintenance performed in the company Maxim Fishing SAS, for this we made a prototype of a mobile application whose functionality governs the objective, this prototype specifies through forms the registered maintenance, pending to run and those that are executed, the development of this app we divide it into two (frontend - backend). The frontend is responsible for writing the code to make visible the site that allows interaction with users and communication with the backend, its main function is to take the maintenance forms of vehicles and pass them to the backend, take data from the backend and display them to users. The backend is responsible for writing the code that creates or uses the services that respond to the maintenance formats and communication with the frontend, its main function is to access the data and services to manage the maintenance and display them in the frontend. The union of these two parts creates our prototype generating an easier and more organized way to perform maintenance having a backup of the forms already managed.

APROBACION DE LA TESIS

Martin Bravo Obando

Nombre Jurado:

Firma:

Germán Martínez

Nombre Jurado:

Firma:

FLOPA
APLICACIÓN MÓVIL PARA EL ÁREA DE MANTENIMIENTO SECTOR
HIDROCARBUROS MAXIM & FISHING SOLUTIONS PROVIDE S.A.S

GABRIELA FLOREZ MORA
JAIRO ANDRES PASTRANA HERNANDEZ

UNIVERSIDAD SURCOLOMBIANA
FACULTAD DE INGENIERIA
PROGRAMA DE INGENIERIA ELECTRONICA
NEIVA – HUILA
2024

FLOPA
APLICACIÓN MÓVIL PARA EL ÁREA DE MANTENIMIENTO SECTOR
HIDROCARBUROS MAXIM & FISHING SOLUTIONS PROVIDE S.A.S

GABRIELA FLOREZ MORA
JAIRO ANDRES PASTRANA HERNANDEZ

Informe proyecto de grado para optar al título de Ingeniero Electrónico

Director:
Ing. Johan Julián Molina Mosquera

Jurados:
Ing. Germán Eduardo Martínez Barreto
Ing. Martín Diomedes Bravo Obando

UNIVERSIDAD SURCOLOMBIANA
FACULTAD DE INGENIERIA

**PROGRAMA DE INGENIERIA ELECTRONICA
NEIVA – HUILA
2024**

Nota de Aceptación

Presidente del Jurado

Jurado

Jurado

Este proyecto está dedicado a todas aquellas personas que estuvieron involucradas en el proceso formativo durante la carrera universitaria, familia, docentes y compañeros de estudio, gracias por brindarnos su apoyo durante este largo camino.

AGRADECIMIENTOS

Agradecemos en primera instancia a la empresa MAXIM & FISHING SOLUTIONS PROVIDE S.A.S, por la oportunidad de trabajar en sus instalaciones en el área de mantenimiento; en segundo lugar, a nuestros asesores Johan Julián Molina (Asesor Interno de la universidad Surcolombiana) y Daniel Felipe Avilés Paipa tecnólogo en software (Asesor Externo), quienes fueron ejes fundamentales de conocimientos y experiencias para la elaboración de este proyecto, por ultimo a nuestras familias y amigos que nos han apoyado en este largo camino de aprendizaje.

RESUMEN

La compañía Maxim Fishing SAS es una empresa que ofrece soluciones integrales de ingeniería al sector de hidrocarburos, relacionadas con trabajos de pesca, adquisición de datos con tecnología WIRELINE, pruebas y mantenimiento de pozos, su principal función es mantener los vehículos y las plantas eléctricas en perfecto estado para que estos puedan estar disponibles y ser utilizados en campo.

La empresa maneja una plataforma Autogest cuya función principal es subir los mantenimientos que estén pendientes, ya hechos y próximos a realizar, lo que previene el mantenimiento electromecánico de los vehículos y plantas eléctricas, sin embargo, esta plataforma no cuenta con información en tiempo real, de igual forma la evidencia de algún mantenimiento se verifica por medio de Google drive y de ahí se genera el formato para ser subido en Autogest ,por lo cual no brinda una información certera, se pierden datos y/o evidencia, ni tampoco se tiene una información clara al momento de buscar algún mantenimiento.

En base a lo anterior este proyecto tiene como finalidad mejorar la organización y garantizar el desarrollo de mantenimientos preventivos y correctivos ejecutados en la empresa Maxim Fishing SAS, mediante la realización de un prototipo de una aplicación móvil, cuya funcionalidad cumpla con sus objetivos.

El prototipo se especificará mediante formularios de mantenimientos que estarán registrados, pendientes a ejecutar y los que se encuentran ejecutados, el desarrollo de esta aplicación se divide en dos el primero es Frontend – Backend que se encarga de la interacción con los usuarios por medio de interfaces y de la comunicación con el Backend, su principal función es registrar los formatos de mantenimiento de los vehículos y guardarlos en el Backend, luego se extraen los datos del Backend para visualizarlos en una interfaz a los usuarios. El segundo es el Backend encargado de registrar y guardar la información que se suministra desde el Frontend, además, se utilizarán los servicios que dan respuesta a los formatos de mantenimiento y de la comunicación con el Frontend, su principal función es acceder a los datos y servicios para dar gestión al mantenimiento, así visualizarlos en el Frontend. La unión de estas dos partes (Frontend y Backend) se crea un prototipo, en donde, facilitará el registro de los formularios de mantenimiento de manera organizada y fácil de interactuar con los usuarios.

PALABRAS CLAVE: Frontend, Backend, Framework, Widget.

CONTENIDO	
INTRODUCCIÓN	8
CAPITULO 1: ASPECTOS INTRODUCTORIOS.....	9
1.1 OBJETIVOS.....	9
1.1.1 OBJETIVO GENERAL	9
1.1.2 OBJETIVOS ESPECÍFICOS.....	9
1.2 PLANTEAMIENTO DEL PROBLEMA	10
1.2.1 DEFINICIÓN DEL PROBLEMA.....	10
1.3 JUSTIFICACIÓN.....	11
1.4 GLOSARIO	12
CAPITULO 2: MARCO TEORICO.....	14
2.1 APPLICACIÓN MOVIL.....	14
2.1.1APLICACIONES NATIVAS	14
2.1.2 APLICACIONES WEB	14
2.1.3 APLICACIONES HÍBRIDAS.....	14
2.2 MANTENIMIENTO	15
2.2.1 MP (MANTENIMIENTO PREVENTIVO)	15
2.2.2 MC (MANTENIMIENTO CORRECTIVO)	15
CAPITULO 3: METODOLOGIA	16
3.1 MATERIALES.....	16
3.2 METODOLOGÍA	16
3.3 DESARROLLO DEL PROYECTO	17
3.3.1 ETAPA DE ANALISIS	18
3.3.2 ETAPA DE DISEÑO	19
CAPITULO 4: PRUEBAS DE FUNCIONAMIENTO.....	54
4.1 VALIDANDO EL LOGIN	55
4.2 VALIDANDO DEL PANEL FRONTAL	63
CRONOGRAMA.....	66
FINANCIACIÓN.....	66
CONCLUSIONES.....	67
BIBLIOGRAFÍA.....	68

INTRODUCCIÓN

En la última década, el auge de la tecnología móvil ha revolucionado las grandes empresas, ya que pueden gestionar grandes cantidades de datos y métricas para tomar mejores decisiones, mediante aplicaciones móviles que bridan acceso instantáneo a la información, por ende, la implementación de una aplicación móvil en la empresa Maxim Fishing SAS mejorara los procesos de mantenimiento preventivo y correctivo en los vehículos y las plantas eléctricas.

Esta aplicación móvil es un programa diseñado específicamente para ser utilizado en dispositivos móviles, como teléfonos inteligentes y tabletas, se puede instalar desde tiendas de aplicaciones oficiales, como App Store para iOS y Google Play para Android, o incluso desde otras fuentes de descarga.

En el presente informe se encontrará el diseño de una aplicación móvil para el área de mantenimiento en el sector de hidrocarburos, con el objetivo de tener los reportes correctivos y preventivos al día, dejando sus equipos operativos y funcionales para ser utilizados en campo, en donde, se realiza la conversión de datos, documentos y procesos analógicos a digitales.

Para llevarse a cabo el prototipo de la aplicación móvil se debe realizar la comunicación entre cliente servidor por medio del ethernet, al realizarse dicha comunicación se procede a la creación de las interfaces (Frontend) que interactuaran y la creación del banco de datos (Backend) para que los usuarios puedan ingresar a la aplicación y diligenciar los reportes de mantenimiento programados por la empresa y ser ejecutado dicho reporte por el operario (mecánico, electricista, etc.), en el cual, el personal pueda subir la evidencia sin tener inconvenientes y en tiempo real para darle cierre a los reportes y con eso prestar un mejor servicio de los equipos que son la mayor fuente de ingresos de la empresa.

CAPITULO 1: ASPECTOS INTRODUCTORIOS

1.1 OBJETIVOS

1.1.1 OBJETIVO GENERAL

Implementar una aplicación que ayude al área de mantenimiento electromecánico a tener los reportes correctivos y preventivos al día, dejando sus equipos operativos y funcionales para ser utilizados en campo.

1.1.2 OBJETIVOS ESPECÍFICOS

- Diseñar el modelo cliente servidor de la aplicación de software.
- Analizar los reportes de mantenimientos ejecutados y programados.
- Diseñar el modelo de Frontend y Backend de la aplicación móvil.
- Validar el funcionamiento del prototipo de aplicación de software.

1.2 PLANTEAMIENTO DEL PROBLEMA

1.2.1 DEFINICIÓN DEL PROBLEMA

La compañía Maxim Fishing SAS cuenta con diversas áreas, este proyecto está enfocado en el mantenimiento electromecánico, el cual se encarga de realizar el mantenimiento a los vehículos y plantas eléctricas.

En esta área se utiliza una plataforma llamada Autogest cuya función principal es subir los mantenimientos que hay pendientes, ya hechos y próximos a realizar, la falla que presenta esta plataforma es que no cuenta con información en tiempo real, a parte para subir la evidencia de algún mantenimiento el encargado hace el cierre por medio de Google drive y de ahí se genera el formato para ser subido en autogest ,por lo cual no brinda una información certera, se pierden datos y/o evidencia, ni tampoco se tiene una información clara al momento de buscar algún mantenimiento.

Por esta razón se plantea la siguiente pregunta: ¿Se puede implementar una aplicación más completa que permita diligenciar, observar e ingresar soportes de mantenimiento en tiempo real?

1.3 JUSTIFICACIÓN

La compañía Maxim Fishing SAS presta diversos servicios principalmente el mantenimiento electromecánico, su principal función es mantener los vehículos y las plantas eléctricas en perfecto estado para que puedan ser utilizados en campo.

La empresa cuenta con un formato del mantenimiento preventivo y correctivo, en el cual se realizan los reportes especificando las inconsistencias encontradas al igual que las soluciones a implementar, a su vez contienen una base de datos de dichos reportes en el servidor, sin embargo no es eficiente debido a que la información se obtiene del drive y el personal que tiene acceso a ello en muchas ocasiones no cuentan con la disponibilidad para subir la evidencia que cierra los reportes, por lo tanto se busca realizar una aplicación móvil en la cual el personal pueda subir la evidencia sin tener inconvenientes en tiempo real, para darle cierre a los reportes y así prestar un mejor servicio de los equipos que son la mayor fuente de ingresos de la empresa.

1.4 GLOSARIO

FRONTEND: El Frontend es la parte del desarrollo web que se dedica a la parte frontal de un sitio web, en pocas palabras del diseño de un sitio web, desde la estructura del sitio hasta los estilos como colores, fondos, tamaños hasta llegar a las animaciones y efectos. Es esa parte de la página con la que interaccionan los usuarios de la misma, es todo el código que se ejecuta en el navegador de un usuario, al que se le denomina una aplicación cliente, es decir, todo lo que el visitante ve y experimenta de forma directa.¹

BACKEND: El Backend procesa la información que alimentará el Frontend de datos. Es la capa de acceso a los datos, ya sea de un software o de un dispositivo en general, es la lógica tecnológica que hace que una página web funcione, lo que queda oculto a ojos del visitante. El Backend de una solución, determina qué tan bien se ejecutará la aplicación y qué experiencia, positiva o negativa, obtendrá el usuario de su uso.²

FRAMEWORK: Un Framework es un marco o esquema de trabajo generalmente utilizado por programadores para realizar el desarrollo de software. Utilizar un Framework permite agilizar los procesos de desarrollo ya que evita tener que escribir código de forma repetitiva, asegura unas buenas prácticas y la consistencia del código.³

KOTLIN: Kotlin es un lenguaje de programación de tipo estático que se ejecuta en la máquina virtual de Java (JVM) y también se puede compilar en JavaScript o código nativo. Fue desarrollado por JetBrains, la compañía detrás del entorno de desarrollo integrado (IDE) IntelliJ IDEA, con el objetivo de mejorar la productividad de Java y proporcionar una alternativa a Java en la plataforma Android. Kotlin es conciso, expresivo y seguro, y tiene muchas funciones que facilitan la escritura de código limpio y fácil de mantener, como seguridad nula, funciones de extensión y lambdas. También es totalmente interoperable con Java, por lo que el código Java existente se puede usar en proyectos de Kotlin y viceversa. Kotlin ha ganado

¹ Servnet. Definicion ¿Qué es Frontend? Párrafo 1 y 2. Ciudad de México marzo 2021.

² HubSpot. Definicion Qué es Backend? Párrafo 1

³ ARIMETRICS. Definición ¿Qué es Framework? Párrafo 1. <https://www.arimetrics.com/glosario-digital/framework>. Madrid.

popularidad en los últimos años y se usa ampliamente para el desarrollo de aplicaciones de Android, la programación del lado del servidor y otros tipos de aplicaciones.⁴

⁴ Elena Canorea. ¿Qué es Kotlin y para qué sirve? España, marzo de 2022.

CAPITULO 2: MARCO TEORICO

2.1 APPLICACIÓN MOVIL

Una aplicación móvil, también llamada app móvil, es un tipo de aplicación diseñada para ejecutarse en un dispositivo móvil, que puede ser un teléfono inteligente o una tableta. Incluso si las aplicaciones suelen ser pequeñas unidades de software con funciones limitadas, se las arreglan para proporcionar a los usuarios servicios y experiencias de calidad.⁵

2.1.1 APPLICACIONES NATIVAS

Las Aplicaciones Nativas están diseñadas para un único sistema operativo móvil. Por eso se denominan nativos: son nativos de una plataforma o dispositivo en particular. La mayoría de las aplicaciones móviles actuales están diseñadas para sistemas como Android o iOS. En pocas palabras, no puedes instalar ni usar una aplicación de Android en iPhone y viceversa.⁶

2.1.2 APPLICACIONES WEB

Las aplicaciones web son aplicaciones de software que se comportan de manera similar a las aplicaciones móviles nativas y funcionan en dispositivos móviles. Sin embargo, existen diferencias significativas entre las aplicaciones nativas y las aplicaciones web. Para empezar, las aplicaciones web utilizan navegadores para ejecutarse y, por lo general, están escritas en los lenguajes CSS, HTML5 o JavaScript. Dichas aplicaciones redirigen al usuario a la URL y luego les ofrecen la opción de instalar la aplicación. Simplemente crean un marcador en su página. Por eso requieren una memoria mínima del dispositivo.⁷

2.1.3 APPLICACIONES HÍBRIDAS

Las aplicaciones híbridas, a diferencia de las nativas, son aquellas capaces de funcionar en distintos sistemas operativos móviles. Entre ellos: Android, iOS y Windows Phone. De esta manera, una misma app puede utilizarse en cualquier Smartphone o Tablet, indistintamente de su marca o fabricante. Para ello, estas

⁵ Luis Herazo. ¿Qué es una aplicación móvil? Párrafo 3. Anincubator.

⁶ Luis Herazo. Aplicaciones Nativas. Párrafo 7. Anincubator.

⁷ Luis Herazo. Aplicaciones Web. Párrafo 10. Anincubator.

aplicaciones tienen componentes que permiten la adaptabilidad de un mismo código a los requerimientos de cada sistema.⁸

2.2 MANTENIMIENTO

El mantenimiento son las acciones necesarias para conservar y restaurar un elemento o equipo en condiciones específicas.

2.2.1 MP (MANTENIMIENTO PREVENTIVO)

El mantenimiento preventivo es el que se realiza con el ánimo de conservar el buen estado del equipamiento o de la instalación sin necesidad de defecto alguno o de que presente un mal funcionamiento.⁹

2.2.2 MC (MANTENIMIENTO CORRECTIVO)

El mantenimiento correctivo es aquel que realiza modificaciones en la instalación o en el equipamiento para devolverlo a su estado original¹⁰

⁸ Berna Network. ¿Que son las aplicaciones híbridas? Párrafo 8.

⁹ Properly. Qué es el mantenimiento preventivo. Párrafo 4.

¹⁰ Properly Qué es el mantenimiento Correctivo. Párrafo 3.

CAPITULO 3: METODOLOGIA.

3.1 MATERIALES

Componentes físicos que se utilizaron fueron:

- Computador
- Teléfono Móvil

Los softwares que se utilizaron fueron:

- Visual Studio Code
- FLUTTER con lenguaje Dart

3.2 METODOLOGÍA

Para llevar a cabo el proyecto se empleará la siguiente metodología:

Fase 1: El modelo cliente servidor, el cual consiste que mediante una aplicación móvil el usuario que ingresa pueda realizar los reportes de mantenimiento preventivos y/o correctivos de los vehículos y las plantas eléctricas, dichos reportes serán enviados por medio de internet y almacenados en un servidor como banco de datos (Cloud Services) con la finalidad de que la persona encargada revise dichos reportes puedan llevar un registro del tipo de mantenimiento que se programó y está próximo a ejecutar.

Fase 2: El diseño del modelo Frontend, este consiste en que la persona encargada de realizar el mantenimiento ingresará al sistema mediante su respectivo usuario y contraseña para acceder al panel principal, el cual mostrará el menú de funciones que necesita realizar el usuario, ya sea el diligenciamiento del tipo de mantenimiento a realizar del respectivo equipo o vehículo, como el proceso de checking con sus soportes.

Fase 3: El diseño del Backend, encontraremos la programación la cual nos permite realizar la comunicación entre el Frontend y el servidor con una velocidad de respuesta óptima, además, se implementará autenticación para el usuario/contraseña de la persona y seguridad para los datos que se almacenan.

Fase 4: Validar funcionamiento de la aplicación, se realizarán pruebas a la aplicación antes del lanzamiento del prototipo, se debe verificar la comunicación cliente servidor, autenticación de la cuenta y contraseña, la velocidad de respuesta de la aplicación siendo la más optima, por último, el adecuado almacenamiento de los datos y soportes que se ingresan a la aplicación.

3.3 DESARROLLO DEL PROYECTO

El desarrollo del proyecto tiene varias etapas en las cuales se van a tener en cuenta para el desarrollo de una aplicación móvil como se muestra en la figura 1, en donde, podemos aplicarlas a nuestra aplicación móvil, además, de tener en cuenta las fases que se mencionaron en la metodología.

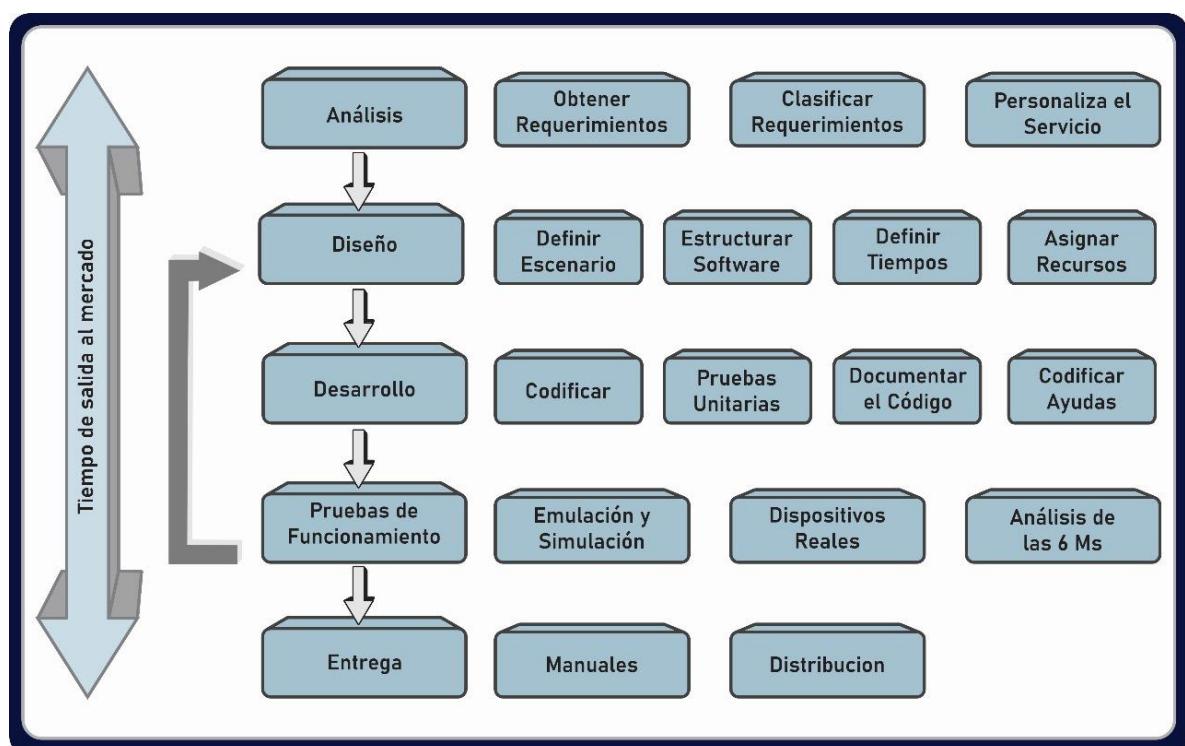


Figura 1.

Fuente: Maira Cecilia Gasca Mantilla¹, Luis Leonardo Camargo Ariza², Byron Medina Delgado³. Metodología para el desarrollo de aplicaciones móviles. Artículo de Investigación. *Tecnura* vol.18 no.40 Bogotá Apr./June 2014. Disponible en: http://www.scielo.org.co/scielo.php?pid=S0123-21X2014000200003&script=sci_arttext#f1.

3.3.1 ETAPA DE ANALISIS

En esta etapa se puede inferir al tipo de servicio que se va a necesitar o a qué modelo cliente servidor se implementará a la aplicación móvil, en este caso, se optó por el modelo cliente servidor de tres etapas, en donde, tenemos la sección 1 de los clientes o usuarios, que son todos los dispositivos que cuente con conexión de internet (teléfonos móviles, computadores, etc.), en la sección 2 pertenece al medio de comunicación que tendrá el cliente con el servidor como se puede observar en la figura 2. Por último, la sección 3 tenemos los servidores, en el cual, se encarga de almacenar todos los datos provenientes de la sección 1 cumpliendo con la fase 1 que se mencionó en la metodología.

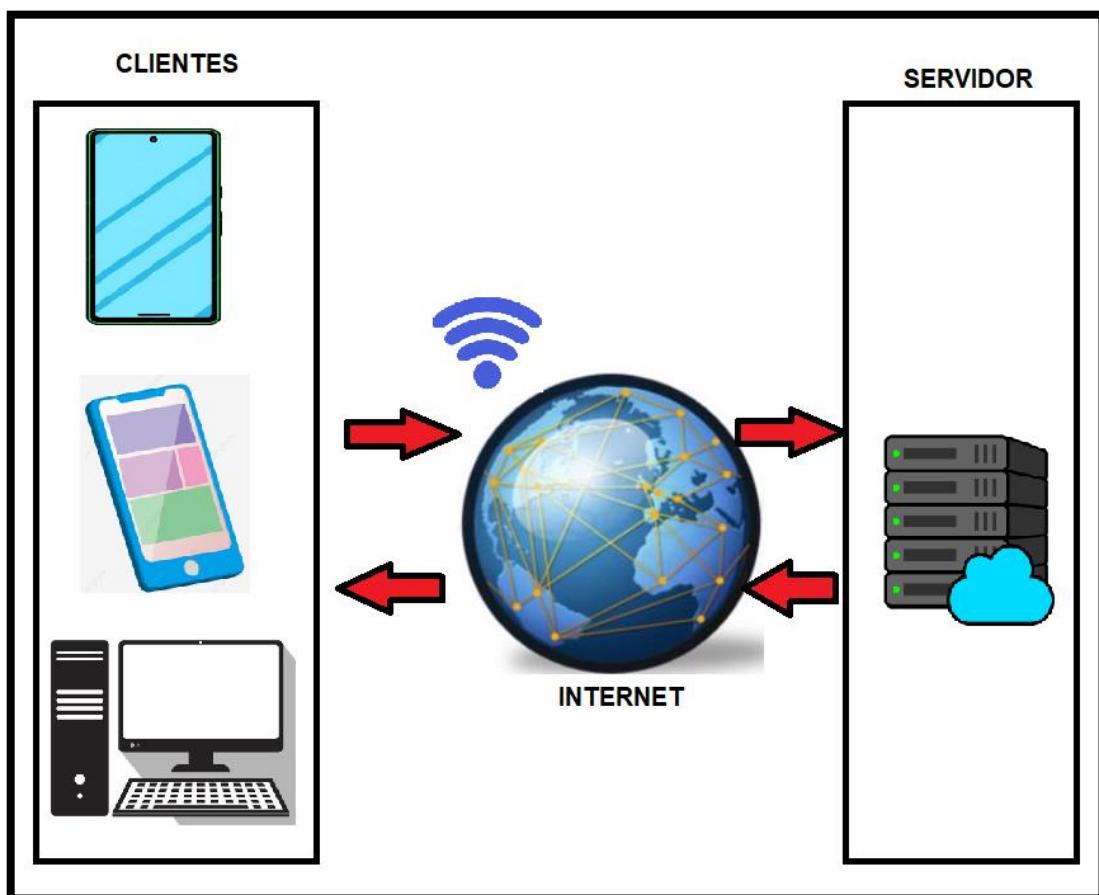


Figura 2. Modelo Cliente Servidor de tres etapas.

Fuente: Elaboración propia.

3.3.2 ETAPA DE DISEÑO

Se realizará el diseño teniendo definido la estructura que va tener la aplicación móvil, para este caso se desarrollará la fase 2 de la metodología que se enfoca en los diseños tanto para Frontend (interfaz donde interactúa el usuario o cliente) como para el Backend (almacenamiento de datos).

3.3.2.1 DISEÑO DEL FRONTEND

El Frontend como se menciona anteriormente son interfaces que el usuario interactúa con la aplicación móvil, para el desarrollo del Frontend se realizó con ayuda del Framework libre llamado FLUTTER, en donde, su principal funcionamiento es la creación de WIDGETS para las interfaces de la aplicación móvil. Un WIDGET es una clase que puede tener argumentos posicionales y argumentos con nombres, en las interfaces residen muchos WIDGETS, pero siempre siguiendo una estructura y a esto se le conoce como ÁRBOL DE WIDGETS, un ejemplo de lo anterior se observa en la siguiente imagen de cómo está construido una interfaz con la estructura ÁRBOL DE WIDGETS.

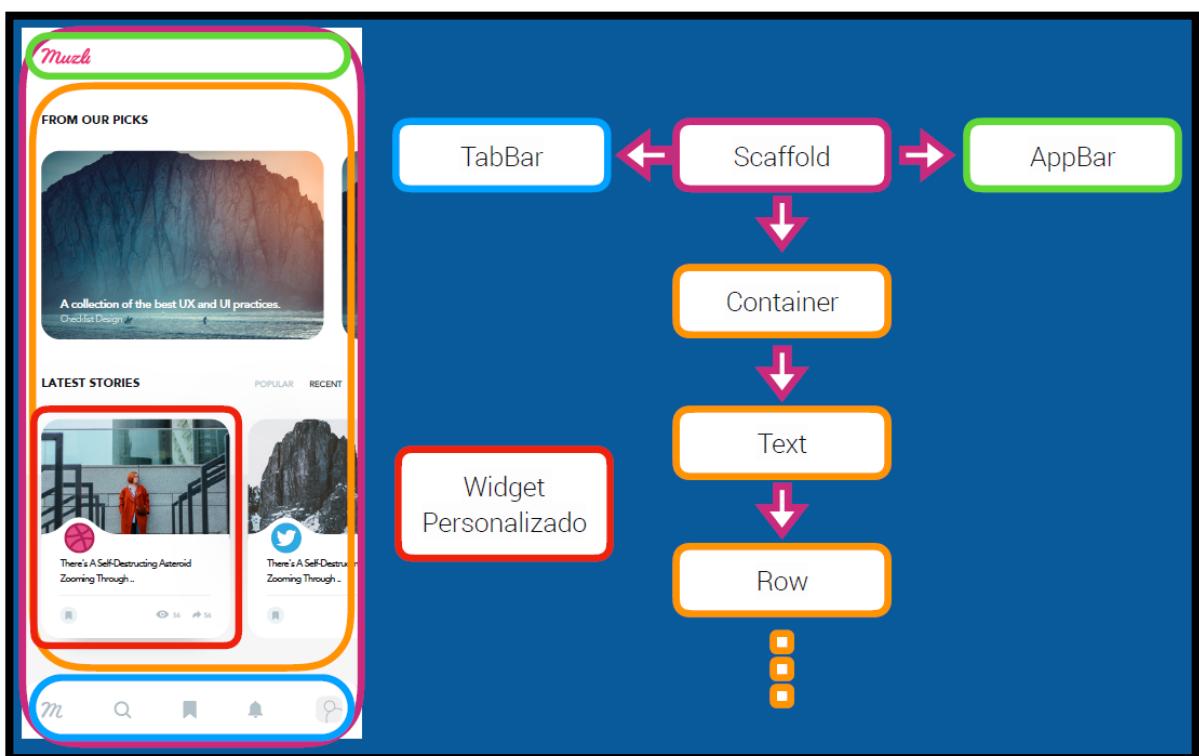


Imagen 1. Estructura de un interfaz hecho en FLUTTER.

En resumen, para realizar una interfaz en FLUTTER se debe tener en cuenta que está creando muchos WIDGETS para la apariencia y contenido de la interfaz, como si fueran bloques de lego que se van apilando según la necesidad que se requiera para la aplicación móvil.



Imagen 2. Vista de widgets como bloque de lego.

El Framework libre FLUTTER utiliza el lenguaje Dart desarrollado por Google, dicho lenguaje para los programadores de aplicaciones móviles es el más sencillo y fácil de aprender, ya que al compilar los cambios realizados en el código se puedan ejecutar inmediatamente. El lenguaje Dart se puede compilar en lenguaje nativo, por lo que no es necesario un entorno especial para ejecutar. En caso de que se haga en desarrollo web, Dart se transpila a JavaScript.

Se decidió realizar la aplicación móvil en el Framework libre FLUTTER con la intención de que es compatible con el desarrollo de aplicaciones en seis plataformas entre estas tenemos: iOS, Android, Web, Windows, MacOS y Linux. En el caso de

las plataformas iOS y Android, el lenguaje Dart se transpila al lenguaje nativo KOTLIN, en donde, el lenguaje nativo KOTLIN es la mejor opción para el desarrollo de aplicaciones móviles destinadas a sistemas operativos como iOS y Android.

3.3.2.1.1 DESARROLLO DEL FRONTEND

El diseño que se va a desarrollar en el Frontend va a tener la siguiente estructura mostrada en el diagrama de flujo.

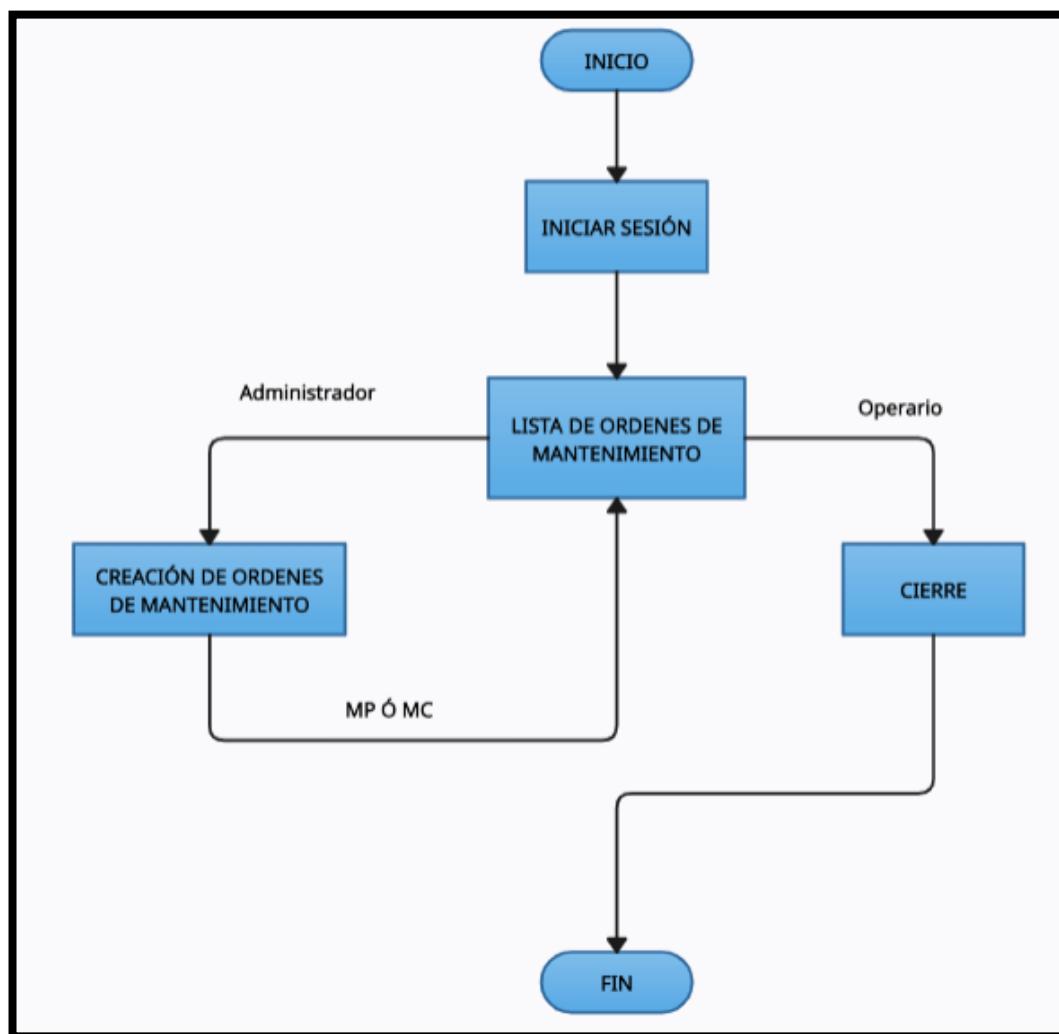


Figura 3. Estructura del Frontend
Fuente: Elaboración propia.

PASO 1: INICIAR SESION (LOGIN)

En el paso 1 tenemos el recuadro de inicio de sesión del Frontend, también conocido como Login, en donde, el Login es el encargado de realizar la fase 2 de la metodología, en el cual consiste, que el usuario ingresará y diligenciará su respectivo usuario y contraseña para acceder al panel principal, con ayuda de la base de datos que tiene el servidor en este caso el Backend, se verifica si el usuario ingresó correctamente los datos. Lo anterior es demostrado por el diagrama de flujo mostrado en la figura 6.

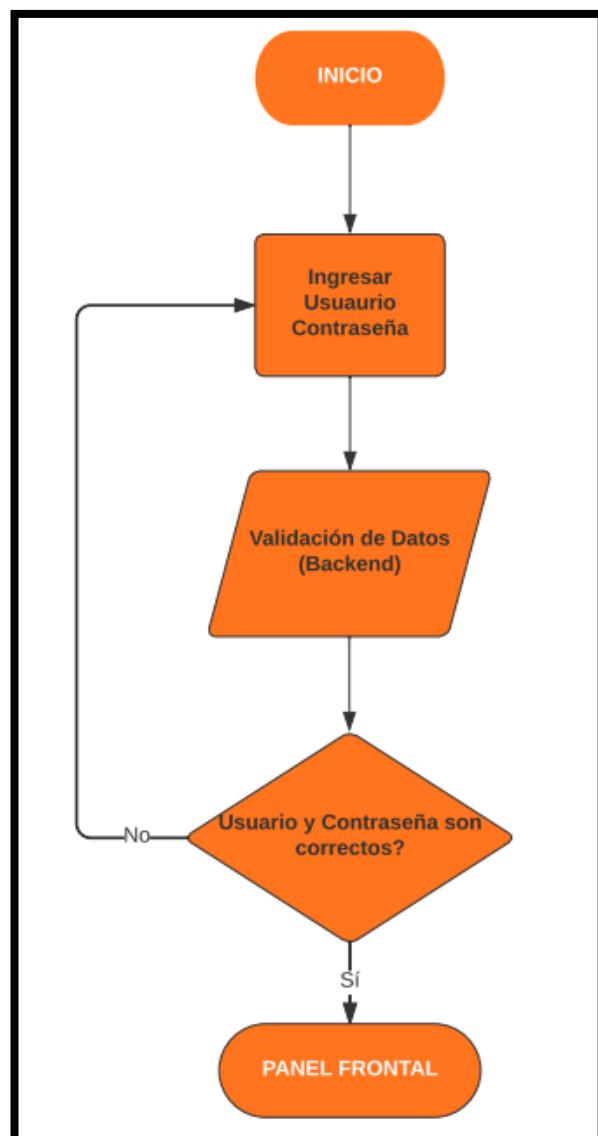


Figura 4. Diagrama de flujo del Login (Inicio de Sesión)

Fuente: Elaboración propia.

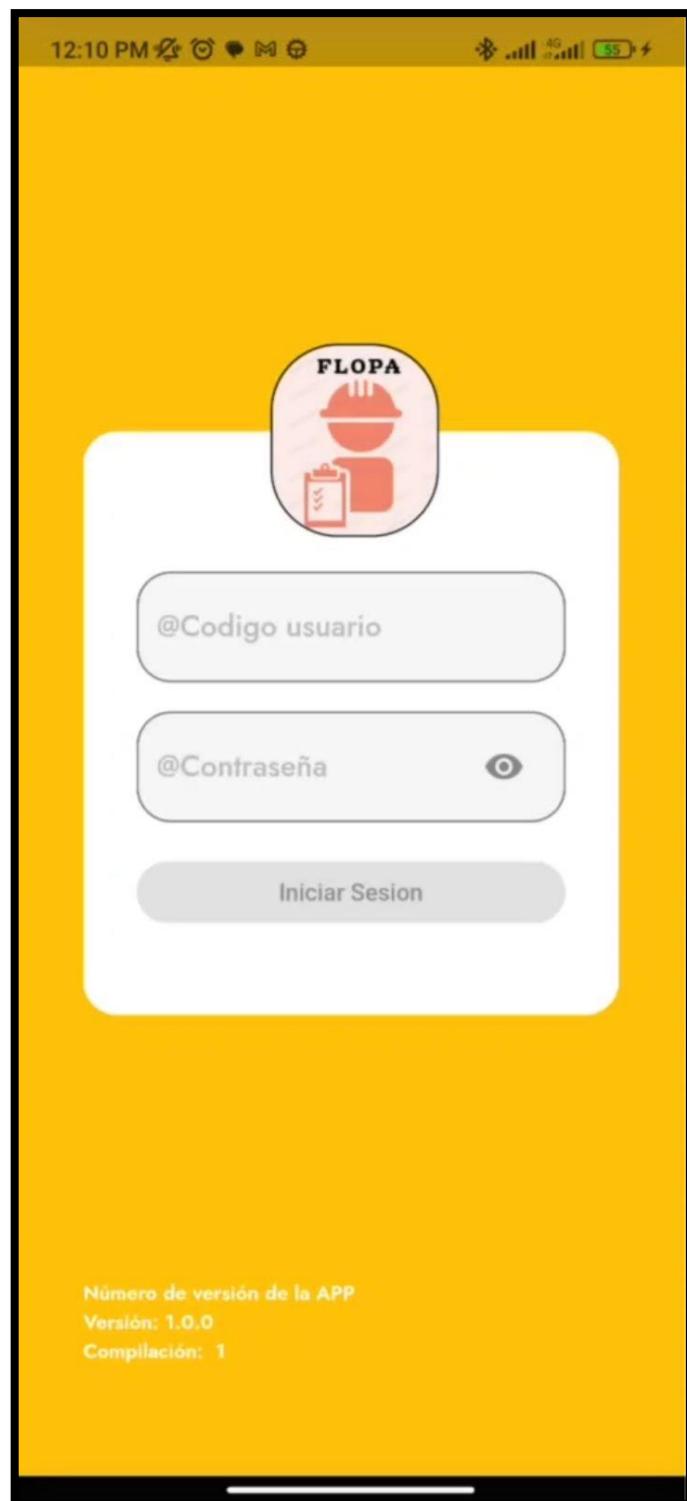


Imagen 3. Interfaz de Login de la App
Fuente: Elaboración propia.

PASO 2: PANEL FRONTAL



Imagen 4. Interfaz del Panel Frontal de la App.
Fuente: Elaboración propia.

En el interfaz del panel frontal se observa la lista de las ordenes de mantenimientos que están pendientes a realizar su respectivo cierre.



Imagen 5. Iconos del Panel Frontal de la App.

Fuente: Elaboración propia.

Se observan dos iconos en la parte superior derecha, en el cual, el icono del lado izquierdo pertenece a la creación del formato nuevo (creación de una nueva orden de mantenimiento), y el icono del lado derecho pertenece al cierre (mantenimiento que realiza el operario), que el operario quiera dar como finalizado una de las ordenes que aparece en la lista del panel frontal.

El icono de creación de formato nuevo, solo aparece cuando al iniciar sesión ingresa el administrador, ya que al momento en el que el operario ingrese no tendrá esta opción, pero le permite interactuar con la lista de los mantenimientos pendientes.

Las listas de mantenimiento que aparecen en el panel frontal son creadas por el administrador como se mencionó anteriormente, con el objetivo de que cada mantenimiento preventivo y correctivo vayan dirigidos a su respectivo operario con la descripción de lo que se le debe hacer, es decir, el administrador deberá elegir al operario competente para realizar el respectivo mantenimiento y hará una breve descripción de lo que se debe hacer, como aparece como ejemplo en la imagen 6.

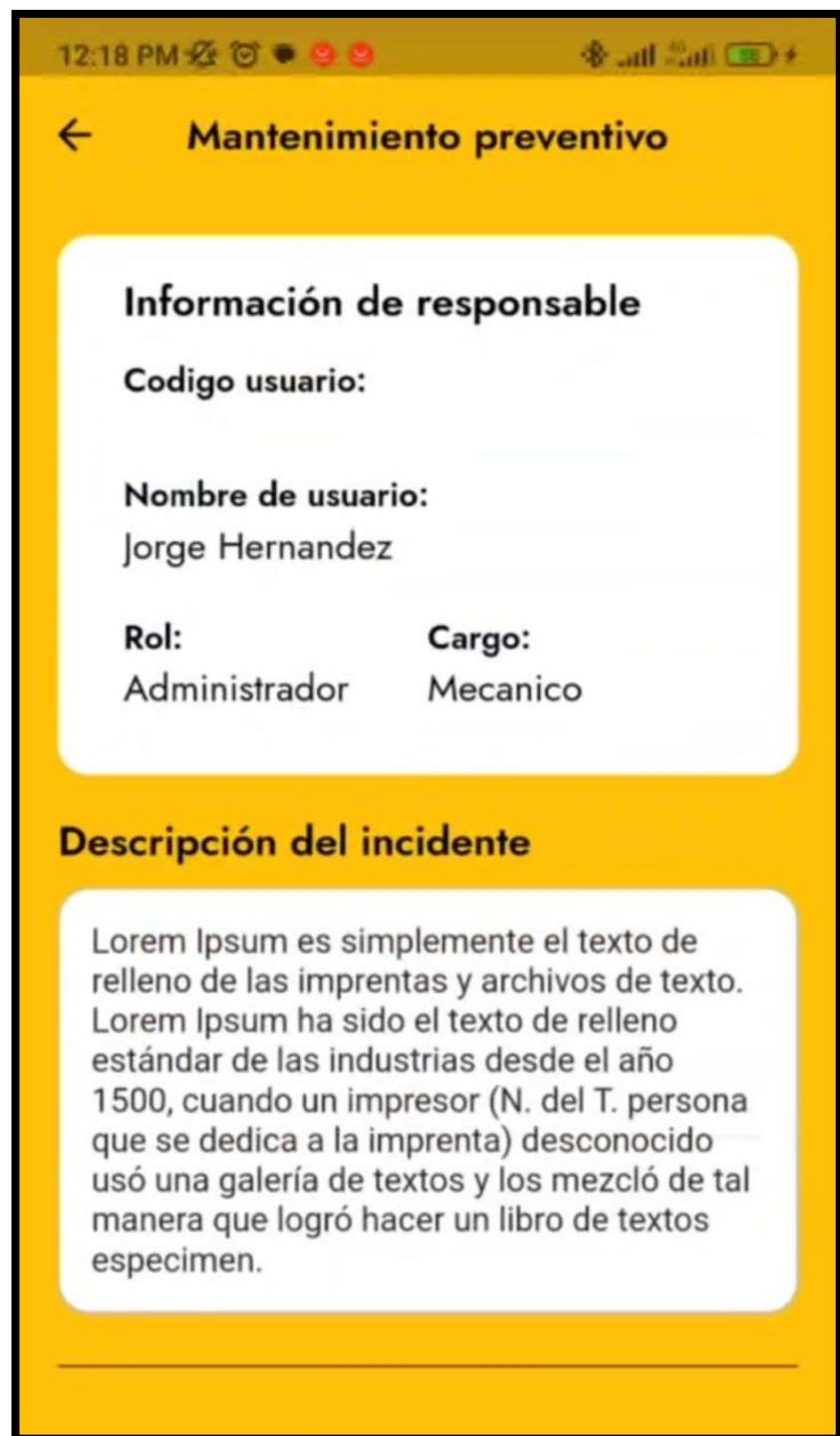


Imagen 6. Iconos del Panel Frontal de la App.
Fuente: Elaboración propia.

PASO 3 CREACIÓN DE ORDEN DE MANTENIMIENTO:

La interfaz “CREACIÓN DE ORDEN DE MANTENIMIENTO”, será realizado por el administrador, en donde, tendrá la obligación de llenar algunos datos para la creación del formato de mantenimiento bien sea MP (Mantenimiento Preventivo) o MC (Mantenimiento Correctivo), dichos datos serían: personal asignada, tipo mantenimiento, el mantenimiento a que va dirigido (Vehículo o Planta), si es para el vehículo automotriz este deberá diligenciar la placa del vehículo y si es para la planta eléctrica deberá diligenciar la marca o modelo de la planta eléctrica, tal como se observa en la siguiente imagen.

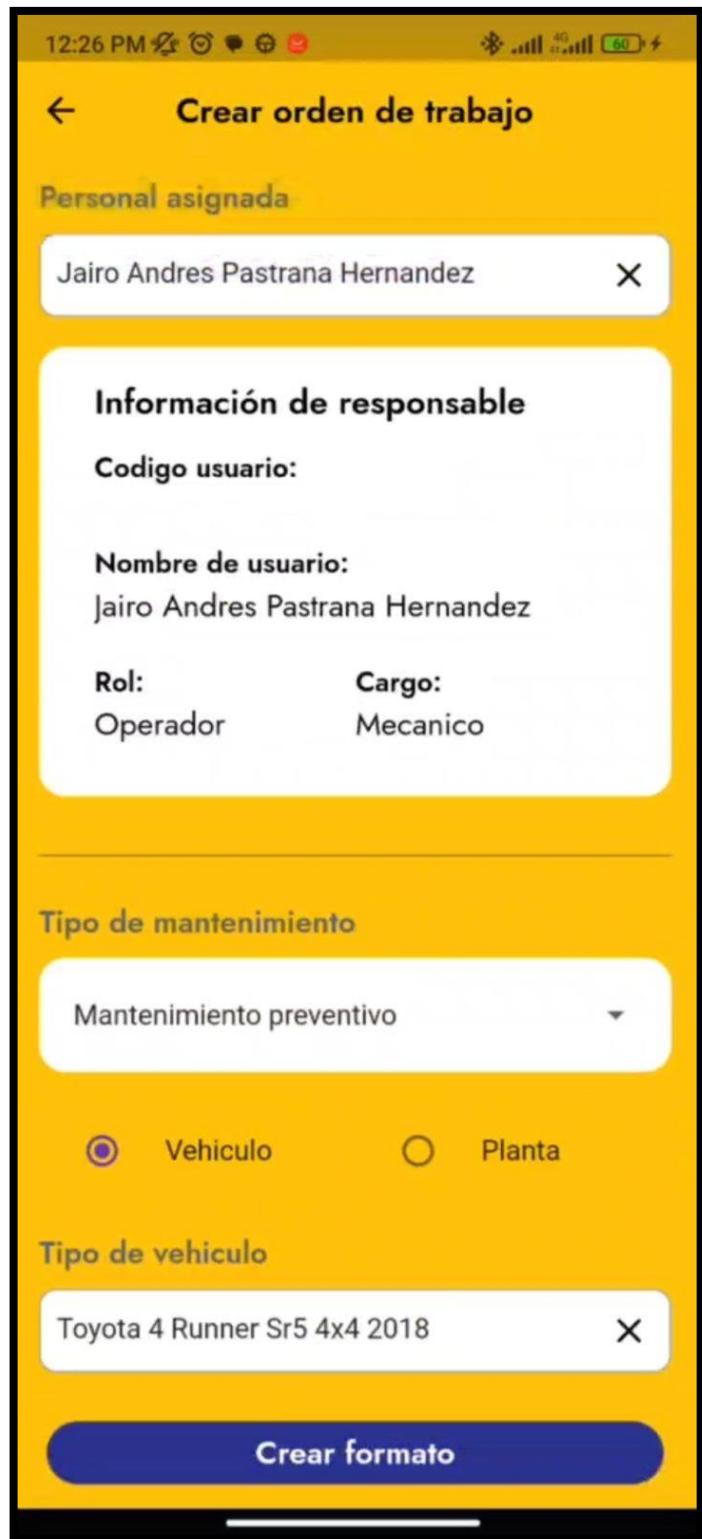


Imagen 7

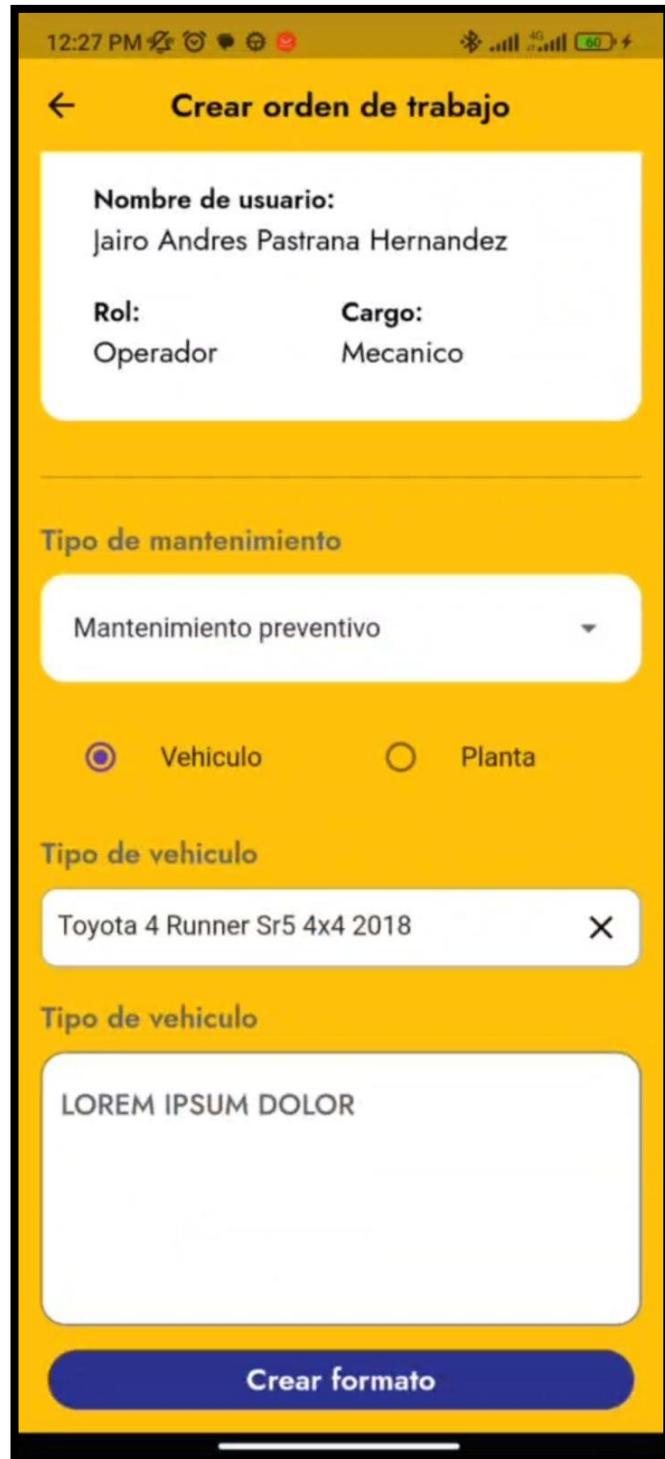


Imagen 8.

Imagen 7 y 8. CREACIÓN DE ORDEN DE MANTENIMIENTO (Vehículo)
Fuente: Elaboración propia.

Luego, de que se diligencie dichos datos y al presionar el botón “CREAR FORMATO”, dicho orden de mantenimiento quedara creada en la lista de pendientes del panel frontal para poder realizar su respectivo cierre que el operario tiene acceso.

PASO 4 CIERRE DE ORDEN DE MP (MANTENIMIENTO PREVENTIVO):

El operario al ingresar a la opción de cierre del panel frontal encontrará una lista de formatos pendientes listos para ser ejecutados, los operarios deben tener en cuenta el tipo de mantenimiento que van a realizar MP o MC, ya que para cada tipo de mantenimiento tiene diferentes interfaces.

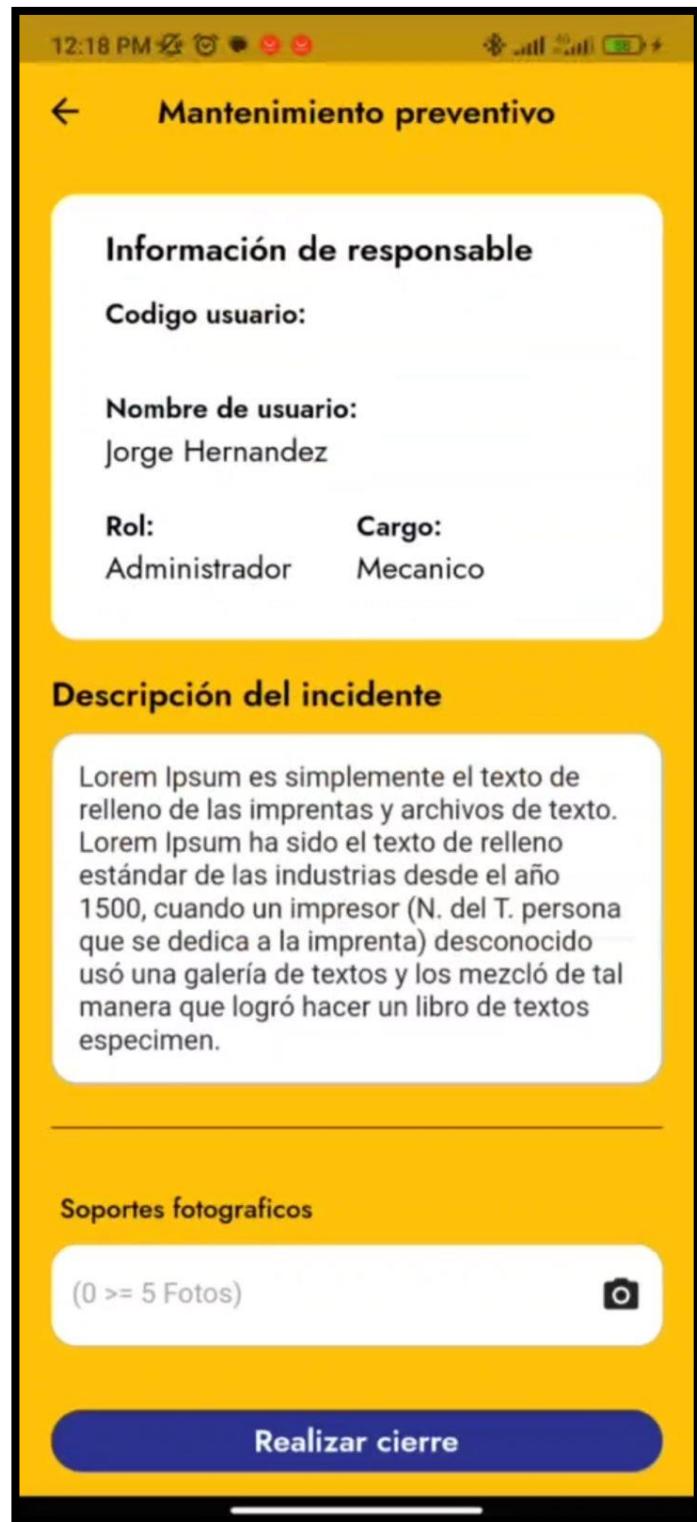


Imagen 9.

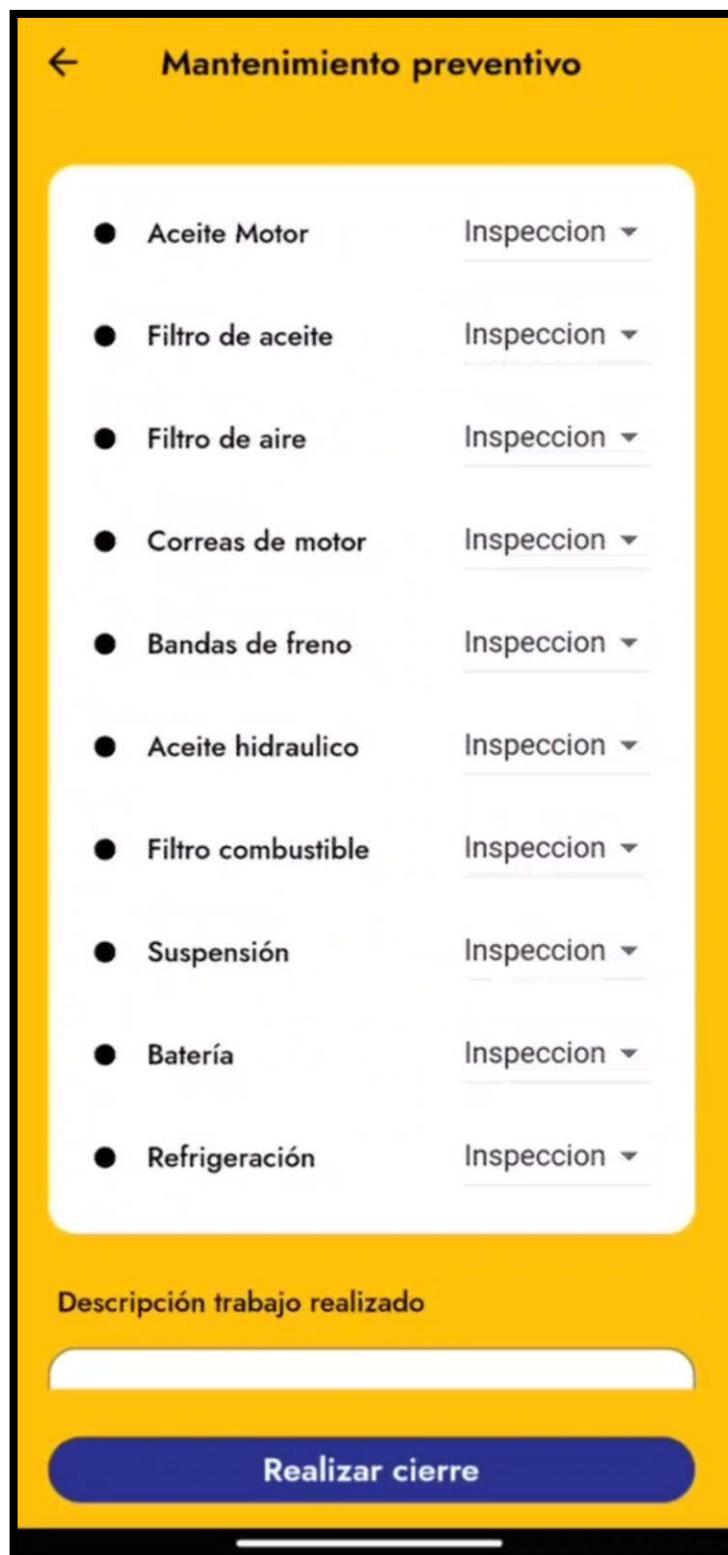


Imagen 10.



Imagen 11.

Imagen 9, 10 y 11. Mantenimiento Preventivo dirigido a Vehículo automotriz.
Fuente: Elaboración propia

En el caso de que el operario haga el cierre del formato de mantenimiento preventivo (MP) dirigido a vehículos, deberá elegir el tipo de actividad que realizara al vehículo automotriz, entre estas actividades tenemos: INSPECCION, CAMBIO, NO APLICA.

Luego hacer una breve descripción de lo que se le hizo al vehículo, claro está con sus respectivos soportes fotográficos. Lo anterior se puede observar en las siguientes imágenes.

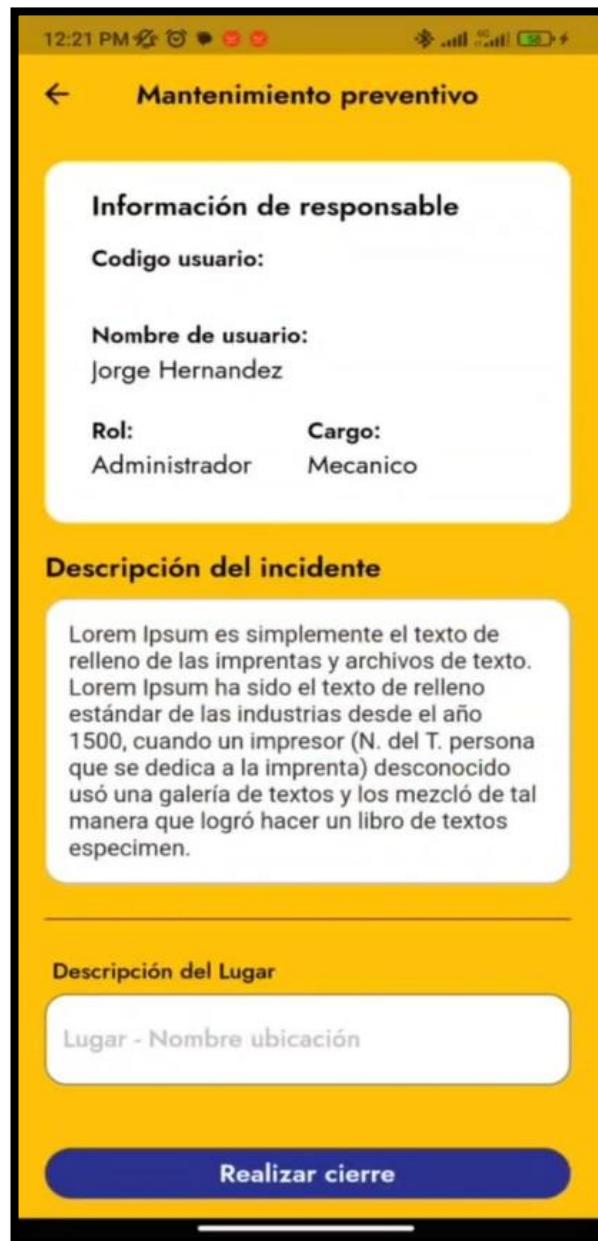


Imagen 12.

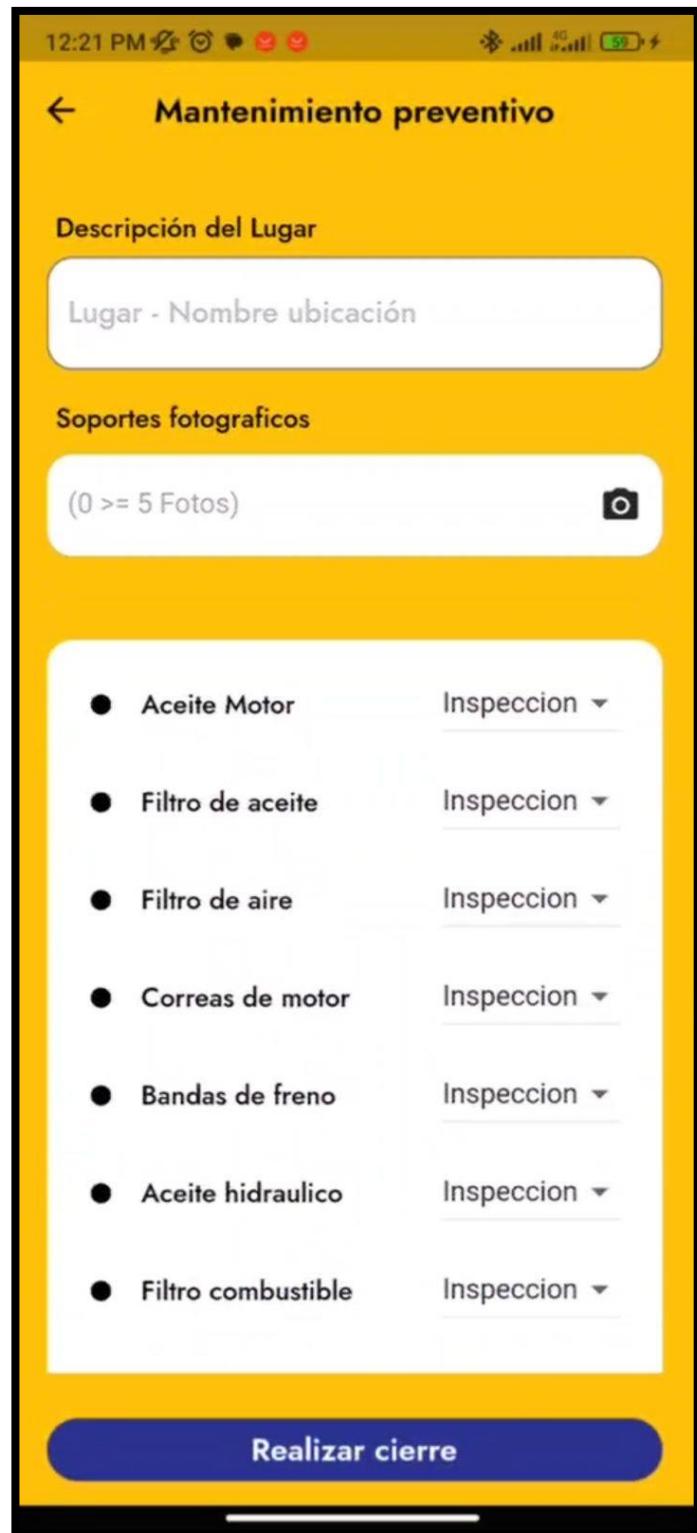


Imagen 13.



Imagen 14.

Imagen 12, 13 y 14. Mantenimiento Preventivo dirigido a Plantas Eléctricas.
Fuente: Elaboración propia

En el caso de que el operario haga el cierre del formato de mantenimiento preventivo (MP) dirigido a Plantas Eléctricas, deberá digitar en el recuadro “LUGAR, donde se encuentra la Planta Eléctrica”. Luego elegir el tipo de actividad que realizara a la Planta, entre estas actividades tenemos: INSPECCION, CAMBIO, NO APLICA. Después, hacer una breve descripción de lo que se le hizo al vehículo, claro está con sus respectivos soportes fotográficos.

Al insertar los soportes fotográficos el operario deberá presionar al icono de la cámara para que el celular entre al módulo de toma de fotografías y pueda tener una vista del listado de las fotos que se va tomando, como se observa en las siguientes imágenes.

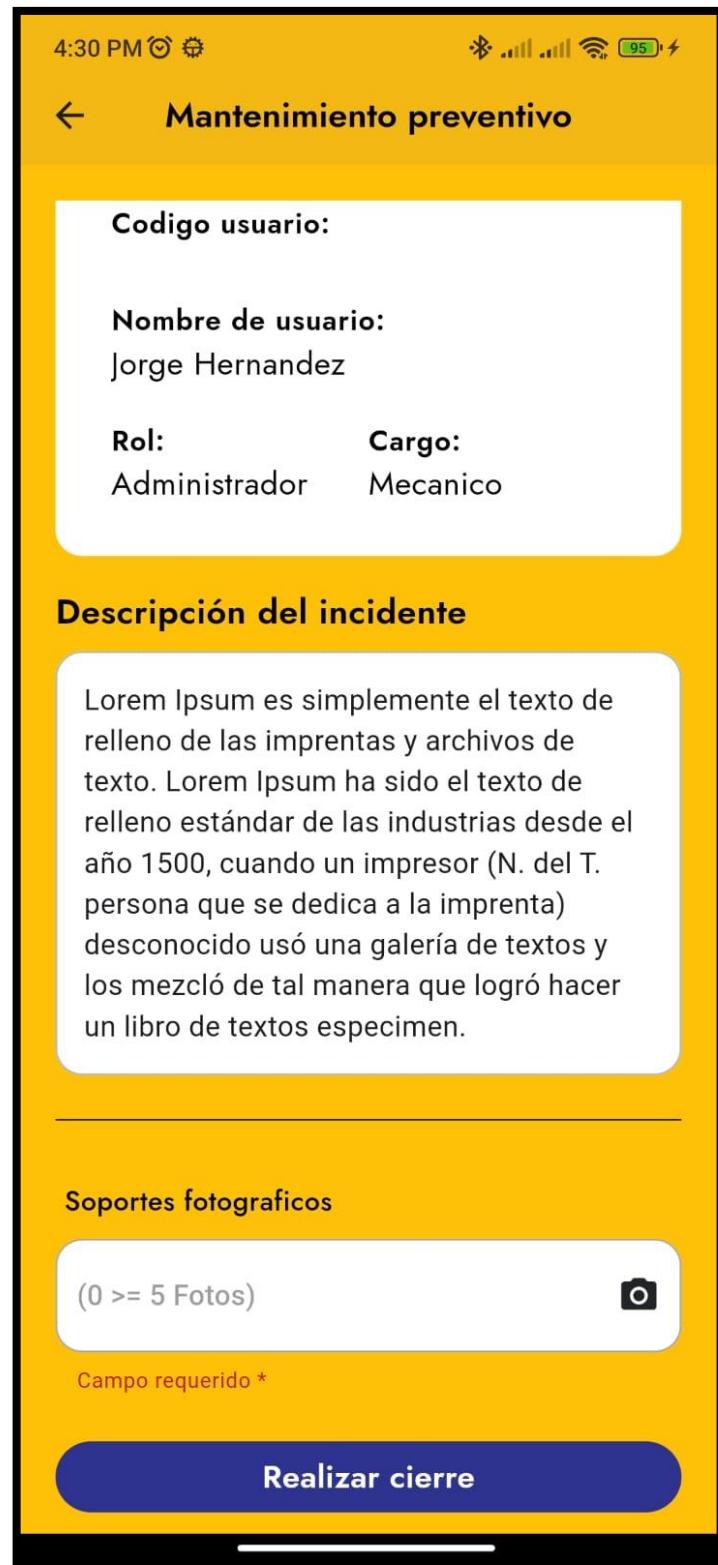


Imagen 15.

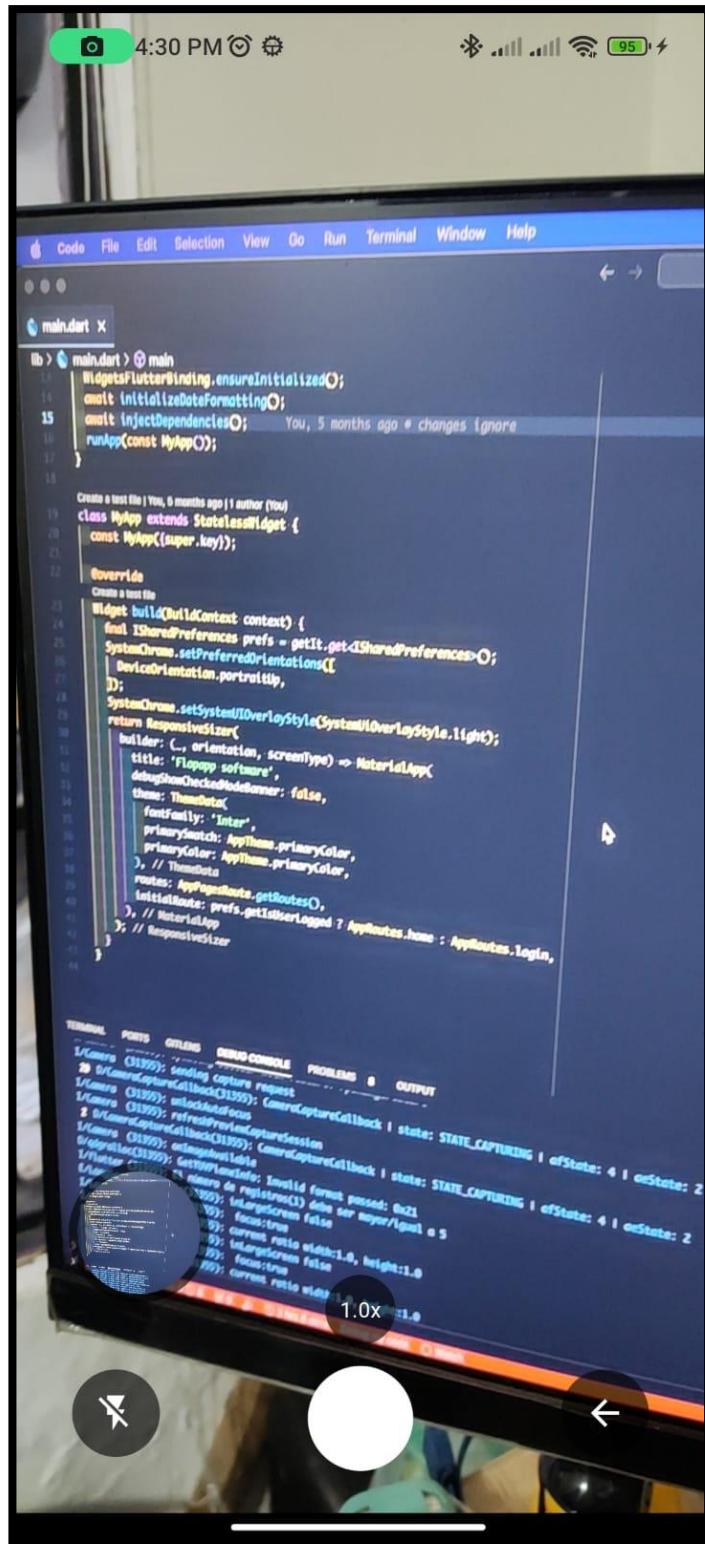


Imagen 16.

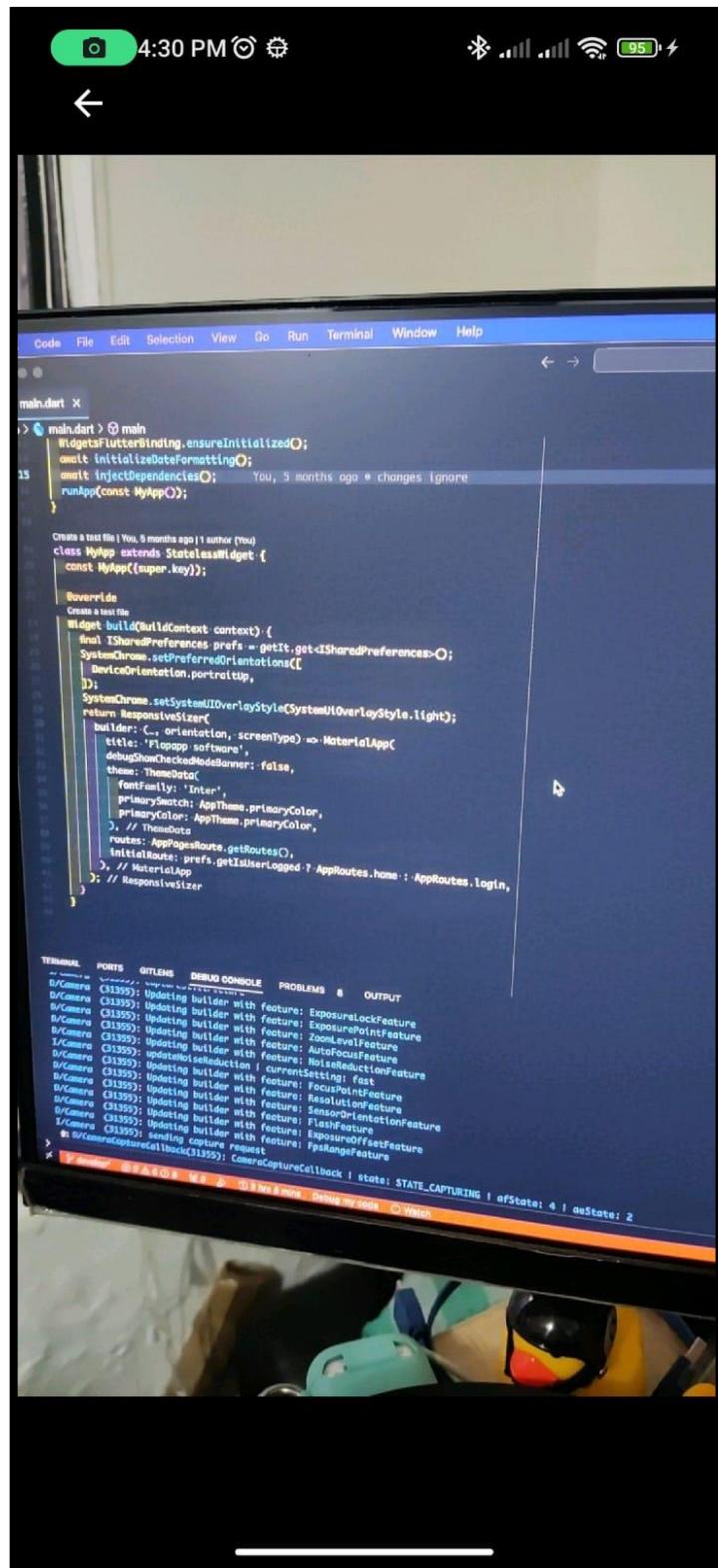


Imagen 17.



Imagen 18.

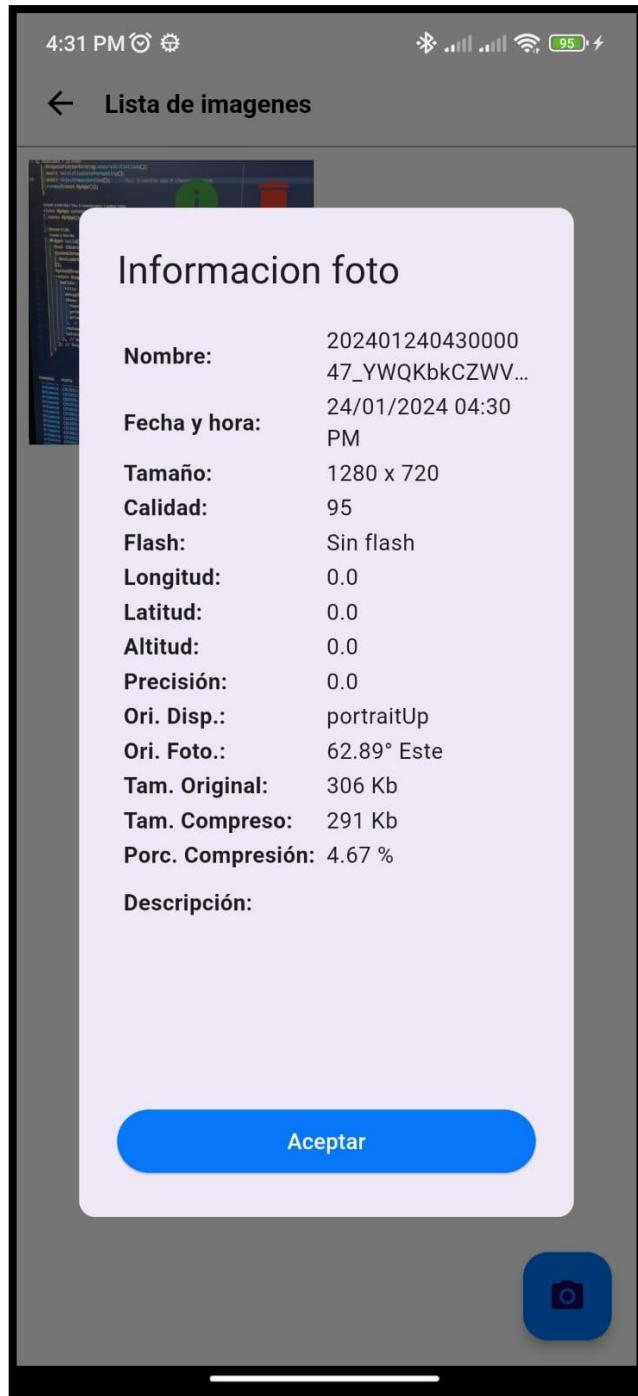


Imagen 19.

Imagen 15, 16, 17, 18 y 19. Módulo de toma de fotografías.
Fuente: Elaboración propia.

Si no se completa con los requisitos que se exigen no se puede realizar el cierre del MP (Mantenimiento Preventivo).

PASO 5 CIERRE DE ORDEN DE MC (MANTENIMIENTO CORRECTIVO):

En el caso de los cierres de orden de MC, su interfaz es más sencillo que los MP, ya que, son repuestos que se cambian por daño o vejez, un ejemplo de MC en los vehículos automotriz es la reparación de motor, en su interfaz solo se tendrá en cuenta una descripción breve de lo que se le hizo al vehículo con sus respectivos soportes fotográficos

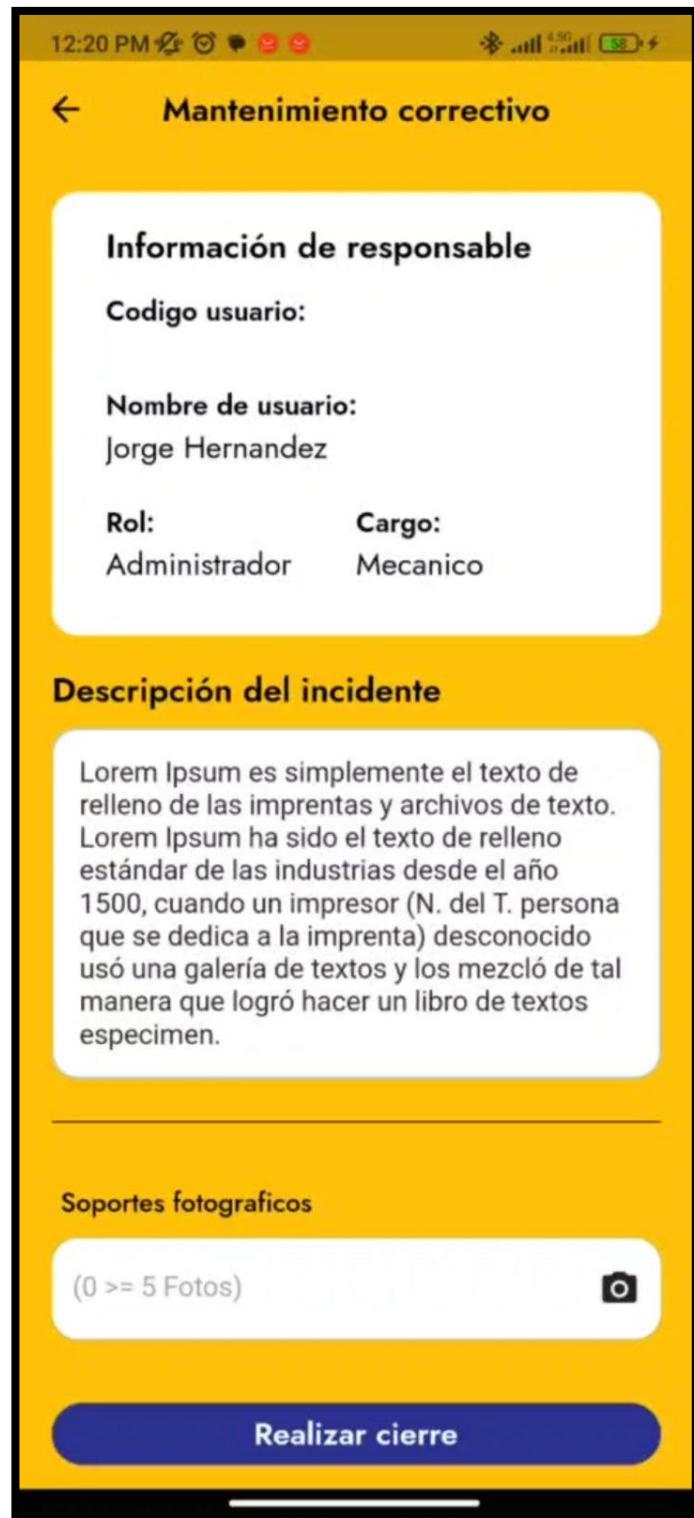


Imagen 18.



Imagen 19.

Imagen 18 y 19. Mantenimiento Correctivo dirigido a Vehículo automotriz.
Fuente: Elaboración propia

El MC de las Plantas Eléctricas es algo similar a las MC para vehículos automotriz, a excepción, que tiene una casilla extra, en el cual, se le agrega el lugar donde está ubicada la planta eléctrica, como se observa en la siguiente imagen.

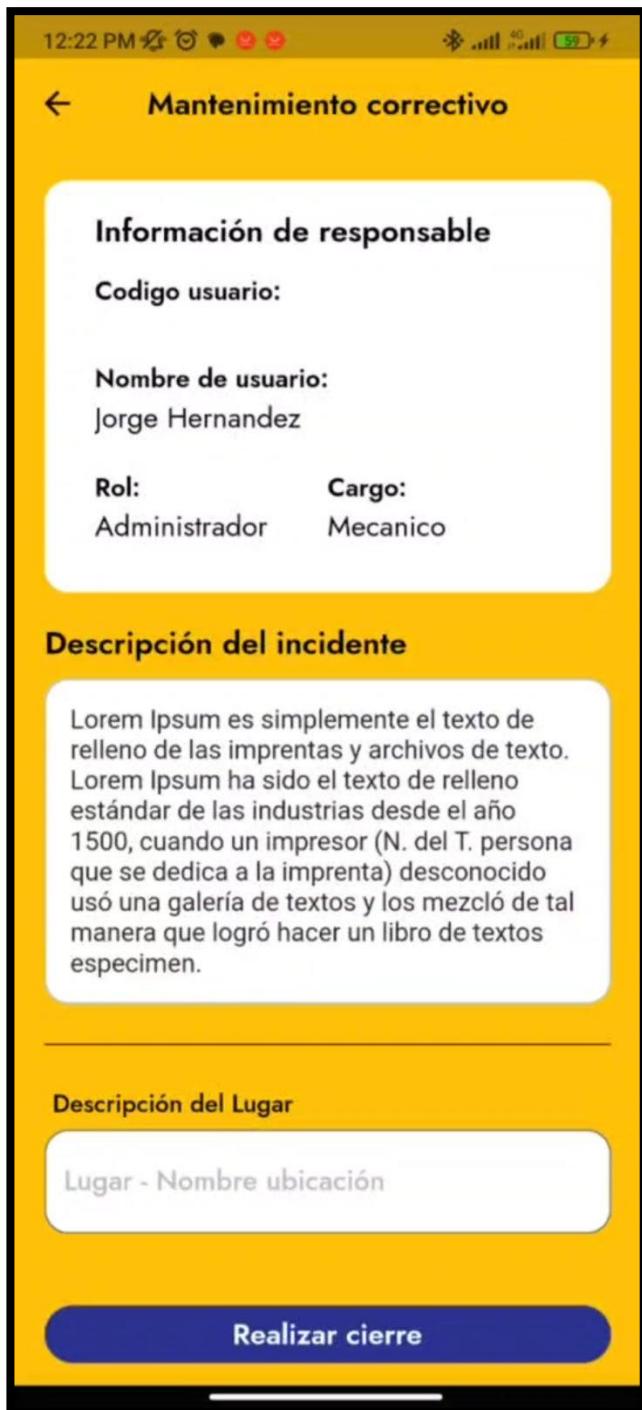


Imagen 20.



Imagen 21.

Imagen 20 y 21. Mantenimiento Correctivo dirigido a Plantas Eléctricas.
Fuente: Elaboración propia.

3.3.2.2 DISEÑO DEL BACKEND

En el Backend se debe tener en cuenta varios aspectos, como es su uso y el tipo de arquitectura. El término "Backend" se refiere a la parte de un sistema informático o de una aplicación que no es visible directamente para los usuarios finales. Es la capa que maneja la lógica de la aplicación, procesa los datos y se encarga de las operaciones del servidor. En contraste, el "Frontend" es la parte de la aplicación con la que los usuarios interactúan directamente, es decir, la interfaz de usuario y la presentación visual.

En el contexto del desarrollo web y de aplicaciones, el Backend generalmente se ejecuta en el servidor y realiza tareas como:

1. Procesamiento de Datos:

- El Backend maneja la lógica de negocio y realiza operaciones de procesamiento de datos. Esto puede incluir cálculos, manipulación de bases de datos y otras operaciones complejas.

2. Almacenamiento de datos:

- Gestiona la persistencia y recuperación de datos. Interactúa con bases de datos para almacenar y recuperar información necesaria para la aplicación.

3. Seguridad:

- Implementa medidas de seguridad, como autenticación y autorización, para proteger los datos y garantizar que solo usuarios autorizados puedan acceder a ciertas partes de la aplicación.

4. Manejo de Usuarios y Sesiones:

- Administra la autenticación y el manejo de sesiones para usuarios. Esto implica la gestión de cuentas de usuario, contraseñas y la gestión de permisos.

5. API (Interfaz de Programación de Aplicaciones):

- Proporciona una interfaz para que el Frontend pueda comunicarse con el Backend. A través de esta interfaz, el Frontend puede solicitar y enviar datos al servidor.

6. Lógica del Servidor:

- Implementa la lógica de negocio de la aplicación. Realizar las operaciones necesarias para satisfacer las solicitudes del cliente y proporcionar una respuesta adecuada.

7. Escalabilidad:

- Se encarga de la escalabilidad del sistema para manejar un gran número de usuarios. Esto puede implicar la implementación de técnicas como la escalabilidad horizontal o vertical, dependiendo de las necesidades de la empresa.

8. Despliegue y Mantenimiento:

- Incluye el despliegue de la aplicación en entornos de producción y el mantenimiento continuo para garantizar el rendimiento y la disponibilidad.

En resumen, el Backend constituye la infraestructura y la lógica que permiten que una aplicación funcione, trabaja en conjunto con el Frontend para ofrecer una experiencia completa al usuario. Mientras que el Frontend se centra en la presentación visual y la interacción del usuario, el Backend gestiona los aspectos internos y el procesamiento de datos que hacen que la aplicación funcione.

3.3.2.2.1 ARQUITECTURA CLIENTE-SERVIDOR

La arquitectura cliente-servidor es un modelo de diseño en el que una aplicación se divide en dos partes principales: el cliente y el servidor. Cada parte desempeña un papel específico en la comunicación y el funcionamiento de la aplicación.

A continuación, se explican los aspectos claves:

1. Cliente:

- El cliente es la interfaz de usuario o la aplicación que utiliza el usuario final. Puede ser un navegador web, una aplicación móvil, una aplicación de escritorio, u otro tipo de interfaz.
- El cliente solicita servicios o recursos al servidor y presenta la información al usuario.

2. Servidor:

- El servidor es una máquina, un software o un conjunto de ellos que almacena la lógica de la aplicación, procesa las solicitudes del cliente y gestiona los recursos, como bases de datos.
- Responde a las solicitudes del cliente proporcionando datos, realizando operaciones o ejecutando la lógica de negocio.

3. Comunicación:

- La comunicación entre el cliente y el servidor se realiza a través de solicitudes y respuestas. El cliente envía solicitudes al servidor, y el servidor responde con los datos solicitados o el resultado de una operación.
- Los protocolos comunes para esta comunicación incluyen HTTP/HTTPS en el contexto de aplicaciones web, TCP/IP en aplicaciones más generales y en el caso de servidores en la nube utilizan microservicios.

4. Independencia:

- La arquitectura cliente-servidor permite la independencia entre el cliente y el servidor, esto significa que pueden evolucionar de manera independiente siempre y cuando se mantenga la compatibilidad con la interfaz de comunicación establecida.

5. Escalabilidad:

- Esta arquitectura es escalable. Puedes escalar el sistema para manejar más usuarios aumentando la capacidad del servidor o distribuyendo la carga entre varios servidores.

6. Seguridad:

- La seguridad puede ser gestionada en ambas partes, el servidor puede implementar medidas de seguridad para proteger los datos y garantizar la autenticación y autorización adecuada.

7. Ejemplo de Flujo:

- El cliente inicia una solicitud al servidor, como solicitar una página web o enviar datos de un formulario.
- El servidor procesa la solicitud, realiza las operaciones necesarias, accede a bases de datos si es necesario y envía la respuesta de vuelta al cliente.
- El cliente interpreta la respuesta y actualiza su interfaz de usuario según sea necesario.

Teniendo en cuenta la información anterior sobre el funcionamiento del Backend, se desarrollará el Backend de la aplicación móvil con ayuda del Framework Spring

Boot, en donde, su lenguaje de programación se trabaja en JAVA, esto con el objetivo de que el Backend sea de forma independiente.



Imagen 22. Logo del Framework Spring Boot

Spring Boot es una herramienta que acelera y simplifica el desarrollo de microservicios y aplicaciones web con Spring Framework gracias a tres funciones principales:

1. Configuración automática
2. Un enfoque de configuración obstinado
3. La capacidad de crear aplicaciones autónomas

Estas características, combinadas, conforman una herramienta que le permite configurar una aplicación basada en Spring con el mínimo de instalación y configuración¹¹, además, de que . Se trata de un tipo de desarrollo que se centra en entregar componentes o piezas pequeñas que sean más administrables por parte de los clientes.¹²

En esencia, se trata de un enfoque que piensa en el diseño de las aplicaciones Java de manera modular, creando pequeñas piezas que funcionan de manera independiente pero que se pueden integrar para funcionar de forma conjunta.

¹¹ IBM. ¿Qué es Java Spring Boot? Párrafo 2.

¹² Tokio New Tecnology School. ¿Para qué sirve Spring Boot? Párrafo 16.

3.3.2.2.2 COMUNICACIÓN ENTRE FRONTEND Y BACKEND

La comunicación entre el Frontend y el Backend en una arquitectura típica cliente-servidor web se realiza a través de solicitudes y respuestas a través de la red.

A continuación, se explica el proceso básico:

1. **Solicitud del Cliente (Frontend):** El usuario interactúa con la interfaz de usuario del Frontend, ya sea a través de un navegador web, una aplicación móvil o cualquier otra interfaz de usuario. Cuando el usuario realiza una acción que requiere datos o servicios del Backend (por ejemplo, hacer clic en un botón para cargar información), el Frontend genera una solicitud.
2. **Envío de la Solicitud al Servidor (Backend):** La solicitud del cliente se envía al servidor a través de la red. Comúnmente, esto se realiza utilizando el protocolo HTTP o su versión segura, HTTPS o en microservicios.
3. **Procesamiento en el Servidor (Backend):** El servidor recibe la solicitud y procesa la acción solicitada. Esto puede implicar operaciones como acceder a una base de datos, realizar cálculos o ejecutar lógica de negocio.
4. **Generación de Respuesta (Backend):** Despues de procesar la solicitud, el servidor genera una respuesta que contiene los datos necesarios o la confirmación de la acción realizada.
5. **Envío de la Respuesta al Cliente (Frontend):** La comunicación que va a tener el Backend con el Frontend para la aplicación móvil “FLOPA” es usando los microservicios RESTFUL, Para entender cómo funcionan los microservicios, volvamos al pasado.

El desarrollo de software tradicional, que aún continúa en muchas organizaciones, utiliza una arquitectura monolítica. Un «monolito» se refiere a una única y gran aplicación que contiene toda su funcionalidad y características y que lo almacena todo en un solo lugar.

Esto significa que todos los componentes de una aplicación, incluida la lógica empresarial, el acceso a los datos y la interfaz de usuario, se almacenan en el mismo lugar. Este desarrollo de software es, de hecho, fácil y resulta natural, por eso muchos siguen optando por él; sin embargo, se vuelve complicado si quieres añadir más funcionalidad a tu aplicación para hacerla atractiva o aumentar su finalidad, usabilidad, seguridad, etc.

Añadir más funcionalidad a la base de código existente puede aumentar la complejidad y el tamaño del monolito, lo que invita a diversos problemas, como:

- El cambio puede afectar a la aplicación en general, incluso si quieres hacer un pequeño cambio, es posible que tengas que volver a desplegar toda la aplicación, lo cual es arriesgado y consume tiempo y recursos.
- Debido a su estructura fuertemente acoplada, los monolitos no son flexibles, por lo tanto, también restringe la pila tecnológica, especialmente cuando la aplicación escala, puede resultar difícil cambiar tu pila tecnológica y pueden verse obligados a utilizar las viejas tecnologías con tantos problemas subyacentes.
- Es arriesgado porque si se deja sin cubrir alguna vulnerabilidad y se compromete esa parte, el ataque puede extenderse por toda la aplicación, comprometiendo toda la aplicación y sus datos.

Por lo tanto, dividir las funciones de una aplicación en diferentes partes parece un enfoque excelente para abordar todas estas cuestiones, que es exactamente lo que hacen los microservicios. Entendamos cómo se pone en marcha la arquitectura de microservicios. En una arquitectura de microservicios, las aplicaciones se estructuran en servicios reutilizables y discretos que se comunican a través de una API. Cada servicio se organiza en torno a un proceso de negocio concreto y se adhiere a un protocolo de comunicación como el HTTP. Estos servicios más pequeños se integran por separado con sus dependencias y otros datos en la aplicación. Así, si quieres hacer algún cambio en una funcionalidad, puedes hacerlo sin afectar a las demás partes de la aplicación con facilidad.

3.3.2.2.3 DESARROLLO DEL BACKEND

En el diseño del Backend se rige por la estructura de datos que van ligado a las interfaces del Frontend, en donde, son almacenados en el servidor, mostrada en la siguiente figura:

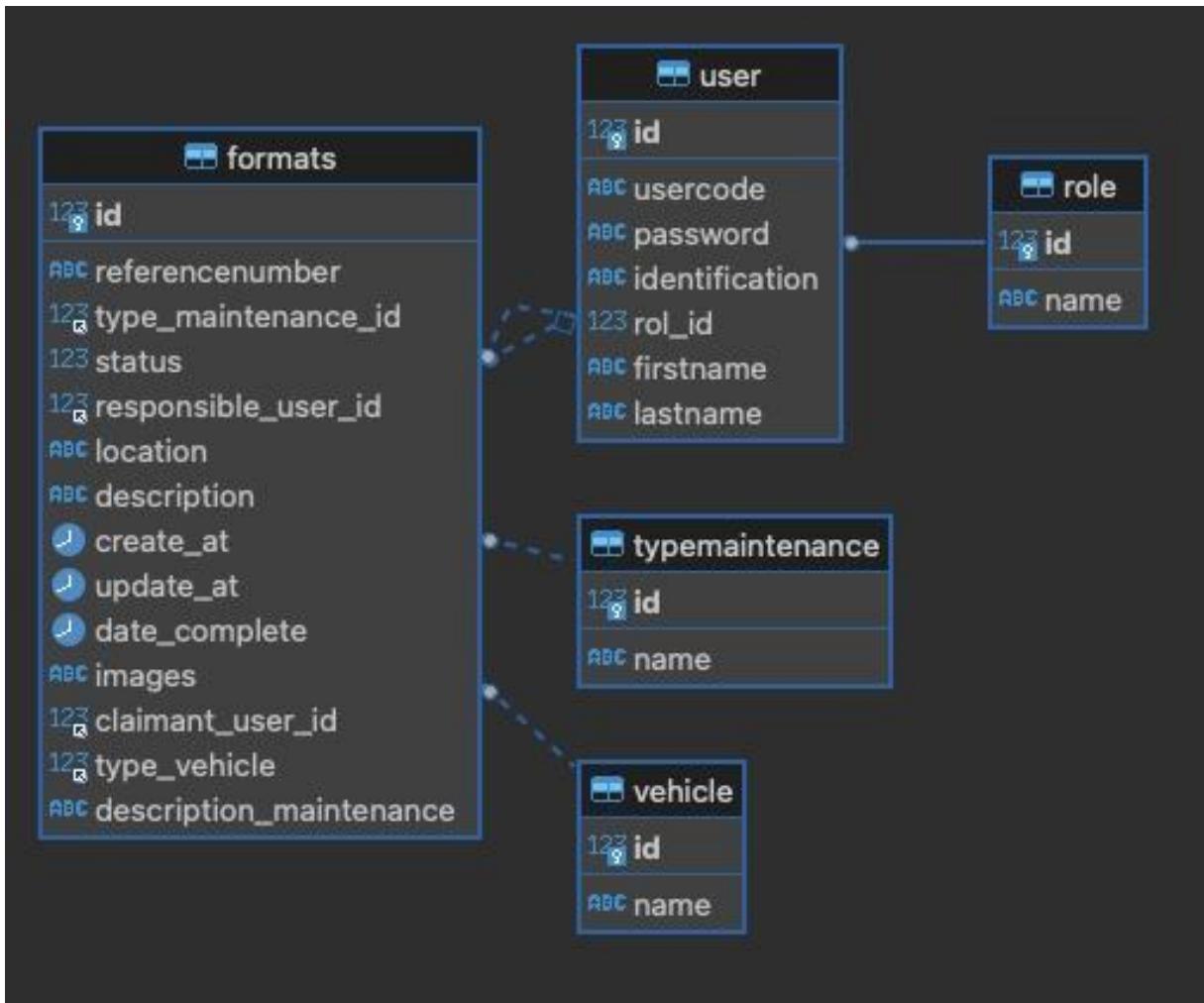


Figura 4. Estructura de Datos.
Fuente: Elaboración propia.

CAPITULO 4: PRUEBAS DE FUNCIONAMIENTO

4.1 VALIDANDO EL LOGIN

Al ingresar el código de usuario deberá cumplir ciertos requisitos tales como:

- Tener una letra Mayúscula.
- 11 números.

Al introducir la contraseña (Password) de debe cumplir:

- Tener letras
- Tener números.
- Tener letras especiales (@"#\$%&/).

La contraseña será proporcionada por la empresa a los operarios (clientes o usuarios), en la siguiente imagen encontramos un ejemplo de cómo la aplicación toma en cuenta lo que mencionamos anteriormente.

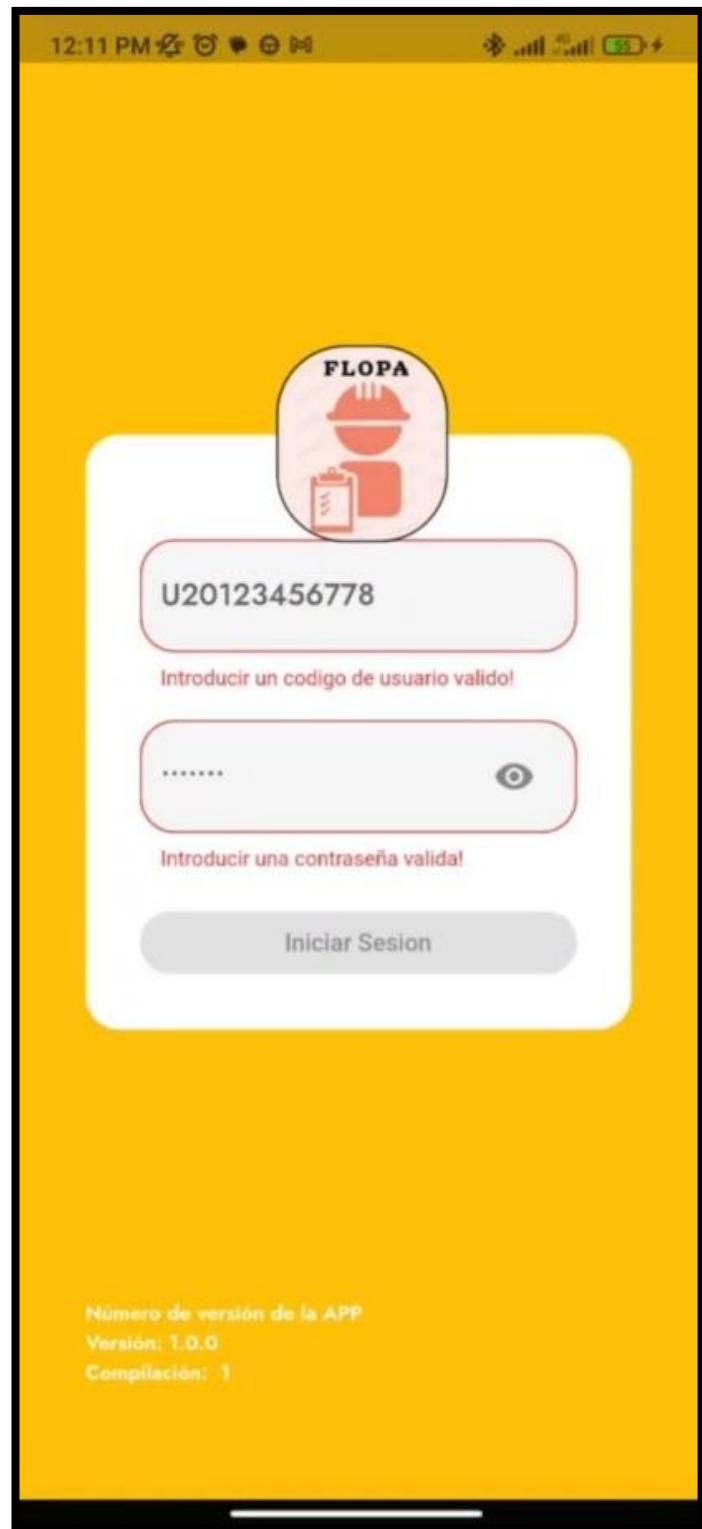


Imagen 23. Interfaz de Login realizando validación de usuario.
Fuente: Elaboración propia.

Si al ingresar el usuario con su respectiva contraseña y al comparar con el Backend no coinciden, no se le permitirá el ingreso, caso contrario obtenemos lo siguiente:



Imagen 24.



Imagen 25.



Imagen 26.



Imagen 27.



Imagen 28.



Imagen 29.

Imagen 24, 25, 26, 27, 28 Y 29. Descargas de recursos que necesita la App para funcionar.

Fuente: Elaboración propia.

Las imágenes anteriores se observa todas las descargas que realiza la aplicación para su óptimo funcionamiento guardados en el Backend, tales como:

- Descargas de recursos: Son los recursos que necesita la aplicación para funcionar óptimamente.
- Descargas de usuarios activos (Backend): Son todos los usuarios que se encuentran en el Backend.
- Descargas de lista de formatos pendientes (Panel Frontal): Son todos los formatos u órdenes de mantenimiento que se han realizado y se muestran en el panel frontal.

4.2 VALIDANDO DEL PANEL FRONTAL

Después de que se hallan realizado las descargas el usuario se encontrara con el panel frontal.



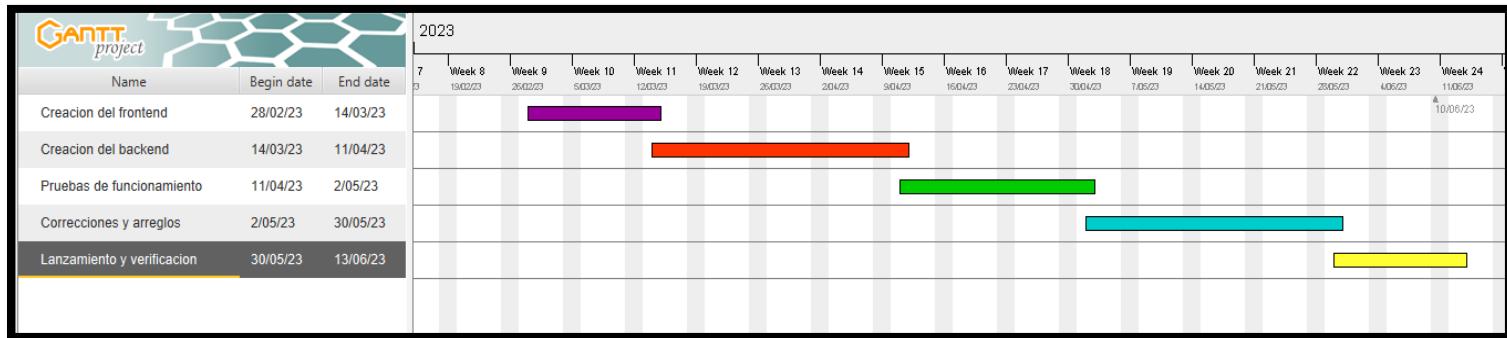
Imagen 30. Panel frontal para el operario



Imagen 31, Panel frontal para el administrador,

CRONOGRAMA

A continuación, se observa el cronograma de las actividades a realizar con su respectiva fecha, este se implementa por medio de Gantt Project:



FINANCIACIÓN

Se presentan cifras tentativas de los insumos necesarios para la creación y desarrollo de la aplicación.

Equipos, insumos y materiales		
Descripción	Cant.	Valor
Computador	1	3'500.000
Celular	2	2'500.000
Servidor	1	500.000 mensual
Software	1	1'000.000

CONCLUSIONES

- La aplicación móvil al asociarse con algún servidor (local o nube), el teléfono tiene que estar por obligación conectado a un servicio de ethernet, para que el usuario pueda ingresar y descargar todos los recursos localizados en el Backend, dado que, si no se encuentra conectado al servicio de ethernet, el usuario no puede ingresar y menos interactuar con las interfaces de la aplicación.
- La empresa Maxim Fishing SAS al apoyarse en las nuevas tecnologías y los desarrollos de software que se están actualizando hoy en día, pueden desarrollar sus actividades de mantenimiento de una forma más eficaz y eficientemente maximizando así los recursos de la empresa.
- Analizar la comunicación Frontend y Backend, ya que dependiendo de cómo van a crear la aplicación móvil según su necesidad, así mismo, es la arquitectura que deben realizar para que el Backend se pueda comunicar con el Frontend sin ningún inconveniente.
- Realizar la validación de cada interfaz del Frontend que se vaya realizando, porque eso conlleva a que se está haciendo una aplicación móvil sin errores y en el caso que haga un Login, la aplicación pueda usarse de manera segura

BIBLIOGRAFÍA

- Servnet. Definición ¿Qué es Frontend? Párrafo 1 y 2. Ciudad de México marzo 2021.
- HubSpot. ¿Definición Qué es Backend? Párrafo 1
- ARIMETRICS. Definición ¿Qué es Framework? Párrafo 1. <https://www.arimetrics.com/glosario-digital/framework>. Madrid
- Elena Canorea. ¿Qué es Kotlin y para qué sirve? España, marzo de 2022. Página Web: <https://www.plainconcepts.com/es/kotlin-android/>. Párrafo 2.
- Luis Herazo. Anincubator. 2022. Página Web: <https://anincubator.com/que-es-una-aplicacion-movil/#:~:text=Una%20aplicaci%C3%B3n%20m%C3%B3vil%2C%20tambi%C3%A9n%20llamada,tel%C3%A9fono%20inteligente%20o%20una%20tableta>.
- Berna Network. ¿Qué son las aplicaciones híbridas? Párrafo 8. Página Web: <https://www.bernanetwork.com/diferencias-entre-app-hibrida-y-nativa>
- Properly. Qué es el mantenimiento preventivo. Párrafo 3 y 4. Página Web: <https://www.properly.es/diferencias-entre-mantenimiento-preventivo-y-correctivo/>
- IBM. ¿Qué es Java Spring Boot? Párrafo 2. Página Web: [https://www.ibm.com/es-es/topics/java-spring-boot/#:~:text=el%20siguiente%20paso,%C2%BFQu%C3%A9%20es%20Java%20Spring%20Boot%3F,m%C3%A1quina%20virtual%20Java%20\(JVM\).](https://www.ibm.com/es-es/topics/java-spring-boot/#:~:text=el%20siguiente%20paso,%C2%BFQu%C3%A9%20es%20Java%20Spring%20Boot%3F,m%C3%A1quina%20virtual%20Java%20(JVM).)

- Tokio New Tecnology School. ¿Para qué sirve Spring Boot? Párrafo 16. Página Web: <https://www.tokioschool.com/noticias/spring-boot/#:~:text=Spring%20Boot%20es%20un%20framework,de%20c%C3%B3digo%20abierto%20y%20gratuito>.
- KINSTA. Microservicios API. Página Web: <https://kinsta.com/es/blog/microservicios-vs-api/#:~:text=Microservicios%3A%20Es%20una%20colecci%C3%B3n%20de,microservicios%20en%20una%20sola%20aplicaci%C3%B3n>.