



UNIVERSIDAD AUTÓNOMA DE CHILE

FACULTAD DE INGENIERÍA

INGENIERÍA CIVIL INFORMÁTICA

DESARROLLO DE APLICACIONES MÓVILES

SEDE ALAMEDA

EVR01 – Informe de Proyecto
Aplicación Móvil “MobiMant”

Autores:

Felipe Arevalo Galaz

Nicolás Fluxá Morán

Abril 2025



Índice

1. Introducción	2
2. Etapa 1 – Diagnóstico y Contexto del Problema	3
2.1. Descripción del Problema	3
2.2. Ejemplos de Recopilación de Información	3
3. Etapa 2 – Propuesta del Proyecto	4
3.1. Planteamiento de la Propuesta	4
3.2. Planteamiento de la Propuesta	5
3.3. Objetivos del Proyecto	6
3.3.1. Objetivo General	6
3.3.2. Objetivos Específicos	6
3.4. Público Objetivo	6
4. Etapa 3 – Diseño de la Interfaz y Desarrollo Inicial	7
4.1. Diseño de Mockups	7
4.2. Primeras Pantallas en Android Studio	14
4.3. Enlace al Repositorio GitHub	14
5. Conclusiones	15
Referencias	16

Índice de figuras

1. Mockup de la pantalla de Registro.	7
2. Mockup de la pantalla de Inicio de Sesión.	8
3. Mockup de la pantalla de Registro de Máquinas.	9
4. Mockup de la pantalla de Gestión de Máquinas.	10
5. Mockup de la pantalla de Consulta de Estado.	11
6. Mockup de la pantalla de Notificaciones.	12
7. Mockup de la pantalla de Configuración.	13



1. Introducción

Este informe presenta el desarrollo del proyecto **MobiMant**, una aplicación móvil para Android diseñada para digitalizar y optimizar el proceso de mantenimiento en pequeñas y medianas empresas del sector industrial. Durante el desarrollo se abordan desde la identificación de la problemática hasta la propuesta de solución y el diseño de la interfaz, integrando tanto los aspectos técnicos como experiencias personales del equipo.



2. Etapa 1 – Diagnóstico y Contexto del Problema

2.1. Descripción del Problema

En muchas empresas medianas y pequeñas del sector manufacturero, los sistemas de control de mantenimiento se gestionan de forma manual o con métodos disociados, lo que reduce la productividad y genera duplicidad de registros.

MobiMant busca ofrecer una trazabilidad clara y confiable, permitiendo a los operadores conocer:

- Las tareas de mantenimiento ya realizadas.
- Las actividades pendientes.
- Las mejoras necesarias en cada máquina.

Mediante interfaces rápidas e intuitivas, los usuarios podrán acceder a la información crítica sin demoras, y las notificaciones automáticas optimizarán el flujo de trabajo al alertar automáticamente sobre eventos críticos, reduciendo el tiempo de respuesta.

2.2. Ejemplos de Recopilación de Información

Además de los métodos tradicionales como:

1. Gestión de Mantenimiento mediante Planillas Excel

En muchas pymes del sector metalúrgico, el uso de Excel para llevar el seguimiento de mantenimiento es común por su bajo costo y facilidad de acceso. (Mobility Work, 2025) No obstante, presenta desafíos importantes:

a) Registro manual:

Cada incidencia debe ser descrita a mano por el operario, lo que introduce errores tipográficos y depende de la atención al detalle de cada usuario.

b) Falta de automatización:

No existen validaciones automáticas ni alertas cuando un campo queda vacío o duplicado, lo que incrementa el riesgo de inconsistencias y datos obsoletos.

c) Limitación de integración:

El archivo suele almacenarse localmente en una única estación de trabajo, impidiendo el acceso simultáneo y la sincronización en tiempo real entre distintos usuarios.

A pesar de estas limitaciones, Excel ofrece algunos beneficios:

a) Bajo costo:

Es una herramienta que la mayoría de las empresas ya tiene instalada, evitando inversiones adicionales en software especializado.

b) Familiaridad:

La mayoría de los operarios conoce su uso básico, lo que reduce el tiempo de capacitación y la resistencia al cambio.



2. Gestión de Mantenimiento a través de Llamadas Telefónicas

- a) **Comunicación verbal:** La incidencia se transmite de manera inmediata mediante una comunicación verbal. Sin embargo, la falta de registro escrito puede resultar en la pérdida de detalles esenciales y en malentendidos, dado que no queda constancia documental de la conversación (Nextiva, 2024).
- b) **Ausencia de registro formal:** Al no contar con un respaldo documental, es difícil realizar seguimientos y auditorías precisas de las intervenciones de mantenimiento, lo cual limita el análisis de incidentes pasados.
- c) **Dependencia de la memoria:** La efectividad de este método depende de la capacidad de retención de la información por parte del operador y del receptor, aumentando el riesgo de omitir datos críticos que pueden afectar el diagnóstico y la resolución de problemas.

3. Registro en Papel de Órdenes de Mantenimiento

- a) **Archivado manual:** Los registros en papel deben organizarse y almacenarse físicamente, lo que genera una carga administrativa elevada. Este proceso conlleva a retrasos en el acceso a la información y aumenta el riesgo de pérdida o extravío de documentos (Sistemas OEE, 2021).
 - b) **Procesos lentos:** La búsqueda, actualización y consolidación de los registros en papel es tediosa y laboriosa, lo que ralentiza la toma de decisiones y dificulta la eficiencia operativa.
 - c) **Propensión a errores:** La introducción manual de datos es susceptible a errores tipográficos, duplicación u omisión de información crítica, comprometiendo la precisión y confiabilidad de los registros.
- a) **Ventajas:**
- 1) **Simplicidad:** Este método es tradicional y de fácil implementación, ya que la mayoría de las empresas cuentan con el papel y los formularios necesarios, sin requerir inversión en tecnología compleja.
 - 2) **Independencia tecnológica:** No depende de dispositivos electrónicos ni de conexión a internet, lo que lo hace viable en entornos con recursos limitados o en situaciones de cortes de energía.

3. Etapa 2 – Propuesta del Proyecto

3.1. Planteamiento de la Propuesta

Para resolver las deficiencias detectadas en los métodos tradicionales, se propone **Mobi-Mant**: una aplicación móvil nativa para Android que centraliza y automatiza el ciclo de vida de las órdenes de mantenimiento.



3.2. Planteamiento de la Propuesta

Para resolver las deficiencias detectadas en los métodos tradicionales, se propone **Mobi-Mant**: una aplicación móvil nativa para Android que centraliza y automatiza el ciclo de vida de las órdenes de mantenimiento.

- **Integración con Firebase:** La app se conectará a Firebase para gestionar la autenticación, los roles y los datos en la nube, asegurando disponibilidad y escalabilidad. Para ello, se utilizarán:
 - **Firestore Authentication** (Authentication, 2023) para el registro e inicio de sesión mediante email/contraseña, permitiendo diferenciar roles (administrador, operador y mecánico).
 - **Cloud Firestore** (Cloud, 2023) como base de datos principal, en la que se almacenará la información estructurada de empresas, usuarios, máquinas y órdenes de mantenimiento. Además, su modo offline permitirá que la aplicación funcione sin conexión y sincronice los cambios cuando se recupere la red.
 - **Firestore Cloud Messaging (FCM)** (Cloud Messaging, 2023) para el envío de notificaciones push, alertando en tiempo real a los operadores y mecánicos sobre nuevas órdenes o eventos críticos.
 - **Firestore Storage** (Storage, 2023) para almacenar de forma segura las imágenes asociadas a las órdenes, permitiendo adjuntar pruebas visuales como fotos de averías.
 - **Firestore Analytics** (Analytics, 2023) para recolectar y analizar eventos clave, entendiendo el comportamiento de los usuarios y facilitando futuras mejoras en la experiencia de la app.
 - **Firestore Security Rules** (Security Rules, 2023) para establecer políticas de acceso rigurosas en Cloud Firestore y Firestore Storage, restringiendo el acceso a los datos según el UID del usuario, la empresa y el rol asignado.
- **Espacio de trabajo personalizado:** Cada empresa tendrá su propio entorno dentro de la aplicación, donde los administradores podrán registrar y editar el catálogo de máquinas, definir umbrales críticos y asignar roles y permisos específicos. Esto asegura que la gestión de la información sea personalizada y aislada para cada compañía.
- **Interfaz dirigida a Operadores y Mecánicos:** Los operadores podrán:
 - Consultar el estado en tiempo real de las máquinas.
 - Generar órdenes de mantenimiento, adjuntando fotos y descripciones detalladas.
 - Recibir notificaciones push inmediatas.

Mientras que los mecánicos visualizarán únicamente las órdenes asignadas, registrarán avances y cerrarán intervenciones. Además, se implementará el escaneo de códigos QR para facilitar el acceso rápido a la ficha técnica de cada máquina.



3.3. Objetivos del Proyecto

3.3.1. Objetivo General

Digitalizar y optimizar el proceso de mantenimiento industrial en pymes mediante una aplicación Android que centralice el registro, seguimiento y análisis de órdenes de mantenimiento, logrando una reducción los tiempos de comunicación entre áreas además de mejorar los tiempos de mantenimientos proporcionando una comunicación efectiva, clara y rápida.

3.3.2. Objetivos Específicos

- Implementar un sistema de autenticación y control de acceso por roles (administrador, operador, mecánico) utilizando Firebase Authentication, finalizando esta funcionalidad antes del cierre del segundo mes de desarrollo.
- Desarrollar la gestión de máquinas y órdenes de mantenimiento en Cloud Firestore, incorporando funcionalidades para crear, editar y cerrar intervenciones—con soporte offline.
- Integrar notificaciones push mediante Firebase Cloud Messaging para alertas de mantenimiento, de modo que se logre que los tiempos de respuestas sean los menores posibles.
- Incluir un sistema de escaneo de códigos QR para otorgar acceso directo a la ficha técnica de cada máquina, reduciendo el tiempo de búsqueda.

3.4. Público Objetivo

Nuestro público objetivo para "MobiMant" se centra principalmente en pequeñas y medianas empresas del sector industria, especialmente aquellas en áreas de manufactura, procesamiento y embalaje, que buscan modernizar y optimizar sus procesos de mantenimiento. Dentro de estas empresas se identifican tres roles claves:

- **Administradores de Mantenimiento:** Responsables de planificar, supervisar y gestionar todas las operaciones de mantenimiento. Necesitan herramientas que les permitan tener una visión completa y en tiempo real del estado de las máquinas.
- **Operadores de Planta:** Técnicos que están en contacto directo con las máquinas. Su labor es reportar incidencias, generar órdenes de mantenimiento y realizar seguimiento del estado de los equipos.
- **Mecánicos de Servicio:** Encargados de ejecutar las intervenciones de mantenimiento. Necesitan un sistema que les asigne órdenes de forma clara, en el cual puedan registrar avances, documentar reparaciones y cerrar intervenciones de manera ordenada.

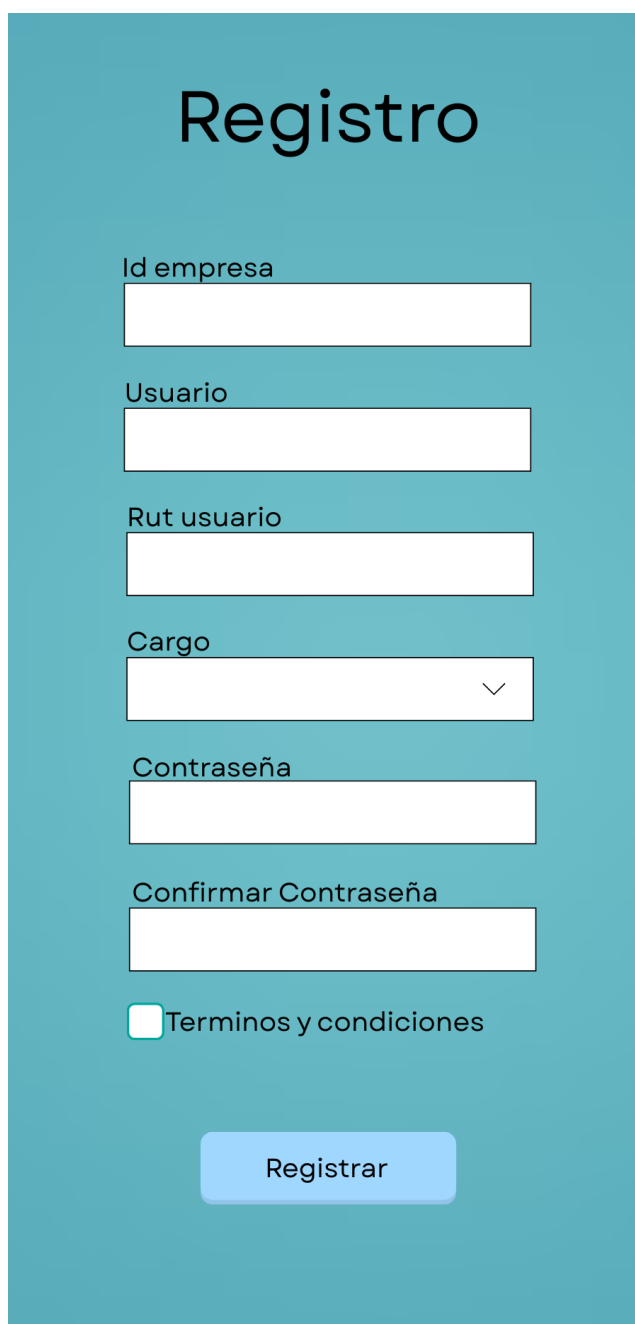


4. Etapa 3 – Diseño de la Interfaz y Desarrollo Inicial

4.1. Diseño de Mockups

A continuación se muestran los mockups que ilustran el flujo de la aplicación **MobiMant**. Se incluyen las siguientes pantallas:

1. **Pantalla de Registro:** Esta vista contiene campos para *Id Empresa*, *Usuario*, *Rut Usuario*, *Cargo*, *Contraseña*, *Confirmar Contraseña* y la opción de aceptar *Términos y Condiciones*, con un botón *Registrar*.



Registro

Id empresa

Usuario

Rut usuario

Cargo

Contraseña

Confirmar Contraseña

☐ Terminos y condiciones

Registrar

Figura 1: Mockup de la pantalla de Registro.



2. **Pantalla de Inicio de Sesión:** En esta vista se muestran los campos para *Id Empresa*, *Usuario* y *Contraseña*, junto con opciones para *Recuperar contraseña*, *Iniciar Sesión* y *Registrarse*.



MobiMant

Id Empresa

Usuario

Contraseña

[Recuperar contraseña](#)

[Iniciar Sesion](#)

[Registrarse](#)

Figura 2: Mockup de la pantalla de Inicio de Sesión.



3. **Pantalla de Registro de Máquinas:** Esta vista permite ingresar datos como *Tipo de Máquina*, *Marca*, *Modelo*, *Motorización* y *Matrícula o Identificador* para registrar una máquina.

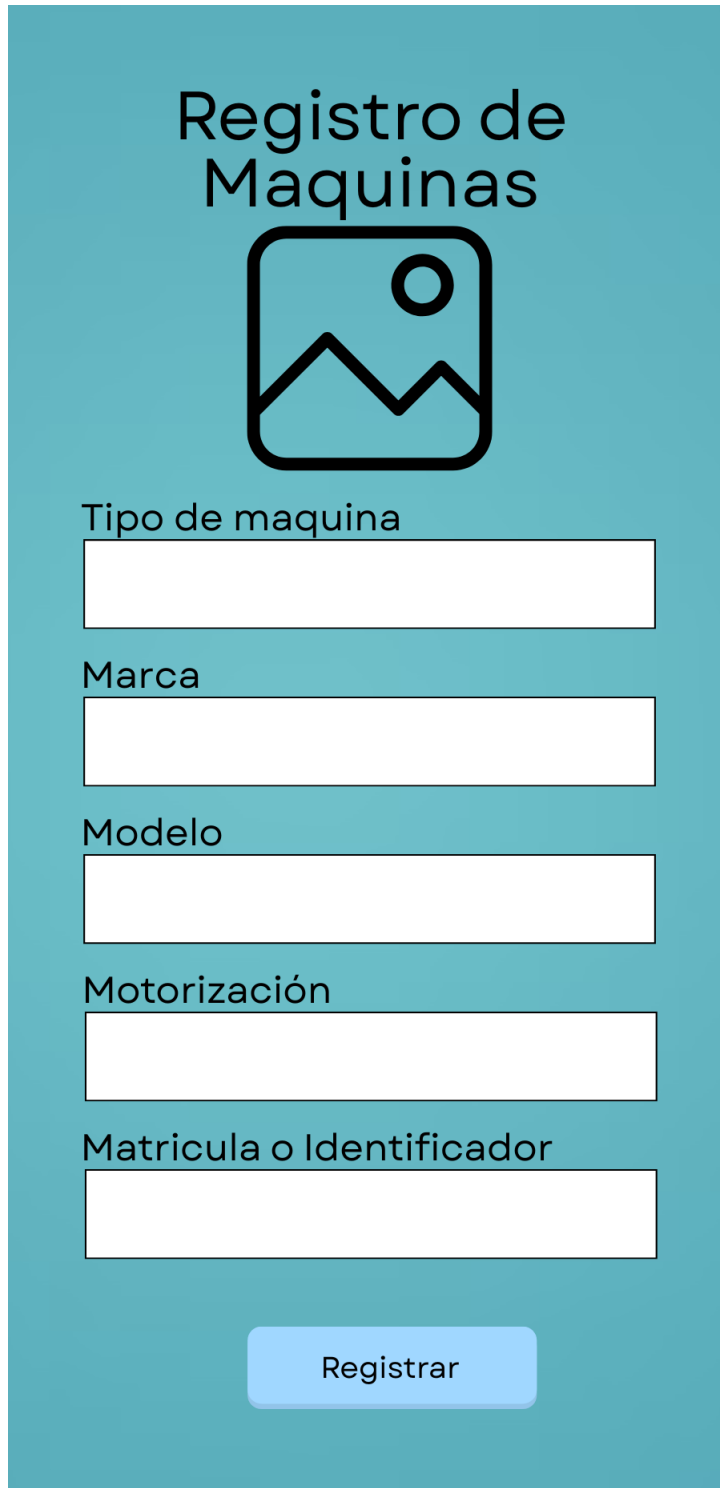



Figura 3: Mockup de la pantalla de Registro de Máquinas.



4. **Pantalla de Gestión de Máquinas:** Esta vista permite agregar y visualizar máquinas, mostrando campos para *Nombre*, *Descripción*, *Último mantenimiento* (en formato dd/mm/aaaa), *Reparaciones realizadas* y *Características de la máquina*.

Maquina:



Ultimo mantenimiento

dd/mm/aaaa

Reparaciones hechas

Características de la maquina

Figura 4: Mockup de la pantalla de Gestión de Máquinas.



5. **Pantalla de Consulta de Estado:** Aquí se permite consultar el estado en tiempo real de las máquinas.

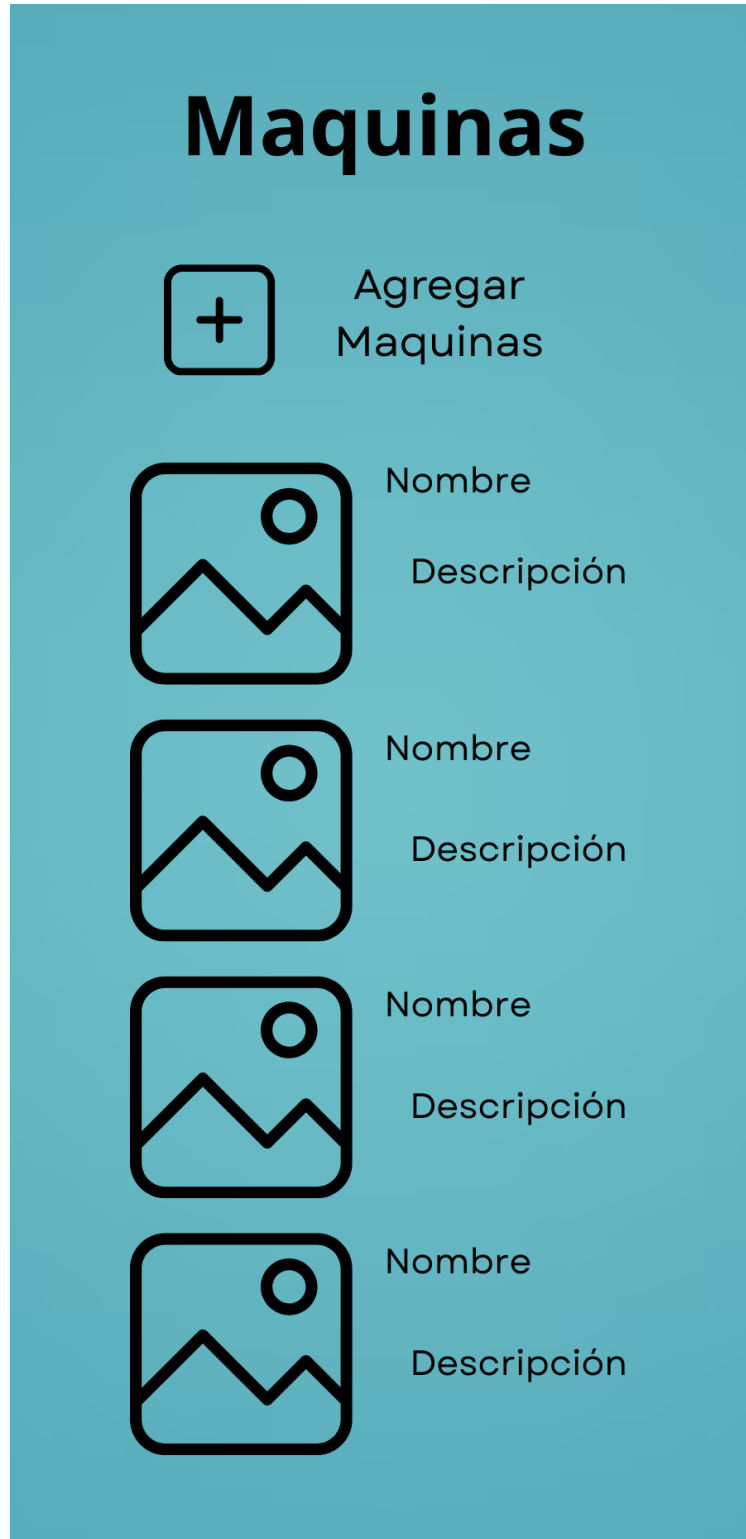


Figura 5: Mockup de la pantalla de Consulta de Estado.



6. **Pantalla de Notificaciones:** Vista que muestra alertas y notificaciones push ante incidencias o alertas críticas de mantenimiento.



Figura 6: Mockup de la pantalla de Notificaciones.



7. **Pantalla de Configuración:** Esta vista permite administrar perfiles, roles y demás ajustes de la aplicación.

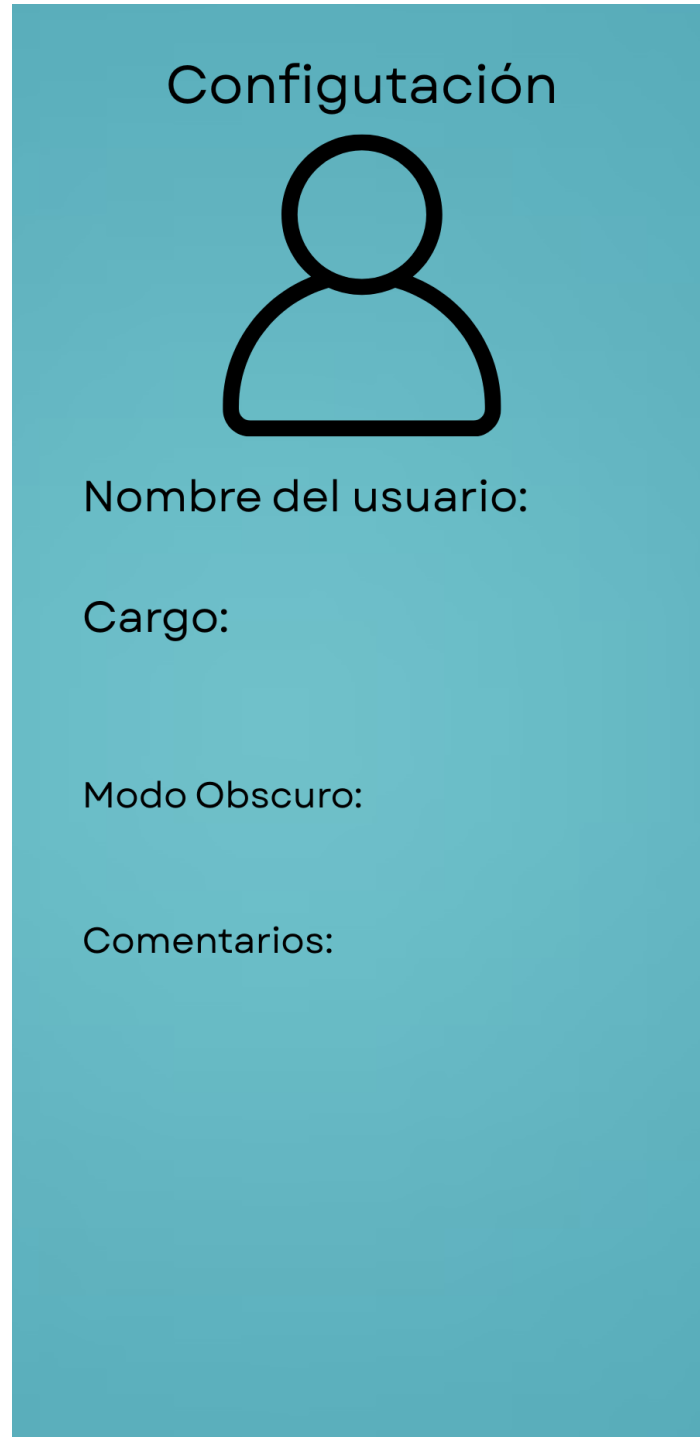


Figura 7: Mockup de la pantalla de Configuración.

Nota: Ninguno de estos templates es definitivo y su objetivo es proporcionar una idea del objetivo que tenemos como desarrolladores de la App.



4.2. Primeras Pantallas en Android Studio

Las primeras pantallas implementadas en Android Studio comprenden funcionalidades básicas que sentaron las bases del desarrollo de **MobiMant**. Entre estas funcionalidades se incluyen:

- **Registro e inicio de sesión:** Se implementó una interfaz, utilizando **Firebase Authentication** para el registro e inicio de sesión mediante email/contraseña. Esta funcionalidad valida a los usuarios y asigna roles diferenciados (administrador, operador y mecánico).
- **Consulta del estado de las máquinas:** Se desarrolló una pantalla que consulta y muestra el estado en tiempo real de las máquinas, extrayendo datos directamente de **Cloud Firestore**. Esta funcionalidad permite a los usuarios conocer de forma inmediata el estado de la máquina.
- **Generación de órdenes de mantenimiento:** Se implementó una interfaz que permite a los usuarios generar requerimientos de mantenimiento, con la capacidad de adjuntar imágenes (almacenadas a través de **Firebase Storage**), lo que permite evidenciar de forma visual los daños o fallos detectados.
- **Simulación de notificaciones push:** Se desarrolló la funcionalidad de envío de notificaciones push, utilizando **Firebase Cloud Messaging (FCM)**, para alertar a los usuarios sobre incidencias en tiempo real, asegurando una respuesta rápida ante eventos críticos.

4.3. Enlace al Repositorio GitHub

El código fuente y la documentación completa del proyecto se encuentran disponibles en el siguiente repositorio:

<https://github.com/NicolasFluxa/MobiMant.git>



5. Conclusiones

El proyecto **MobiMant** representa una solución integral para la gestión del mantenimiento en entornos industriales. Al digitalizar y centralizar el registro, seguimiento y análisis de las órdenes de mantenimiento, la aplicación no solo optimiza la productividad, sino que también reduce considerablemente los errores y mejora la trazabilidad en el proceso.

Durante el desarrollo se destacó la importancia de identificar una problemática real y traducirla en objetivos claros y medibles. La integración de servicios en la nube mediante Firebase y la implementación inicial en Android Studio han demostrado la viabilidad técnica de la propuesta. Además, el trabajo en equipo y la asunción de roles rotativos han enriquecido nuestro aprendizaje, permitiéndonos enfrentar y superar desafíos tecnológicos y de coordinación.



Referencias

- Analytics. (2023). *Firebase Analytics*. <https://firebase.google.com/docs/analytics?hl=es-419>. (Abril)
- Authentication. (2023). *Firebase Authentication*. <https://firebase.google.com/docs/auth?hl=es-419>. (Abril)
- Cloud. (2023). *Cloud Firestore*. <https://firebase.google.com/docs/firestore?hl=es-419>. (Abril)
- Cloud Messaging. (2023). *Firebase Cloud Messaging (FCM)*. <https://firebase.google.com/docs/cloud-messaging?hl=es-419>. (Abril)
- Mobility Work. (2025). *GMAO o Excel: ¿Cuál es la mejor opción para el mantenimiento?* <https://mobility-work.com/es/blog/gmao-o-excel-mantenimiento/>. (Abril)
- Nextiva. (2024). *VoIP Advantages and Disadvantages*. http://www-nextiva-com.translate.goog/blog/voip-advantages-and-disadvantages.html?_x_tr_sl=en&_x_tr_tl=es&_x_tr_hl=es&_x_tr_pto=sge. (Abril)
- Security Rules. (2023). *Firebase Security Rules*. <https://firebase.google.com/docs/rules?hl=es-419>. (Abril)
- Sistemas OEE. (2021). *Captura de datos manual en papel para órdenes de mantenimiento*. <https://www.sistemasoe.com/captura-datos-manual-papel/>. (Abril)
- Storage. (2023). *Firebase Storage*. <https://firebase.google.com/docs/storage?hl=es-419>. (Abril)