

**UNIVERSIDAD TECNOLÓGICA DE LA**

**COSTA GRANDE DE GUERRERO**

MEMORIA DE ESTADÍA

**“SISTEMA DE GESTIÓN DE TAREAS PARA QUBIT MANAGED SERVICES**”.

EMPRESA:

**“QUBIT MANAGED SERVICES”**

PRESENTADO POR:

**ANDRES PINEDA DE LA O**

PARA OBTENER EL TÍTULO DE:

**INGENIERÍA EN DESARROLLO Y GESTIÓN DE SOFTWARE**

ASESOR EMPRESARIAL: ING. MARCO ANTONIO SERRANO GARCÍA

ASESOR ACADÉMICO: MTRO. JOSÉ DAVID CASTREJÓN ORTEGA

**GENERACIÓN: 2023-2025**

Elaborada El 03 De enero De 2025 A Zihuatanejo GRO. 3 de mayo 2025

# Agradecimientos:

***Quiero agradecer profundamente a mi familia por ser mi base y mi mayor motivación, a mis maestros y asesor por su paciencia y por compartir su conocimiento, y a mis amigos y compañeros por hacer más llevadero este camino.***

***También agradezco a todas las personas que, de alguna forma, estuvieron presentes en este proceso, brindándome ánimo, consejos o simplemente escuchándome cuando más lo necesitaba. Cada palabra. ¡Gracias!***

# Dedicatoria:

***A mis padres, por su amor, apoyo incondicional y por ser mi mayor inspiración siempre.***

#### ÍNDICE

[Agradecimientos: II](#_90jt50kxq735)

[Dedicatoria: III](#_t79bg9x5vdxy)

[Introducción: VI](#_z66mqm31w47n)

[CAPÍTULO I. Marco contextual o referencial 7](#_lcjn778dqqrx)

[1.1 Antecedentes Históricos de la empresa: 7](#_4yncl46yr3x3)

[1.2 Descripción De La Empresa: 8](#_attqmmewn5t)

[1.3 Descripción Del Área Donde Se Realizarán Las Prácticas Profesionales: 11](#_ilmefmvyr1s4)

[Capítulo II. Marco teórico 12](#_gjk57aj85i06)

[2.1 ¿Qué Es Una Aplicación Móvil?: 12](#_7giaioqbjvux)

[2.2 Ventajas De Una Aplicación Móvil: 12](#_3u50s6tfmfwk)

[2.3 Bases de datos NoSQL: 13](#_tvkg6nto6cvz)

[2.4 Firebase: 13](#_m28uz8rsqtdh)

[2.5 Virtualización: 13](#_8uf30v371ygh)

[2.6 Android Studio: 14](#_tnhbkol3t43p)

[2.7 Kotlin: 14](#_cn3v40ptrfqh)

[2.8 Pruebas De Software: 15](#_2agjmh5uys7l)

[2.9 Modelo De Desarrollo De Software XP: Metodología XP (Extreme Programming) 15](#_nvbn72lx43o4)

[2.10 Manual De Usuario: 16](#_jok1t3rvchb5)

[2.11 Manual Técnico: 16](#_7vqh6yya6ie1)

[Capítulo III. Marco metodológico 17](#_gw5az8ojb7yr)

[3.1 Planteamiento del Problema: 17](#_wopqu7fmknae)

[3.2 Delimitación del problema: 18](#_iqdqw2uvo0fh)

[3.3 Justificación: 19](#_97fc6viamd02)

[3.4 Objetivos: 20](#_7zli62x0it50)

[3.5 Tipo De Investigación: 21](#_nhh37146ch7)

[Propuesta del proyecto: 24](#_61np5q4wzs93)

[Resumen: 24](#_gaxq9fyhzq0)

[Abstract: 25](#_xot0x0j8ierq)

[Desarrollo: 26](#_oon8ckay6zuc)

[Conclusiones Y Recomendaciones: 57](#_sr7xwz953kgu)

[Referencias BIBLIOGRÁFICAS 58](#_w0rb3kwho23b)

[Índice De Figuras 60](#_y6465l2s5tmu)

[Anexos 62](#_5a30fqrfne6)

[MANUAL DE USUARIO: 1](#_4qomkxwyv452)

[MANUAL TÉCNICO 1](#_nnp0bn8uc0ej)

# Introducción:

Este proyecto ha sido desarrollado con el objetivo de mejorar la organización y la eficiencia de los trabajadores de la empresa Qubit Managed Services. A través de esta herramienta, se facilita la gestión de tareas diarias como revisiones e instalaciones, permitiendo un control detallado de los reportes y actividades realizadas por cada empleado. De esta forma, se optimiza el flujo de trabajo y se favorece la productividad de cada trabajador.

El desarrollo de este documento se encuentra estructurado en cuatro apartados principales.

* **Capítulo I, Marco Contextual o Referencial**, se describe la situación actual de la empresa, su entorno, misión, visión y organigrama.
* **Capítulo II, Marco Teórico**, se abordan los conceptos y antecedentes técnicos necesarios para comprender la base del desarrollo.
* **Capítulo III, Marco Metodológico**, explica la metodología aplicada durante todo el proceso, desde el análisis hasta la implementación.
* **Propuesta del Proyecto**, se presenta la solución tecnológica diseñada, detallando su funcionamiento, características técnicas, evidencias del desarrollo (como diagramas, pantallas de diseño y fragmentos de código), así como los resultados obtenidos tras su implementación

# CAPÍTULO I. Marco contextual o referencial

## Antecedentes Históricos de la empresa:

La empresa es un proveedor de servicios de Tecnología de la Información que ofrece soluciones I.T. administradas y tradicionales. Cuenta con una amplia experiencia en gestión de redes, seguridad informática, servicios en la nube, respaldo y recuperación ante desastres, dirección de proyectos y más.

Ofrece un amplio catálogo de servicios I.T., que incluyen la administración de infraestructura tecnológica, gestión de proyectos técnicos, asignación de recursos FTE, soporte para usuarios finales, mantenimiento y reparación de hardware, así como despliegues y migraciones de hardware y software para clientes. Su equipo de profesionales altamente comprometidos se distingue por su pasión y dedicación.



## Descripción De La Empresa:

* **Nombre de la empresa**

“QUBIT MANAGED SERVICES”.

* **Dirección**

“Av. Teopanzolco No.669 Col. Lomas de Cortes Cuernavaca, Morelos.CP. 62240”.

* **Sector y giro empresarial**

“Tecnología de la Información”.

* **Servicios que ofrecen**

“Soluciones I.T. administradas y tradicionales, administración de infraestructura tecnológica, gestión de proyectos técnicos, asignación de recursos FTE, soporte para usuarios finales, mantenimiento y reparación de hardware, despliegues y migraciones de hardware y software.”

* **Nombre de la memoria**

“SISTEMA DE GESTIÓN DE TAREAS PARA QUBIT MANAGED SERVICES”

* **Asesor de la empresa**

“Marco Antonio Serrano García”.

* **Cargo del Asesor Empresarial**

Ingeniero

### Misión

"Proveer soluciones tecnológicas innovadoras y servicios administrados de clase mundial, mediante un equipo altamente calificado y procesos optimizados, contribuyendo al éxito de nuestros clientes y al desarrollo profesional de nuestros colaboradores."

### Visión

"Consolidarnos como el partner tecnológico preferido en el mercado de servicios administrados de TI, reconocidos por nuestra excelencia operativa y capacidad de innovación constante."

### Objetivos:

El objetivo principal es impulsar la inclusión financiera, atendiendo a los segmentos de la población que tradicionalmente han sido desatendidos por la banca comercial, especialmente los estratos medios y bajos.

### Estructura orgánica general:

## Descripción Del Área Donde Se Realizarán Las Prácticas Profesionales:

### Nombre del área:

Departamento de Operaciones: Field Services

### Objetivos:

Ofrecer soluciones tecnológicas integrales y de alta calidad que optimicen los procesos operativos y administrativos de nuestros clientes, garantizando eficiencia, seguridad y soporte técnico profesional.

### Funciones:

* Realizar visitas técnicas a clientes
* Ejecutar mantenimientos preventivos y correctivos
* Implementar nuevas soluciones en sitio

### Organigrama.

# Capítulo II. Marco teórico

## ¿Qué Es Una Aplicación Móvil?:

Una app es una aplicación de software diseñada para funcionar en dispositivos móviles, como smartphones y tabletas.

A diferencia de las web apps, que son accesos directos a páginas web y no requieren de instalación alguna, las apps se instalan en el dispositivo. Su popularidad se disparó tras el lanzamiento de tiendas de aplicaciones como la App Store de Apple y el Android Market (ahora Google Play) en 2008, convirtiéndose en un término cotidiano y en un tipo de software con el que la gran mayoría de la población interactúa a diario.

Calvo, L. (2024, 26 noviembre). *¿Qué es una app? Guía sobre aplicaciones móviles*. GoDaddy Resources - Spain. <https://www.godaddy.com/resources/es/tecnologia/que-es-una-app-y-para-que-se-utiliza>

## Ventajas De Una Aplicación Móvil:

* Ofrece una accesibilidad mucho mayor
* La experiencia de usuario es veloz, sencilla y ágil
* La navegabilidad y usabilidad es mejor
* Adaptadas a toda clase de dispositivos y para numerosos usuarios

Admin\_Gunka. (2025, 12 enero). ¿Conoces las ventajas y desventajas de una aplicación móvil? | Gunka Studios. *Gunka Studios*. <https://gunkastudios.com/ventajas-y-desventajas-aplicacion-movil/>

## Bases de datos NoSQL:

Una base de datos NoSQL, o base de datos no relacional, permite almacenar y manipular datos no estructurados y semiestructurados (a diferencia de una base de datos relacional, que define cómo se deben componer todos los datos insertados en la base de datos). Las bases de datos NoSQL se hicieron populares a medida que las aplicaciones web se volvían más comunes y complejas.

*¿Qué es una base de datos?* (s. f.). <https://www.oracle.com/mx/database/what-is-database/>

## Firebase:

Firebase de Google es una plataforma en la nube para el desarrollo de aplicaciones web y móvil. Está disponible para distintas plataformas (iOS, Android y web), con lo que es más rápido trabajar en el desarrollo.

Su función esencial es hacer más sencilla la creación de tanto aplicaciones webs como móviles y su desarrollo, procurando que el trabajo sea más rápido, pero sin renunciar a la calidad requerida.

Mora, S. L. (2022, 4 octubre). *Firebase: qué es, para qué sirve, funcionalidades y ventajas*. DIGITAL55. <https://digital55.com/blog/que-es-firebase-funcionalidades-ventajas-conclusiones/>

## Virtualización:

La virtualización es una tecnología que se puede usar para crear representaciones virtuales de servidores, almacenamiento, redes y otras máquinas físicas. El software virtual imita las funciones del hardware físico para ejecutar varias máquinas virtuales a la vez en una única máquina física. Las empresas recurren a la virtualización para utilizar sus recursos de hardware de manera eficiente y obtener retornos mayores de sus inversiones.

*¿Qué es la virtualización? - Explicación de la virtualización de la computación en la nube - AWS*. (s. f.). Amazon Web Services, Inc. <https://aws.amazon.com/es/what-is/virtualization/>

## Android Studio:

Android Studio es el entorno de desarrollo integrado (IDE) oficial que se usa en el desarrollo de apps para Android. Basado en el potente editor de código y las herramientas para desarrolladores de IntelliJ IDEA, Android Studio ofrece aún más funciones que mejoran tu productividad cuando compilas apps para Android, como las siguientes:

* Un sistema de compilación flexible basado en Gradle
* Un emulador rápido y cargado de funciones
* Un entorno unificado donde puedes desarrollar para todos los dispositivos Android
* Ediciones en vivo para actualizar elementos componibles en emuladores y dispositivos físicos, en tiempo real
* Integración con GitHub y plantillas de código para ayudarte a compilar funciones de apps comunes y también importar código de muestra

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

## Kotlin:

Kotlin es un lenguaje de programación moderno que fue diseñado para ser una alternativa más concisa e intuitiva a Java. Es un lenguaje de tipado estático que se ejecuta en la máquina virtual Java (JVM) y compila tanto en bytecode Java como en JavaScript.

Kotlin es totalmente compatible con Java, por lo que el código Java existente puede convertirse a Kotlin sin ninguna modificación. A partir de Android Studio 3.0, Kotlin es ahora el lenguaje por defecto para el desarrollo de Android.

Moraguez, E. R. (2023, 2 marzo). *¿Qué es Kotlin: ¿Cómo funciona y para qué sirve?‍ | LovTechnology*. LovTechnology. <https://lovtechnology.com/que-es-kotlin-como-funciona-y-para-que-sirve/>

## Pruebas De Software:

Las pruebas de software son el proceso de evaluar y verificar que un producto o aplicación de software hace lo que se supone que debe. Los beneficios de unas buenas pruebas incluyen la prevención de errores y la mejora del rendimiento

Las pruebas de software hoy en día son más efectivas cuando son continuas, lo que indica que las pruebas se inician durante el diseño, continúan a medida que se crea el software e incluso se producen cuando se despliega en producción.

Ibm. (2024, 9 mayo). ¿Qué son las pruebas de software y cómo funcionan? | IBM. *S.N*. <https://www.ibm.com/mx-es/topics/software-testing>

## Modelo De Desarrollo De Software XP: Metodología XP (Extreme Programming)

La programación extrema es una metodología ágil de gestión de proyectos que se centra en la velocidad y la simplicidad con ciclos de desarrollo cortos y con menos documentación. La estructura del proceso está determinada por 5 valores fundamentales, 5 reglas y 12 prácticas de XP.

Al igual que otras metodologías ágiles, la programación extrema es un método de desarrollo de software dividido en Sprint de trabajo. Los marcos ágiles siguen un proceso iterativo, en el que se completa y revisa el marco al final de cada sprint, refinándolo para adaptarlo a los requisitos cambiantes y alcanzar la eficiencia máxima.

Raeburn, A. (2025, 13 febrero). ¿Qué es la programación extrema (XP)? [2025] • Asana. *Asana*. <https://asana.com/es/resources/extreme-programming-xp>

## Manual De Usuario:

Un manual de usuario, también llamado guía de usuario, es un documento de comunicación técnica destinado a dar asistencia a las personas que utilizan un sistema en particular. Por lo general, este documento está redactado por un escritor técnico, como por ejemplo los programadores del sistema o los directores de proyectos implicados en su desarrollo, o el personal técnico, especialmente en las empresas más pequeñas.

Los manuales de usuario son más comúnmente asociados con los productos electrónicos, como computadoras y programas.

colaboradores de Wikipedia. (2023, 8 noviembre). *Manual de usuario*. Wikipedia, la Enciclopedia Libre. <https://es.wikipedia.org/wiki/Manual_de_usuario>

## Manual Técnico:

Un manual técnico es aquel que va dirigido a un público con conocimientos técnicos sobre algún área. La documentación de proyectos es importante para identificar más fácilmente los aspectos y características que forman parte de un proyecto. Una adecuada documentación le proporciona identidad y "personalidad" a un proyecto, de manera que los usuarios irresponsables del mismo podrán reconocer más fácilmente las ventajas y desventajas, características y funcionalidades, funciones y ventajas, así como costos y beneficios que impliquen el desarrollo del proyecto.

*Manual Técnico | evedg17*. (s. f.). Evedg17. <https://miyuvampaia17.wixsite.com/evedg17/blank-4>

# Capítulo III. Marco metodológico

## Planteamiento del Problema:

En muchas empresas que brindan servicios de mantenimiento, seguimiento técnico o soporte a sucursales, los empleados enfrentan dificultades para consultar y gestionar sus tareas diarias de manera eficiente. La falta de herramientas móviles adecuadas complica la visualización de información actualizada, lo que puede provocar retrasos, duplicación de esfuerzos o incumplimiento de plazos.

Aunque existen plataformas de gestión de tareas, muchas de ellas no están optimizadas para dispositivos móviles o no se adaptan a las necesidades específicas de cada organización. Por ello, surge la necesidad de desarrollar una aplicación móvil que permita a los empleados consultar de forma clara y rápida sus tareas asignadas, con información como título, descripción, prioridad, estatus y fecha límite.

## Delimitación del problema:

Este proyecto se enfoca exclusivamente en la visualización de tareas desde dispositivos móviles mediante una aplicación desarrollada en Android Studio. Las funciones desarrolladas incluyen la consulta de tareas almacenadas en Firebase Firestore, su despliegue en una interfaz ordenada y el filtrado por criterios como prioridad o estatus.

No se abordan en esta etapa funcionalidades como la creación, edición o eliminación de tareas, ni aspectos relacionados con la administración de usuarios o reportes estadísticos. El alcance está limitado al módulo móvil y se orienta principalmente al personal técnico que ejecuta tareas asignadas por la empresa.

## Justificación:

Las razones principales por las cuales se decide desarrollar una aplicación para la gestión de reportes de actividades en la empresa son las siguientes:

• Mejora en el control y seguimiento: Permite a la empresa tener un registro más preciso y actualizado de las actividades realizadas.

• Facilidad y rapidez: Los empleados podrán registrar sus actividades de forma ágil y desde cualquier lugar.

• Incremento en la eficiencia operativa: Permite a la empresa optimizar la gestión de la información y reducir errores.

Por estas razones, la implementación de una aplicación de reportes contribuye a mejorar la organización, la eficiencia y el control de procesos de la empresa.

## Objetivos:

### Objetivo General:

Desarrolla e implementar una aplicación que facilite la administración y la documentación de informes de actividades en la organización.

### Objetivos Específicos

* Establecer un sistema digital para la recopilación y registro eficiente de las actividades llevadas a cabo por el equipo.
* Promover la utilización y acceso a la aplicación desde aparatos móviles para que los trabajadores puedan informar en tiempo real.
* Optimizar la comunicación interna a través de la transformación digital de los procesos de informe y monitoreo de actividades.

## Tipo De Investigación:

Desarrollar una investigación de tipo aplicada con enfoque cuantitativo, al tener como finalidad ofrecer una solución tecnológica práctica a una problemática identificada en la empresa Qubit Managed Services, relacionada con la organización y el seguimiento de tareas del personal técnico.

Realizar un estudio no experimentalydescriptivo, al no manipular variables, sino observar y analizar la situación actual para proponer una herramienta que permita mejorar la gestión de actividades. Describir el funcionamiento del sistema propuesto y evaluar su impacto en los procesos internos mediante el desarrollo de una aplicación móvil y una plataforma de administración conectadas a una base de datos en la nube.

Implementar el diseño en un entorno real para observar sus efectos en la productividad y la organización del equipo de trabajo, y así validar la efectividad de la solución tecnológica planteada.

### 3.5.1 Análisis e interpretación de los resultados

Se plantearon preguntas enfocadas en el tema, para poder obtener respuestas breves, con el objetivo de obtener información que facilitará el entendimiento de la situación.

¿Consideran útil contar con una aplicación móvil para gestionar tareas?

¿Consideran útil contar con una aplicación móvil para gestionar tareas?

¿Tiene la empresa interés en implementar esta solución?

*Figura 6 Respuestas 3*

¿Los supervisores han identificado fallas en la comunicación?

# Propuesta del proyecto:

## Resumen:

En esta memoria de estadías se presenta el desarrollo de una aplicación móvil dirigida a la empresa Qubit Managed Services.

El primer aspecto a analizar corresponde a la recopilación de datos, donde se examinan el enfoque de la aplicación, sus objetivos generales y específicos, la metodología empleada y los requerimientos necesarios para su implementación.

El segundo aspecto se centra en el proceso de creación, detallando el equipo utilizado tanto en su parte física como digital, así como los pasos seguidos para la elaboración del proyecto.

## Abstract:

This report presents the development of a mobile application aimed at the company Qubit Managed Services.

The first aspect to be analyzed corresponds to data collection, where the approach of the application, its general and specific objectives, the methodology used and the necessary requirements for its implementation are examined.

The second aspect focuses on the creation process, detailing the equipment used both in its physical and digital parts, as well as the steps followed for the development of the project.

## Desarrollo:



### Cronograma De Actividades:

### Planificación:

La etapa de planificación implica la definición de los objetivos del proyecto y los requerimientos funcionales

### Entrevista:

* **¿Cuál es el objetivo principal del programa?**
* *R = Establecer un orden a las tareas de los trabajadores para que puedan desarrollar de mejor manera sus actividades diarias*
* **¿Qué funciones específicas necesita?**
* *R = Que pueda asignarles tareas del día a mis empleados y ellos puedan mandarme sus tareas completadas en un reporte y pueda estar al tanto de si todo se realizó con éxito o hubo algún inconveniente*
* **¿Creé que este sistema pueda darles mayor eficiencia a todos?**
* *R = Si, ya que a si todos los empleados podrán llevar su propio orden especifico*

### Requerimientos Funcionales Prioritarios:

Desarrollar una aplicación móvilque optimice la organización de tareas técnicas (instalaciones, revisiones, mantenimientos) para el personal de Qubit.

Visualización jerárquica de tareas**,** filtrables por: prioridad (alta/medio/baja), estado (pendiente/en progreso/completado) y técnico asignado.

Sistema de notificaciones integrado para alertar sobre tareas próximas a vencer (vía correo electrónico y notificaciones push internas).

### Diseño:

La etapa de diseño implica la descripción de las herramientas y el hardware que se usará, además de la creación del diseño de interacción.

### Tecnologías que se usaran:

* Computadora.
* Android Studio.
* Firebase.
* Smartphone.
* Cable.
* Conexión a internet.

### Especificaciones Técnicas:

**Requisitos Mínimos:**

* **Frontend:**
  + Chrome 98+
  + 2GB RAM
  + Conexión 3G
* **Backend:**
  + PHP 8.1+
  + 2 vCPUs
  + 4GB RA

### Diagrama De Flujo

### Diagrama De Caso De Uso:

### Diccionario De Datos:

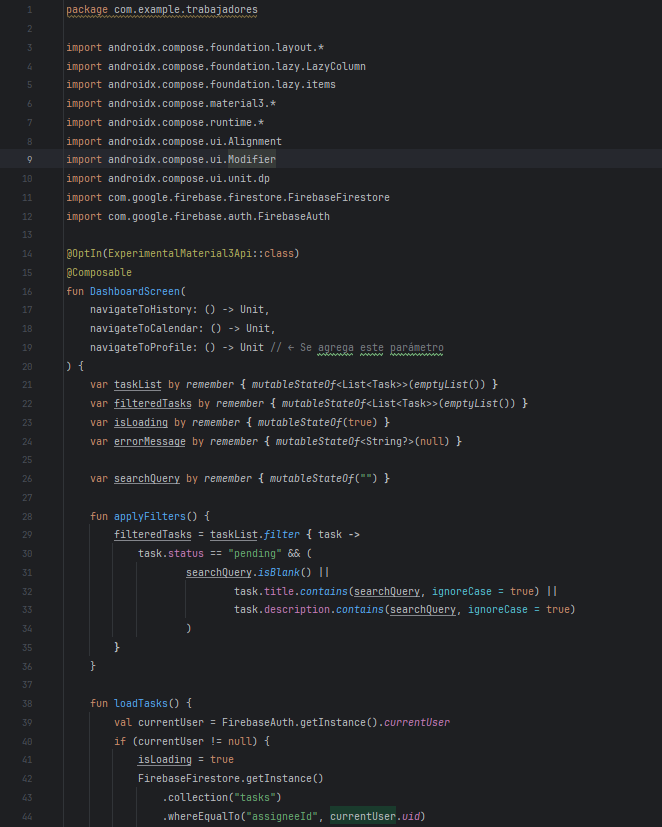
| Campo | Descripción | Tipo de dato |
| --- | --- | --- |
| Base de datos | Sistema de tareas | S.D |
| Admin | Tabla de Administradores | S.D |
| Nombre | Nombre del admin | String |
| Email | Correo electrónico | String |
| Password | Contraseña | String |
| CreatedAt | Fecha de creación | timestamp |
| users | Tabla de trabajadores | S.D |
| Nombre | Registra información del nombre completo del trabajador | string |
| Email | Registra su correo electrónico de la empresa | string |
| Password | Se le asigna una contraseña | string |
| Role | Se registra el dato del roll especifico | string |
| CreatedAt | Se registra cuando fue ingresado el usuario | Timestamp |
| Tasks | Tabla de tareas | S.D |
| Id | Se asigna una id única a la tarea | string |
| CreatedAt | Fecha de creación de la tarea | Timestamp |
| deadline | Fecha límite para completar la tarea | Timestamp |
| Description | Descripción a detalle de la tarea | string |
| Priority | Nivel de prioridad que debe atribuirse a una tarea | string |
| Status | Estatus del estado de la tarea /pendiente/completo | string |
| Title | Título principal de la tarea | string |

### Codificación:

Desarrollar esta etapa conlleva transformar los diseños previos y los requerimientos funcionales en un conjunto de instrucciones que permitan la ejecución del sistema de manera efectiva. Escribir el código fuente de la aplicación utilizando las tecnologías seleccionadas, en este caso Android Studio como entorno de desarrollo y Firebase como plataforma de backend, permite implementar las funcionalidades clave como autenticación, gestión de tareas y control de accesos.

Aplicar buenas prácticas de programación y estructurar el código siguiendo una arquitectura ordenada es fundamental para garantizar la mantenibilidad y escalabilidad del sistema. Utilizar funciones reutilizables, validar datos y manejar excepciones son aspectos esenciales para asegurar el correcto funcionamiento de la aplicación en diferentes escenarios.

*DashboardScreen:*

La primera vista de la aplicación en la cual se muestra las tareas en forma de lista, barra de búsqueda, botón para poder marcar las tareas como completadas y un BotonBar en la parte inferior para poder navegar entre cada página.



### TaskHistoryScreen

Esta vista muestra en forma de lista todas las tareas que fueron marcadas como completadas en la ventana de DashboarScreen, esta vista tiene la función de mantener un control de las tareas. 

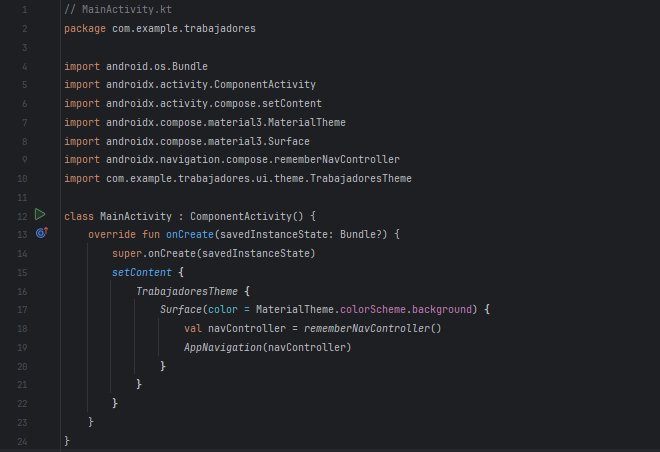


*NavigationScreen*

Esta vista es la que permite moverse entre las diferentes ventanas de la aplicación, facilitando al usuario poder cambiar de ventanas.

*Figura 19 NavigationScreen1*

*MainActivity*

Es la vista principal de la aplicación, esta da la orden de ejecución y a su vez redirige hacia la primera vista de la aplicación. 

### Pruebas

Es esencial llevar a cabo pruebas durante el desarrollo de una aplicación móvil para garantizar la calidad del producto final.

Estas pruebas posibilitan identificar y rectificar fallos de manera oportuna, optimizar la experiencia del usuario, asegurar la compatibilidad con diferentes dispositivos, y confirmar que el sistema opere conforme a los requisitos propuestos al inicio.

| **TÍTULO DE LA PRUEBA** | | | **PRIORIDAD** | **ID DE CASO DE PRUEBA** | | **NÚMERO DE PRUEBA** | | | **FECHA DE LA PRUEBA** |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Pruebas unitarias | | | Alta | #1 | | #1 | | | 30/mzo/2025 |
| **DESCRIPCIÓN DE LA PRUEBA** | | | | **PRUEBA DISEÑADA POR** | | **PRUEBA EJECUTADA POR** | | | **FECHA DE EJECUCIÓN** |
| Examinar El Funcionamiento Adecuado De Cada Elemento Individual De La Aplicación | | | | Andres Pineda | | Andres Pineda | | | 31/mzo/2025 |
|  | | | | | | | | | |
|  | | | | | | | | | |
| **ID DE PASO** | **DESCRIPCIÓN DEL PASO** | **FECHA DE LA PRUEBA** | **RESULTADOS ESPERADOS** | | **RESULTADOS REALES** | | **APROBAR / REPROBAR** | **NOTAS ADICIONALES** | |
| U1 | Comprobación De Campos Necesarios | 31/mzo/2025 | Aplicación Funcional | | * Error Al Cargar La Aplicación * Error En La Lectura De Datos | | APROBADA | Errores corregidos | |
| U2 | Verificar Las Vistas Y Todas Sus Funciones | 31/mzo/2025 | Opciones Funcionales | | * Error Al Cambiar De Vista | | APROBADA | Errores corregidos | |

**PLANIFICACIÓN Y EJECUCIÓN DE CASOS DE PRUEBA**

**PLANIFICACIÓN Y EJECUCIÓN DE CASOS DE PRUEBA**

| **TÍTULO DE LA PRUEBA** | | | **PRIORIDAD** | **ID DE CASO DE PRUEBA** | | **NÚMERO DE PRUEBA** | | | **FECHA DE LA PRUEBA** |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Pruebas De Rendimiento | | | Media | #2 | | #2 | | | 1/abr/2025 |
| **DESCRIPCIÓN DE LA PRUEBA** | | | | **PRUEBA DISEÑADA POR** | | **PRUEBA EJECUTADA POR** | | | **FECHA DE EJECUCIÓN** |
| Facilitar La Detección De Pequeños Fallos De Rendimiento En La Aplicación | | | | Andres Pineda | | Andres Pineda | | | 2/abr/2025 |
|  | | | | | | | | | |
|  | | | | | | | | | |
| **ID DE PASO** | **DESCRIPCIÓN DEL PASO** | **FECHA DE LA PRUEBA** | **RESULTADOS ESPERADOS** | | **RESULTADOS REALES** | | **APROBAR / REPROBADA** | **NOTAS ADICIONALES** | |
| R1 | Tiempo De Carga De La Aplicación | 2/abr/2025 | 2 Segundos Máximo | | 1.5 Segundos En La Carga Inicial | | APROBADA | Sin Errores | |
| R2 | Rapidez Y Velocidad De Sincronización De Datos Con El Servidor. | 2/abr/2025 | 4 Segundos Máximo | | * Tiempo Mínimo 2 Segundos * Tiempo Máximo 4 Segundos | | APROBADA | Sin Errores | |

**PLANIFICACIÓN Y EJECUCIÓN DE CASOS DE PRUEBA**

| **TÍTULO DE LA PRUEBA** | | | **PRIORIDAD** | **ID DE CASO DE PRUEBA** | | **NÚMERO DE PRUEBA** | | | **FECHA DE LA PRUEBA** |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Pruebas De Validación De Datos | | | Alta | #3 | | #3 | | | 3/abr/2025 |
| **DESCRIPCIÓN DE LA PRUEBA** | | | | **PRUEBA DISEÑADA POR** | | **PRUEBA EJECUTADA POR** | | | **FECHA DE EJECUCIÓN** |
| Facilitar La Detección Fallos De Validación | | | | Andres Pineda | | Andres Pineda | | | 4/abr/2025 |
|  | | | | | | | | | |
| **ID DE PASO** | **DESCRIPCIÓN DEL PASO** | **FECHA DE LA PRUEBA** | **RESULTADOS ESPERADOS** | | **RESULTADOS REALES** | | **APROBAR / REPROBAR** | **NOTAS ADICIONALES** | |
| V1 | Verificar datos de los usuarios introducidos en la base de datos | 4/abr/2025 | Mostrar Los Datos Almacenados En La Base De Datos | | * Datos Del Usuario Mostrados Correctamente | | APROBADA | Sin Errores | |
| V2 | Seguir la estructura establecida (título, descripción, fecha, responsable, etc.). | 4/abr/2025 | Mostrar Los Datos Almacenados En La Base De Datos | | * Datos De Las Tareas Mostrados Correctamente | | APROBADA | Sin Errores | |

### Lanzamiento:

Una vez completados los pasos anteriores para el desarrollo de la aplicación móvil de tareas, se procede a su implementación final.

La aplicación es empaquetada en un archivo ejecutable en formato APK (Android Package), lo que permite su instalación manual en dispositivos Android compatibles.

El archivo APK es generado directamente desde el entorno de desarrollo. De esta forma, se facilita la instalación de la aplicación en los dispositivos de los empleados o usuarios finales sin necesidad de pasar por una plataforma de distribución.

Este método de entrega es adecuado para proyectos de uso interno, donde se requiere un control más directo sobre la instalación y el acceso a la aplicación.

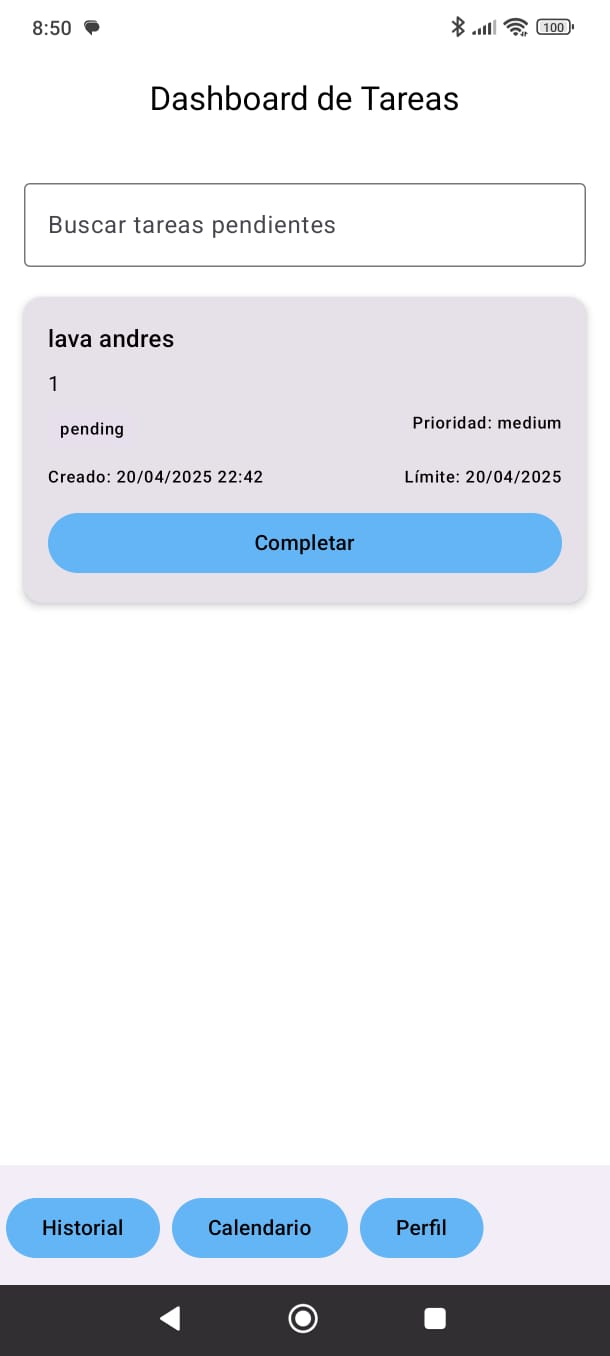
### Resultados

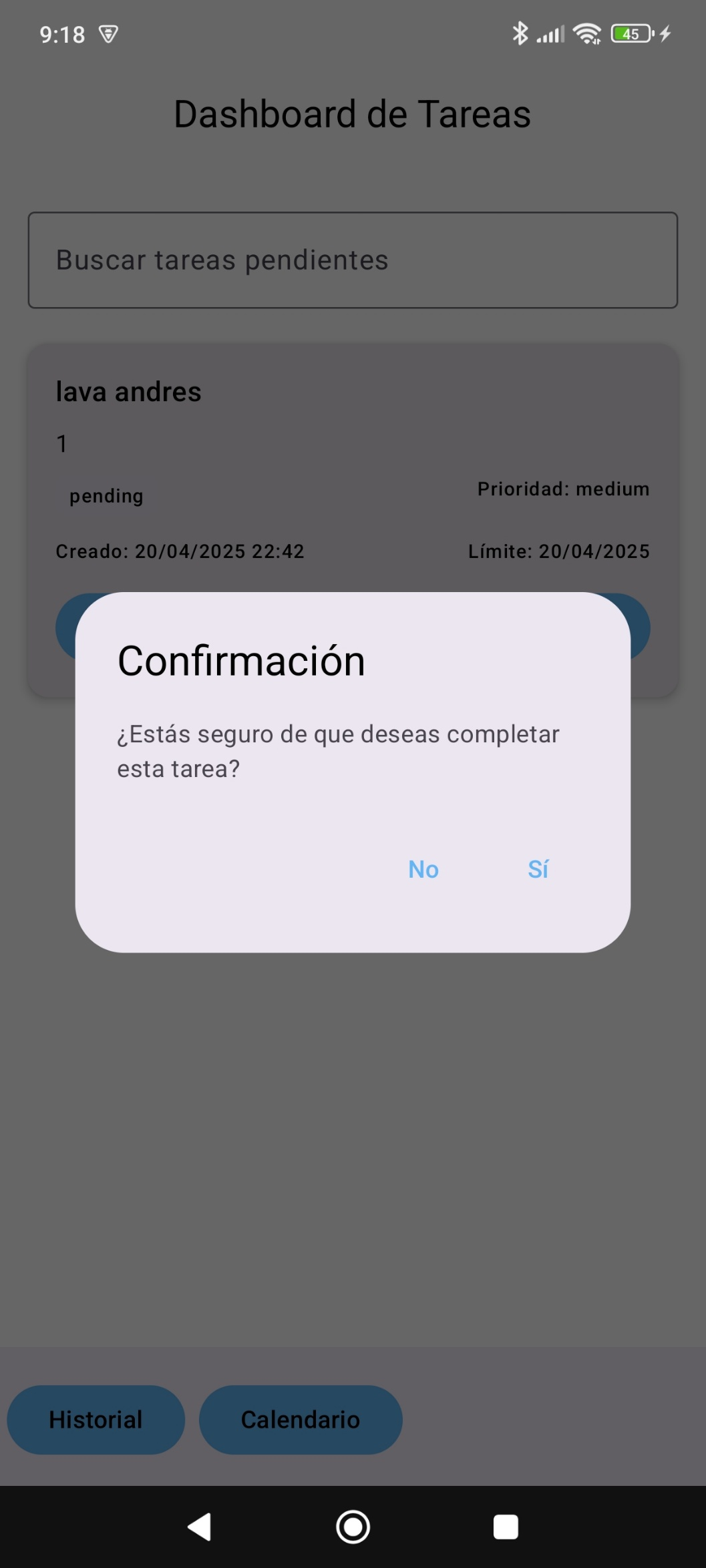
### Etapa de Planificación:

Se definieron roles técnicos (frontend, backend, base de datos). Se usó Trello para distribuir tareas semanalmente.

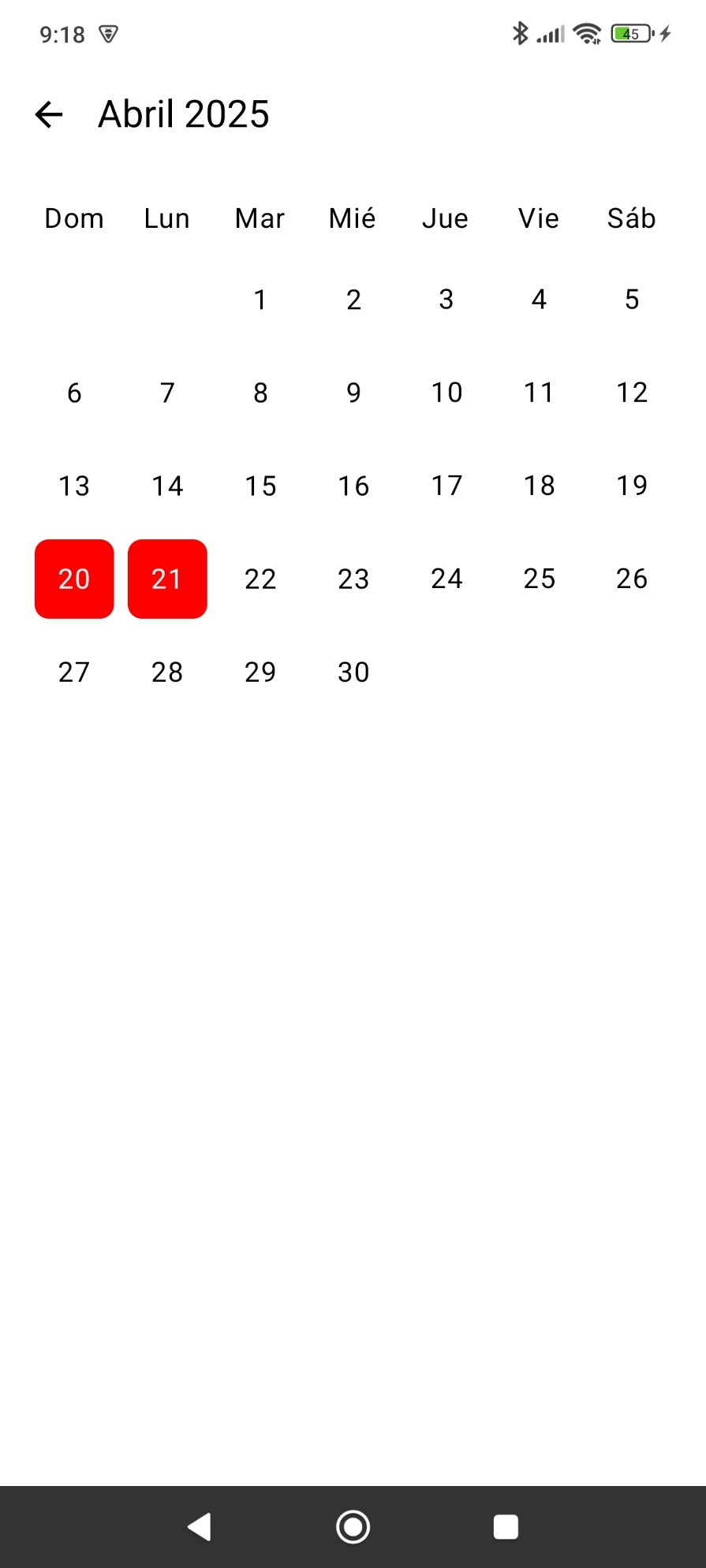
### Etapa de Diseño

Los diseños fueron creados en libreta y validados con el equipo. Se elaboró un flujo de navegación de pantallas.

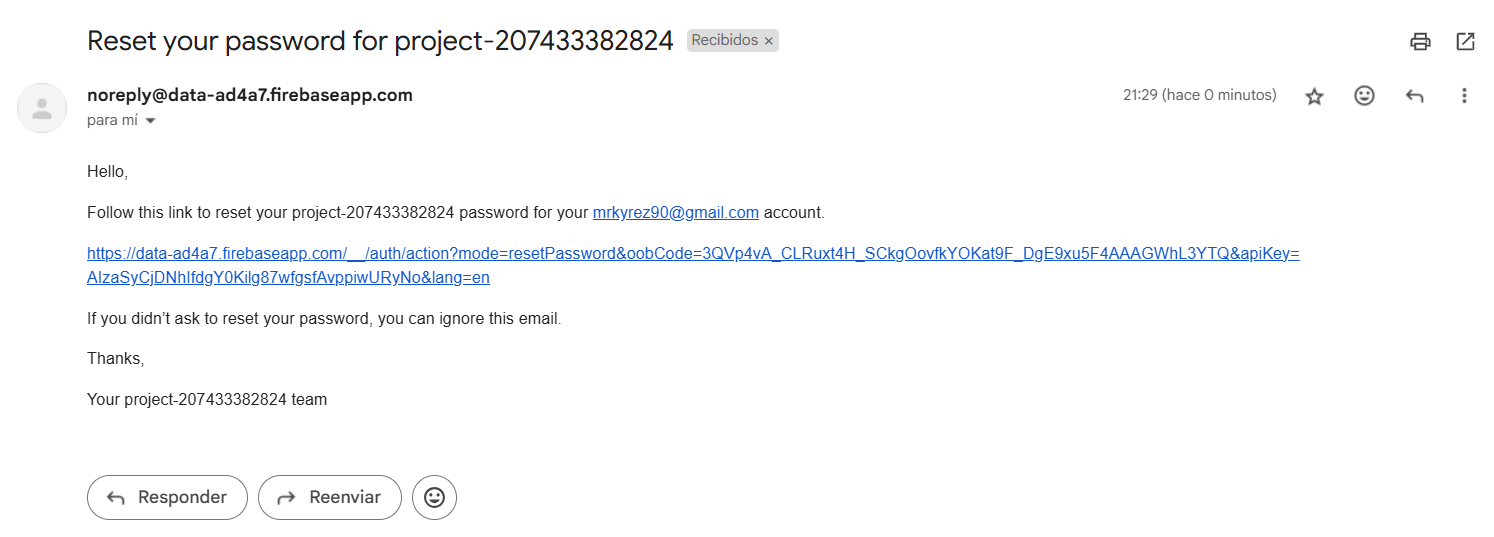


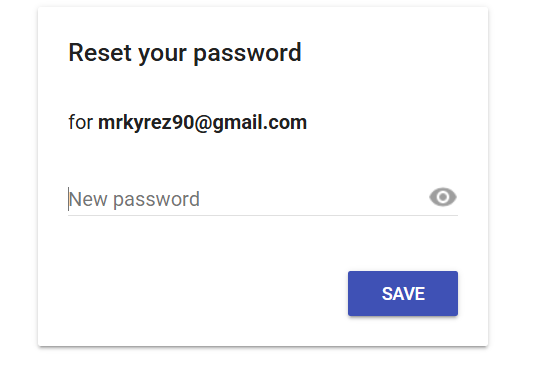






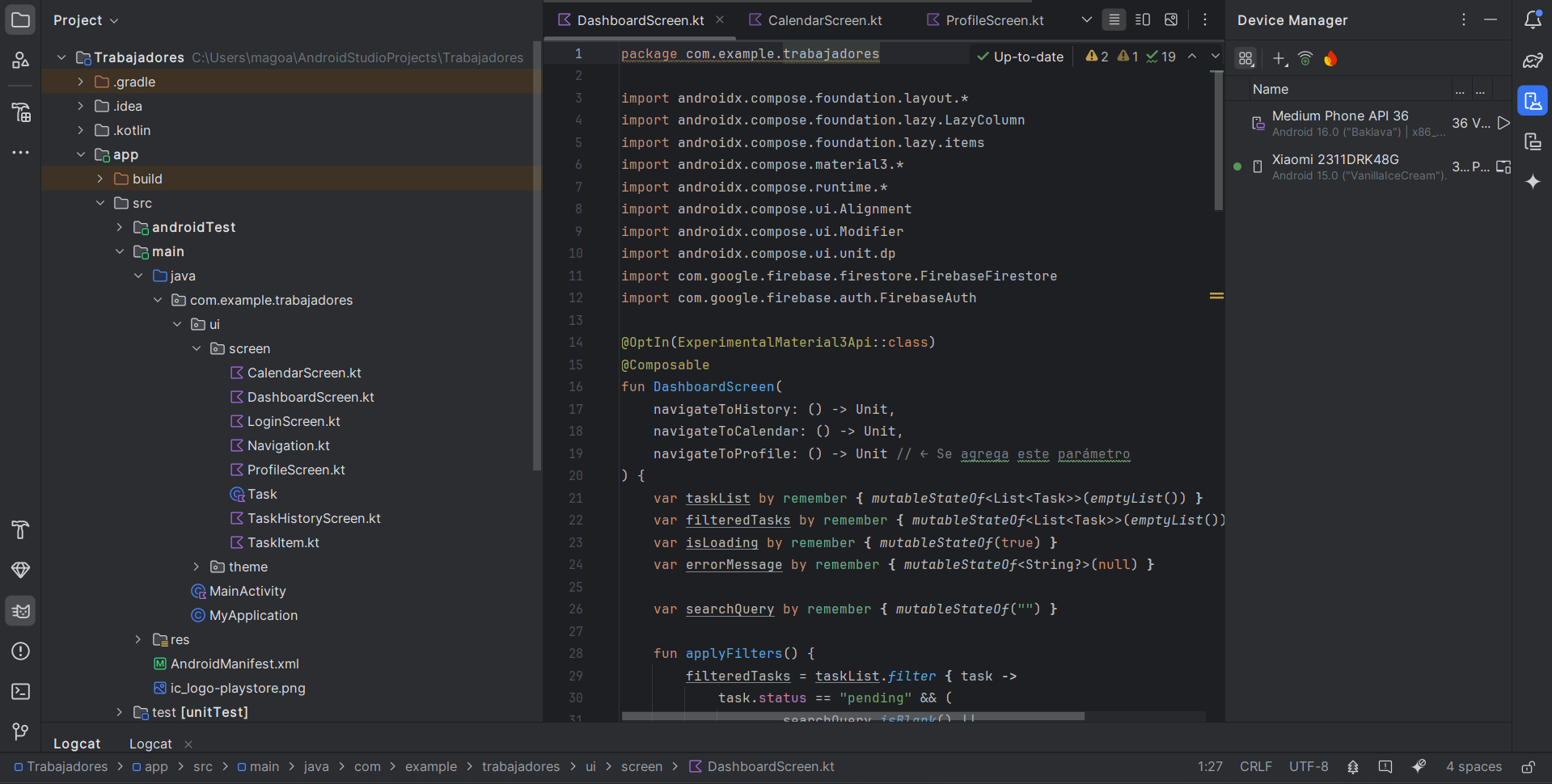








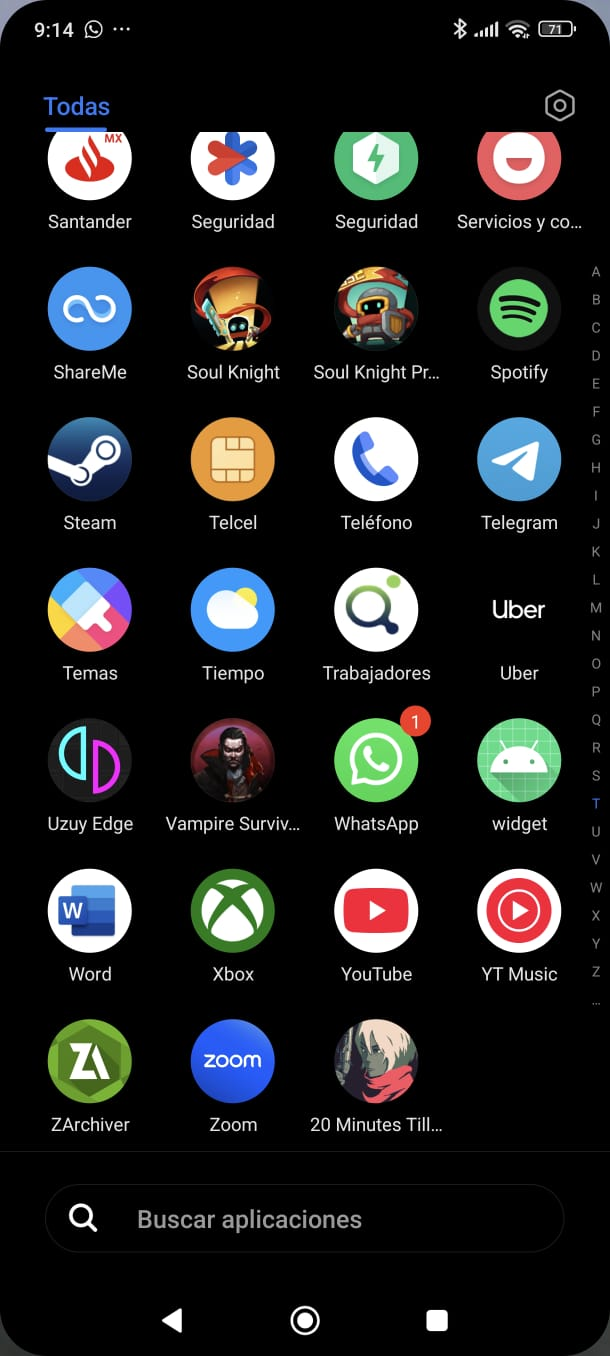
### Etapa De Codificación

Se comenzó con el módulo de login, usando Firebase Auth. Posteriormente se desarrolló la pantalla de dashboard, que carga las tareas filtradas por assigneeId. El código fue escrito en Kotlin bajo arquitectura MVVM.

### Etapa de Pruebas

Se ejecutaron pruebas manuales sobre validaciones de campos vacíos, errores de autenticación, carga incorrecta de tareas y comportamiento fuera de línea. Se documentaron 3 errores, todos corregidos antes de la entrega.

### Etapa de Implementación

La aplicación fue entregada en APK en dispositivos Android a los técnicos. Se levantaron observaciones sobre posibles mejoras futuras.

# Conclusiones Y Recomendaciones:

Gracias a esta herramienta, los empleados pueden visualizar de forma clara información relevante como el título, la descripción, la prioridad, el estado y la fecha límite de cada actividad, lo que mejora la organización y el cumplimiento de sus responsabilidades.

La aplicación desarrollada demostró ser efectiva para organizar las tareas del personal de manera más eficiente, lo que permitió a Qubit Managed Services supervisar mejor las actividades y optimizar los procedimientos internos. Como resultado, la empresa, ahora presenta beneficios en el rendimiento y la coordinación del equipo.

Como parte de las recomendaciones a futuro, se detectaron posibles mejoras para futuras versiones del proyecto, tales como incorporar un sistema de notificaciones, filtros más precisos, un módulo de estadísticas. Además, agregar un buscador de tareas por palabra clave podría mejorar aún más la usabilidad.

# Referencias BIBLIOGRÁFICAS

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

*Moraguez, E. R. (2023, 2 marzo). ¿Qué es Kotlin: ¿Cómo funciona y para qué sirve?‍ | LovTechnology. LovTechnology.* <https://lovtechnology.com/que-es-kotlin-como-funciona-y-para-que-sirve/>

*¿Qué es una base de datos? (s. f.).* <https://www.oracle.com/mx/database/what-is-database/>

*¿Qué es la virtualización? - Explicación de la virtualización de la computación en la nube - AWS. (s. f.). Amazon Web Services, Inc.* <https://aws.amazon.com/es/what-is/virtualization/>

*Mora, S. L. (2022, 4 octubre). Firebase: qué es, para qué sirve, funcionalidades y ventajas. DIGITAL55.* <https://digital55.com/blog/que-es-firebase-funcionalidades-ventajas-conclusiones/>

*Aprende el lenguaje de programación Kotlin. (s. f.). Android Developers.* <https://developer.android.com/kotlin/learn?hl=es-419>

*Cómo agregar dependencias de compilación. (s. f.). Android Developers.* <https://developer.android.com/build/dependencies?hl=es-419>

*Cómo descargar Android Studio y App Tools - Android Developers. (s. f.). Android Developers.* <https://developer.android.com/studio?hl=es-419>

*Firebase. (s. f.). Firebase.* <https://firebase.google.com/?hl=es-419>

*Raeburn, A. (2025, 13 febrero). ¿Qué es la programación extrema (XP)? [2025] • Asana. Asana.* <https://asana.com/es/resources/extreme-programming-xp>

*Admin\_Gunka. (2025, 12 enero). ¿Conoces las ventajas y desventajas de una aplicación móvil? | Gunka Studios. Gunka Studios. https://gunkastudios.com/ventajas-y-desventajas-aplicacion-movil/*

*Calvo, L. (2024, 26 noviembre). ¿Qué es una app? Guía sobre aplicaciones móviles. GoDaddy Resources - Spain.* <https://www.godaddy.com/resources/es/tecnologia/que-es-una-app-y-para-que-se-utiliza>

*colaboradores de Wikipedia. (2023, 8 noviembre). Manual de usuario. Wikipedia, la Enciclopedia Libre.* <https://es.wikipedia.org/wiki/Manual_de_usuario>

*Ibm. (2024, 9 mayo). ¿Qué son las pruebas de software y cómo funcionan? | IBM. S.N.* <https://www.ibm.com/mx-es/topics/software-testing>

*Manual Técnico | evedg17. (s. f.). Evedg17.* <https://miyuvampaia17.wixsite.com/evedg17/blank-4>

# Índice De Figuras

[Figura 1 Logo De La Empresa 7](#_9wlu81nopnuf)

[Figura 2 Organigrama De La Empresa 10](#_d6dpeadm3oia)

[Figura 3 Organigrama Del Área Donde Se Realizarán Las Prácticas Profesionales 11](#_ze2rrmc1vp8o)

[Figura 4 Respuestas 1 22](#_z7784rplpuvc)

[Figura 5 Respuestas 2 22](#_1oh1tmejauws)

[Figura 6 Respuestas 3 23](#_xqqknhhth9if)

[Figura 7 Respuestas 4 23](#_gzahwwfunp91)

[Figura 8 Metodología De Desarrollo 26](#_3vcd6t4uwutz)

[Figura 9 Cronograma De Actividades 27](#_592mmxq1w2cf)

[Figura 10 Diagrama De Flujo 30](#_m9ht9zgtf9ss)

[Figura 11 Diagrama De Caso De Uso 31](#_dco304bgiam2)

[Figura 12 Código DashboardScreen 1 35](#_s5sm9e12w3eh)

[Figura 13 DashboardScreen 2 36](#_wuh394hq1ha4)

[Figura 14 DashboardScreen 3 36](#_j7zbtifmt54j)

[Figura 15 TaskhistoryScreen1 37](#_4iu99b5eyd3u)

[Figura 16 TaskhistoryScreen2 37](#_o4k9tzmmr04s)

[Figura 17 TaskhistoryScreen3 38](#_5u8p77o2srg)

[Figura 18 HistoryScreen4 38](#_sipntvtcebyo)

[Figura 19 NavigationScreen1 39](#_m8gno15dixx6)

[Figura 20 MainActivity1 40](#_ei19mfphjpgg)

[Figura 21 Tablero Kanban 2](#_tcx79yeto4sk)

[Figura 22 Inicio De Sesión 3](#_ogkd4ylyle34)

[Figura 23 Dashboard De Tareas 4](#_shk57frvxij0)

[Figura 24 Confirmación De Tarea 5](#_dnjmipagdk31)

[Figura 25 Historial De Tareas 6](#_6zbyzmy8bhgp)

[Figura 26 Calendario 7](#_fp9u0bywl31r)

[Figura 27 Perfil De Usuario 8](#_xtd29sh8umpf)

[Figura 28 Restaurar Contraseña 9](#_rqnkubeuq2c0)

[Figura 29 Nueva Contraseña 9](#_gd32ca1rew3l)

[Figura 30 Base De Datos 10](#_1z9ur0ph38lt)

[Figura 31 Android Studio 11](#_1vgknvtc5gzs)

[Figura 32 Aplicación Instalada 12](#_er9wgmvgg4bi)

[Figura 33 Inicio Sesión Manual Usuario 21](#_pl0wj0y9nfyb)

[Figura 34 Dashboard De Tareas Manual Usuario 22](#_pur6onlfvmff)

[Figura 35 Confirmación De Tareas Manual Usuario 23](#_kese7hqqf6ho)

[Figura 36 Historial De Tareas Manual Usuario 24](#_pabe3n603h90)

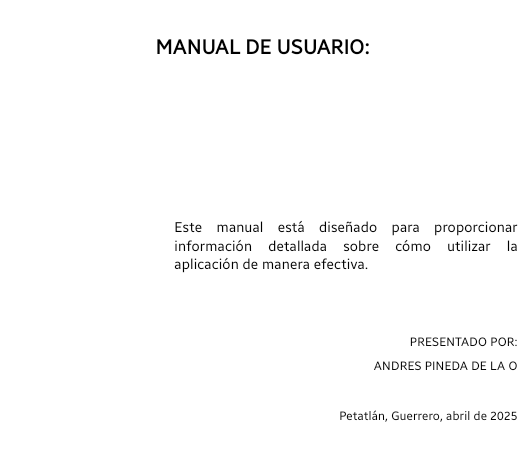
[Figura 37 Calendario De Tareas Manual Usuario 25](#_yh5y8uzdteep)

[Figura 38 Perfil De Usuario Manual Usuario 26](#_k3d669o6dut5)

[Figura 39 Restablecer Contraseña Manual Usuario 27](#_binynbeobbeq)

[Figura 40 Nueva Contraseña Manual Usuario 27](#_ydol9micj8xz)

# Anexos

#### ÍNDICE

Iniciar sesión 3

Tabla de tareas 4

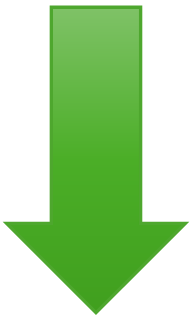
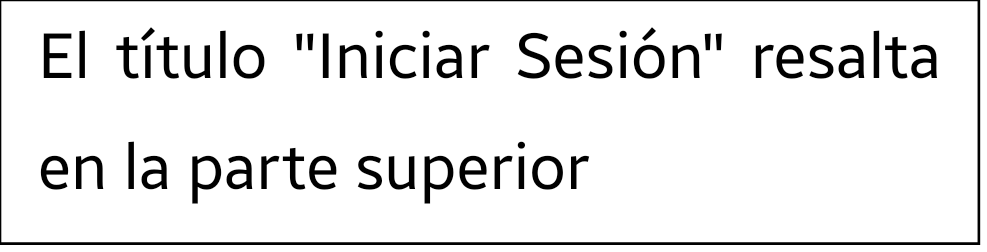
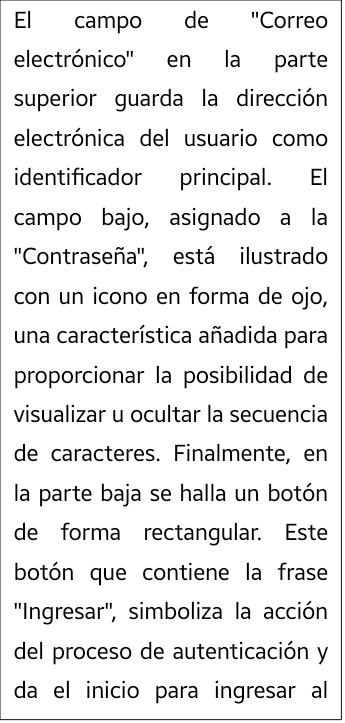
Historial 6

Calendario 7

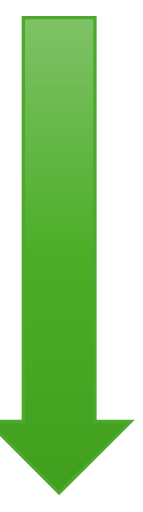
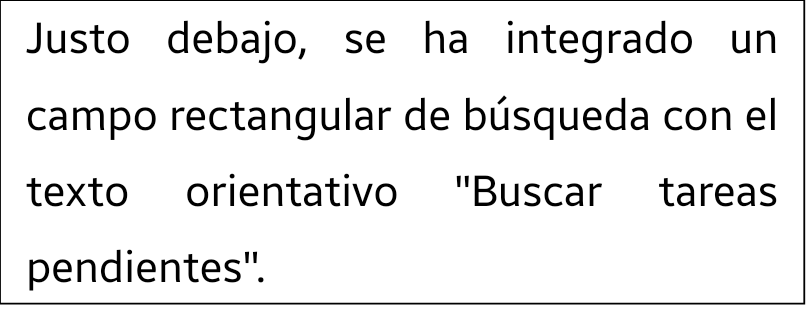
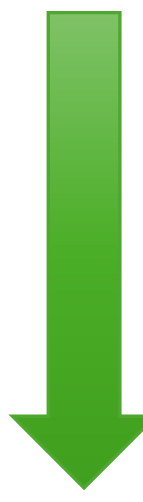
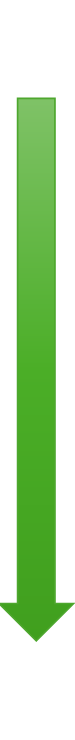
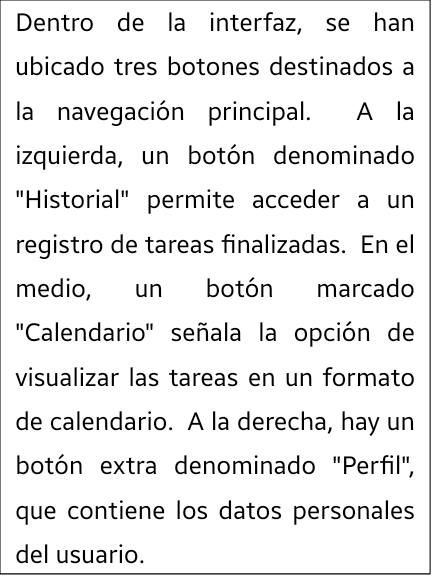
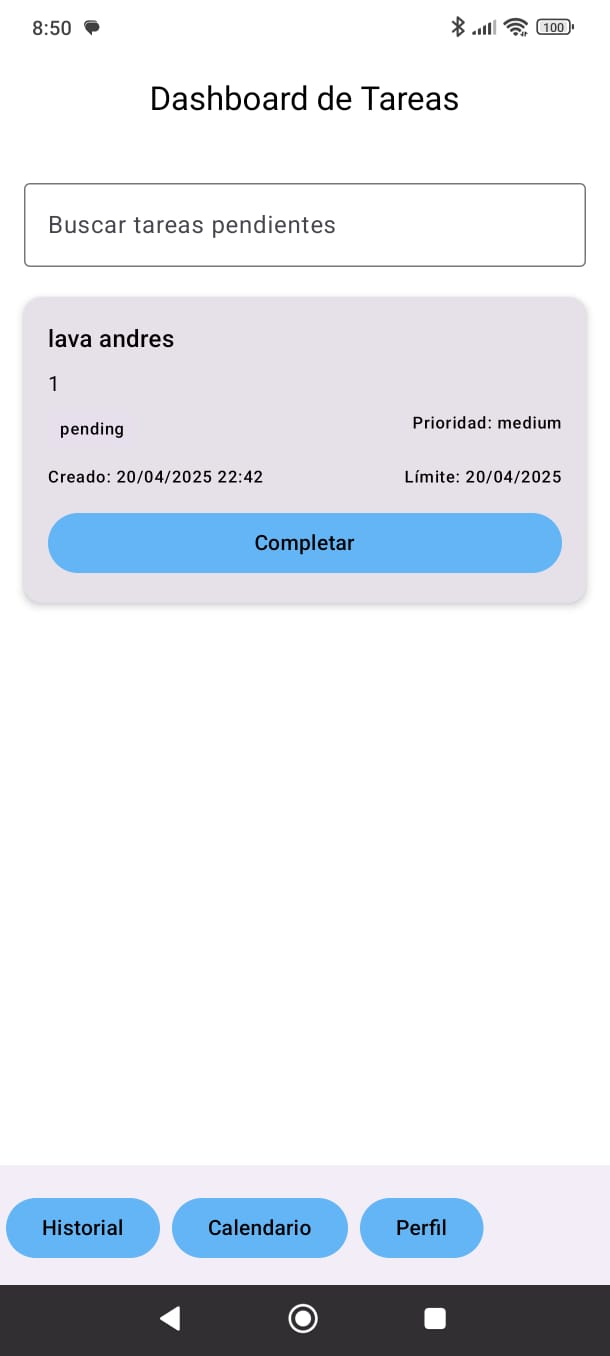
Perfil 8

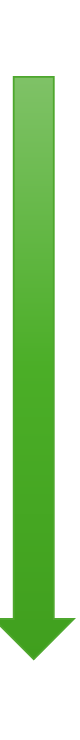
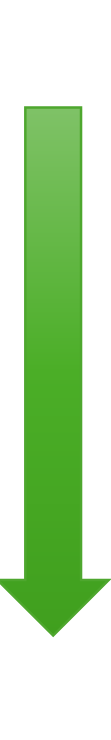
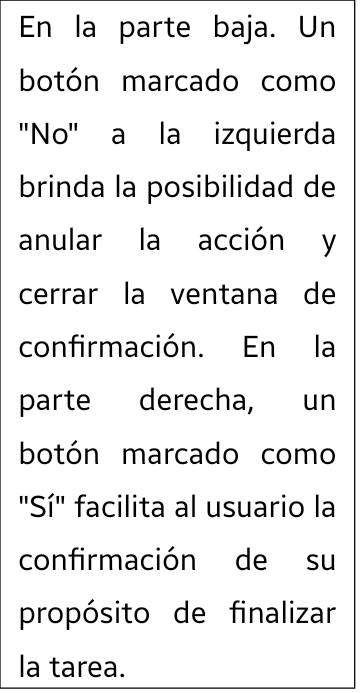
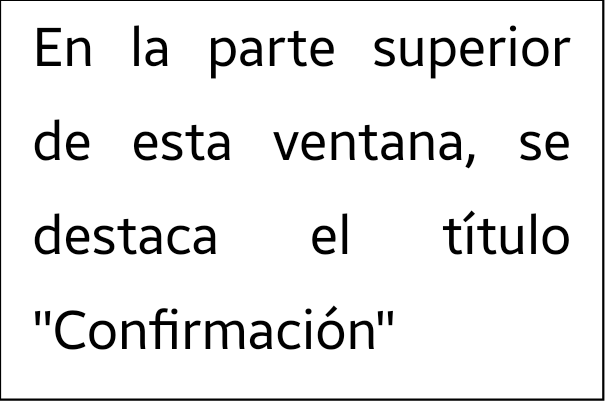
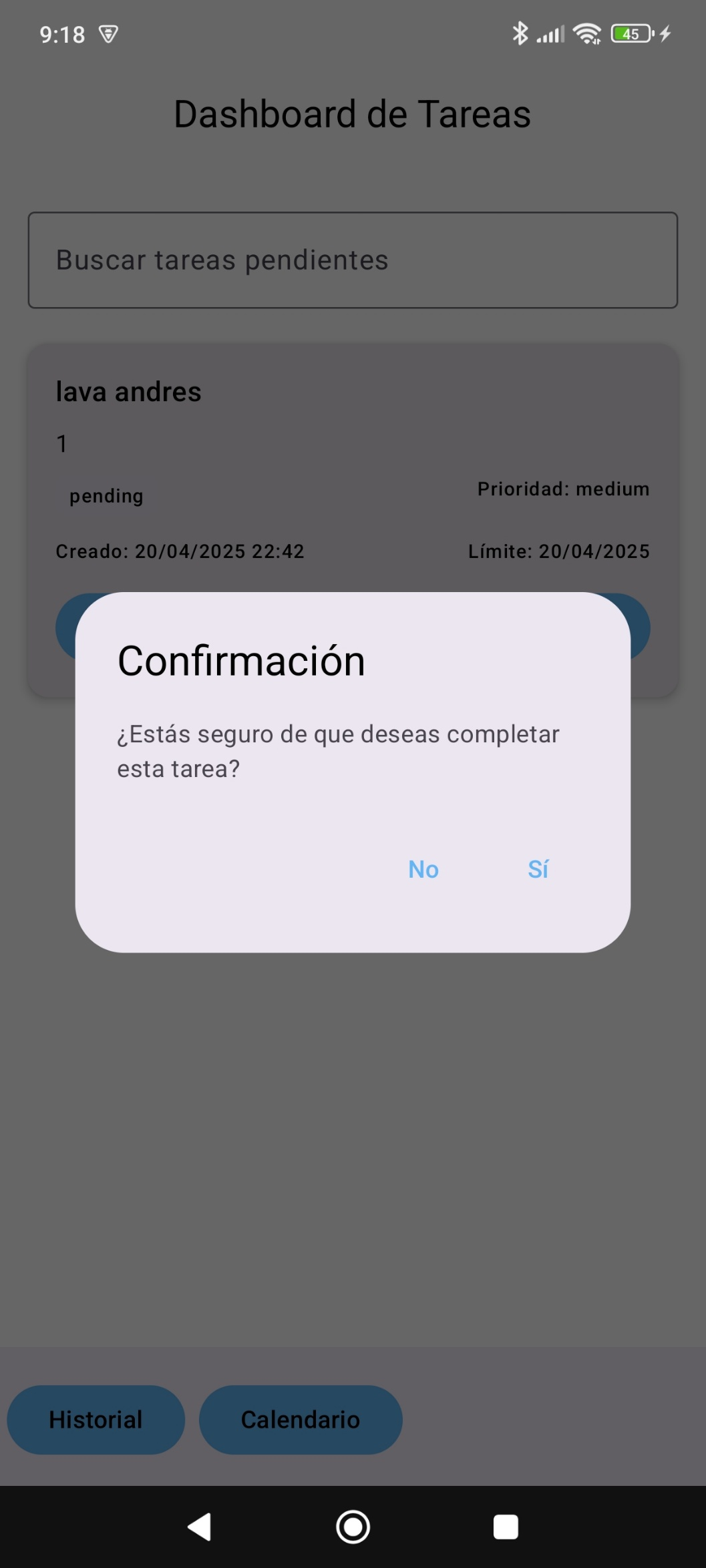
Restablecer contraseña 9

#### Iniciar sesión



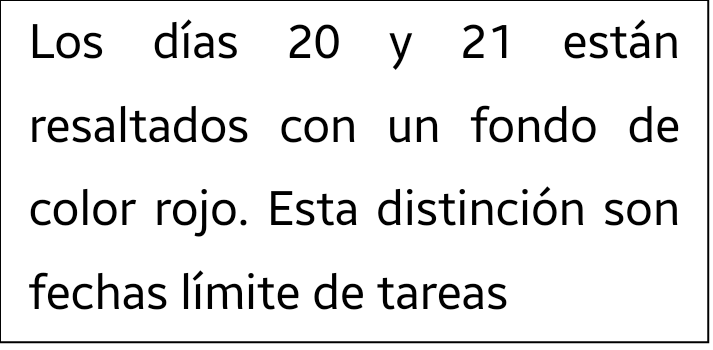
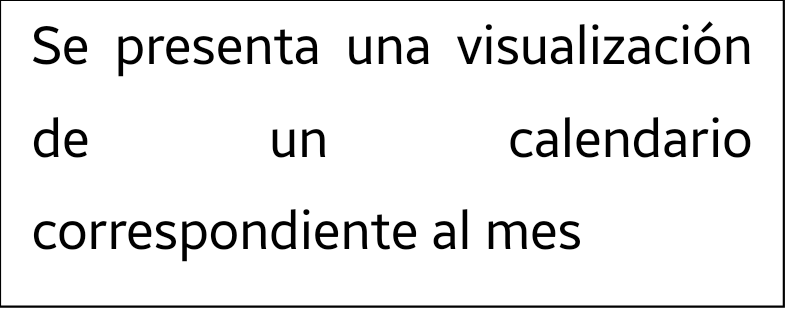
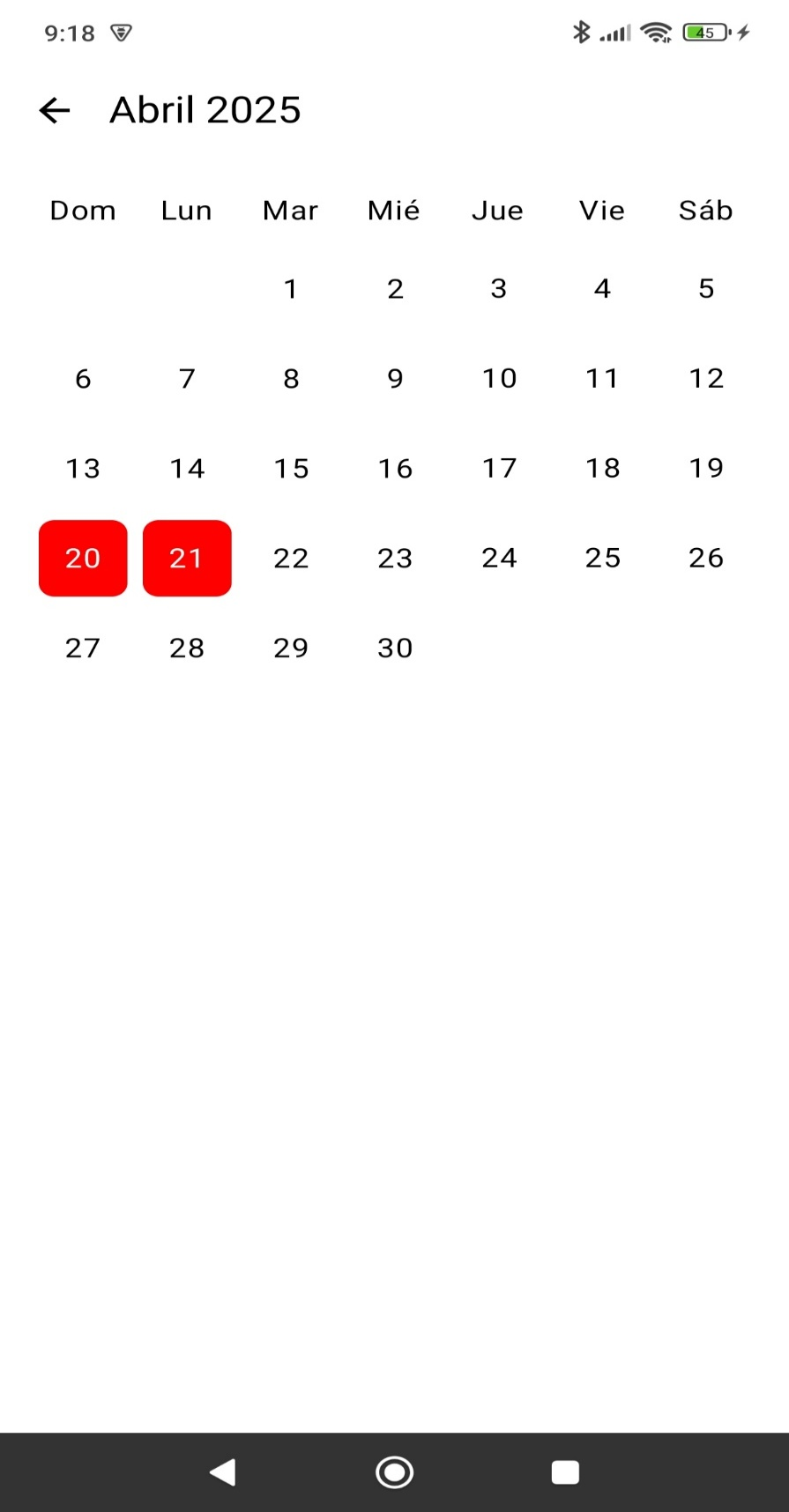
#### Tabla de tareas



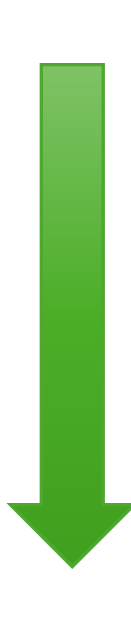
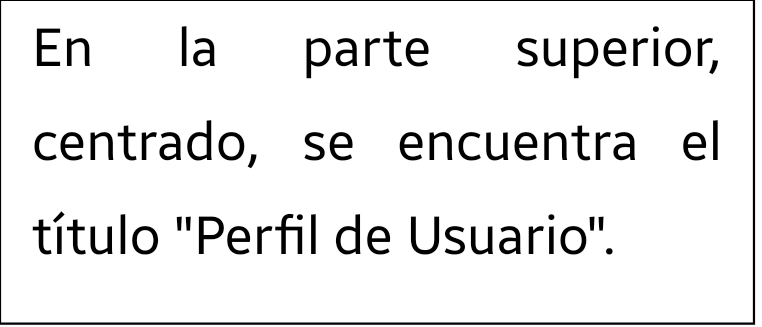
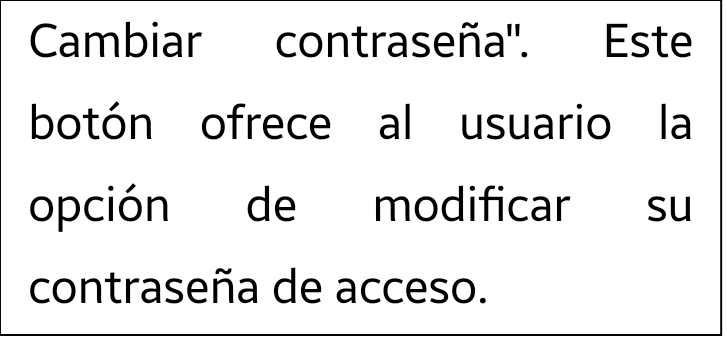
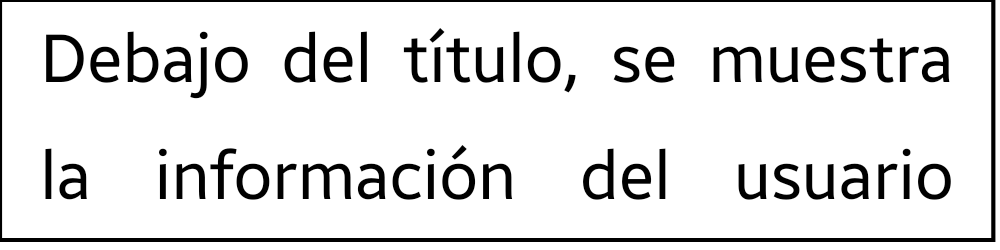
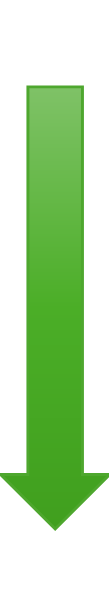
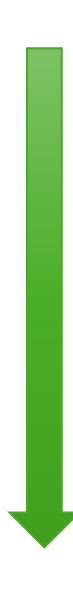


#### Historial

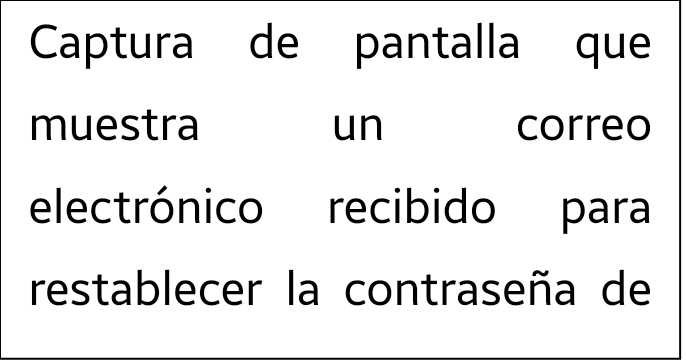
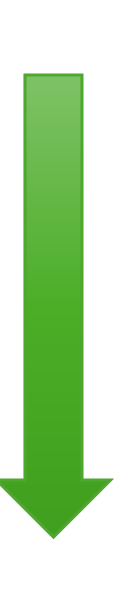
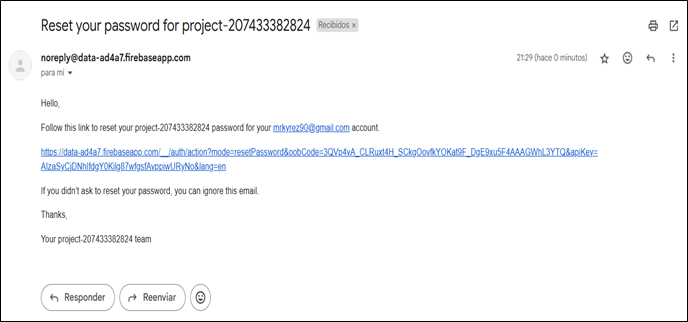
#### Calendario

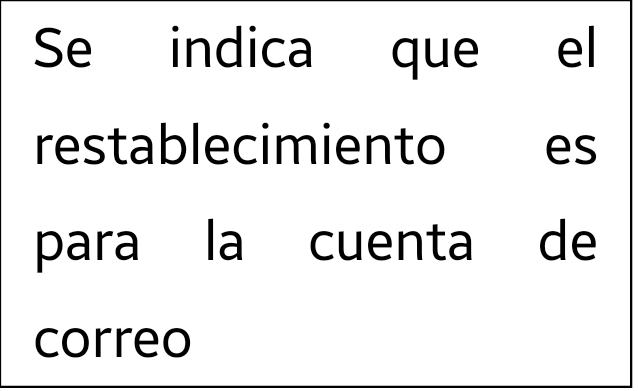
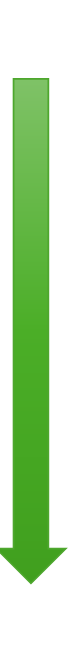
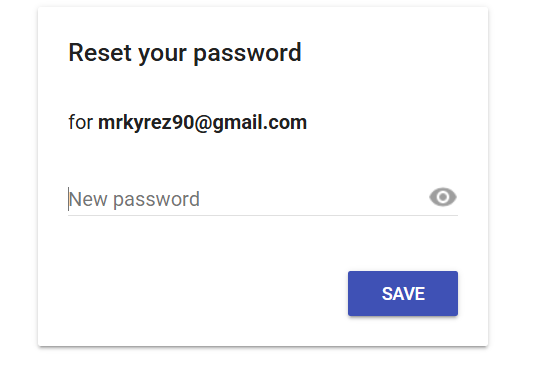


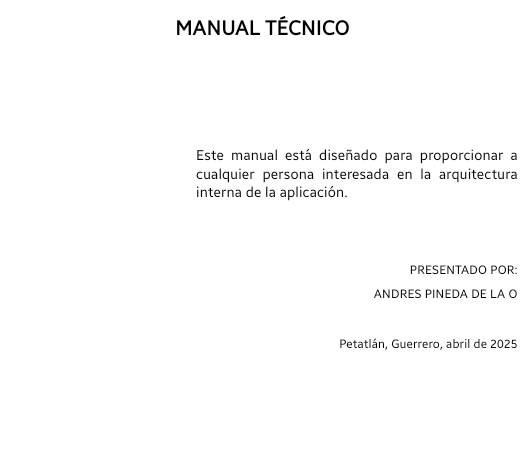
#### Perfil



#### Restablecer contraseña







#### índice

Estructura del proyecto 3

MainActivity 4

LoginScreen.kt 4

DashboardScreen.kt 9

Task.tk 14

TaskItem.kt 16

TaskHistoryScreen.kt 21

Navigation 26

CalendarScreen.kt 28

ProfileScreen.kt 33

#### Estructura del proyecto

#### MainActivity

// MainActivity.kt  
package com.example.trabajadores  
  
import android.os.Bundle  
import androidx.activity.ComponentActivity  
import androidx.activity.compose.setContent  
import androidx.compose.material3.MaterialTheme  
import androidx.compose.material3.Surface  
import androidx.navigation.compose.rememberNavController  
import com.example.trabajadores.ui.theme.TrabajadoresTheme  
  
class MainActivity : ComponentActivity() {  
 override fun onCreate(savedInstanceState: Bundle?) {  
 super.onCreate(savedInstanceState)  
 *setContent* **{** TrabajadoresTheme **{** Surface(color = MaterialTheme.colorScheme.background) **{** val navController = rememberNavController()  
 AppNavigation(navController)  
 **}  
 }  
 }** }  
}

#### LoginScreen.kt

package com.example.trabajadores  
  
import androidx.compose.foundation.background  
import androidx.compose.foundation.layout.\*  
import androidx.compose.material.icons.Icons  
import androidx.compose.material.icons.filled.*Visibility*import androidx.compose.material.icons.filled.*VisibilityOff*import androidx.compose.material3.\*  
import androidx.compose.runtime.\*  
import androidx.compose.ui.Alignment  
import androidx.compose.ui.Modifier  
import androidx.compose.ui.graphics.Color  
import androidx.compose.ui.text.input.PasswordVisualTransformation  
import androidx.compose.ui.text.input.VisualTransformation  
import androidx.compose.ui.unit.dp  
import androidx.navigation.NavController  
import com.google.firebase.auth.FirebaseAuth  
import com.google.firebase.firestore.FirebaseFirestore  
  
@Composable  
fun LoginScreen(navController: NavController) {  
 var email by remember **{** *mutableStateOf*("") **}** var password by remember **{** *mutableStateOf*("") **}** var passwordVisible by remember **{** *mutableStateOf*(false) **}** var errorMessage by remember **{** *mutableStateOf*("") **}** val auth = FirebaseAuth.getInstance()  
  
 Column(  
 modifier = Modifier  
 .*fillMaxSize*()  
 .*background*(Color.White) // Fondo blanco forzado  
 .*padding*(24.*dp*),  
 verticalArrangement = Arrangement.Center,  
 horizontalAlignment = Alignment.CenterHorizontally  
 ) **{** Text("Iniciar Sesión", style = MaterialTheme.typography.headlineMedium)  
  
 Spacer(modifier = Modifier.*height*(32.*dp*))  
  
 OutlinedTextField(  
 value = email,  
 onValueChange = **{** email = **it }**,  
 label = **{** Text("Correo electrónico") **}**,  
 modifier = Modifier.*fillMaxWidth*()  
 )  
  
 Spacer(modifier = Modifier.*height*(16.*dp*))  
  
 OutlinedTextField(  
 value = password,  
 onValueChange = **{** password = **it }**,  
 label = **{** Text("Contraseña") **}**,  
 visualTransformation = if (passwordVisible) VisualTransformation.None else PasswordVisualTransformation(),  
 modifier = Modifier.*fillMaxWidth*(),  
 trailingIcon = **{** IconButton(onClick = **{** passwordVisible = !passwordVisible **}**) **{** Icon(  
 imageVector = if (passwordVisible) Icons.Filled.*Visibility* else Icons.Filled.*VisibilityOff*,  
 contentDescription = null  
 )  
 **}  
 }** )  
  
 if (errorMessage.*isNotEmpty*()) {  
 Spacer(modifier = Modifier.*height*(8.*dp*))  
 Text(errorMessage, color = MaterialTheme.colorScheme.error)  
 }  
  
 Spacer(modifier = Modifier.*height*(24.*dp*))  
  
 Button(  
 onClick = **{** if (email.*isBlank*() || password.*isBlank*()) {  
 errorMessage = "Completa todos los campos"  
 } else {  
 errorMessage = ""  
 auth.signInWithEmailAndPassword(email, password)  
 .addOnCompleteListener **{** task **->** if (task.*isSuccessful*) {  
 val userId = auth.*currentUser*?.*uid* if (userId != null) {  
 val db = FirebaseFirestore.getInstance()  
 db.collection("users").document(userId).get()  
 .addOnSuccessListener **{** document **->** val role = document.getString("role")  
 if (role == "employee") {  
 navController.navigate("dashboard")  
 } else {  
 errorMessage = "No tienes permisos para ingresar"  
 auth.signOut()  
 }  
 **}** .addOnFailureListener **{** errorMessage = "Error al verificar el rol"  
 **}** } else {  
 errorMessage = "No se pudo obtener el usuario"  
 }  
 } else {  
 errorMessage = "Correo o contraseña incorrectos"  
 }  
 **}** }  
 **}**,  
 modifier = Modifier.*fillMaxWidth*()  
 ) **{** Text("Ingresar")  
 **}  
  
 }**}

#### DashboardScreen.kt

package com.example.trabajadores  
  
import androidx.compose.foundation.layout.\*  
import androidx.compose.foundation.lazy.LazyColumn  
import androidx.compose.foundation.lazy.items  
import androidx.compose.material3.\*  
import androidx.compose.runtime.\*  
import androidx.compose.ui.Alignment  
import androidx.compose.ui.Modifier  
import androidx.compose.ui.unit.dp  
import com.google.firebase.firestore.FirebaseFirestore  
import com.google.firebase.auth.FirebaseAuth  
  
@OptIn(ExperimentalMaterial3Api::class)  
@Composable  
fun DashboardScreen(  
 navigateToHistory: () -> Unit,  
 navigateToCalendar: () -> Unit,  
 navigateToProfile: () -> Unit // ← Se agrega este parámetro  
) {  
 var taskList by remember **{** *mutableStateOf*<List<Task>>(*emptyList*()) **}** var filteredTasks by remember **{** *mutableStateOf*<List<Task>>(*emptyList*()) **}** var isLoading by remember **{** *mutableStateOf*(true) **}** var errorMessage by remember **{** *mutableStateOf*<String?>(null) **}** var searchQuery by remember **{** *mutableStateOf*("") **}** fun applyFilters() {  
 filteredTasks = taskList.*filter* **{** task **->** task.status == "pending" && (  
 searchQuery.*isBlank*() ||  
 task.title.*contains*(searchQuery, ignoreCase = true) ||  
 task.description.*contains*(searchQuery, ignoreCase = true)  
 )  
 **}** }  
  
 fun loadTasks() {  
 val currentUser = FirebaseAuth.getInstance().*currentUser* if (currentUser != null) {  
 isLoading = true  
 FirebaseFirestore.getInstance()  
 .collection("tasks")  
 .whereEqualTo("assigneeId", currentUser.*uid*)  
 .get()  
 .addOnSuccessListener **{** result **->** taskList = result.*mapNotNull* **{** doc **->** try {  
 doc.toObject(Task::class.*java*)?.copy(id = doc.*id*)  
 } catch (e: Exception) {  
 null  
 }  
 **}** applyFilters()  
 isLoading = false  
 **}** .addOnFailureListener **{** e **->** errorMessage = "Error al cargar tareas: ${e.*localizedMessage*}"  
 isLoading = false  
 **}** } else {  
 errorMessage = "Usuario no autenticado"  
 isLoading = false  
 }  
 }  
  
 fun updateTaskStatus(task: Task) {  
 FirebaseFirestore.getInstance()  
 .collection("tasks")  
 .document(task.id)  
 .update("status", task.status, "completedAt", task.completedAt)  
 .addOnSuccessListener **{** loadTasks() **}** }  
  
 LaunchedEffect(Unit) **{** loadTasks()  
 **}** Scaffold(  
 topBar = **{** CenterAlignedTopAppBar(title = **{** Text("Dashboard de Tareas") **}**)  
 **}**,  
 bottomBar = **{** BottomAppBar **{** Button(onClick = navigateToHistory) **{** Text("Historial")  
 **}** Spacer(modifier = Modifier.*width*(8.*dp*))  
 Button(onClick = navigateToCalendar) **{** Text("Calendario")  
 **}** Spacer(modifier = Modifier.*width*(8.*dp*))  
 Button(onClick = navigateToProfile) **{** // ← Botón nuevo  
 Text("Perfil")  
 **}  
 }  
 }** ) **{** paddingValues **->** Column(  
 modifier = Modifier  
 .*padding*(paddingValues)  
 .*padding*(16.*dp*)  
 .*fillMaxSize*()  
 ) **{** OutlinedTextField(  
 value = searchQuery,  
 onValueChange = **{** searchQuery = **it** applyFilters()  
 **}**,  
 label = **{** Text("Buscar tareas pendientes") **}**,  
 modifier = Modifier.*fillMaxWidth*()  
 )  
  
 Spacer(modifier = Modifier.*height*(12.*dp*))  
  
 if (isLoading) {  
 Box(Modifier.*fillMaxSize*(), contentAlignment = Alignment.Center) **{** CircularProgressIndicator()  
 **}** } else if (errorMessage != null) {  
 Text("Error: $errorMessage", color = MaterialTheme.colorScheme.error)  
 } else if (filteredTasks.isEmpty()) {  
 Text("No hay tareas pendientes con esos criterios")  
 } else {  
 LazyColumn **{** *items*(filteredTasks) **{** task **->** TaskItem(task = task, onStatusChange = **{** updateTaskStatus(**it**) **}**)  
 **}  
 }** }  
 **}  
 }**}

#### Task.tk

package com.example.trabajadores  
  
import com.google.firebase.Timestamp  
import java.text.SimpleDateFormat  
import java.util.\*  
  
data class Task(  
 val id: String = "",  
 val assigneeId: String = "",  
 val createdAt: Timestamp = Timestamp.now(),  
 val createdBy: String = "",  
 val deadline: Timestamp? = null,  
 val description: String = "",  
 val priority: String = "",  
 val status: String = "pending",  
 val title: String = "",  
 val completedAt: Timestamp? = null  
) {  
 fun getFormattedCreatedAt(): String {  
 return try {  
 SimpleDateFormat("dd/MM/yyyy HH:mm", Locale.getDefault())  
 .format(createdAt.toDate())  
 } catch (e: Exception) {  
 "Fecha inválida"  
 }  
 }  
  
 fun getFormattedDeadline(): String {  
 return deadline?.*let* **{** try {  
 SimpleDateFormat("dd/MM/yyyy", Locale.getDefault())  
 .format(**it**.toDate())  
 } catch (e: Exception) {  
 "Fecha inválida"  
 }  
 **}** ?: "Sin fecha límite"  
 }  
  
 fun getFormattedCompletedAt(): String {  
 return completedAt?.*let* **{** try {  
 SimpleDateFormat("dd/MM/yyyy HH:mm", Locale.getDefault())  
 .format(**it**.toDate())  
 } catch (e: Exception) {  
 "Fecha inválida"  
 }  
 **}** ?: "Sin fecha de finalización"  
 }  
}

#### TaskItem.kt

package com.example.trabajadores  
  
import android.widget.Toast  
import androidx.compose.foundation.background  
import androidx.compose.foundation.layout.\*  
import androidx.compose.material3.\*  
import androidx.compose.runtime.\*  
import androidx.compose.ui.Alignment  
import androidx.compose.ui.Modifier  
import androidx.compose.ui.graphics.Color  
import androidx.compose.ui.unit.dp  
import com.google.firebase.Timestamp  
import java.text.SimpleDateFormat  
import java.util.\*  
  
@OptIn(ExperimentalMaterial3Api::class)  
@Composable  
fun TaskItem(task: Task, onStatusChange: (Task) -> Unit = **{}**) {  
 var taskStatus by remember **{** *mutableStateOf*(task.status) **}** var showDialog by remember **{** *mutableStateOf*(false) **}** if (showDialog) {  
 AlertDialog(  
 onDismissRequest = **{** showDialog = false **}**,  
 title = **{** Text("Confirmación") **}**,  
 text = **{** Text("¿Estás seguro de que deseas completar esta tarea?") **}**,  
 confirmButton = **{** TextButton(  
 onClick = **{** taskStatus = "complete"  
 val updatedTask = task.copy(  
 status = taskStatus,  
 completedAt = Timestamp.now()  
 )  
 onStatusChange(updatedTask)  
 showDialog = false  
 **}** ) **{** Text("Sí")  
 **}  
 }**,  
 dismissButton = **{** TextButton(onClick = **{** showDialog = false **}**) **{** Text("No")  
 **}  
 }** )  
 }  
  
 Card(  
 modifier = Modifier  
 .*fillMaxWidth*()  
 .*padding*(vertical = 8.*dp*),  
 elevation = CardDefaults.cardElevation(defaultElevation = 4.*dp*)  
 ) **{** Column(modifier = Modifier.*padding*(16.*dp*)) **{** Text(  
 text = task.title,  
 style = MaterialTheme.typography.titleMedium  
 )  
  
 Spacer(modifier = Modifier.*height*(8.*dp*))  
  
 if (task.description.*isNotEmpty*()) {  
 Text(  
 text = task.description,  
 style = MaterialTheme.typography.bodyMedium  
 )  
 Spacer(modifier = Modifier.*height*(8.*dp*))  
 }  
  
 Row(  
 modifier = Modifier.*fillMaxWidth*(),  
 horizontalArrangement = Arrangement.SpaceBetween  
 ) **{** Box(  
 contentAlignment = Alignment.Center,  
 modifier = Modifier  
 .*background*(  
 color = when (taskStatus.*lowercase*(Locale.getDefault())) {  
 "complete" -> MaterialTheme.colorScheme.primaryContainer  
 else -> MaterialTheme.colorScheme.surfaceVariant  
 },  
 shape = MaterialTheme.shapes.small  
 )  
 .*padding*(horizontal = 8.*dp*, vertical = 4.*dp*)  
 ) **{** Text(  
 text = taskStatus,  
 style = MaterialTheme.typography.labelSmall  
 )  
 **}** Text(  
 text = "Prioridad: ${task.priority}",  
 style = MaterialTheme.typography.labelSmall  
 )  
 **}** Spacer(modifier = Modifier.*height*(12.*dp*))  
  
 Row(  
 modifier = Modifier.*fillMaxWidth*(),  
 horizontalArrangement = Arrangement.SpaceBetween  
 ) **{** Text(  
 text = "Creado: ${task.getFormattedCreatedAt()}",  
 style = MaterialTheme.typography.labelSmall  
 )  
 Text(  
 text = "Límite: ${task.getFormattedDeadline()}",  
 style = MaterialTheme.typography.labelSmall  
 )  
 **}** Spacer(modifier = Modifier.*height*(12.*dp*))  
  
 if (task.status != "complete") {  
 Button(  
 onClick = **{** showDialog = true **}**,  
 modifier = Modifier.*fillMaxWidth*()  
 ) **{** Text("Completar")  
 **}** } else {  
 task.completedAt?.*let* **{** Text(  
 text = "Completada: ${SimpleDateFormat("dd/MM/yyyy HH:mm", Locale.getDefault()).format(**it**.toDate())}",  
 style = MaterialTheme.typography.labelSmall  
 )  
 **}** }  
 **}  
 }**}

#### TaskHistoryScreen.kt

package com.example.trabajadores  
  
import androidx.compose.foundation.layout.\*  
import androidx.compose.foundation.lazy.LazyColumn  
import androidx.compose.foundation.lazy.items  
import androidx.compose.material.icons.Icons  
import androidx.compose.material.icons.automirrored.filled.*ArrowBack*import androidx.compose.material.icons.filled.*ArrowBack*import androidx.compose.material3.\*  
import androidx.compose.runtime.\*  
import androidx.compose.ui.Alignment  
import androidx.compose.ui.Modifier  
import androidx.compose.ui.unit.dp  
import androidx.navigation.NavController  
import com.google.firebase.firestore.FirebaseFirestore  
import com.google.firebase.auth.FirebaseAuth  
import com.google.firebase.firestore.ktx.toObject  
  
@OptIn(ExperimentalMaterial3Api::class)  
@Composable  
fun TaskHistoryScreen(navController: NavController) {  
 var taskList by remember **{** *mutableStateOf*<List<Task>>(*emptyList*()) **}** var isLoading by remember **{** *mutableStateOf*(true) **}** var errorMessage by remember **{** *mutableStateOf*<String?>(null) **}** fun loadCompletedTasks() {  
 val currentUser = FirebaseAuth.getInstance().*currentUser* if (currentUser != null) {  
 FirebaseFirestore.getInstance()  
 .collection("tasks")  
 .whereEqualTo("assigneeId", currentUser.*uid*)  
 .whereEqualTo("status", "complete")  
 .get()  
 .addOnSuccessListener **{** result **->** val tasks = result.*mapNotNull* **{** document **->** try {  
 document.*toObject*<Task>().copy(id = document.*id*)  
 } catch (e: Exception) {  
 null  
 }  
 **}** taskList = tasks.*sortedByDescending* **{ it**.completedAt **}** isLoading = false  
 **}** .addOnFailureListener **{** e **->** errorMessage = "Error al cargar tareas: ${e.*localizedMessage*}"  
 isLoading = false  
 **}** } else {  
 errorMessage = "Usuario no autenticado"  
 isLoading = false  
 }  
 }  
  
 LaunchedEffect(Unit) **{** loadCompletedTasks()  
 **}** Scaffold(  
 topBar = **{** TopAppBar(  
 title = **{** Text("Historial de Tareas") **}**,  
 navigationIcon = **{** IconButton(onClick = **{** navController.popBackStack() **}**) **{** Icon(  
 imageVector = Icons.Default.*ArrowBack*,  
 contentDescription = "Volver"  
 )  
 **}  
 }**,  
 colors = TopAppBarDefaults.centerAlignedTopAppBarColors(  
 containerColor = MaterialTheme.colorScheme.primaryContainer,  
 titleContentColor = MaterialTheme.colorScheme.onPrimaryContainer  
 )  
 )  
 **}** ) **{** paddingValues **->** Box(modifier = Modifier  
 .*fillMaxSize*()  
 .*padding*(paddingValues)) **{** when {  
 isLoading -> {  
 CircularProgressIndicator(modifier = Modifier.*align*(Alignment.Center))  
 }  
 errorMessage != null -> {  
 Column(  
 modifier = Modifier.*align*(Alignment.Center),  
 horizontalAlignment = Alignment.CenterHorizontally  
 ) **{** Text(  
 text = errorMessage ?: "Error desconocido",  
 color = MaterialTheme.colorScheme.error  
 )  
 Spacer(modifier = Modifier.*height*(16.*dp*))  
 Button(onClick = **{** isLoading = true  
 errorMessage = null  
 loadCompletedTasks()  
 **}**) **{** Text("Reintentar")  
 **}  
 }** }  
 taskList.isEmpty() -> {  
 Text(  
 text = "No hay tareas completadas",  
 modifier = Modifier.*align*(Alignment.Center)  
 )  
 }  
 else -> {  
 LazyColumn(  
 modifier = Modifier  
 .*fillMaxSize*()  
 .*padding*(horizontal = 16.*dp*)  
 ) **{** *items*(taskList) **{** task **->** Card(  
 modifier = Modifier  
 .*fillMaxWidth*()  
 .*padding*(vertical = 8.*dp*),  
 elevation = CardDefaults.cardElevation(defaultElevation = 4.*dp*),  
 colors = CardDefaults.cardColors(  
 containerColor = MaterialTheme.colorScheme.surfaceVariant  
 )  
 ) **{** Column(modifier = Modifier.*padding*(16.*dp*)) **{** Text(  
 text = task.title,  
 style = MaterialTheme.typography.titleMedium  
 )  
 Spacer(modifier = Modifier.*height*(4.*dp*))  
 Text(  
 text = task.description,  
 style = MaterialTheme.typography.bodyMedium  
 )  
 Spacer(modifier = Modifier.*height*(8.*dp*))  
 Text(  
 text = "Finalizada: ${task.getFormattedCompletedAt()}",  
 style = MaterialTheme.typography.labelSmall  
 )  
 **}  
 }  
 }  
 }** }  
 }  
 **}  
 }**}

#### Navigation

package com.example.trabajadores  
  
import androidx.compose.runtime.Composable  
import androidx.navigation.NavHostController  
import androidx.navigation.compose.NavHost  
import androidx.navigation.compose.composable  
import com.example.trabajadores.ui.screen.ProfileScreen  
  
@Composable  
fun AppNavigation(navController: NavHostController) {  
 NavHost(navController = navController, startDestination = "login") **{** *composable*("login") **{** LoginScreen(navController)  
 **}** *composable*("dashboard") **{** DashboardScreen(  
 navigateToHistory = **{** navController.navigate("task\_history") **}**,  
 navigateToCalendar = **{** navController.navigate("calendar") **}**,  
 navigateToProfile = **{** navController.navigate("profile") **}** // ← Nuevo  
 )  
 **}** *composable*("task\_history") **{** TaskHistoryScreen(navController)  
 **}** *composable*("calendar") **{** CalendarScreen(  
 navigateBackToDashboard = **{** navController.popBackStack() **}** )  
 **}** *composable*("profile") **{** ProfileScreen(  
 navigateBack = **{** navController.popBackStack() **}** // ← Por si quieres volver al dashboard  
 )  
 **}  
 }**}

#### CalendarScreen.kt

package com.example.trabajadores  
  
import androidx.compose.foundation.background  
import androidx.compose.foundation.layout.\*  
import androidx.compose.foundation.lazy.grid.GridCells  
import androidx.compose.foundation.lazy.grid.LazyVerticalGrid  
import androidx.compose.foundation.lazy.grid.items  
import androidx.compose.material.icons.Icons  
import androidx.compose.material.icons.filled.*ArrowBack*import androidx.compose.material3.\*  
import androidx.compose.runtime.\*  
import androidx.compose.ui.Alignment  
import androidx.compose.ui.Modifier  
import androidx.compose.ui.graphics.Color  
import androidx.compose.ui.text.style.TextAlign  
import androidx.compose.ui.unit.dp  
import com.google.firebase.auth.FirebaseAuth  
import com.google.firebase.firestore.FirebaseFirestore  
import com.google.firebase.firestore.ktx.toObject  
import java.text.SimpleDateFormat  
import java.util.\*  
  
@OptIn(ExperimentalMaterial3Api::class)  
@Composable  
fun CalendarScreen(navigateBackToDashboard: () -> Unit) {  
 val today = remember **{** Calendar.getInstance() **}** val currentYear = today.get(Calendar.*YEAR*)  
 val currentMonth = today.get(Calendar.*MONTH*) // Enero = 0  
  
 var deadlineDays by remember **{** *mutableStateOf*(*setOf*<Int>()) **}** // Cargar fechas límite desde Firebase  
 LaunchedEffect(Unit) **{** val currentUser = FirebaseAuth.getInstance().*currentUser* ?: return@LaunchedEffect  
 val db = FirebaseFirestore.getInstance()  
  
 db.collection("tasks")  
 .whereEqualTo("assigneeId", currentUser.*uid*)  
 .get()  
 .addOnSuccessListener **{** result **->** val deadlines = result.*mapNotNull* **{ it**.*toObject*<Task>().deadline?.toDate() **}** .*filter* **{** val cal = Calendar.getInstance()  
 cal.*time* = **it** cal.get(Calendar.*MONTH*) == currentMonth && cal.get(Calendar.*YEAR*) == currentYear  
 **}** .*map* **{** val cal = Calendar.getInstance()  
 cal.*time* = **it** cal.get(Calendar.*DAY\_OF\_MONTH*)  
 **}** .*toSet*()  
  
 deadlineDays = deadlines  
 **}  
 }** Scaffold(  
 topBar = **{** TopAppBar(  
 title = **{** val monthFormat = SimpleDateFormat("MMMM yyyy", Locale.getDefault())  
 Text(text = monthFormat.format(today.*time*).*replaceFirstChar* **{ it**.*uppercase*() **}**)  
 **}**,  
 navigationIcon = **{** IconButton(onClick = **{** navigateBackToDashboard() **}**) **{** Icon(Icons.Default.*ArrowBack*, contentDescription = "Volver")  
 **}  
 }** )  
 **}** ) **{** paddingValues **->** Column(  
 modifier = Modifier  
 .*padding*(paddingValues)  
 .*padding*(16.*dp*)  
 .*fillMaxSize*()  
 ) **{** // Nombres de días  
 val daysOfWeek = *listOf*("Dom", "Lun", "Mar", "Mié", "Jue", "Vie", "Sáb")  
 Row(  
 Modifier.*fillMaxWidth*(),  
 horizontalArrangement = Arrangement.SpaceBetween  
 ) **{** daysOfWeek.*forEach* **{** Text(  
 text = **it**,  
 modifier = Modifier.*weight*(1f),  
 textAlign = TextAlign.Center  
 )  
 **}  
 }** Spacer(modifier = Modifier.*height*(8.*dp*))  
  
 // Mostrar días del mes  
 val calendar = Calendar.getInstance()  
 calendar.set(Calendar.*YEAR*, currentYear)  
 calendar.set(Calendar.*MONTH*, currentMonth)  
 calendar.set(Calendar.*DAY\_OF\_MONTH*, 1)  
 val startDayOfWeek = calendar.get(Calendar.*DAY\_OF\_WEEK*) - 1 // para ajustar al índice (Domingo = 1)  
 val daysInMonth = calendar.getActualMaximum(Calendar.*DAY\_OF\_MONTH*)  
  
 val totalCells = startDayOfWeek + daysInMonth  
 val dayNumbers = (1..totalCells).*map* **{** index **->** if (index <= startDayOfWeek) null else index - startDayOfWeek  
 **}** LazyVerticalGrid(  
 columns = GridCells.Fixed(7),  
 modifier = Modifier.*fillMaxSize*(),  
 horizontalArrangement = Arrangement.Center  
 ) **{** *items*(dayNumbers) **{** day **->** Box(  
 modifier = Modifier  
 .*aspectRatio*(1f)  
 .*padding*(4.*dp*)  
 .*background*(  
 color = if (day != null && day in deadlineDays) Color.Red else Color.Transparent,  
 shape = MaterialTheme.shapes.small  
 ),  
 contentAlignment = Alignment.Center  
 ) **{** Text(  
 text = day?.toString() ?: "",  
 color = if (day != null && day in deadlineDays) Color.White else Color.Unspecified  
 )  
 **}  
 }  
 }  
 }  
 }**}

#### ProfileScreen.kt

package com.example.trabajadores.ui.screen  
  
import androidx.compose.foundation.layout.\*  
import androidx.compose.material3.\*  
import androidx.compose.runtime.\*  
import androidx.compose.ui.Alignment  
import androidx.compose.ui.Modifier  
import androidx.compose.ui.unit.dp  
import com.google.firebase.auth.FirebaseAuth  
import com.google.firebase.firestore.FirebaseFirestore  
  
data class UserProfile(  
 val email: String = "",  
 val name: String = ""  
)  
  
@OptIn(ExperimentalMaterial3Api::class)  
@Composable  
fun ProfileScreen(navigateBack: () -> Boolean) {  
 var userProfile by remember **{** *mutableStateOf*<UserProfile?>(null) **}** var isLoading by remember **{** *mutableStateOf*(true) **}** var errorMessage by remember **{** *mutableStateOf*<String?>(null) **}** var successMessage by remember **{** *mutableStateOf*<String?>(null) **}** val currentUser = FirebaseAuth.getInstance().*currentUser* val userId = currentUser?.*uid* LaunchedEffect(userId) **{** if (userId != null) {  
 FirebaseFirestore.getInstance()  
 .collection("users")  
 .document(userId)  
 .get()  
 .addOnSuccessListener **{** doc **->** if (doc.exists()) {  
 userProfile = doc.toObject(UserProfile::class.*java*)  
 isLoading = false  
 } else {  
 errorMessage = "Usuario no encontrado"  
 isLoading = false  
 }  
 **}** .addOnFailureListener **{** errorMessage = "Error al cargar perfil: ${**it**.*localizedMessage*}"  
 isLoading = false  
 **}** } else {  
 errorMessage = "Usuario no autenticado"  
 isLoading = false  
 }  
 **}** Scaffold(  
 topBar = **{** CenterAlignedTopAppBar(title = **{** Text("Perfil de Usuario") **}**)  
 **}** ) **{** paddingValues **->** Box(  
 modifier = Modifier  
 .*padding*(paddingValues)  
 .*fillMaxSize*(),  
 contentAlignment = Alignment.Center  
 ) **{** when {  
 isLoading -> CircularProgressIndicator()  
  
 errorMessage != null -> Text(errorMessage ?: "", color = MaterialTheme.colorScheme.error)  
  
 userProfile != null -> {  
 Column(  
 modifier = Modifier  
 .*fillMaxWidth*()  
 .*padding*(24.*dp*),  
 verticalArrangement = Arrangement.spacedBy(16.*dp*)  
 ) **{** Text("📧 Correo: ${userProfile!!.email}")  
 Text("👤 Nombre: ${userProfile!!.name}")  
  
 Button(onClick = **{** currentUser?.*email*?.*let* **{** email **->** FirebaseAuth.getInstance()  
 .sendPasswordResetEmail(email)  
 .addOnSuccessListener **{** successMessage = "Se envió un correo para restablecer la contraseña."  
 **}** .addOnFailureListener **{** errorMessage = "Error al enviar correo: ${**it**.*localizedMessage*}"  
 successMessage = null  
 **}  
 }  
 }**) **{** Text("Cambiar contraseña")  
 **}** successMessage?.*let* **{** Text(**it**, color = MaterialTheme.colorScheme.primary)  
 **}  
 }** }  
 }  
 **}  
 }**}