



Parcial II: Aplicaciones Móviles
Noviembre 2025

Realizado por: *Carelis Fagundez Frías*
DNI: 19.094.687

MisQuehaceres- App Mobile

Carrera: Análisis de Sistemas
Comisión: ACN3BV
Profesor: Sergio Medina

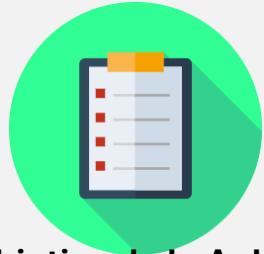




MisQuehaceres - Presentación

Este documento describe el funcionamiento, diseño, flujo de uso e implementación técnica de la aplicación móvil **MisQuehaceresApp**, desarrollada en Android Studio utilizando Java, XML y Firebase Firestore.

La aplicación permite gestionar tareas, organizarlas según su estado y mantener la información sincronizada en la nube.



MisQuehaceres - Funcionalidades

2. Objetivo de la Aplicación

El objetivo de MisQuehaceresApp es brindar una herramienta simple y ordenada para gestionar quehaceres diarios. La aplicación busca que el usuario pueda:

Crear una nueva tarea

Cambiar su estado (pendiente, prioridad, terminada)

Eliminar tareas

Mantener todas las tareas guardadas aun cuando cierre la app

Navegar entre pantallas de forma intuitiva.

3. HomeActivity (Pantalla de Inicio)

Componentes:

Título de bienvenida

Botón “Ir a Mis Tareas”

Botón “Acerca de la App”

Funcionalidad:

Es la pantalla inicial de la aplicación. Permite navegar a la gestión de tareas o acceder a la sección informativa.

Flujo:

El usuario abre la app

Selecciona entre:

Ver tareas (MainActivity)

Ver información (AcercaActivity)



MisQuehaceres - Funcionalidades

MainActivity (Gestión de Tareas)

Componentes:

EditText para ingresar una tarea

Botón “Agregar”

Contenedores dinámicos:

Tareas Pendientes

Tareas Importantes

Tareas Terminadas

ScrollView para visualizar todas las categorías

Popup menu en cada tarea para cambiar el estado o eliminar

Funcionalidad:

Agregar tareas y mostrarlas dinámicamente

Guardar las tareas en Firebase Firestore

Cargar todas las tareas al iniciar la pantalla

Cambiar el estado de una tarea, actualizando Firestore

Eliminar una tarea en Firestore y en la interfaz

Mantener sincronización entre interfaz y base de datos

Eventos Implementados:

onClick para agregar

onClick en cada tarea para abrir menú contextual

onMenuItemClick para cambiar estado o eliminar

Comportamiento Dinámico:

Las tareas se crean en tiempo real como TextView

Se mueven visualmente entre categorías

Las categorías se actualizan instantáneamente



MisQuehaceres - Funcionalidades

MainActivity (Gestión de Tareas)

Componentes:

EditText para ingresar una tarea

Botón “Agregar”

Contenedores dinámicos:

Tareas Pendientes

Tareas Importantes

Tareas Terminadas

ScrollView para visualizar todas las categorías

Popup menu en cada tarea para cambiar el estado o eliminar

AcercaActivity (Información)

Componentes:

Título, Descripción breve del propósito de la app

Número de versión

Funcionalidad:

Mostrar información institucional de la aplicación.

Funcionalidad:

Agregar tareas y mostrarlas dinámicamente

Guardar las tareas en Firebase Firestore

Cargar todas las tareas al iniciar la pantalla

Cambiar el estado de una tarea, actualizando Firestore

Eliminar una tarea en Firestore y en la interfaz

Mantener sincronización entre interfaz y base de datos

Eventos Implementados:

onClick para agregar

onClick en cada tarea para abrir menú contextual

onMenuItemClick para cambiar estado o eliminar

Comportamiento Dinámico:

Las tareas se crean en tiempo real como TextView

Se mueven visualmente entre categorías

Las categorías se actualizan instantáneamente



MisQuehaceres - Funcionalidades

Diseño y Layouts

La aplicación utiliza:

ConstraintLayout como contenedor principal

LinearLayout vertical para listas de tareas

LinearLayout horizontal para encabezados y secciones

ScrollView para manejar contenido extenso

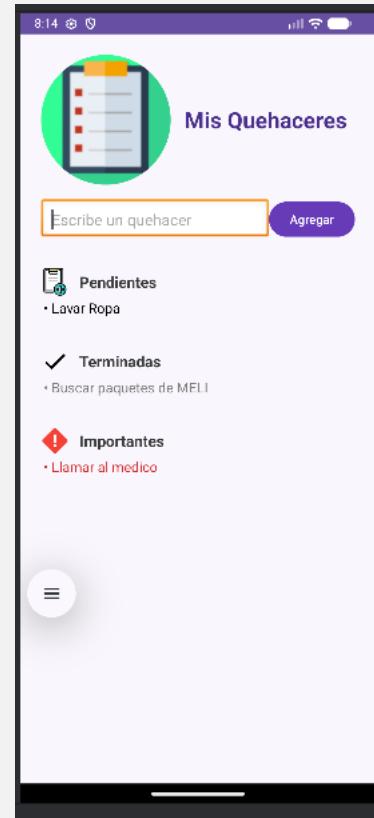
Imagenes reales en los encabezados de cada sección

Variables separadas en: strings.xml,

colors.xml

dimens.xml

El diseño prioriza claridad visual, jerarquía y legibilidad.



Persistencia con Firebase Firestore Configuración

Se integró Firebase mediante google-services.json

Se agregaron dependencias de Firestore en build.gradle

Se utilizó la colección "**tareas**"

The screenshot shows the Firebase Cloud Firestore interface. On the left, the sidebar includes 'Firebase' (selected), 'MisQuehaceresApp', 'Cloud Firestore', 'Base de datos' (selected), 'Datos', 'Reglas', 'Índices', 'Recuperación ante desastres', 'Uso', and 'Extensões'. In the main area, under 'Base de datos', there's a note about protecting resources from abuse. Below it, the 'tareas' collection is shown with three documents. The first document has fields: 'estado: "pendiente"' and 'texto: "Lavar Ropa"'. The other two documents have similar structures.

ID	Estado	Tarea
hixUUcknyHt9u61Qphx9	pendiente	Lavar Ropa
s0rx3aMFNpm1KY8iIve	pendiente	
utetZuzuJHtxY1lvOHAL	pendiente	



MisQuehaceres - Funcionalidades

Flujo General del Usuario

El usuario abre la app → HomeActivity

Selecciona “Ir a Mis Tareas”

MainActivity carga automáticamente todas las tareas desde Firestore

El usuario:

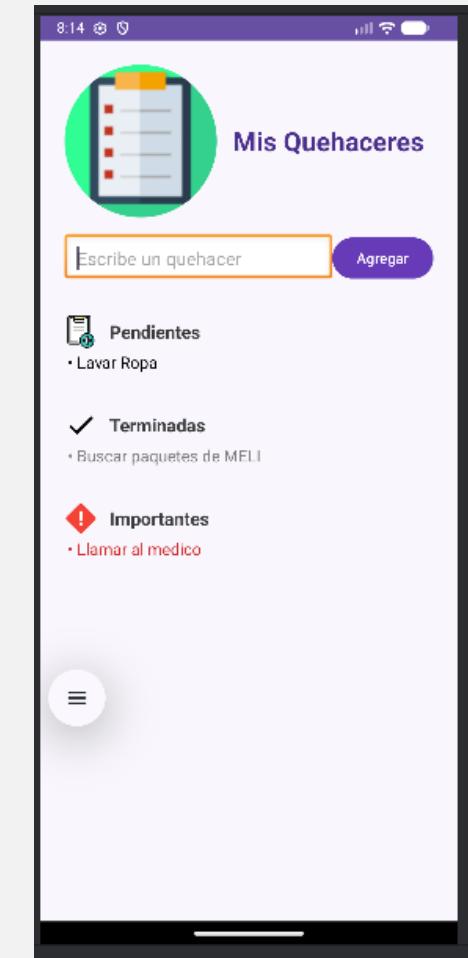
- Agrega nuevas tareas

- Cambia estado

- Elimina

Todas las acciones se reflejan en la interfaz y en Firestore

Si vuelve a abrir la app, el contenido se conserva.





MisQuehaceres - MockUp

Vista principal de la app
con secciones visibles y
ejemplo de tarea

Mis Quehaceres

Ingrese actividad... Button

	<ul style="list-style-type: none">• Lavar ropa• Reunión de padres
	<p>Limpiar el living</p>
	<p><input type="checkbox"/> Llamar al médico</p>

Mis Quehaceres

Ingrese actividad... Button

	<ul style="list-style-type: none">• Lavar ropa• Reunión de padres
	<p>Limpiar el living</p>
	<p><input type="checkbox"/> Llamar al médico</p>

- Marcar tarea Terminada
- Marcar tarea Prioridad
- Marcar tarea Pendiente
- Eliminar Tarea