



Instituto Politécnico Nacional  
Escuela Superior de Cómputo  
Ingeniería en  
Sistemas Computacionales



Unidad de Aprendizaje:

Inteligencia Artificial

Grupo: 7CV2

Práctica: 5

## Sistema de Notificaciones con Firebase (FCM)

Alumno:

Bautista Ríos Alfredo

Profesor: Gabriel Hurtado Avilés

# Indice

1. Introducción	3
2. Desarrollo: Funcionalidades Implementadas	3
2.1 Autenticación con Google (Firebase)	3
2.2 Exploración de Personajes (API & Retrofit)	5
2.3 Detalle del Personaje	7
2.4 Gestión de Perfil y Favoritos (Room)	9
3. Arquitectura y Mejores Prácticas	12
4. Conclusiones	12
Retos Principales	12
Logros Alcanzados	12

# 1. Introducción

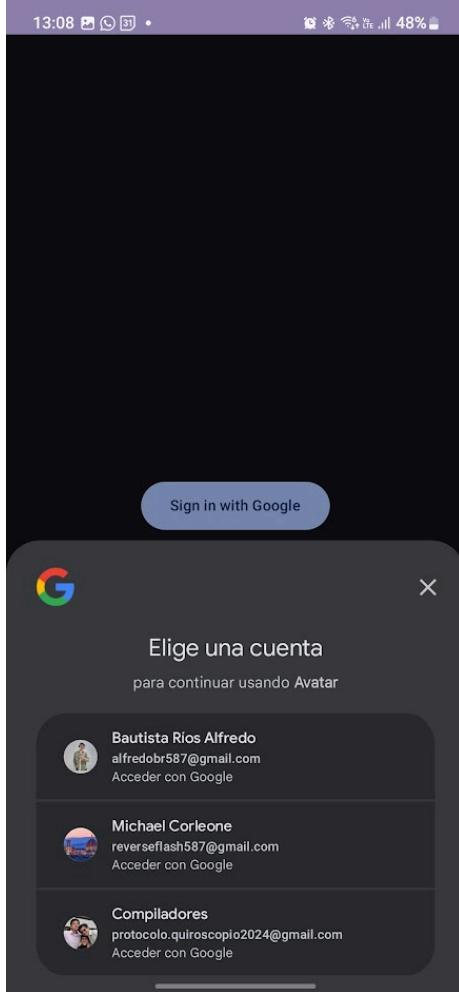
Este proyecto consiste en una aplicación móvil desarrollada para la plataforma Android que permite explorar el universo de la serie "Avatar: The Last Airbender". La aplicación integra el consumo de una API REST externa, persistencia de datos local, y un sistema de autenticación robusto, proporcionando una experiencia de usuario moderna y fluida siguiendo los lineamientos de Material Design 3.

## 2. Desarrollo: Funcionalidades Implementadas

### 2.1 Autenticación con Google (Firebase)

Se implementó un flujo de acceso seguro utilizando Firebase Authentication y Google Sign-In. El usuario puede identificarse para acceder a sus datos personalizados.

- Tecnología: Google Auth UI Client, Firebase SDK.



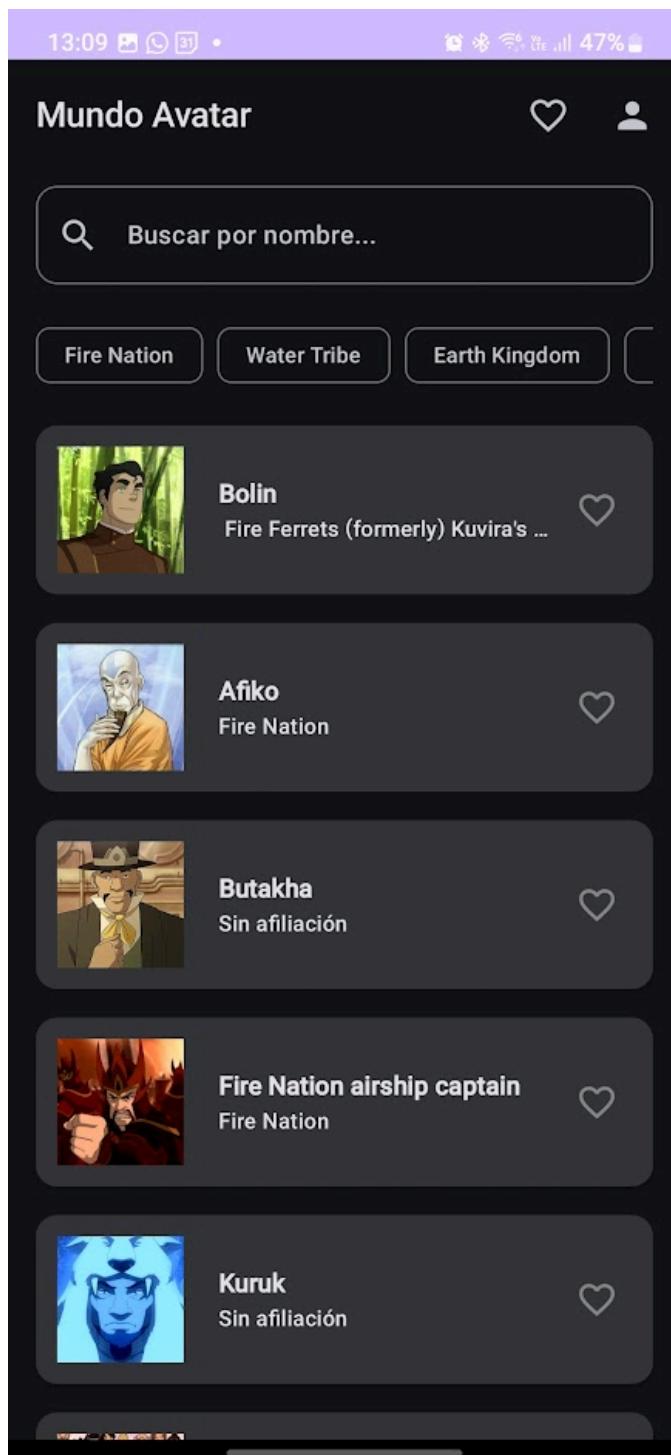
Descripción: Pantalla de bienvenida con el botón de "Sign in with Google".

- Prueba Realizada: Se validó que, tras un inicio de sesión exitoso, el token de usuario se recupere correctamente y se redirija al usuario a la pantalla principal automáticamente.

## 2.2 Exploración de Personajes (API & Retrofit)

La funcionalidad principal permite listar personajes obtenidos de la API last-airbender-api. Se utiliza Retrofit para las peticiones de red y Coil para la carga eficiente de imágenes.

- Tecnología: Retrofit2, Gson, Coil.



Descripción: Listado de personajes con tarjetas que muestran su nombre y foto.

- Prueba Realizada: Verificación de la carga de datos en segundo plano y manejo de estados de carga (loading) y error.

## 2.3 Detalle del Personaje

Al seleccionar un personaje, se navega a una pantalla de detalle que expone información técnica como su afiliación, enemigos, aliados y más.

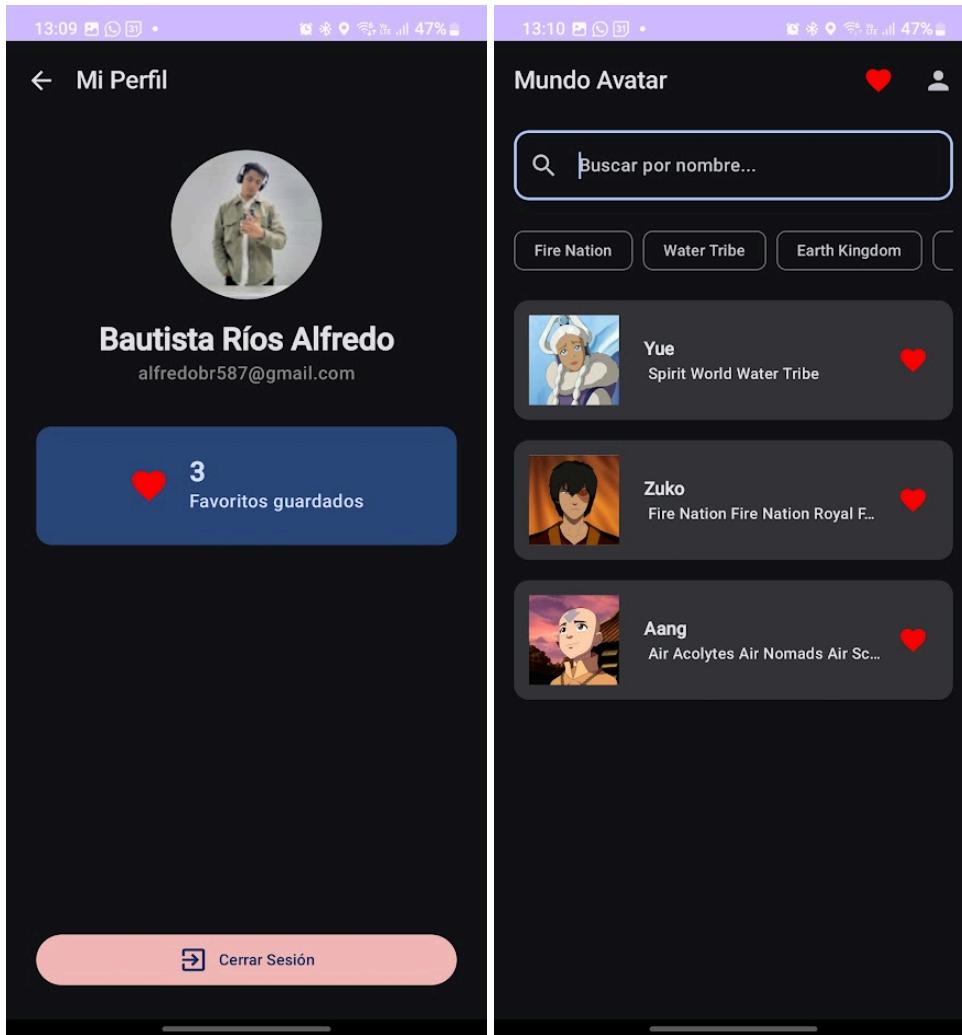


Descripción: Información detallada de un personaje seleccionado (ej. Aang o Zuko).

## 2.4 Gestión de Perfil y Favoritos (Room)

El usuario dispone de una sección de perfil donde puede ver su información de Google y gestionar sus personajes favoritos, los cuales se almacenan localmente mediante Room.

- Tecnología: Room Database, Jetpack Compose.



Descripción: Pantalla de perfil con la foto del usuario y contador de favoritos.

- Prueba Realizada: Se comprobó que al marcar un personaje como favorito, este persiste incluso después de cerrar y reiniciar la aplicación.

## 3. Arquitectura y Mejores Prácticas

El proyecto sigue la arquitectura recomendada por Google:

- MVVM (Model-View-ViewModel): Separación clara entre la lógica de negocio y la interfaz de usuario.
- Inyección de Dependencias (Hilt): Para un código más modular y fácil de testear.
- Jetpack Compose: Desarrollo de UI declarativa y reactiva.
- Navigation Compose: Gestión centralizada de las rutas de navegación.

## 4. Conclusiones

### Retos Principales

1. Integración de Google Auth: Configurar correctamente los SHA-1 y el archivo google-services.json para permitir la autenticación en diferentes entornos.
2. Manejo de Estados Complejos: Coordinar la carga de datos desde la API y la base de datos local simultáneamente para asegurar que la UI siempre muestre información actualizada.

### Logros Alcanzados

1. Interfaz de Usuario Moderna: Se logró una experiencia visual atractiva utilizando componentes de Material 3 y animaciones fluidas.
2. Persistencia Eficiente: La implementación de Room permite que la app sea funcional incluso con conectividad limitada para los datos ya guardados.
3. Código Limpio: El uso de Hilt y Clean Architecture facilita el mantenimiento futuro del proyecto.