Índice

1. Descripción y Justificación del Proyecto

LoopTube es una aplicación Android que permite a los usuarios reproducir música y podcasts de YouTube en segundo plano, incluso cuando la aplicación se minimiza. El objetivo es ofrecer una alternativa a los usuarios que no estén conformes con Youtube Music, con funcionalidades de gestión de listas de reproducción, marcadores de favoritos e historial de reproducción personalizado.

La aplicación surge de la necesidad de los usuarios de escuchar contenido de YouTube sin depender de una suscripción premium o tener que mantener la aplicación abierta. Además, busca ofrecer un entorno limpio y optimizado, sin publicidad invasiva, y con sincronización de datos entre la nube y el almacenamiento local.

LoopTube se dirige a un público amplio: usuarios que consumen música o podcasts más de nicho desde YouTube y buscan una forma más eficiente y personalizable de gestionarlo. La aplicación combina Firebase y SQLite para garantizar que no haya perdida de datos por parte del usuario.

2. Alcance del Proyecto

El proyecto abarca el desarrollo de una aplicación funcional que incluya autenticación de usuarios, búsqueda y reproducción de música mediante la API de YouTube, gestión de listas de reproducción y favoritos, historial de reproducción y sincronización entre Firebase y SQLite.

El alcance no incluye funcionalidades de descarga de contenido ni interacción social entre usuarios, por motivos legales y de viabilidad técnica.

3. Valoración de Alternativas Existentes

Aplicación	Ventajas	Limitaciones
YouTube Music	Aplicación oficial, alta calidad, integración completa con YouTube.	Requiere suscripción Premium para reproducir en segundo plano.
NewPipe	Gratuita, sin anuncios, ligera.	No sincroniza datos en la nube ni ofrece autenticación.
YMusic	Permite reproducción en segundo plano y modo sin conexión.	No cuenta con soporte oficial ni gestión de usuarios.

LoopTube busca ofrecer una experiencia equilibrada, combinando la libertad de reproducción en segundo plano con sincronización con base de datos y almacenamiento local, algo que las alternativas existentes no lo consiguen completamente.

4. Stack Tecnológico

- **Lenguaje:** Java
- **Entorno de desarrollo:** Android Studio
- **Backend/Nube:** Firebase (Authentication, Firestore, Storage)
- **Base de datos local:** SQLite (modo offline y caché)
- **API externa:** YouTube Data API v3
- **Control de versiones:** Git y GitHub
- **Roles:** Usuario estándar y administrador (para configuración general)
- **Pruebas:** Emulador Android y dispositivo físico

5. Objetivos

Objetivo general: Desarrollar una aplicación Android que permita reproducir música y podcasts de YouTube en segundo plano, con autenticación de usuarios y sincronización de datos entre la nube y el almacenamiento local.

Objetivos específicos:

- Implementar autenticación mediante Firebase Authentication.
- Permitir la búsqueda y reproducción de contenido con la API de Youtube.
- Sincronizar listas, favoritos e historial con Firebase Firestore.
- Implementar almacenamiento local con SQLite.
- Diseñar una interfaz moderna, intuitiva y accesible.

6. Requisitos de la Aplicación

Requisitos funcionales:

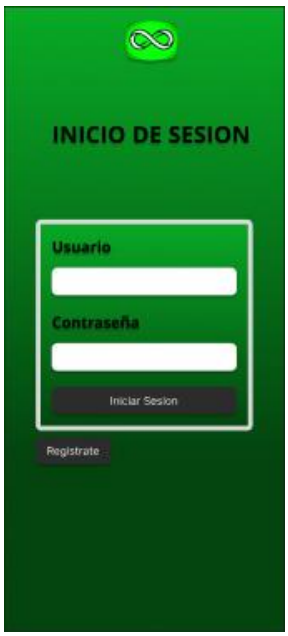
- **RF1:** Registrar e iniciar sesión mediante Firebase Authentication.
- **RF2:** Buscar videos por nombre o palabra clave.
- **RF3:** Reproducir contenido en segundo plano.
- **RF4:** Crear, editar y eliminar listas de reproducción.
- **RF5:** Agregar canciones a lista de favoritos.
- **RF6:** Guardar historial de reproducción.
- **RF7:** Sincronizar datos entre Firebase y SQLite.
- **RF8:** Gestionar los usuarios mediante un rol de administrador.

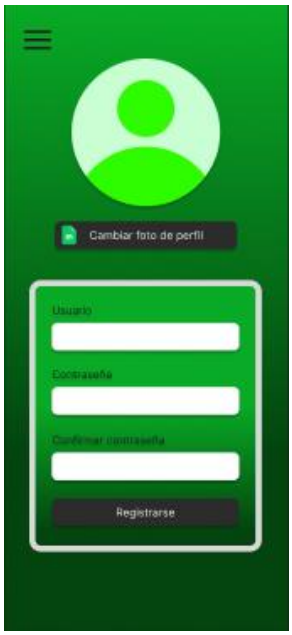
Requisitos no funcionales:

- **RNF1:** Compatibilidad con Android 8.0 o superior.
- **RNF2:** Interfaz intuitiva y accesible.
- **RNF3:** Seguridad en el manejo de datos.
- **RNF4:** Eficiencia energética en segundo plano.
- **RNF5:** Escalabilidad y mantenimiento.

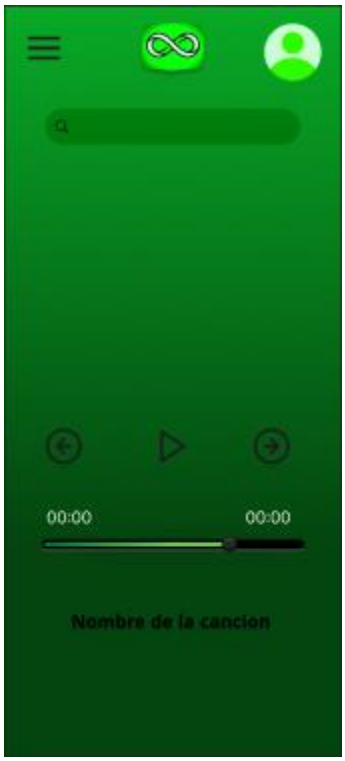
Requisitos de interfaz:

- Pantalla de inicio de sesión/registro.

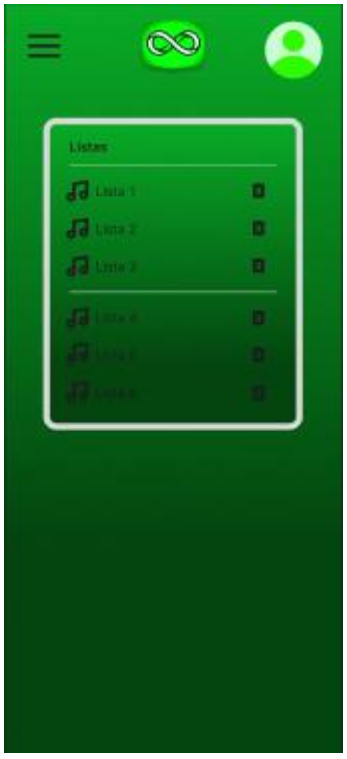


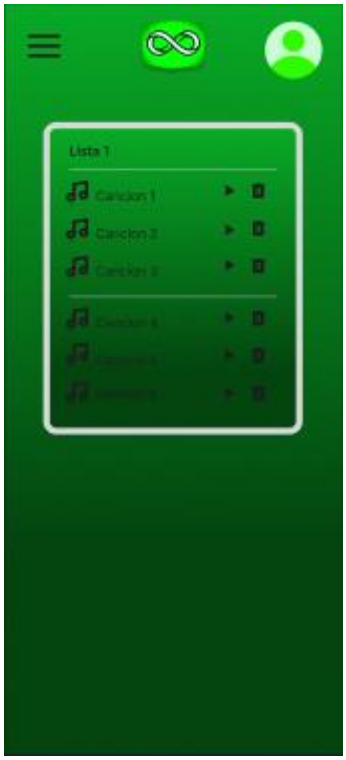


- Pantalla principal con búsqueda y reproducción.

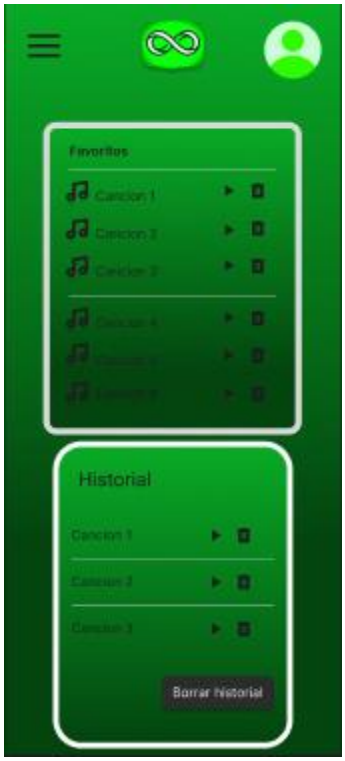


- Sección de listas de reproducción.

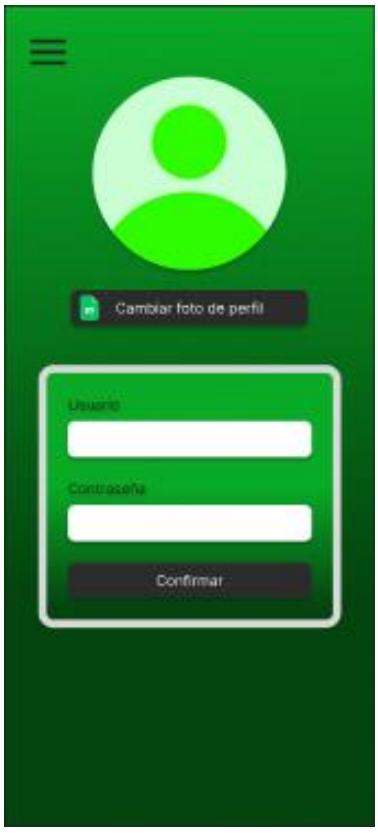




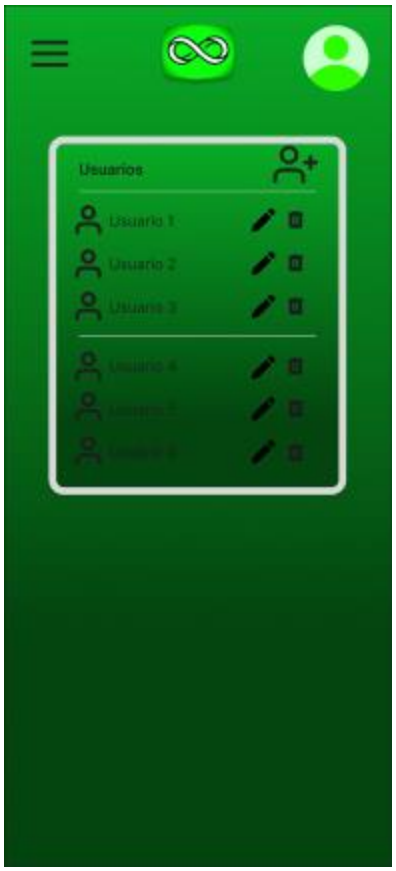
- Pantalla de historial y favoritos.



- Configuración y perfil de usuario.



-Administrador



7. Casos de Uso Principales

1. CU1 – Iniciar sesión: El usuario introduce sus credenciales y Firebase valida el acceso.

2. CU2 – Buscar y reproducir música: El usuario realiza una búsqueda, se consultan resultados mediante la API de YouTube y se reproduce el contenido.

3. CU3 – Siguiente canción: El usuario hace clic en el botón de siguiente canción y salta a la siguiente canción

4. CU4 – Canción anterior: El usuario hace clic en el botón de canción anterior y reproduce la canción que se ha reproducido antes de la que este escuchando en ese momento.

5. CU5 – Añadir a favoritos: El usuario selecciona una canción y la marca como favorita agregándose a la lista de canciones favoritas.

6. CU6 – Crear lista de reproducción: El usuario hace clic en el icono de crear la lista de reproducción introduce un nombre y se añade a la sección de listas de reproducción

7. CU7 – Eliminar lista de reproducción: El usuario hace clic en el icono eliminar la lista de reproducción y la lista de reproducción se elimina correctamente.

8. CU8 –Agregar canción a lista de reproducción: El usuario hace clic en el icono de agregar a lista de reproducción y la canción se agrega correctamente a la lista de reproducción determinada.

9. CU9 – Eliminar canción de la lista de reproducción: El usuario hace clic en el icono de eliminar en una canción de la lista de reproducción en la que este y se elimina la canción de esa lista de reproducción

10. CU10 – Canción aparece en el historial: Al escuchar el usuario una canción, se dirige a la sección de historial y la canción que escucho aparece listada.

11. CU11 – Modo offline: Si no hay conexión, la app utiliza los datos almacenados en SQLite.

12. CU12 – Agregar Usuario (Administrador): El usuario administrador hace clic en el botón de crear usuario, le redirige a la pantalla de registro para posteriormente escribir las credenciales de nuevo usuario. Por último, el usuario se ha creado y añadido correctamente a la lista de usuarios.

13. CU13 – Eliminar usuario (Administrador): El usuario administrador hace clic en el botón de eliminar usuario y el usuario se elimina correctamente de la lista de usuarios.

14. CU14 – Editar usuario (Administrador): El usuario administrador hace clic en el botón de editar usuario y le redirige a la página de editar perfil con los datos del usuario que quiere editar y al cambiarlos e iniciar sesión con ese usuario se han efectuado los cambios correctamente.

8. Modelo y Diseño de la Base de Datos

El modelo de datos se compone de las siguientes entidades:
- Usuarios (id, nombre, email, contraseña_hash, rol)
- Listas (id_lista, id_usuario, nombre)
- Canciones (id_video, título, canal, url_miniatura)
- Favoritos (id_usuario, id_video)
- Historial (id_usuario, id_video, fecha_reproducción)

El diseño del modelo se representará mediante un diagrama entidad-relación incluido en el repositorio.

9. Repositorio GitHub

Repositorio público: https://github.com/2DAM-josesan303/Looptube.git