



UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

**FACULTAD DE CONTADURÍA Y
ADMINISTRACIÓN**

**Aplicación Móvil; Acortador
de URLs**

Proyecto Final Apps Móviles

P R E S E N T A

González de Santiago Itiel

PROFESOR

Cristian Cardoso Arellano

ASIGNATURA

Desarrollo de aplicaciones móviles



Ciudad Universitaria, Cd. Mx., 2025

1. Objetivo General

Se buscará diseñar y desarrollar una aplicación móvil para dispositivos Android que permita a los usuarios generar versiones acortadas de enlaces web de manera rápida y segura, integrando funcionalidades clave como autenticación mediante Google y gestión de cuentas personales. La aplicación ofrecerá funcionalidades diferentes según el tipo de usuario: los usuarios gratuitos podrán generar hasta cinco URLs por mes, mientras que los usuarios con suscripción premium tendrán acceso ilimitado a la funcionalidad de acortamiento.

El sistema también permitirá a los usuarios visualizar un historial completo de URLs generadas y eliminar enlaces individuales. Toda la información generada será almacenada de forma persistente utilizando Room como solución de base de datos local, y se comunicará con un backend real a través de Retrofit para manejar la lógica del acortamiento y la autenticación.

El desarrollo de esta aplicación se ajustará estrictamente a las restricciones técnicas del proyecto, excluyendo el uso de tecnologías y herramientas como RX, Data Binding, Jetpack Compose o librerías externas adicionales, garantizando una solución eficiente, escalable y dentro del marco definido por los lineamientos del curso.

2. Funcionalidades Principales

2.1 Autenticación

La aplicación implementa un sistema de autenticación que permite a los usuarios iniciar sesión exclusivamente mediante su cuenta de Google, esta funcionalidad es obligatoria ya que es el único método de acceso, garantizando una experiencia de usuario segura, sin necesidad de gestionar contraseñas locales ni otras formas de autenticación.

2.2 Gestión de URLs

La funcionalidad principal de la aplicación es la generación y gestión de enlaces acortados, dependiendo del tipo de usuario, la funcionalidad será diferente para cada caso.

Usuarios Free Tier	Usuarios Premium (mediante suscripción simulada)
Podrán generar un máximo de cinco URLs acortadas por mes.	Capacidad ilimitada para acortar URLs sin restricciones mensuales.
Al alcanzar el límite mensual, la funcionalidad de acortamiento se bloquea temporalmente hasta el siguiente ciclo mensual.	

Tanto usuarios Free como Premium pueden:

- Visualizar un historial de URLs generadas.
- Eliminar enlaces específicos de su historial.

2.3 Restricciones Técnicas

Con el fin de garantizar el cumplimiento de los lineamientos académicos y técnicos establecidos para este proyecto, se definen las siguientes restricciones obligatorias:

- **No se permite el uso de bibliotecas externas**, con excepción de Retrofit para la comunicación HTTP. (como Glide, OkHttp directamente, Gson por separado, etc.)
- **No se permite el uso de Jetpack Compose** para la construcción de interfaces gráficas. La interfaz debe desarrollarse exclusivamente mediante archivos XML tradicionales, manteniendo la estructura clásica de Android con Activities y Views.
- **Está prohibido el uso de DataBinding**, el acceso y manipulación de vistas debe realizarse mediante findViewById u otras alternativas tradicionales. ViewBinding tampoco está permitido.
- **No se permite el uso de programación reactiva con RX (RxJava o RxKotlin)**. La lógica debe implementarse utilizando Kotlin Coroutines o callbacks simples.
- **No se permite copiar o reutilizar código generado por Inteligencia Artificial**. Todo el código implementado debe ser completamente original y desarrollado por el responsable del proyecto.

3. Requisitos Funcionales

ID	Descripción	Prioridad
RF01	El usuario podrá iniciar sesión exclusivamente con Google.	Alta
RF02	El usuario Free podrá acortar hasta 5 URLs por mes.	Alta
RF03	El usuario Premium podrá acortar URLs sin límite.	Alta
RF04	El sistema deberá impedir nuevas URLs a usuarios Free que superen el límite.	Alta
RF05	El usuario podrá ver su historial de URLs acortadas.	Media
RF06	El usuario podrá eliminar URLs del historial.	Media

3.1 Requisitos no Funcionales

- **Rendimiento:** la app debe responder a cada solicitud en menos de 3 segundos.
- **Compatibilidad:** debe funcionar en dispositivos Android 9 o superior.
- **Seguridad:** se deben verificar tokens JWT de Firebase para cada petición.
- **Usabilidad:** interfaz limpia, sin elementos innecesarios.

4. Requisitos Técnicos

4.1 Tecnologías Obligatorias

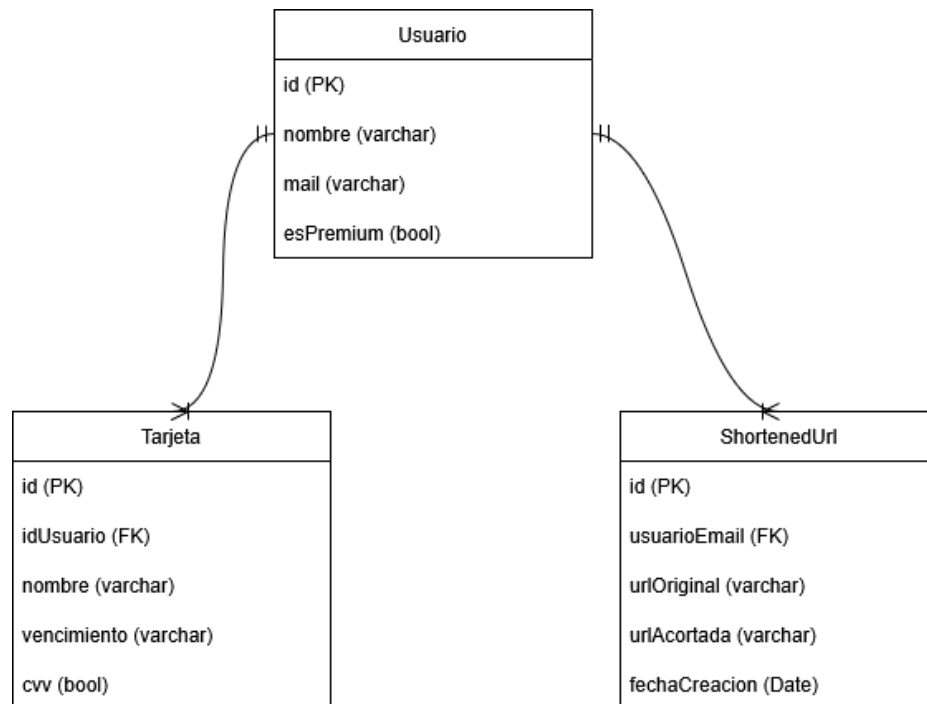
- **Lenguaje:** Java
- **UI:** XML tradicional (sin Compose ni DataBinding)
- **Base de datos:** Room
- **Librería de red:** Retrofit
- **Login:** Firebase Authentication (con Google Sign-In)
- **Backend:** Node.js + PostgreSQL (Railway)

5. Alcance del Proyecto

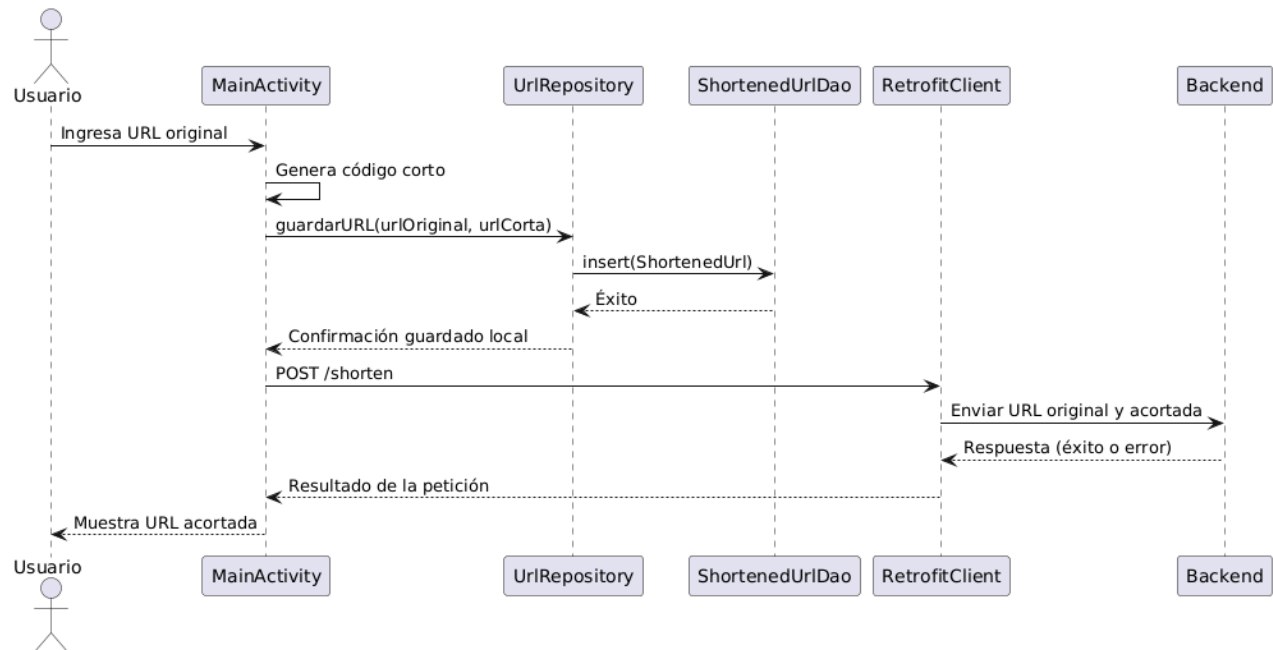
Este proyecto se enfoca en el desarrollo de una aplicación móvil para Android que permite el acortamiento de URLs mediante un backend real, la app se limita a dispositivos Android 9 o superior, y no contempla su implementación en iOS ni en web, además estará fuera de alcance lo siguiente:

- Edición de URLs acortadas.
- Otras formas de autenticación (correo, redes sociales, etc.).
- Soporte multiplataforma (iOS, web, etc.).
- Pago real para usuarios premium (es una validación simulada).

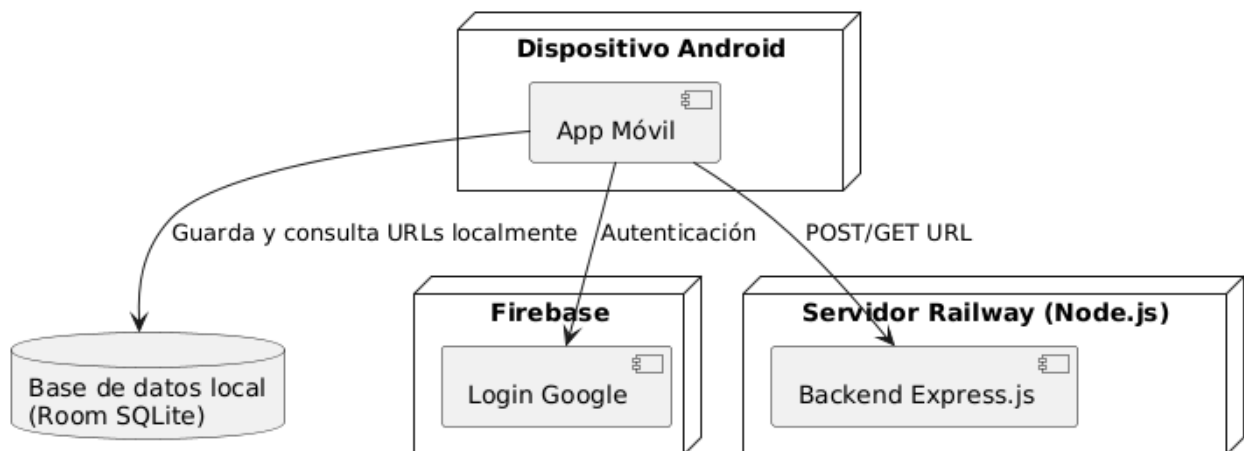
6. Diagrama de datos



7. Diagramas de secuencia



8. Diagrama general de web



9. Requisitos para el Funcionamiento de la Aplicación

Para garantizar el correcto funcionamiento de la aplicación móvil de acortamiento de URLs, se requiere cumplir con los siguientes aspectos técnicos y operativos:

9.1. Instalación del APK

La aplicación ha sido compilada en formato APK (Android Package) y debe instalarse manualmente en los dispositivos móviles con sistema operativo Android. Para ello, es necesario:

- Habilitar la instalación de aplicaciones desde orígenes desconocidos en el dispositivo.
- Contar con un dispositivo con Android 7.0 (Nougat) o superior.
- Espacio de almacenamiento mínimo: 30 MB.

9.2. Conectividad a Internet

La aplicación requiere una conexión estable a Internet (Wi-Fi o datos móviles) para ejecutar correctamente funciones como:

- Ingreso con cuenta de Google mediante Firebase Authentication.
- Comunicación con el servidor backend para guardar o redireccionar URLs.
- Sincronización de datos entre la base de datos local y el servidor en la nube.

9.3. Servicios de Firebase

Se utiliza Firebase Authentication para permitir el inicio de sesión con cuentas de Google. Esto implica:

- Acceso a los servicios de Google Play.
- Contar con una cuenta de Google válida en el dispositivo.
- Tener habilitados los servicios de autenticación de Firebase en el backend del proyecto.

9.4. Backend y Servidor Web

Para la funcionalidad de redirección y almacenamiento en la nube, la aplicación se comunica con un backend desarrollado en Node.js y desplegado en Railway, el cual ofrece:

- Un endpoint para almacenar nuevas URLs acortadas.
- Un endpoint para redireccionar cuando un usuario accede a una URL corta.
- Un middleware que valida el origen y destino de las peticiones.

Conexión a una base de datos MongoDB en la nube.

Nota: Es importante que el backend esté activo y disponible durante el uso de la aplicación. Si el servidor está caído, ciertas funciones no estarán disponibles (como el acortamiento de nuevas URLs o la redirección).

9.5. Base de Datos Local

La aplicación implementa una base de datos local utilizando Room (SQLite) para:

- Guardar URLs acortadas generadas por el usuario.
- Consultar el historial de URLs sin depender de la conexión al backend.

Esta arquitectura híbrida permite un funcionamiento básico incluso en momentos de baja conectividad, aunque se requieren datos móviles para sincronización completa.

9.6. Configuraciones Técnicas del Proyecto

- Lenguaje principal: Java
- Arquitectura: MVM
- Librerías utilizadas: Room, Retrofit, Firebase, RxJava
- No se ha utilizado Jetpack Compose ni DataBinding, de acuerdo con las restricciones del proyecto.