



I E S A L B A R R E G A S

TÉCNICO SUPERIOR EN DESARROLLO DE
APLICACIONES MULTIPLATAFORMA

Departamento de Informática

PROYECTO

ChatNow

Manual Técnico

Autor/es: Jorge Solana Holgado


Curso Académico: Desarrollo de Aplicaciones Multiplataforma 2º B

Índice

1.Introducción	3
2.Arquitectura de la aplicación	3
2.1. Android Studio	3
2.2. Firebase Authentication:	4
2.3. Firebase Realtime Database:	4
3.Flujos de la aplicación	5
3.1. Flujo de Registro de Usuario	5
3.2. Flujo de Inicio de Sesión.....	6
3.3. Flujo de Envío de Mensajes	7
3.4. Flujo de Recuperación de Contraseña.....	8
3.5. Flujo de Gestión de Perfil	9
4.Diseño de la Base de Datos	10
5.Propuestas de mejoras.....	10
6.Manual instalación de la aplicación.....	11

1. Introducción

ChatNow es una aplicación para dispositivos móviles que tiene como objetivo desarrollar un chat en tiempo real que permita a los usuarios comunicarse de manera instantánea y eficiente. La aplicación estará diseñada para ofrecer una experiencia de usuario fluida y segura, facilitando la interacción en tiempo real entre múltiples usuarios.

Repositorio de Github con el código de la aplicación → 

2. Arquitectura de la aplicación

2.1. Android Studio

Es el entorno de desarrollo integrado (IDE) utilizado para crear la aplicación ChatNow. Proporciona todas las herramientas necesarias para desarrollar, depurar y probar aplicaciones Android de manera eficiente.

- **Desarrollo en Kotlin:** Android Studio es la plataforma utilizada para escribir el código de la aplicación en Kotlin, un lenguaje moderno y eficiente para el desarrollo de aplicaciones Android.
- **Gestión de Dependencias:** A través de Gradle, Android Studio maneja las dependencias de la aplicación, incluidas las librerías de Firebase.
- **Pruebas y Depuración:** Android Studio permite probar la aplicación en un emulador o dispositivo físico, y ofrece herramientas de depuración para solucionar problemas en el código.
- **Diseño de la Interfaz:** Android Studio ofrece un diseño visual para construir la interfaz de usuario de manera interactiva, permitiendo arrastrar y soltar componentes visuales en el layout.



2.2. Firebase Authentication:

Se utiliza para manejar la autenticación de los usuarios. Permite a los usuarios registrarse, iniciar sesión y recuperar sus contraseñas de manera segura. Este servicio es fundamental para asegurar que los datos y la comunicación dentro de la aplicación sean privados y estén asociados a un usuario específico.

- **Registro e Inicio de Sesión:** Los usuarios pueden registrarse utilizando su correo electrónico y contraseña, y luego iniciar sesión con estos datos.
- **Recuperación de Contraseña:** Si los usuarios olvidan su contraseña, pueden solicitar un enlace de recuperación enviado a su correo electrónico registrado.
- **Seguridad:** Firebase Authentication ofrece una solución segura para la autenticación sin necesidad de manejar contraseñas directamente en el backend.

2.3. Firebase Realtime Database:

Se utiliza para almacenar y sincronizar los datos en tiempo real, lo que permite una comunicación instantánea entre los usuarios. Los datos de los usuarios (nombre, correo, etc.) y los mensajes enviados por los usuarios se almacenan en esta base de datos.

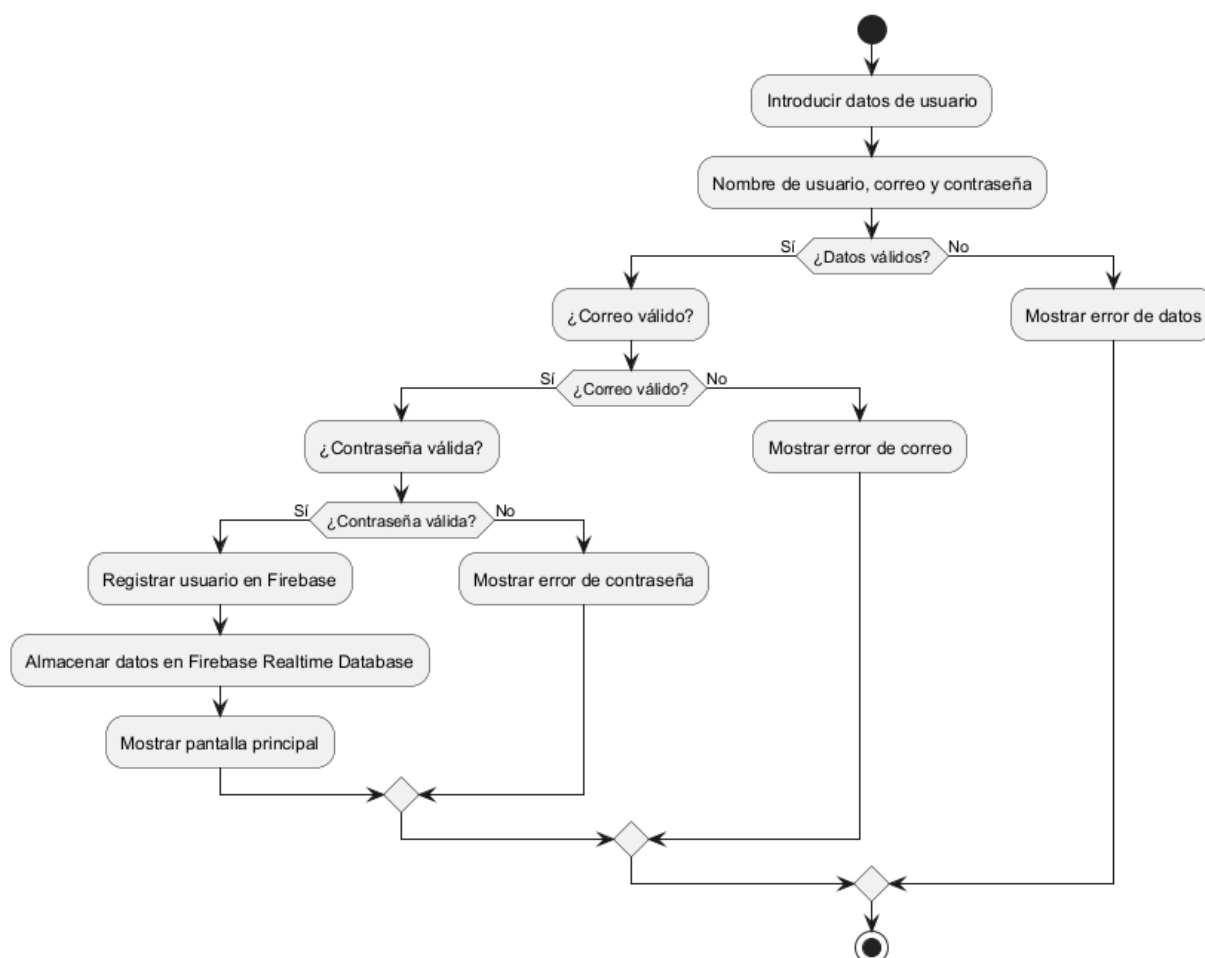
- **Almacenamiento de Mensajes:** Los mensajes de texto o imágenes se almacenan en Firebase Realtime Database. Cada mensaje se guarda bajo un nodo específico para cada chat y se sincroniza automáticamente entre los dispositivos de los usuarios.
- **Estructura de la Base de Datos:** La estructura de la base de datos está organizada de manera que los mensajes se almacenen en nodos por cada chat entre usuarios. Los datos de perfil de los usuarios también se guardan bajo su identificador único (UID).
- **Sincronización en Tiempo Real:** Los mensajes se actualizan en tiempo real sin necesidad de refrescar la aplicación, lo que ofrece una experiencia de chat fluida.



3. Flujos de la aplicación

3.1. Flujo de Registro de Usuario

Cuando un usuario se registra, la aplicación recibe los datos del formulario (nombre de usuario, correo electrónico y contraseña), valida la información, la guarda en Firebase y muestra la pantalla principal.

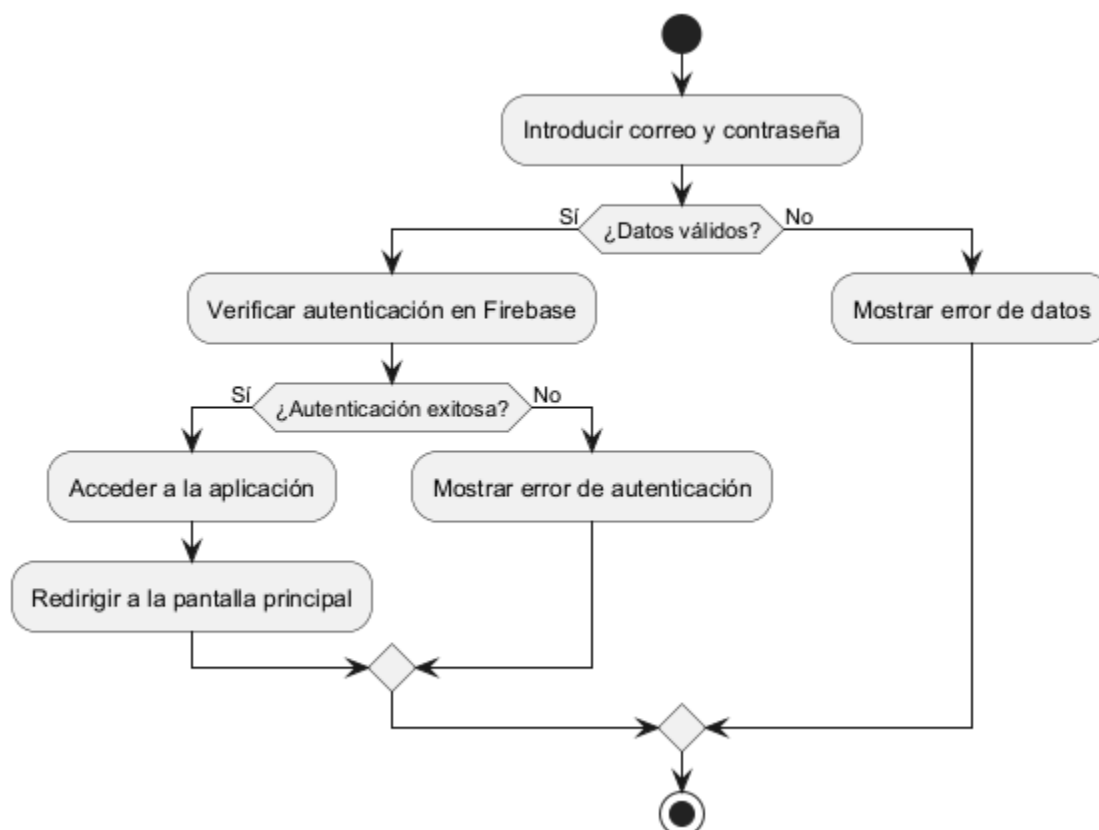


Descripción técnica:

- **Validación de Datos:** El sistema valida si los campos (correo, nombre de usuario y contraseña) no están vacíos y cumplen con las políticas de validación (por ejemplo, correo válido y contraseña segura).
- **Firestore Authentication:** Se usa FirebaseAuth para registrar al usuario.
- **Firestore Realtime Database:** Los datos del usuario (nombre de usuario, correo, UID, etc.) se almacenan en Firestore Realtime Database.

3.2. Flujo de Inicio de Sesión

El inicio de sesión se realiza con un correo electrónico y contraseña. Si los datos son correctos, el sistema autentica al usuario y lo redirige a la pantalla principal.

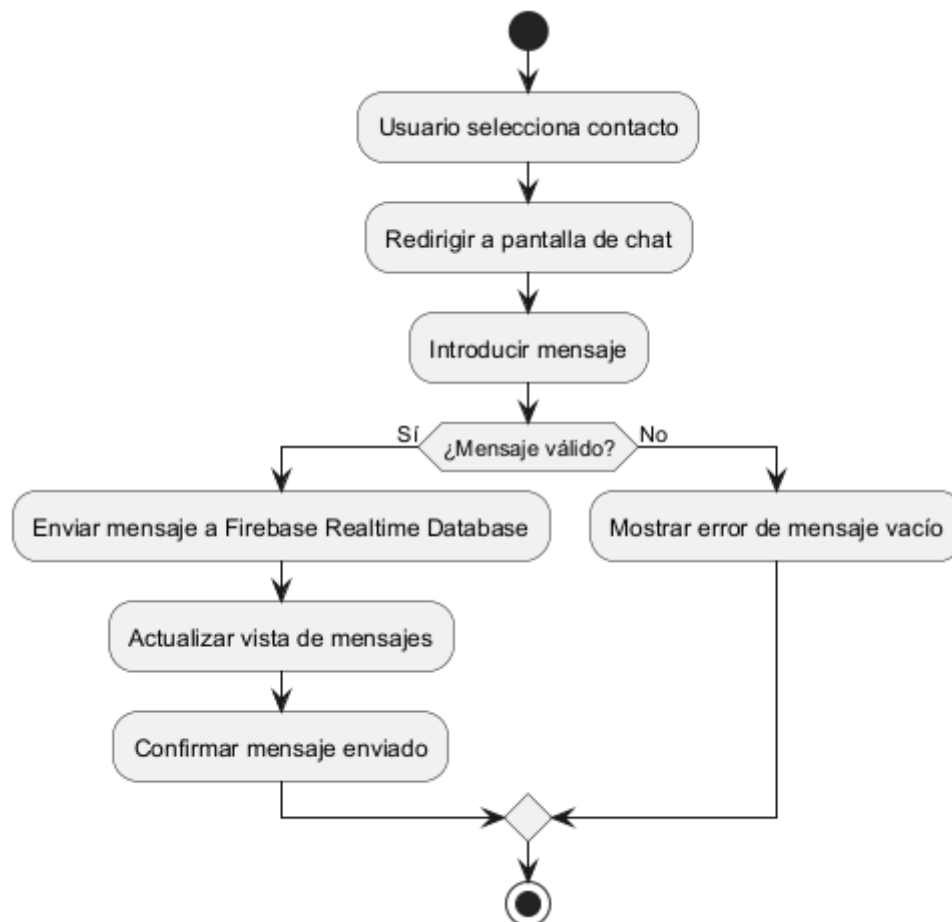


Descripción técnica:

- **Autenticación con Firebase:** Usando FirebaseAuth, se verifica la existencia del usuario con el correo y la contraseña proporcionados.
- **Redirección:** Si la autenticación es exitosa, el usuario es redirigido a la pantalla principal. Si falla, se muestra un mensaje de error.

3.3. Flujo de Envío de Mensajes

El envío de mensajes es uno de los procesos principales de la aplicación. Cuando un usuario escribe y envía un mensaje, este se guarda en Firebase y se muestra en el chat.

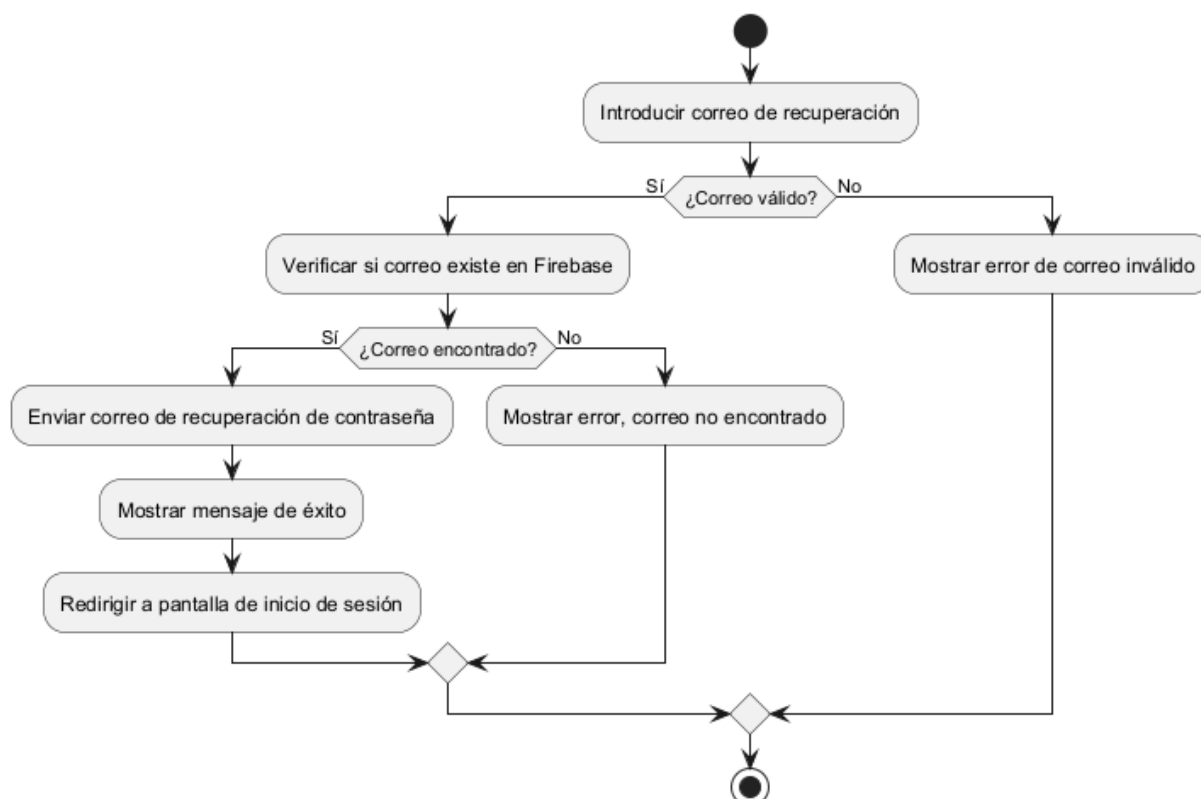


Descripción técnica:

- **Selección de Contacto:** El usuario selecciona un contacto para iniciar un chat.
- **Envío a Firebase:** El mensaje es enviado a **Firebase Realtime Database**, donde se almacena en el nodo correspondiente al chat entre los dos usuarios.
- **Actualizar UI:** Después de enviar el mensaje, la interfaz de usuario se actualiza para mostrar el nuevo mensaje.

3.4. Flujo de Recuperación de Contraseña

Cuando un usuario olvida su contraseña, puede iniciar un proceso para recuperarla. La aplicación envía un enlace de recuperación a su correo electrónico registrado.

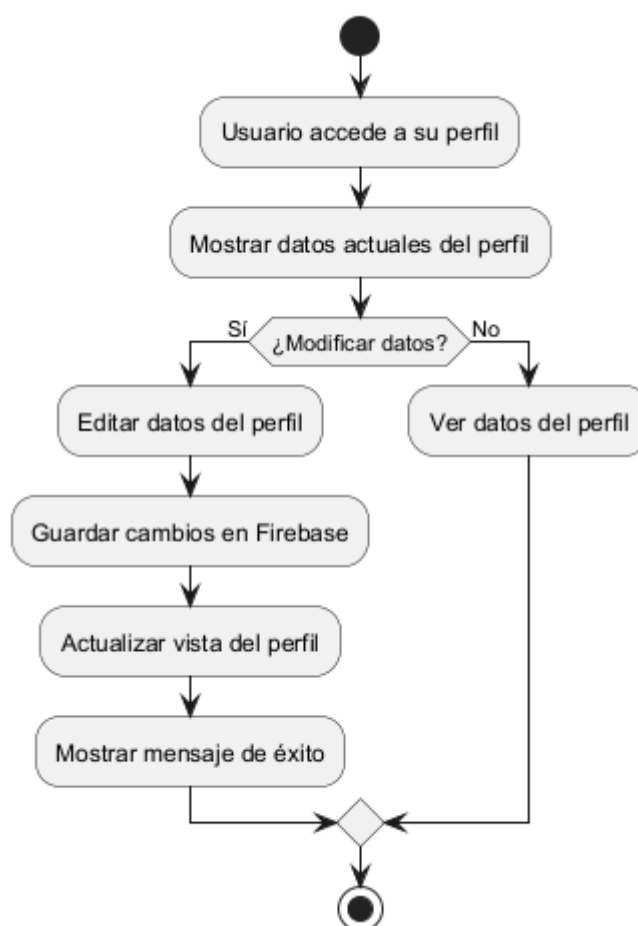


Descripción técnica:

- **Correo de Recuperación:** El sistema solicita al usuario ingresar su correo electrónico.
- **Verificación en Firebase:** Se verifica si el correo existe en la base de datos de usuarios de Firebase.
- **Enlace de Recuperación:** Si el correo existe, se envía un enlace de recuperación utilizando `FirebaseAuth.sendPasswordResetEmail`.

3.5. Flujo de Gestión de Perfil

El usuario puede ver y editar su perfil. Los cambios realizados en el perfil se reflejan en Firebase y se actualizan en la aplicación.

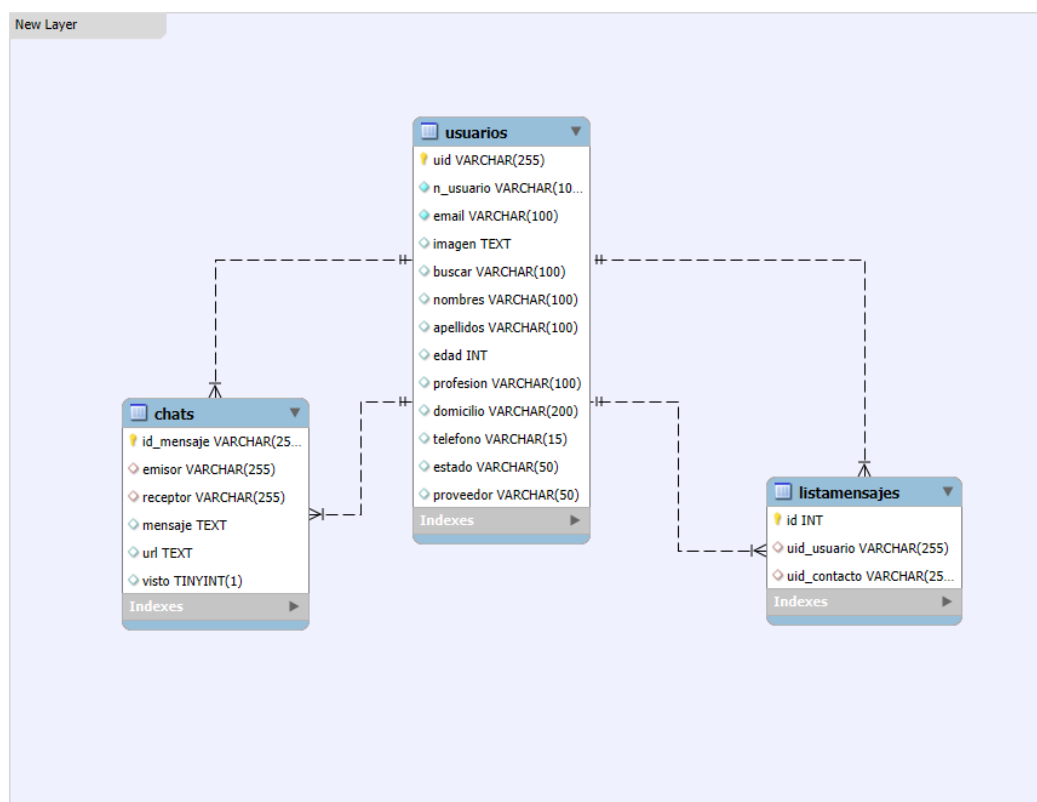


Descripción técnica:

- **Acceso al Perfil:** El usuario accede a su perfil, donde puede ver sus datos almacenados en Firebase.
- **Actualización de Datos:** Si el usuario decide editar su perfil, los nuevos datos se guardan en **Firestore Realtime Database**.
- **Interfaz Actualizada:** La interfaz se actualiza para reflejar los nuevos datos.

4. Diseño de la Base de Datos

La base de datos se ha creado en Firebase Realtime Database, tenemos tres tablas: Usuarios, Chats y Listamensaje



5. Propuestas de mejoras

Algunas mejoras a considerar para futuras actualizaciones del programa incluyen:

- **Notificaciones Push:** Implementar notificaciones push para alertar al usuario receptor cuando reciba un mensaje, asegurando que esté informado en tiempo real.
- **Envío de Videos:** Ampliar la funcionalidad actual de envío de imágenes para permitir el envío de videos, mejorando las opciones de intercambio multimedia.
- **Chats Grupales:** Habilitar la creación de grupos con múltiples usuarios, facilitando la comunicación colectiva dentro de un solo chat.

6. Manual instalación de la aplicación

Para instalar la aplicación es necesario tener un dispositivo móvil Android y que tenga la versión de software Android 11 o superior.

Lo único que necesitamos es la apk de la aplicación que podemos encontrarla en el repositorio de [GitHub](#) o en esta carpeta de [Google Drive](#).

7. Bibliografía

Para desarrollar esta aplicación se ha recogido información de estos sitios web:

- [Stack Overflow](#)
- [Android Studio Developers](#)
- [Documentación de Firebase](#)