	UNIVERSIDAD DON BOSCO FACULTAD DE INGENIERÍA ESCUELA DE COMPUTACIÓN	
	GUIA DE LABORATORIO #04	
	Nombre de la Práctica:	Guia #04- Login y RealTime Database Firebase
	MATERIA:	Desarrollo de Software para Móviles

I. OBJETIVOS

1. Implementar un Login Google en Firebase
2. Obtener información del usuario directamente desde Firebase
3. Conocer los beneficios de usar Firebase en Android
4. Conocer los fundamentos de una base de datos en tiempo real de Firebase.
5. Desarrollar una aplicación CRUD que utilice RealTime Database de Firebase.

II. INTRODUCCION TEORICA

Introducción a Firebase

Firebase se desarrolló en 2011 para crear y brindar soporte para aplicaciones web y móviles. Posteriormente, en 2014, fue adquirido por Google. Le brinda una serie de facilidades que se pueden utilizar para el desarrollo rápido de sus aplicaciones. Por lo tanto, todo lo que necesita hacer es crear aplicaciones rápidamente sin pensar en administrar la infraestructura. A continuación, se muestran los productos de Firebase que se pueden usar en cualquier aplicación:

- **Cloud Firestore:** es una base de datos NoSQL que se usa para almacenar, sincronizar y consultar fácilmente datos para aplicaciones.
- **ML Kit:** brinda el poder del aprendizaje automático a sus aplicaciones.
- **Authentication:** se utiliza para autenticar a los usuarios de sus aplicaciones.
- **Realtime Database:** se utiliza para almacenar y recuperar datos en tiempo real.
- **Crashlytics:** le proporciona los informes de fallos de su aplicación.
- **Test Labs:** probará sus aplicaciones en los dispositivos alojados por Google.
- **Performance Monitoring:** supervisa el rendimiento de su aplicación en cada segundo.
- **In-App Messaging:** le ayuda a enviar mensajes dentro de la aplicación que ayudan a sus usuarios a interactuar con usted.
- **Google Analytics:** generará un informe de su aplicación. Por ejemplo, un informe de usuarios aumentó en el último mes o usuarios activos, etc.
- **Remote Config:** le ayuda a cambiar o agregar alguna función en su aplicación incluso sin poner ninguna actualización.

Además de los productos de Firebase mencionados anteriormente, algunos de los otros productos de Firebase son Cloud Functions , Hosting , Cloud Storage , App Distribution , Predictions , A / B Testing , Cloud Messaging y Dynamic Links .

Inicio de sesión y autenticación en Firebase

La autenticación de Firebase se utiliza para autenticar a los usuarios de aplicaciones de una manera muy sencilla. No solo para los usuarios, sino también para los desarrolladores, proporciona un flujo muy fácil para el proceso de autenticación e inicio de sesión que está presente en casi todas las aplicaciones.

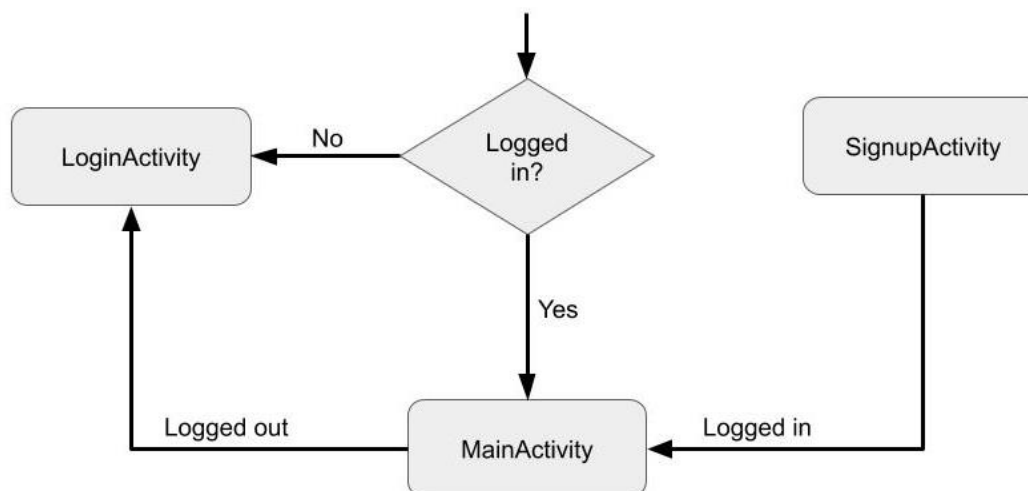
Firebase admite la autenticación mediante correo electrónico y contraseña, números de teléfono o incluso puede usar **Facebook, Google, Twitter, Github**, etc.

Para autenticar a sus usuarios, todo lo que necesita hacer es obtener las credenciales de autenticación del usuario y luego pasar esta credencial al SDK de autenticación de Firebase. Estas credenciales pueden ser una contraseña de correo electrónico o un número de teléfono móvil o cualquier **token** de proveedores de identidad como Facebook, Google, Twitter, Github, etc. Después de pasar las credenciales, Firebase verificará las credenciales y, a cambio, recibirá una respuesta que le indicará si la autenticación es exitosa o no.

De forma predeterminada, los usuarios autenticados pueden leer / escribir los datos de la base de datos en tiempo real o del almacenamiento en la nube.

Inicio de sesión y registro por correo electrónico

Aprenderemos cómo iniciar sesión y registrarse usando correo electrónico y contraseña con la ayuda de Firebase. En este ejemplo, tendremos cuatro actividades, es decir, una para iniciar sesión, una para registrarse, una para restablecer la contraseña y una será nuestra actividad principal. La siguiente imagen describirá el flujo de actividades:



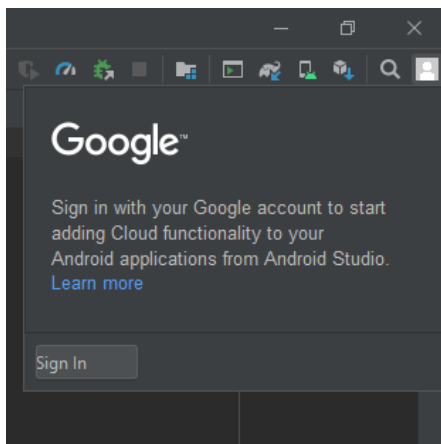
IV. PROCEDIMIENTO

Inicio de sesión y autenticación de Firebase:

Para usar Firebase **Authentication** en nuestra aplicación, necesitamos conectar nuestro proyecto, es decir, el proyecto de Android Studio a Firebase. Los siguientes son los pasos que se utilizan para conectar un proyecto de Android a Firebase

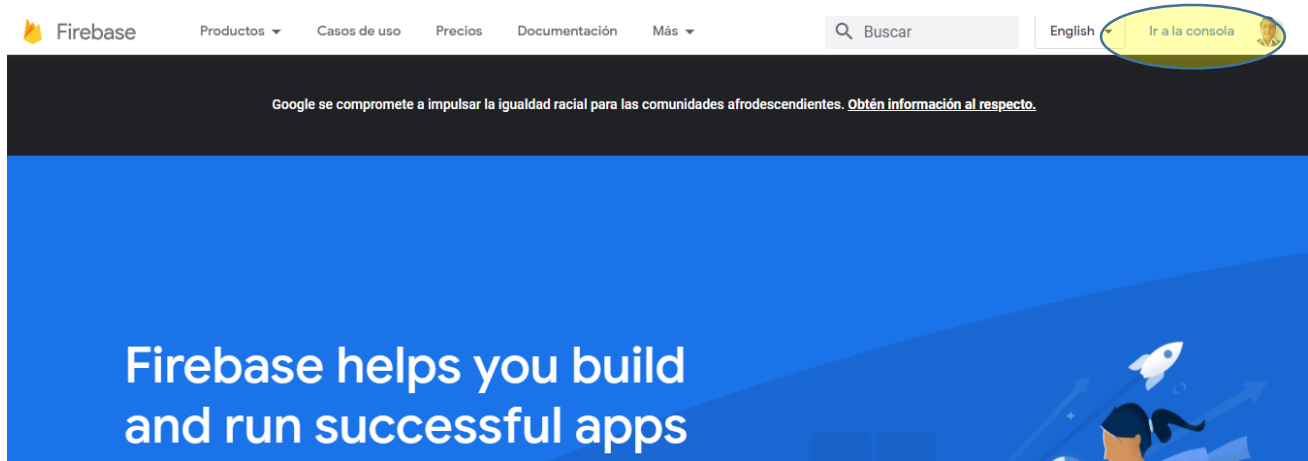
Paso 1: Ejecuta Android Studio y crea un nuevo proyecto. Selecciona **Empty Activity** y presiona **Next**. Como nombre **Guia07App**, en Package name escribe **sv.edu.udb.guia07app** en **Save location** escoge la carpeta de tu preferencia, en Language elige **Java** y el Minimum SDK selecciona API 16: **API 23: Android 6.0 (Marshmallow)** y Presiona Finish.

Paso 2: En Android Studio, inicie sesión con su correo electrónico. Puede encontrar el botón de inicio de sesión en la esquina superior derecha de Android Studio.



Paso 3: Abra el **sitio web de Firebase** e inicie sesión en él. (use la misma identificación de correo electrónico que se usa en Android Studio para iniciar sesión) <https://firebase.google.com/>


Paso 4: Después de iniciar sesión, haga clic en el botón "**Ir a la consola**" que está presente en la parte superior derecha del sitio web.



Paso 5: haga clic en " **Agregar proyecto** ".



Paso 6: Ingrese los detalles requeridos del proyecto y haga clic en enviar.



× Crear un proyecto(paso 1 de 3)

Empieza por ponerle un nombre al proyecto ⓘ

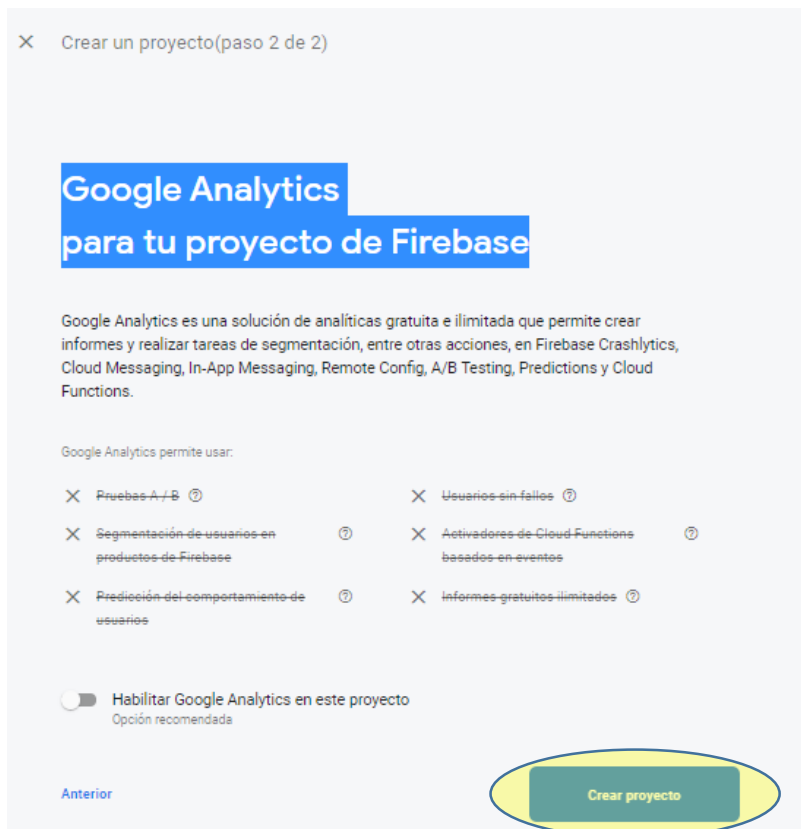
Nombre del proyecto

LoginFirebaseApp

mi-impresionante-identificación-del-proyecto

Continuar

Paso 6.1: Google Analytics para tu proyecto de Firebase, Dejar deshabilitada la opción.



× Crear un proyecto(paso 2 de 2)

Google Analytics para tu proyecto de Firebase

Google Analytics es una solución de analíticas gratuita e ilimitada que permite crear informes y realizar tareas de segmentación, entre otras acciones, en Firebase Crashlytics, Cloud Messaging, In-App Messaging, Remote Config, A/B Testing, Predictions y Cloud Functions.

Google Analytics permite usar:

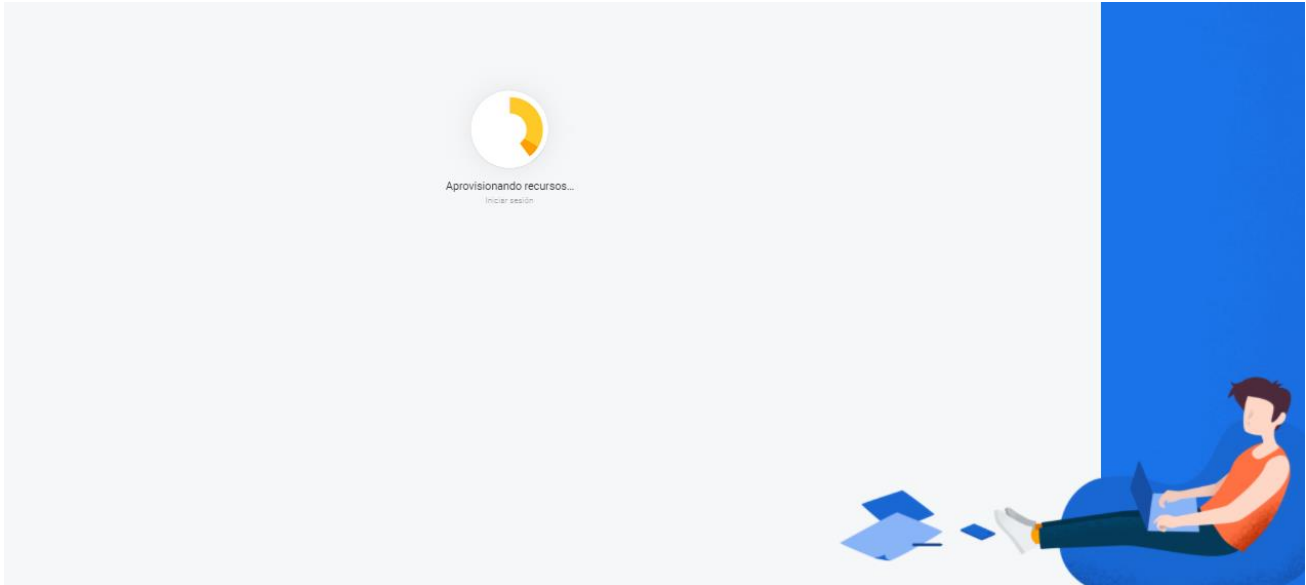
- × Pruebas A/B ⓘ
- × Segmentación de usuarios en productos de Firebase ⓘ
- × Predicción del comportamiento de usuarios ⓘ
- × Usuarios sin fallos ⓘ
- × Activadores de Cloud Functions basados en eventos ⓘ
- × Informes gratuitos ilimitados ⓘ

☐ Habilitar Google Analytics en este proyecto
Opción recomendada

Anterior

Crear proyecto

Paso 6.2: Se iniciará a construir el proyecto.



Paso 7: Después de crear un proyecto, verá la siguiente imagen del panel de control de su proyecto.

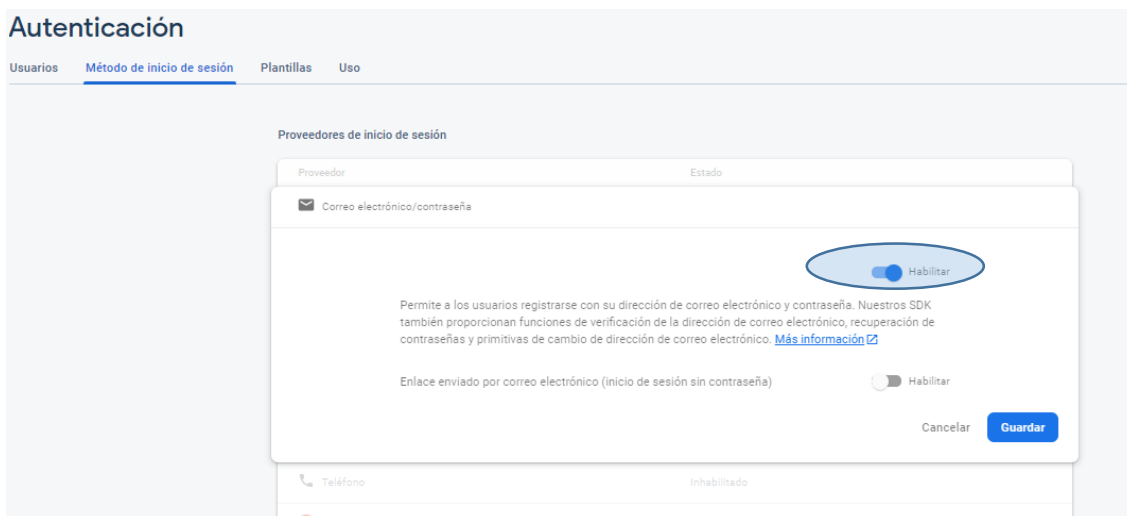


Aquí se muestran todos los servicios de Firebase y puedes usar cualquiera de ellos.

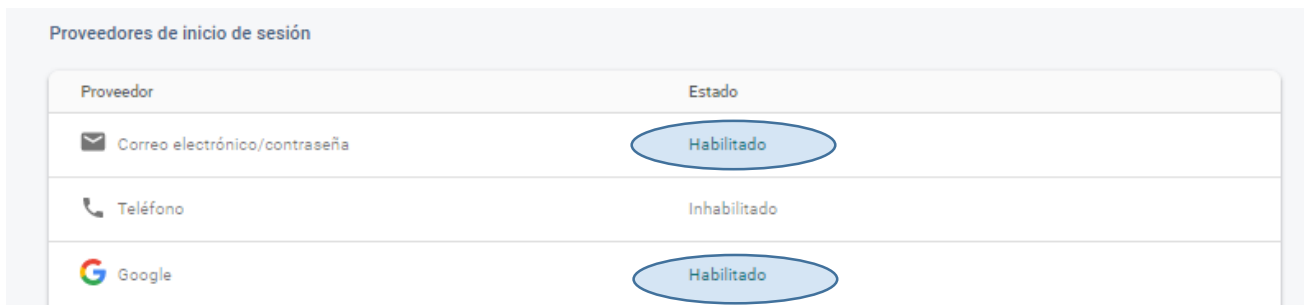
Paso 8: En la parte de autenticación, haga clic en el botón "**Autenticación**" y luego cambie a la pestaña "**Método de inicio de sesión**".



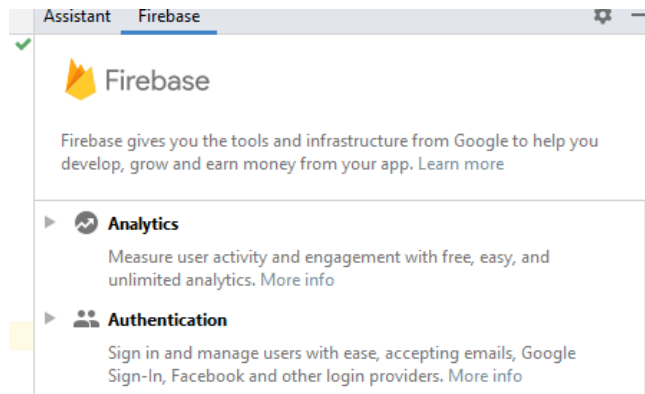
Paso 9: Haga clic en el botón editar junto a la opción "**Correo electrónico / Contraseña**" y verá la siguiente pantalla.



Paso 10: Habilite el inicio de sesión **correo electrónico/contraseña** y **Google** haga clic en "**Guardar**". Ha terminado con la parte del sitio web de Firebase.



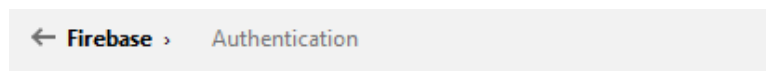
Paso 11: Regrese a su proyecto en Android Studio y haga clic en **Herramientas> Firebase> Autenticación**.



Paso 12: Agregar el proyecto Android Studio con el proyecto que hemos creado en **Firebase**. Clic en " **Autenticación de correo electrónico y contraseña**". Aquí, tenemos dos opciones:

1. **Conectarse a Firebase**
2. **Agregar autenticación de Firebase a su aplicación**

En primer lugar, tenemos que conectar nuestro proyecto a Firebase, entonces, haga clic en el botón " **Conectar a Firebase** ".



Email and password authentication

You can use Firebase Authentication to let your users sign in with their email and password-based accounts. This tutorial helps you set up an email and password authentication for your app.

[Launch in browser](#)

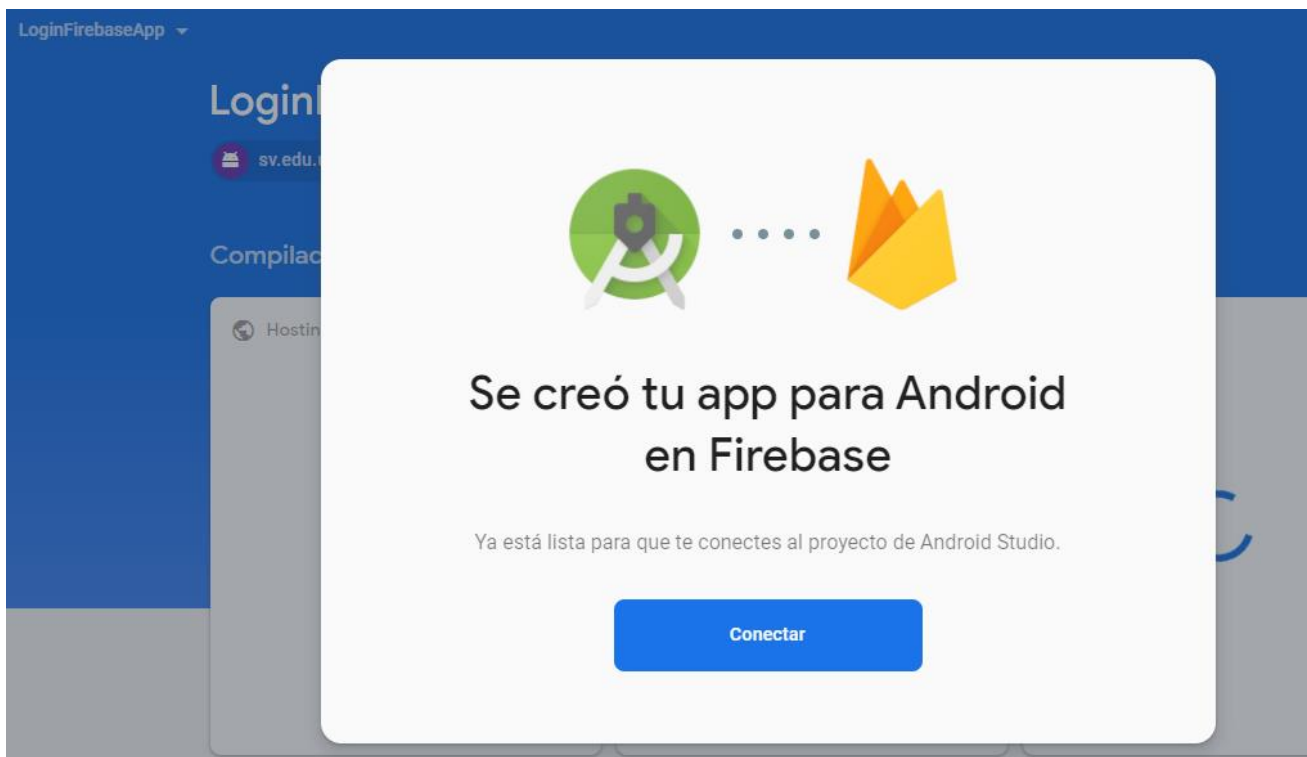
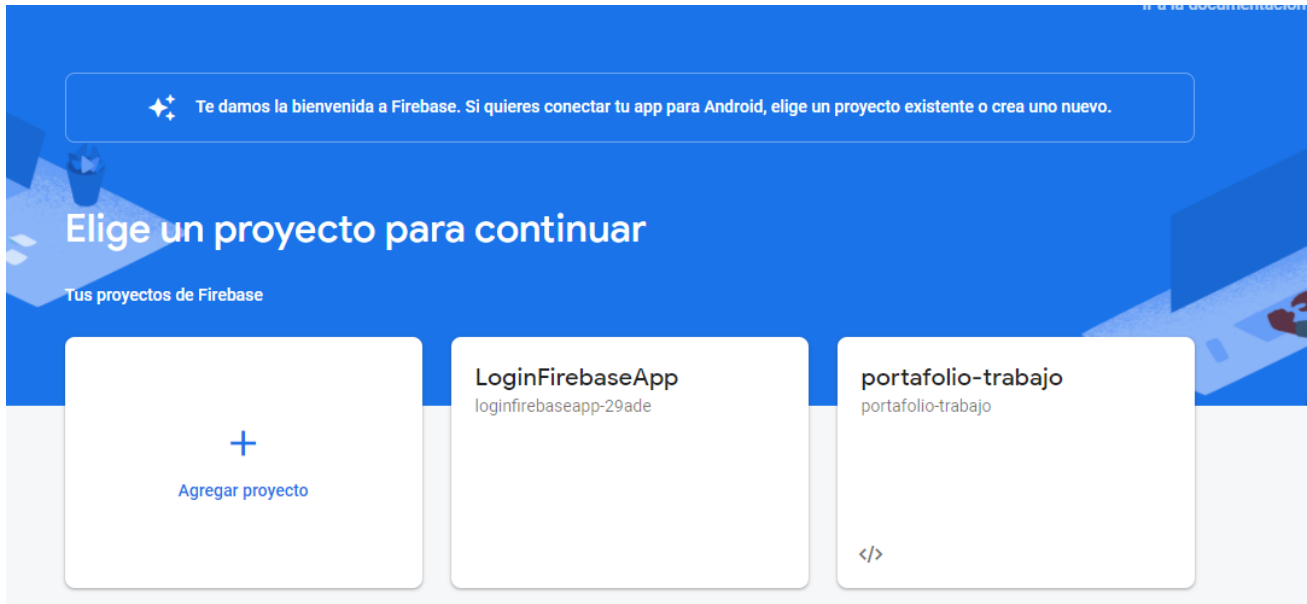
① Connect your app to Firebase

Connect to Firebase

② Add Firebase Authentication to your app

Add Firebase Authentication to your app

Paso 13: Se mostrará una lista de proyectos asociados con su correo electrónico y deberá seleccionar el proyecto que ha creado anteriormente en Firebase y luego hacer clic en " **Conectar a Firebase** ".

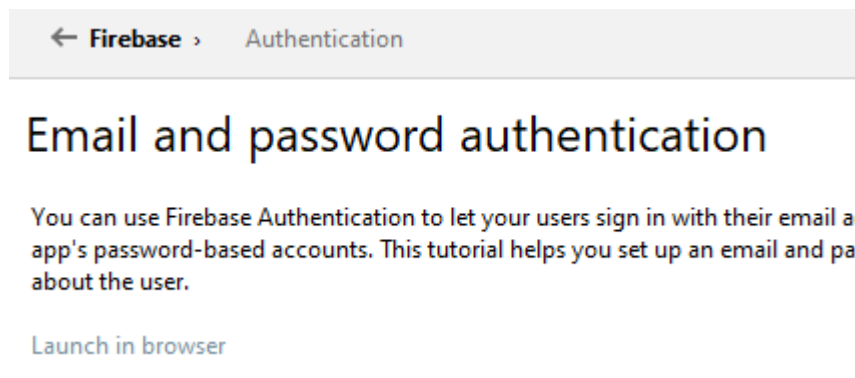




Your Android Studio project is connected to your Firebase Android app

You can now use Firebase in your project! Go back to Android Studio to start using one of the Firebase SDKs.

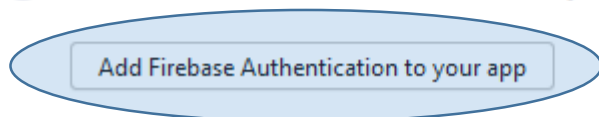
Paso 14: Ahora, su proyecto en Android Studio está conectado con el presente en Firebase. Agregar algunas dependencias a nuestros proyectos. Entonces, haga clic en "**Agregar autenticación de Firebase a su aplicación**" (esta es la segunda opción que encontramos en el paso número 12). Se abrirá un cuadro de diálogo. Haga clic en "**Aceptar cambios**" y automáticamente agregará todas las dependencias a su proyecto.

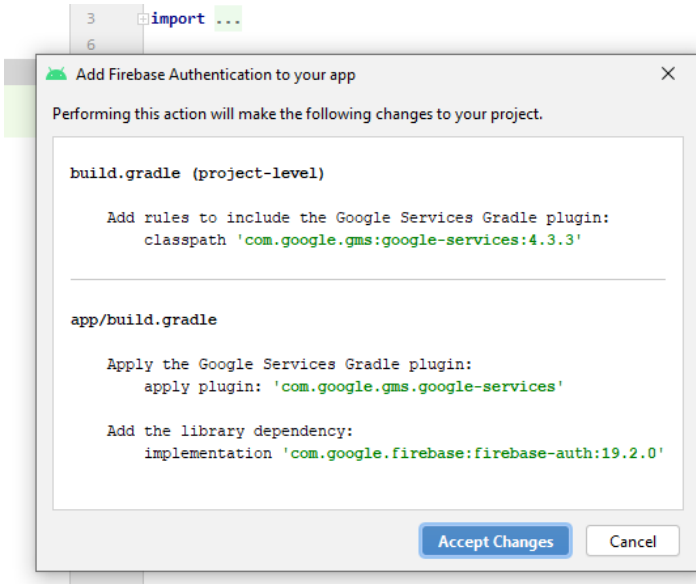


① Connect your app to Firebase

✓ Connected

② Add Firebase Authentication to your app





build.gradle (project-level)

Add rules to include the Google Services Gradle plugin:
classpath 'com.google.gms:google-services:4.3.3'

app/build.gradle

Apply the Google Services Gradle plugin:
apply plugin: 'com.google.gms:google-services'

Add the library dependency:
implementation 'com.google.firebase:firebase-auth:19.2.0'

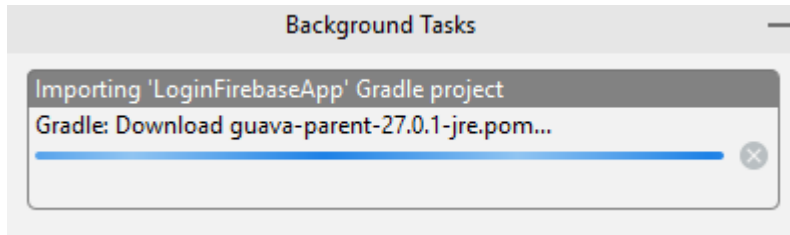
Buttons: Accept Changes, Cancel

Email and password authentication

You can use Firebase Authentication to let your users sign in with their email and password-based accounts. This tutorial helps you set up an email and password authentication provider.

[Launch in browser](#)

- 1 Connect your app to Firebase
✓ Connected
- 2 Add Firebase Authentication to your app
Add Firebase Authentication to your app
- 3 Check current auth state
Declare an instance of `FirebaseAuth`



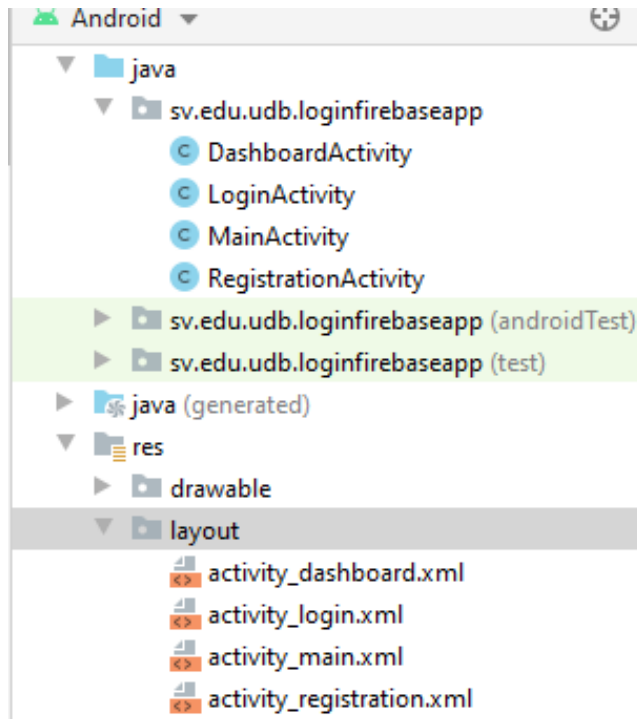
```
dependencies {

    implementation 'androidx.appcompat:appcompat:1.2.0'
    implementation 'com.google.android.material:material:1.3.0'
    implementation 'androidx.constraintlayout:constraintlayout:2.0.4'
    implementation 'com.google.firebase:firebase-auth:19.2.0'
    testImplementation 'junit:junit:4.+'
    androidTestImplementation 'androidx.test.ext:junit:1.1.2'
    androidTestImplementation 'androidx.test.espresso:espresso-core:3.3.0'
}
```

Finalmente, hemos terminado con todos los pasos necesarios para conectar nuestro proyecto de **Android Studio con Firebase for Authentication**. Ahora, pasemos a la parte de codificación.

Explicación de las actividades:

Repositorio Código : <https://github.com/AlexanderSiguenza/LoginFirebaseApp.git>



1. **activity_main** : Esta es la actividad principal de la aplicación, presenta las dos opciones de la aplicación, registrarse o login.
2. **activity_login** : Esta actividad contiene dos EditText para recibir correo electrónico y contraseña del usuario. Dos botones para iniciar sesión y registrarse.
3. **activity_registration**: Esta actividad se utiliza para registrar al usuario en la aplicación y, después de un registro exitoso. Contiene dos EditText para tomar el correo electrónico y la contraseña del usuario para registrarse.
4. **activity_dashboard**: El usuario será redirigido a esta actividad si ha iniciado sesión correctamente o ha creado una nueva cuenta

Nota: Ahora está listo para comenzar a codificar. Lo primero que debe considerar es incluir la solicitud de permiso de Internet en su archivo de manifiesto:

```
<uses-permission android:name="android.permission.INTERNET"/>
```

Pasos para crear la aplicación:

Para registrar un usuario con correo electrónico y contraseña, en primer lugar, debe declarar una instancia de **FirebaseAuth** .

```
private FirebaseAuth mAuth;
```

Ahora, en el método **onCreate ()** , inicialice la instancia de FirebaseAuth.

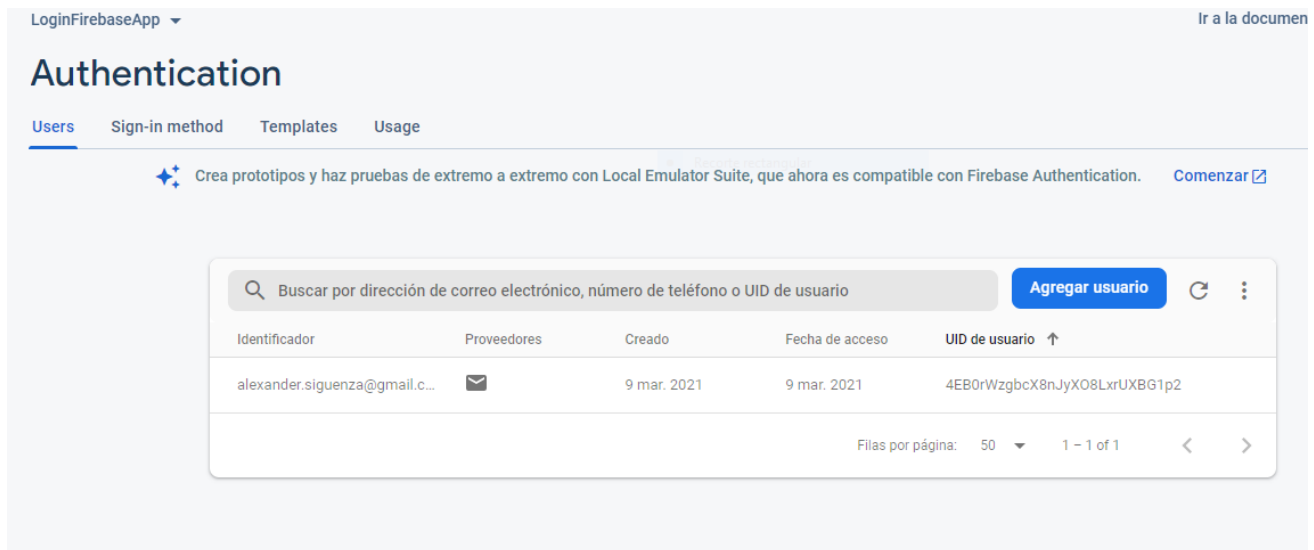
```
mAuth = FirebaseAuth.getInstance();
```

Para registrar un usuario con correo electrónico y contraseña, podemos tomar la ayuda del método **createUserWithEmailAndPassword ()** . Este método toma el correo electrónico y la contraseña como parámetro, los valida y luego crea un nuevo usuario.

```
mAuth.createUserWithEmailAndPassword(email, password)
    .addOnCompleteListener(new OnCompleteListener<AuthResult>() {
        @Override
        public void onComplete(@NonNull Task<AuthResult> task) {
            if (task.isSuccessful()) {
                Toast.makeText(getApplicationContext(), "Registration successful!", Toast.LENGTH_LONG).show();
                progressBar.setVisibility(View.GONE);

                Intent intent = new Intent(RegistrationActivity.this, LoginActivity.class);
                startActivity(intent);
            }
            else {
                Toast.makeText(getApplicationContext(), "Registration failed! Please try again later",
                Toast.LENGTH_LONG).show();
                progressBar.setVisibility(View.GONE);
            }
        }
    });
```

Eso es. Su usuario se registrará y podrá ver al usuario registrado en el sitio web de Firebase en la sección Autenticación y luego en la pestaña Usuarios. Si está intentando registrarse con el mismo correo electrónico más de una vez, recibirá un error o, en palabras simples, su tarea no se realizó correctamente.



Iniciar sesión como usuario con correo electrónico y contraseña

En la sección anterior, vimos cómo registrar un usuario con correo electrónico y contraseña. Ahora, necesitamos iniciar sesión como usuario autenticando el correo electrónico y la contraseña ingresados por el usuario.

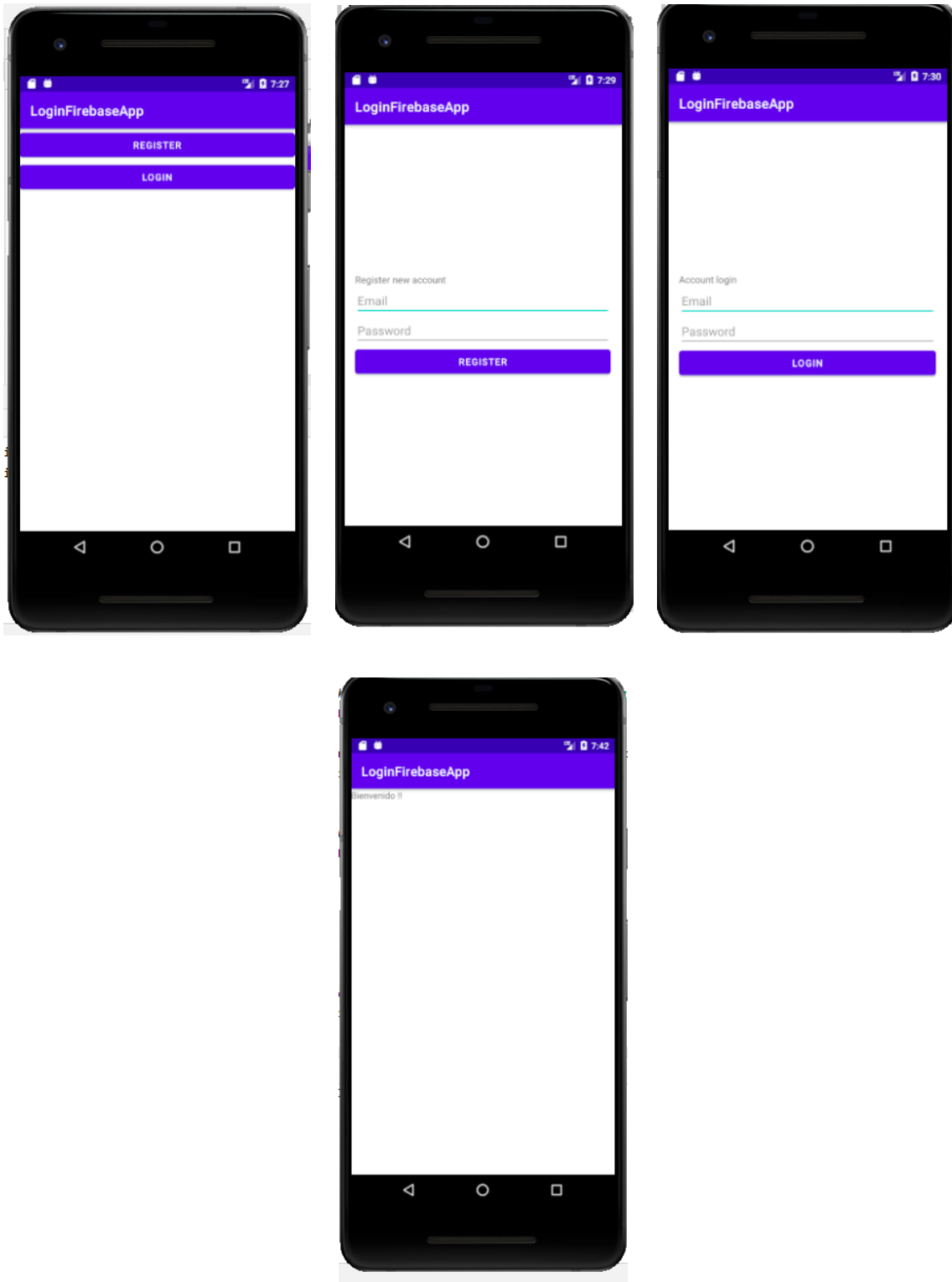
Para registrar un usuario con correo electrónico y contraseña, tenemos un método llamado **signInWithEmailAndPassword ()**.

Este método toma el correo electrónico y la contraseña como parámetro, los valida y luego registra a un usuario en su aplicación si la validación es exitosa. A continuación, se muestra el código para el mismo:

```
mAuth.signInWithEmailAndPassword(email, password)
    .addOnCompleteListener(new OnCompleteListener<AuthResult>() {
        @Override
        public void onComplete(@NonNull Task<AuthResult> task) {
            if (task.isSuccessful()) {
                Toast.makeText(getApplicationContext(), "Login successful!", Toast.LENGTH_LONG).show();
                progressBar.setVisibility(View.GONE);

                Intent intent = new Intent(LoginActivity.this, DashboardActivity.class);
                startActivity(intent);
            }
            else {
                Toast.makeText(getApplicationContext(), "Login failed! Please try again later",
                    Toast.LENGTH_LONG).show();
                progressBar.setVisibility(View.GONE);
            }
        }
    });
```

La aplicación debería de tener una apariencia similar:



II. INTRODUCCION TEORICA

Introducción a RealTime Database de Firebase

Firebase Realtime Database es una base de datos alojada en la nube. Los datos se almacenan en formato JSON y se sincronizan en tiempo real con cada cliente conectado. Todos los clientes comparten una instancia de Realtime Database y reciben actualizaciones automáticamente con los datos más recientes.

En lugar de solicitudes HTTP típicas, Firebase Realtime Database usa la sincronización de datos (cada vez que cambian los datos, los dispositivos conectados reciben esa actualización en milisegundos). Proporciona experiencias colaborativas y envolventes sin pensar en el código de red.

Las apps de Firebase continúan respondiendo, incluso sin conexión, dado que el SDK de Firebase Realtime Database hace que los datos persistan. Cuando se restablece la conexión, el dispositivo cliente recibe los cambios que faltaban y los sincroniza con el estado actual del servidor.

Realtime Database es una base de datos NoSQL y, como tal, tiene diferentes optimizaciones y funcionalidades en comparación con una base de datos relacional. La API de Realtime Database está diseñada para permitir solo operaciones que se puedan ejecutar rápidamente. Eso permite crear una excelente experiencia de tiempo real que puede servir a millones de usuarios sin afectar la capacidad de respuesta.

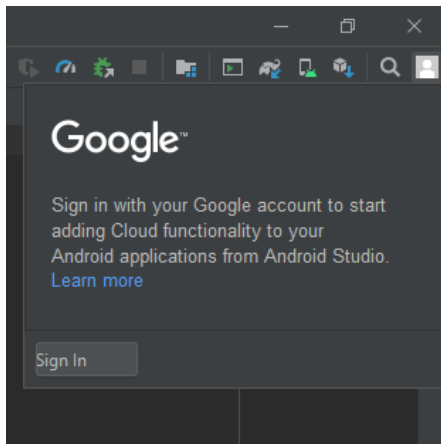
III. PROCEDIMIENTO

Inicio de sesión y RealTime Database de Firebase:

Para usar Firebase **RealTime Database** en nuestra aplicación, necesitamos conectar nuestro proyecto, es decir, el proyecto de Android Studio a Firebase. Los siguientes son los pasos que se utilizan para conectar un proyecto de Android a Firebase

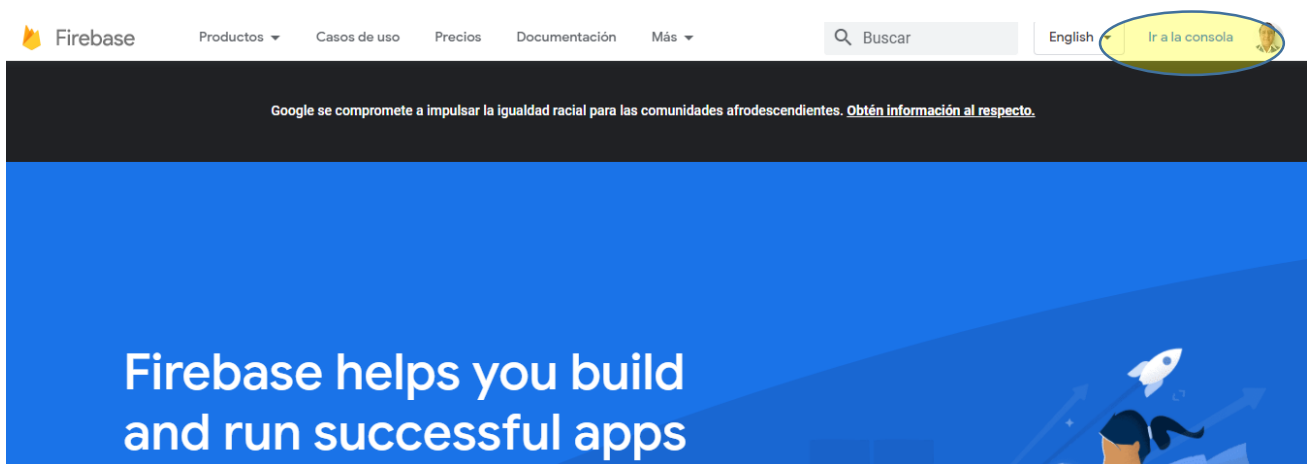
Paso 1: Ejecuta Android Studio y crea un nuevo proyecto. Selecciona **Empty Activity** y presiona **Next**. Como nombre **Guia08App**, en Package name escribe **sv.edu.udb.guia08app** en **Save location** escoge la carpeta de tu preferencia, en Language elige **Java** y el Minimum SDK selecciona API 19 y Presiona Finish.

Paso 2: En Android Studio, inicie sesión con su correo electrónico. Puede encontrar el botón de inicio de sesión en la esquina superior derecha de Android Studio.



Paso 3: Abra el **sitio web de Firebase** e inicie sesión en él. (use la misma identificación de correo electrónico que se usa en Android Studio para iniciar sesión) <https://firebase.google.com/>

Paso 4: Después de iniciar sesión, haga clic en el botón " **Ir a la consola** " que está presente en la parte superior derecha del sitio web.



Paso 5: haga clic en " **Agregar proyecto** ".



Paso 6: Ingrese los detalles requeridos del proyecto y haga clic en enviar.

Comencemos con el nombre de tu proyecto [?]

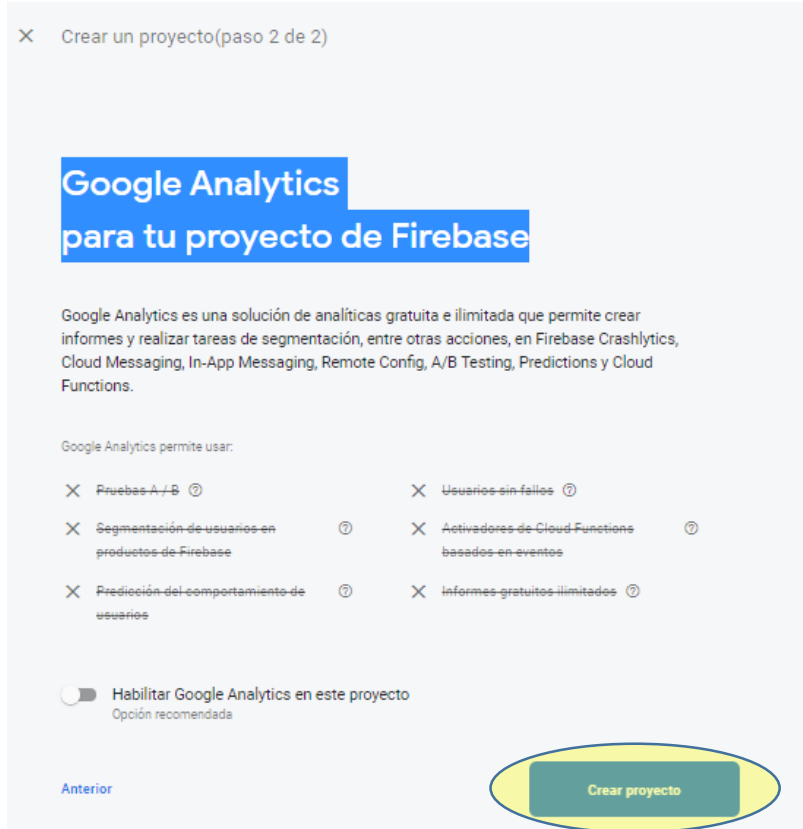
Nombre del proyecto

FirebaseDBRealTime

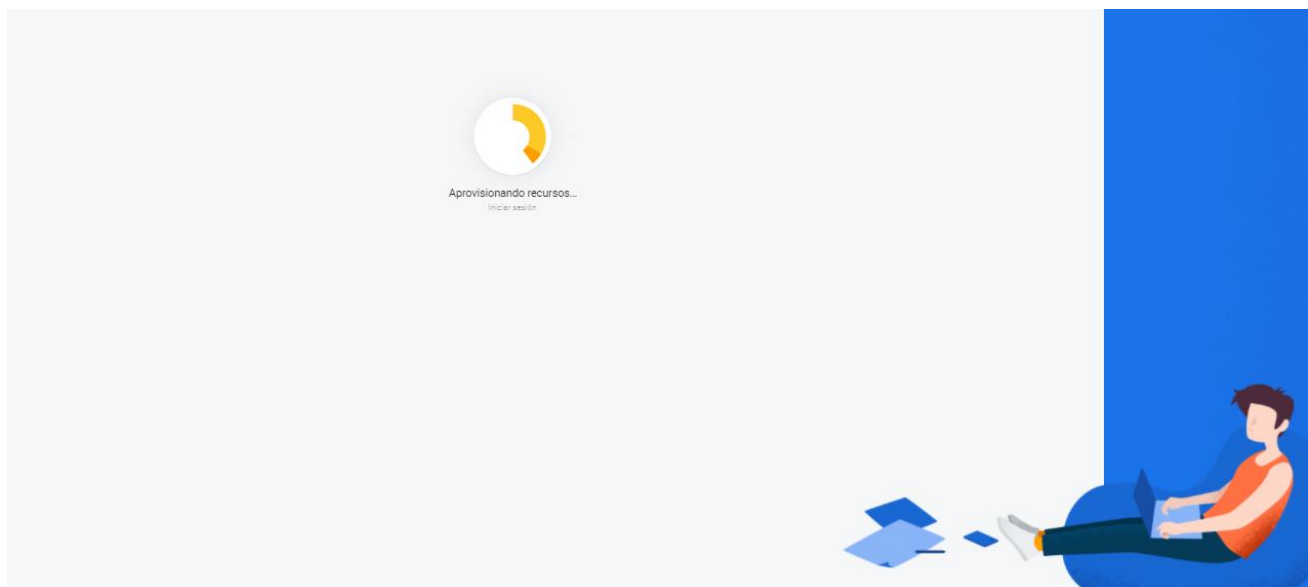
fir-dbrealttime-b4f26

Continuar

Paso 6.1: Google Analytics para tu proyecto de Firebase, Dejar deshabilitada la opción.



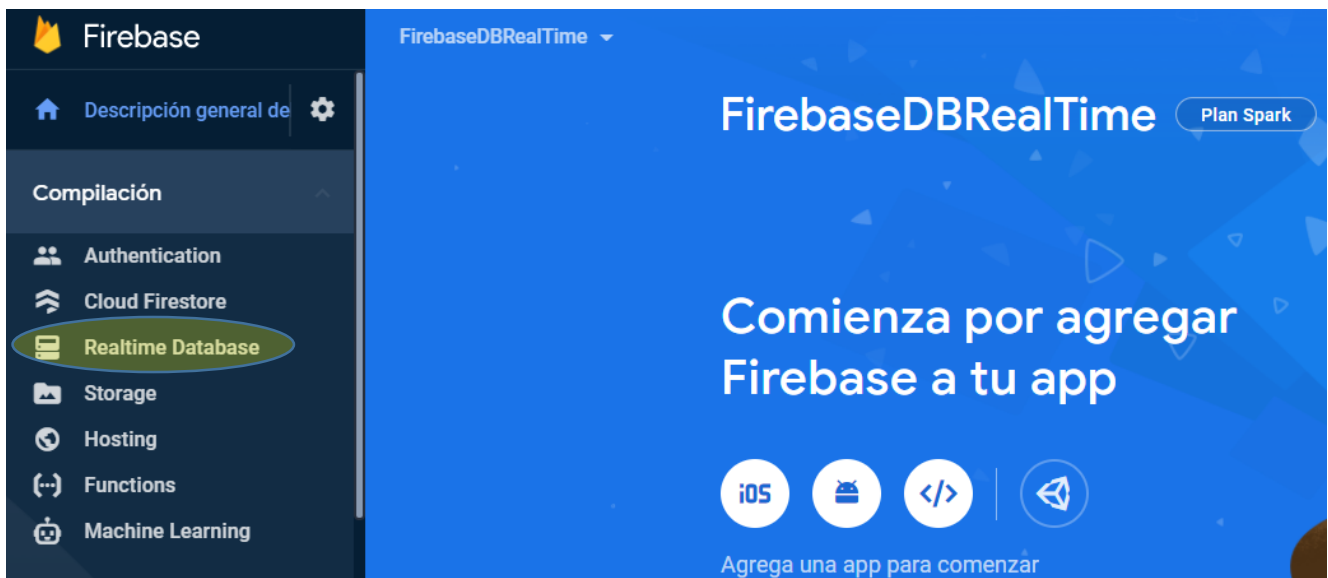
Paso 6.2: Se iniciará a construir el proyecto.



Paso7: Cuando se crea tu proyecto aparecerá una ventana como la siguiente: (Selecciona **Continuar**)

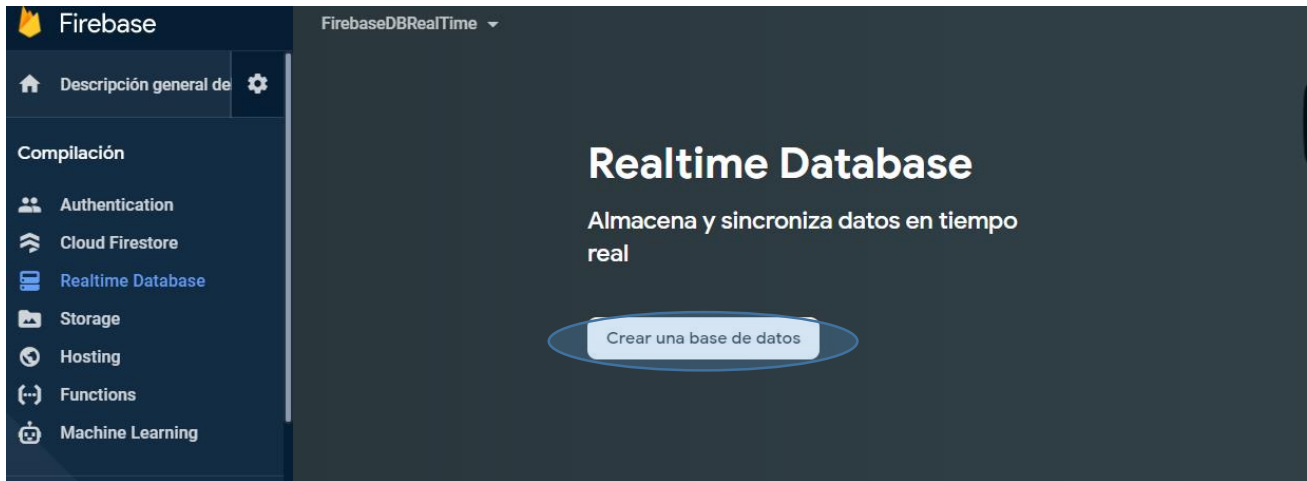


Paso 8: Después de crear un proyecto, verá la siguiente imagen del panel de control de su proyecto.

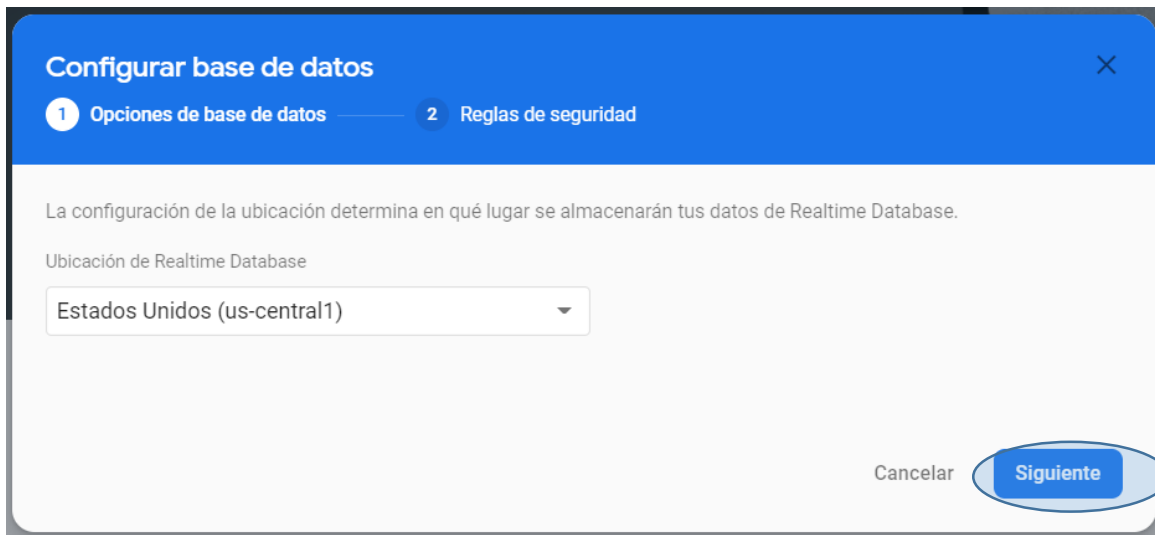


Aquí se muestran todos los servicios de Firebase y puedes usar cualquiera de ellos, selecciona **RealTime Database**

Paso 9: En la siguiente ventana selecciona la opción **“Crear una base de datos”**



Paso 10: Haga clic en el botón Siguiente, dejando la ubicación por default que aparece.



Paso 11: Habilite la opción “Comenzar en modo prueba” y luego selecciona “Habilitar”.

Configurar base de datos

1 Opciones de base de datos — 2 Reglas de seguridad

Cuando defines la estructura de los datos, deberás crear reglas para protegerlos.
[Más información](#)

☐ Comenzar en modo bloqueado
 De forma predeterminada, tus datos son privados. El acceso de lectura/escritura de los clientes solo se otorgará como se indica en las reglas de seguridad.

☒ Comenzar en modo de prueba
 Para permitir una configuración rápida, los datos se abren de forma predeterminada. Sin embargo, debes actualizar las reglas de seguridad dentro de 30 días a fin de habilitar el acceso de lectura/escritura a largo plazo para los clientes.

```

{
  "rules": {
    ".read": "now < 1618380000000", // 2021-4-14
    ".write": "now < 1618380000000", // 2021-4-14
  }
}

```

! Las reglas de seguridad predeterminadas del modo de prueba permiten que cualquier usuario con acceso a tu referencia de base de datos pueda ver, editar y borrar todos los datos durante los siguientes 30 días.

Cancelar **Habilitar**

Esto es necesario para que podamos acceder a nuestra base de datos más adelante sin restricción desde nuestra app Android. (en modo prueba)

Ahora ya cuentas con una base de datos en tiempo real:

Realtime Database

Datos Reglas Copias de seguridad Uso

✦ Crea prototipos y haz pruebas de extremo a extremo con Local Emulator Suite, que ahora es compatible con Firebase Authentication. [Comenzar](#)

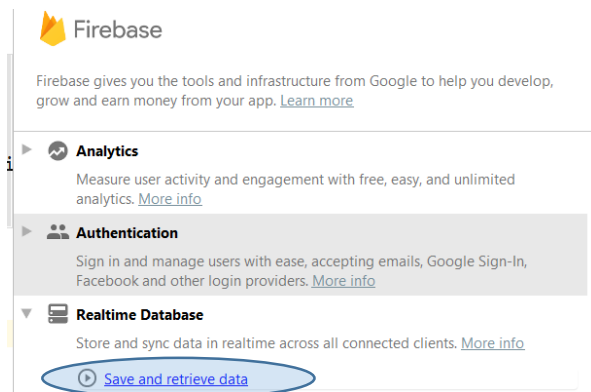
🔗 https://fir-dbrealttime-b4f26-default-rtdb.firebaseio.com/

fir-dbrealttime-b4f26-default-rtdb:	null	+	×
-------------------------------------	------	---	---

Paso 12: Edita las reglas de acceso a la base de datos para que sean públicas sin restricción de tiempo de la siguiente forma: Selecciona Reglas, Luego Editar reglas y editarlas como se ve en pantalla dejando los valores de .write y .read ambos en true Y luego selecciona Publicar.



Paso 13: Regrese a su proyecto en Android Studio y haga clic en **Herramientas > Firebase > RealTime Database > Save and Retrieve data**



Paso 14: Agregar el proyecto Android Studio con el proyecto que hemos creado en **Firebase**.

Contamos con 2 opciones

3. **Conectarse a Firebase**

4. **Agregar RealTime Database a su aplicación**

En primer lugar, tenemos que conectar nuestro proyecto a Firebase, entonces, haga clic en el botón **"Conectar a Firebase"**.

1 Connect your app to Firebase

Paso 15: Se mostrará una lista de proyectos asociados con su correo electrónico y deberá seleccionar el proyecto que ha creado anteriormente en Firebase y luego hacer clic en " **Conectar a Firebase** ". (Tal como lo hizo en la guía anterior)



Your Android Studio project is connected to your Firebase Android app

You can now use Firebase in your project! Go back to Android Studio to start using one of the Firebase SDKs.

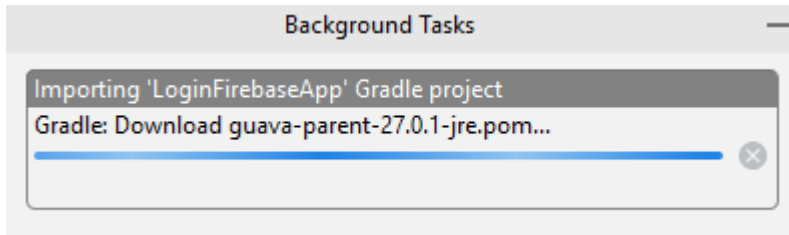
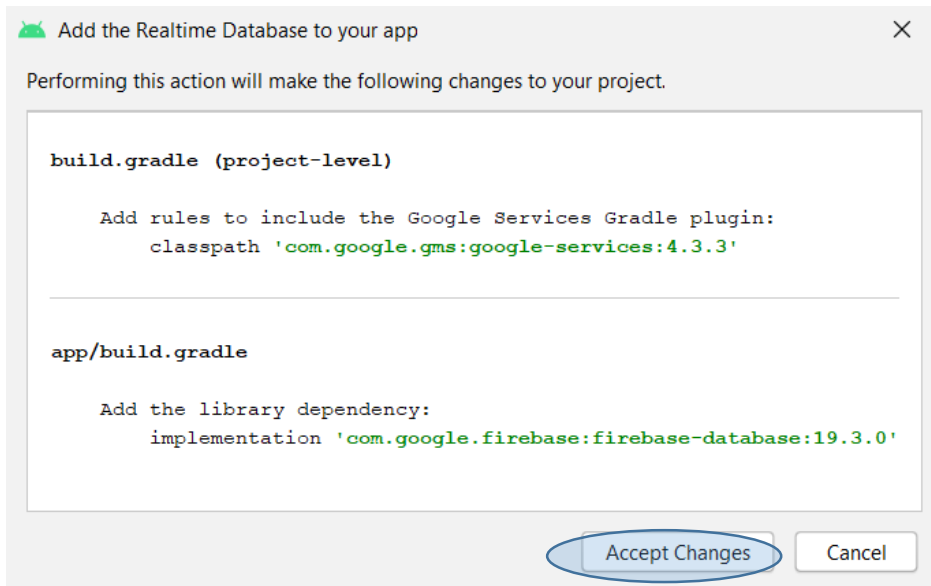
Paso 14: Ahora, su proyecto en Android Studio está conectado con el presente en Firebase. Agregar algunas dependencias a nuestros proyectos. Entonces, haga clic en " **Agregar RealTime Database en tu app** ". Se abrirá un cuadro de diálogo. Haga clic en " **Aceptar cambios** " y automáticamente agregará todas las dependencias a su proyecto.

① Connect your app to Firebase

✓ Connected

② Add the Realtime Database to your app

Add the Realtime Database to your app



Se ha actualizado el archivo **Build.Gradle** (Nivel de app) con el objeto de contar con las clases que nos permitirán hacer nuestra codificación de la app Android y conectarnos a la base de datos.

```

26 dependencies {
27     implementation fileTree(dir: "libs", include: ["*.jar"])
28     implementation 'androidx.appcompat:appcompat:1.2.0'
29     implementation 'androidx.constraintlayout:constraintlayout:2.0.4'
30     implementation 'com.google.firebase:firebase-database:16.0.4'
31     implementation 'com.google.android.material:material:1.3.0'
32     testImplementation 'junit:junit:4.12'
33     androidTestImplementation 'androidx.test.ext:junit:1.1.2'
34     androidTestImplementation 'androidx.test.espresso:espresso-core:3.3.0'
35 }
36

```

Finalmente, hemos terminado con todos los pasos necesarios para conectar nuestro proyecto de **Android Studio con RealTime Database**. Ahora, pasemos a la parte de codificación.

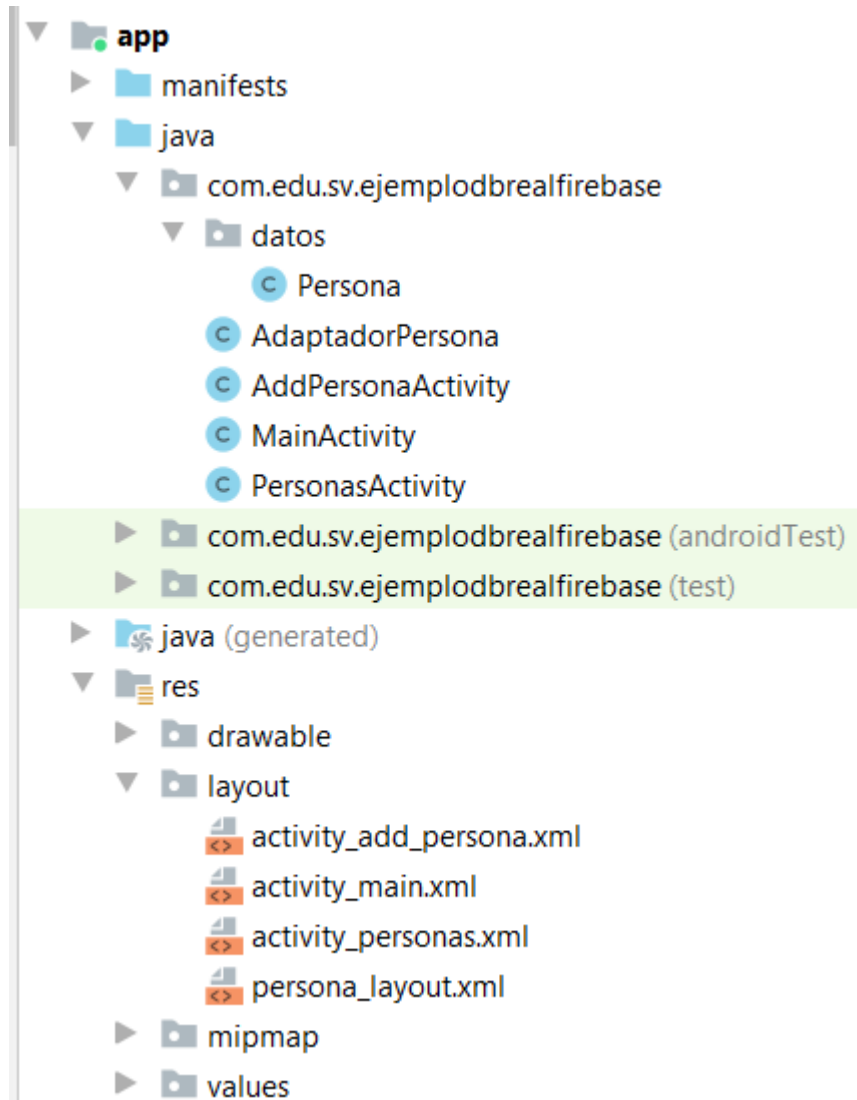
Aplicacion de registro de personas.

Nuestra aplicación tiene por objetivo llevar un control de personas, pudiendo registrar por cada persona su DUI y su nombre.

La aplicación podrá realizar lo que se conoce como un CRUD (Create, Read, Update, Delete) sobre la base de datos de Firebase.

Explicación de las actividades y clases.

Repositorio Código : <https://github.com/Alexjimenez7/RealTimeDataBase/tree/master>



5. **MainActivity:** Esta es la actividad principal de la aplicación que se crea por default, pero no la utilizaremos para este ejemplo.

6. **PersonasActivity** : Esta actividad contiene una lista para mostrar todas las personas que vamos a guardar en nuestra base de datos
7. **AddPersonaActivity** : Esta actividad se utiliza agregar o editar una persona (Tiene una doble función)
8. **AdaptadorPersona**: Representa a un adaptador personalizado para formar una lista que mostrara el DUI y el nombre de cada persona de nuestra base de datos.
9. **datos.Persona**: Representa el modelo que contiene los campos o nodos que se guardaran por cada registro en nuestra base de datos.

Nota : Asegúrate que la actividad que inicie la aplicación sea **PersonaActivity**. Esto lo puedes corregir en tu archivo **AndroidManifest.xml** (Provisto en el repositorio de **github**)

```
<activity android:name=".MainActivity">

</activity>
<activity android:name=".PersonasActivity">
    <intent-filter>
        <action android:name="android.intent.action.MAIN" />

        <category android:name="android.intent.category.LAUNCHER" />
    </intent-filter>
</activity>
<activity android:name=".AddPersonaActivity"/>
```

Elementos importantes en una app que utilice RealTime Database de Firebase.

```
public static FirebaseDatabase database = FirebaseDatabase.getInstance();
public static DatabaseReference refPersonas = database.getReference("personas");
```

Primero será necesario crear un objeto de la clase FirebaseDatabase Y luego apuntar a una referencia o nodo dentro de nuestra base de datos. A partir de este nodo vamos a poder comenzar a agregar sub nodos en la base de datos.

Cuando contemos con la referencia podremos agregar un objeto que estará pendiente de los cambios que se produzcan en la base de datos. De esta forma podremos actualizar la lista de las personas que estaremos mostrando.

```
refPersonas.addValueEventListener(new ValueEventListener() {
    @Override
    public void onDataChange(DataSnapshot dataSnapshot) {
        // Procedimiento que se ejecuta cuando hubo algun cambio
```

```

// en la base de datos
// Se actualiza la coleccion de personas
personas.removeAll(personas);
for (DataSnapshot dato : dataSnapshot.getChildren()) {
    Persona persona = dato.getValue(Persona.class);
    persona.setKey(dato.getKey());
    personas.add(persona);
}

AdaptadorPersona adapter = new AdaptadorPersona(PersonasActivity.this,
    personas );
listaPersonas.setAdapter(adapter);

}
@Override
public void onCancelled(DatabaseError databaseError) {
}
});

```

A la hora de querer agregar o editar un nodo o registro en nuestra base de datos podremos proceder de la siguiente forma:

```

if (accion.equals("a")) { //Agregar usando push()
    PersonasActivity.refPersonas.push().setValue(persona);
}
else // Editar usando setValue
{
    PersonasActivity.refPersonas.child(key).setValue(persona);
}

```

push().setValue(persona) es la forma en que podemos pasar un objeto para ser agregado a nuestra base de datos.

Child(key).setValue(persona) es la forma en que podemos acceder a un nodo conociendo su identificador (key) y actualizando los datos con un objeto persona.

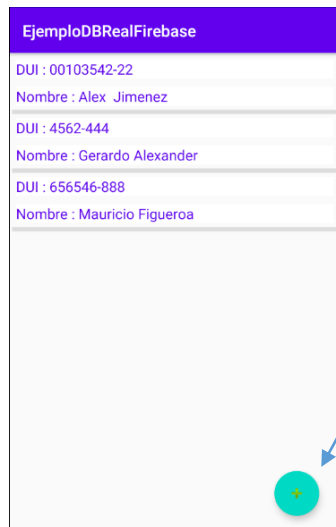
También será posible eliminar un nodo conociendo su identificador (key) y usando el método `removeValue()`

```

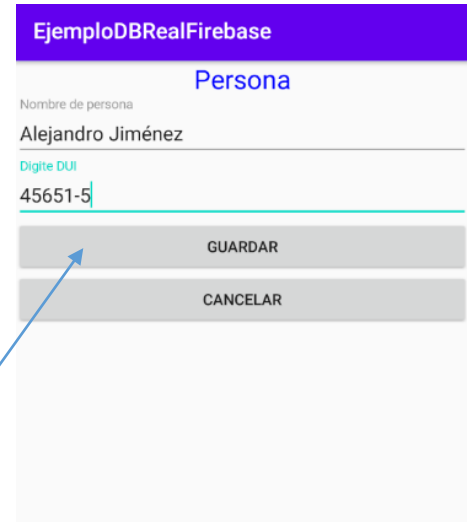
PersonasActivity.refPersonas
    .child(personas.get(position).getKey()).removeValue();

```

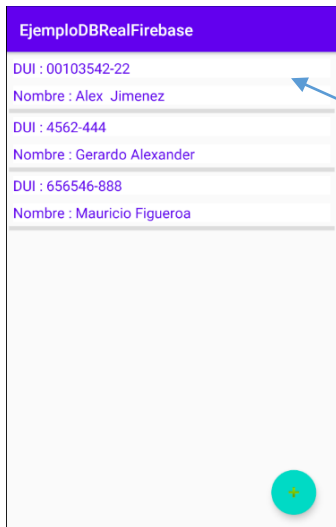
La aplicación debería de tener una apariencia similar:



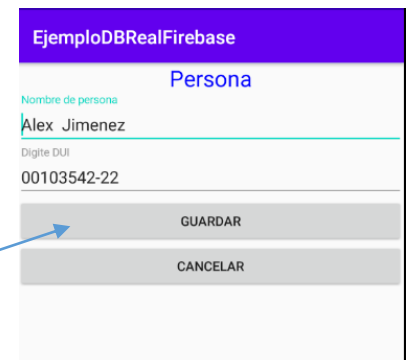
Puedes usar este botón para agregar nuevos registros de personas



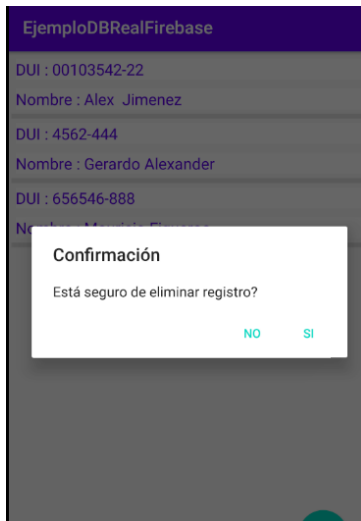
Posteriormente podrás agregar una nueva persona y dar clic en el botón guardar



Tambien puedes dar en en cada registro con el objeto de poderlo editar

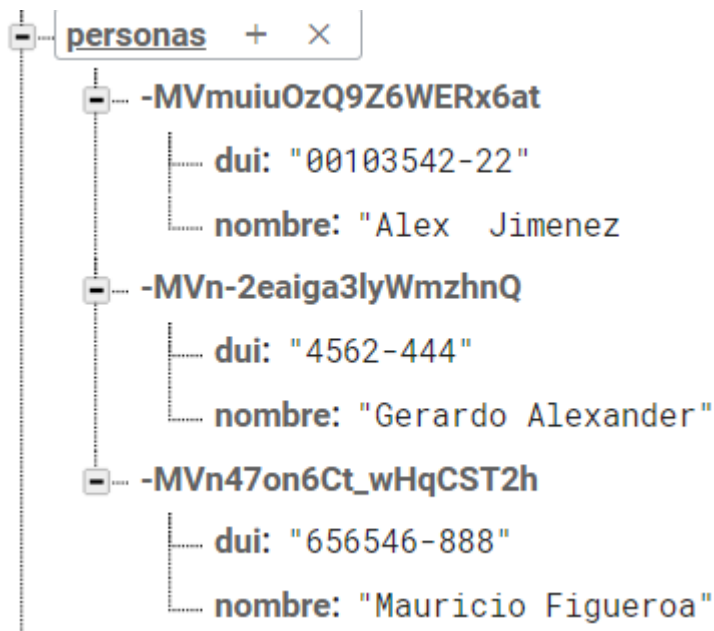


Posteriormente podrás modificar a la persona seleccionada y seleccionar guardar para actualizar los datos



También podrás mantener presionado uno de los registros por más de 2 segundos y se presentara una pregunta si estás seguro de eliminar el registro. (Esta solo es una opción para hacerlo, tu podrás decidir cómo seleccionar un registro o eliminarlo)

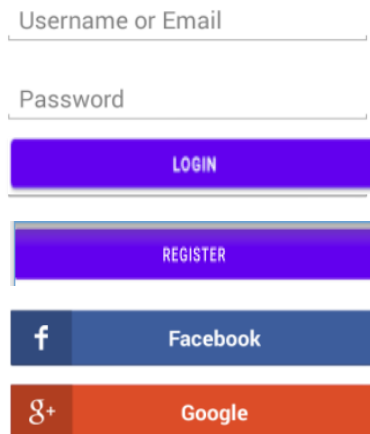
La base de datos en Firebase debería verse aproximadamente de la siguiente forma:
(Cada persona tiene una llave o key generado por firebase que identifica a cada nodo como único)



V. DISCUSION DE RESULTADOS

1. Modificar el ejemplo anterior y darle una apariencia como la siguiente imagen, agregar conectarse con Google y Facebook

LoginFirebaseApp



A login form with the following elements:

- A text input field labeled "Username or Email".
- A text input field labeled "Password".
- A blue button labeled "LOGIN".
- A blue button labeled "REGISTER".
- A Facebook login button with the Facebook 'f' logo and the text "Facebook".
- A Google login button with the Google 'G+' logo and the text "Google".

2. Modificar el ejemplo anterior y agregar los ítems: Fecha de nacimiento, Genero de la persona, peso y altura de la persona.

VII. BIBLIOGRAFIA

- firebase.google. Autentica con Firebase en Android. Recuperado 6 de marzo de 2021, de <https://firebase.google.com/docs/auth/android/custom-auth?hl=es>
- RealTime Database de Firebase. Recuperado 15 de marzo de 2021, de <https://firebase.google.com/docs/database?authuser=0>