	UNIVERSIDAD DON BOSCO FACULTAD DE INGENIERÍA ESCUELA DE COMPUTACIÓN	
CICLO: I/2021	GUIA DE LABORATORIO #08	
	Nombre de la Práctica:	Guía #08- Firebase RealTime Database
	MATERIA:	Desarrollo de Software para Móviles

I. OBJETIVOS

1. Conocer los fundamentos de una base de datos en tiempo real de Firebase.
2. Desarrollar una aplicación CRUD que utilice RealTime Database de Firebase.

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.

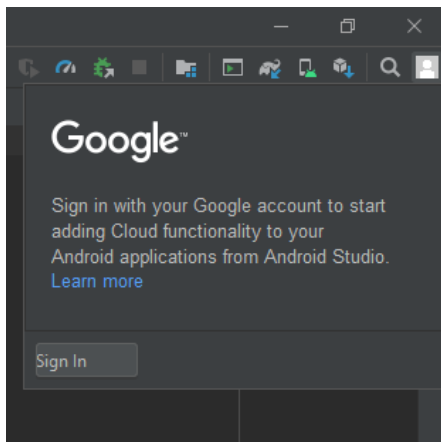
III. PROCEDIMIENTO

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

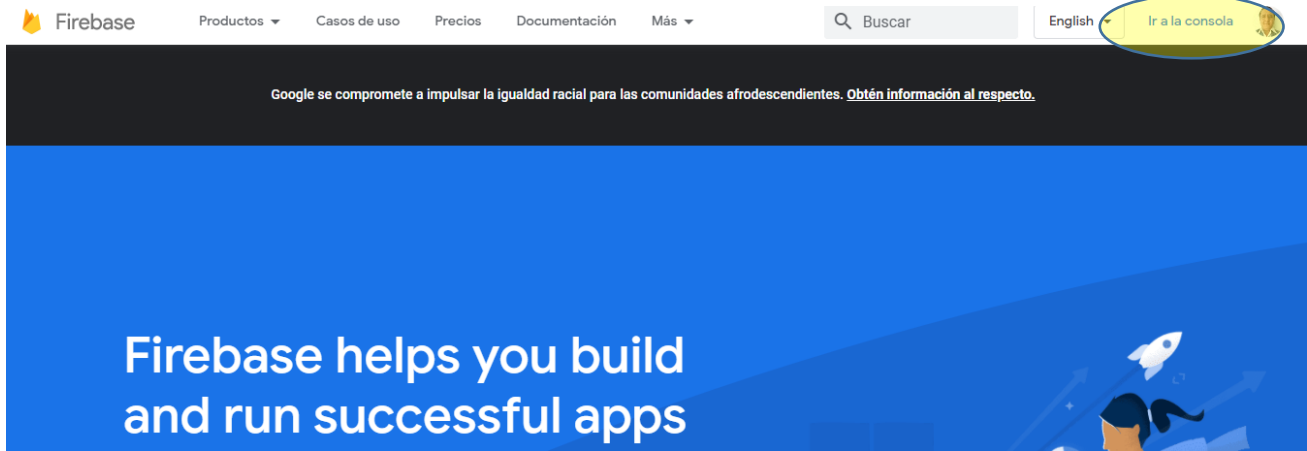
Paso 1: Ejecuta Android Studio y crea un nuevo proyecto. Selecciona **Empty Activity** y presiona **Next**. Como nombre **Guia08App**, en Package name escribe **sv.edu.udb.guia08app** en **Save location** escoge la carpeta de tu preferencia, en Language elige **Java** y el Minimum SDK selecciona API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

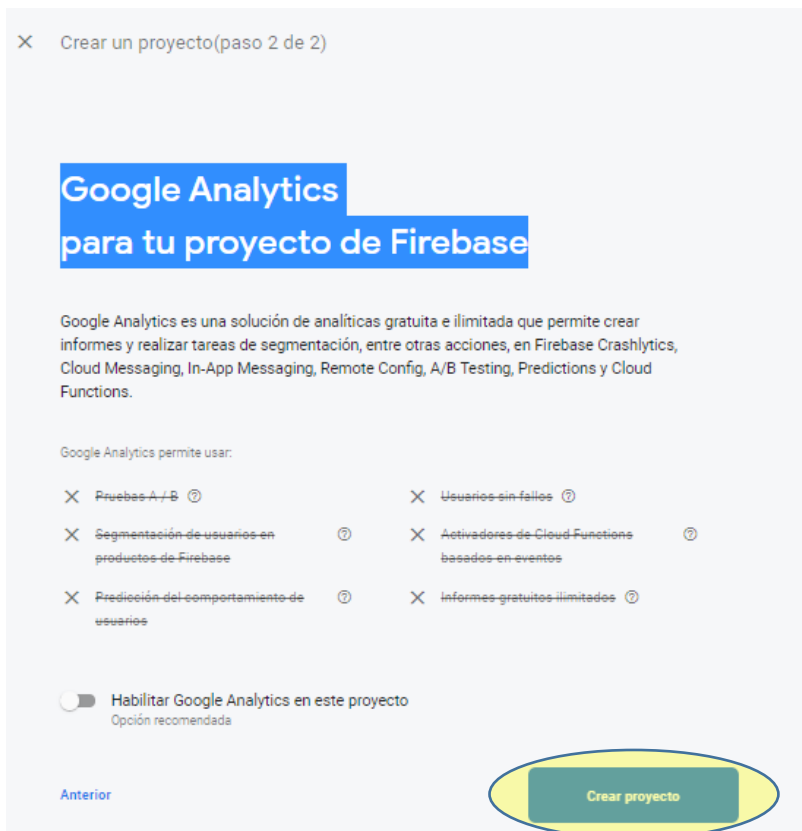
Nombre del proyecto

FirebaseDBRealTime

 fir-dbrealttime-b4f26

Continuar

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



✕ 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:

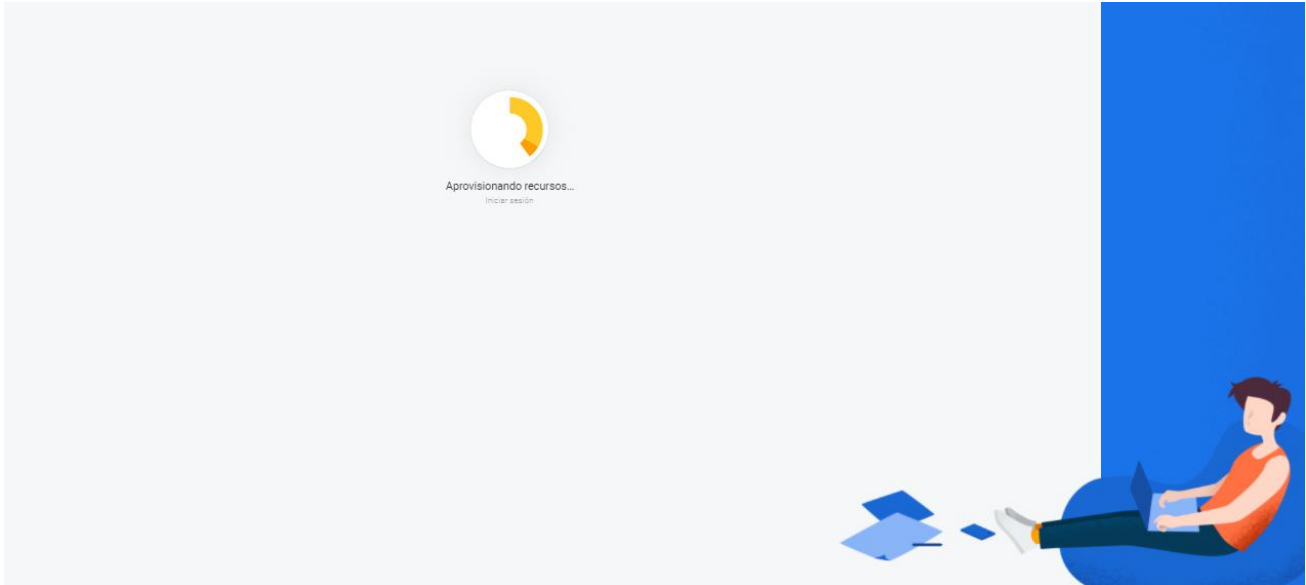
- ✕ Pruebas A/B[?]
- ✕ Usuarios sin fallos[?]
- ✕ Segmentación de usuarios en productos de Firebase[?]
- ✕ Activadores de Cloud Functions basados en eventos[?]
- ✕ Predicción del comportamiento de usuarios[?]
- ✕ Informes gratuitos ilimitados[?]

☐ Habilitar Google Analytics en este proyecto
Opción recomendada

[Anterior](#)

Crear proyecto

Paso 6.2: Se iniciará a construir el proyecto.



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

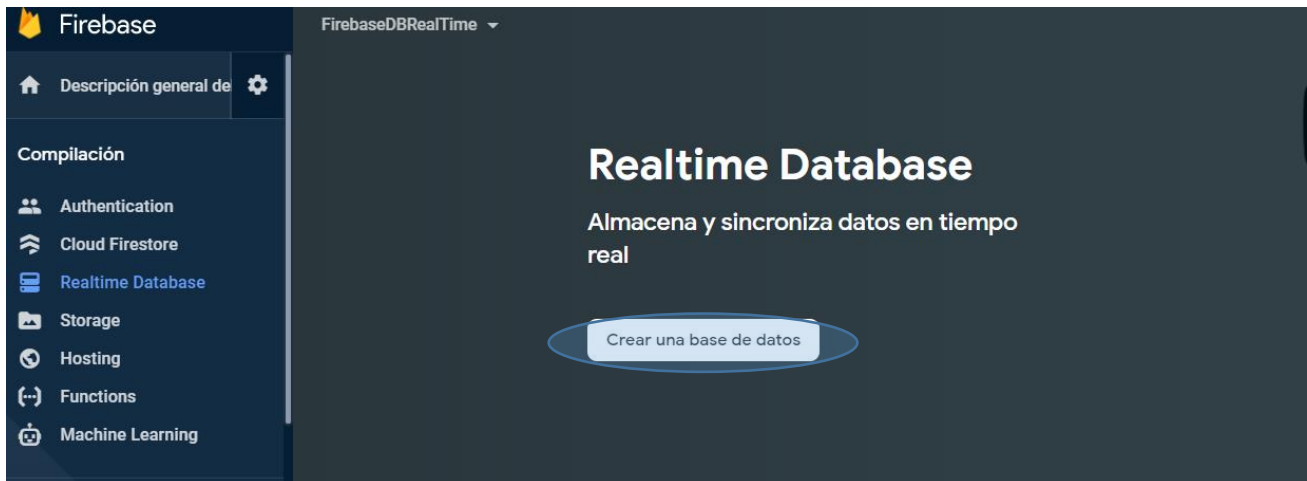


Paso 8: Después de crear un proyecto, verá la siguiente imagen del panel de control de su proyecto.

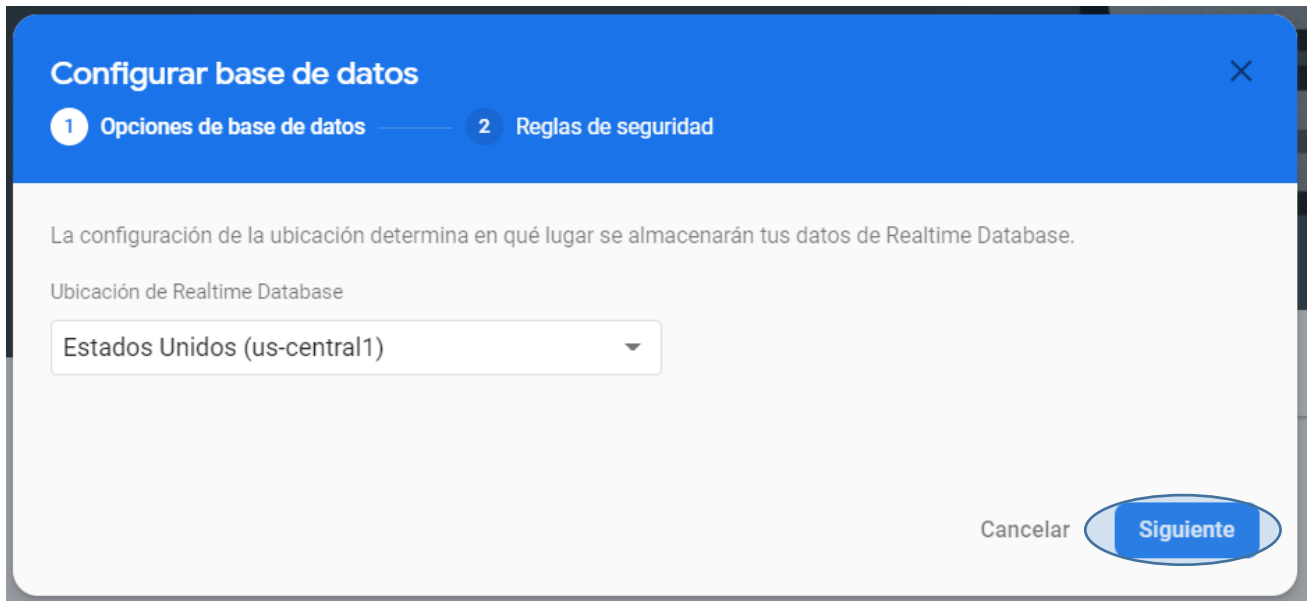


Aquí se muestran todos los servicios de Firebase y puedes usar cualquiera de ellos, selecciona **RealTime Database**

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”.



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

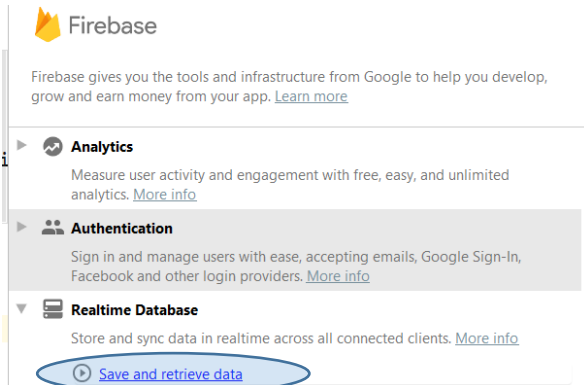
Ahora ya cuentas con una base de datos en tiempo real:



Paso 12: Edita las reglas de acceso a la base de datos para que sean publicas 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

1. **Conectarse a Firebase**
2. **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.

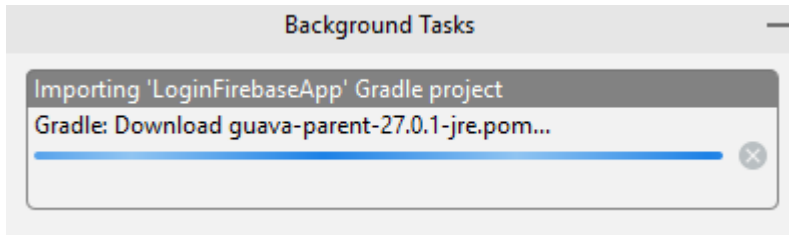
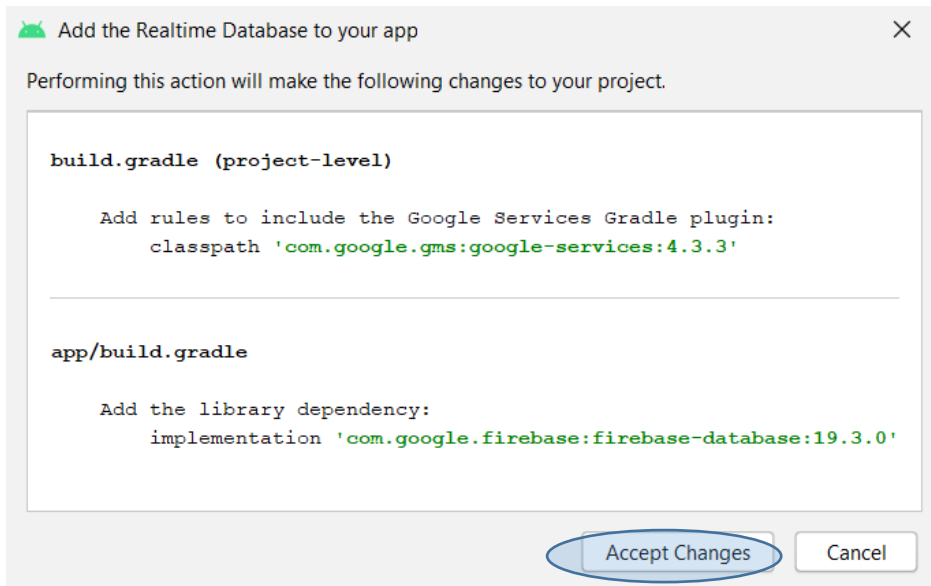
Paso 14: Ahora, su proyecto en Android Studio está conectado con el presente en Firebase. Agregar algunas dependencias a nuestros proyectos. Entonces, haga clic en "**Agregar RealTime Database en tu app**". Se abrirá un cuadro de diálogo. Haga clic en "**Aceptar cambios**" y automáticamente agregará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

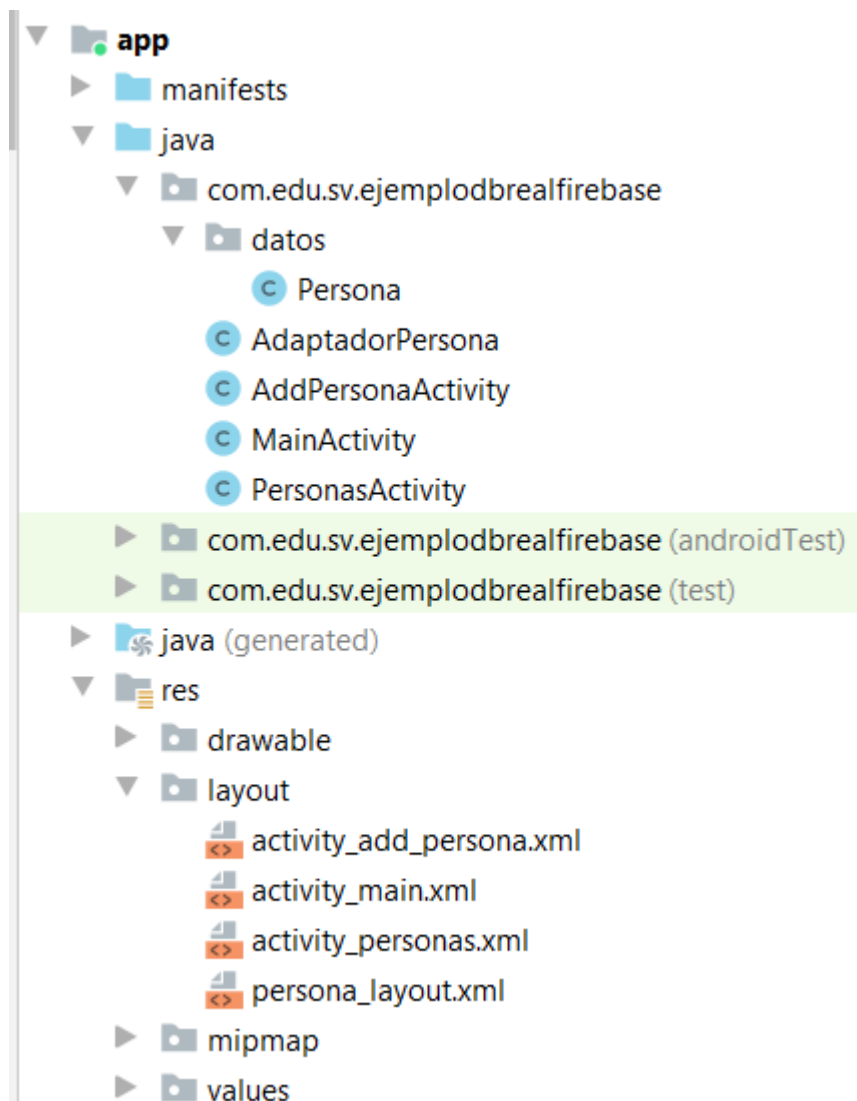
Aplicación de registro de personas.

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>



1. **MainActivity**: Esta es la actividad principal de la aplicación que se crea por default, pero no la utilizaremos para este ejemplo.
2. **PersonasActivity** : Esta actividad contiene una lista para mostrar todas las personas que vamos a guardar en nuestra base de datos
3. **AddPersonaActivity** : Esta actividad se utiliza agregar o editar una persona (Tiene una doble función)
4. **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.
5. **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);
}
```

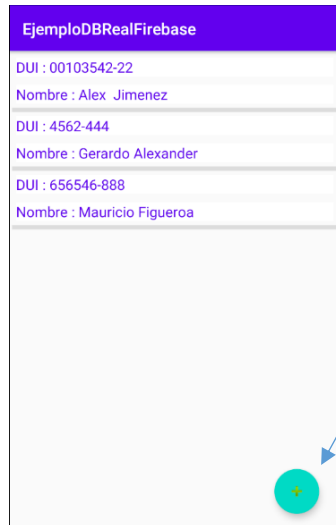
push().setValue(persona) es la forma en que podemos pasar un objeto para ser agregado a nuestra base de datos.

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.

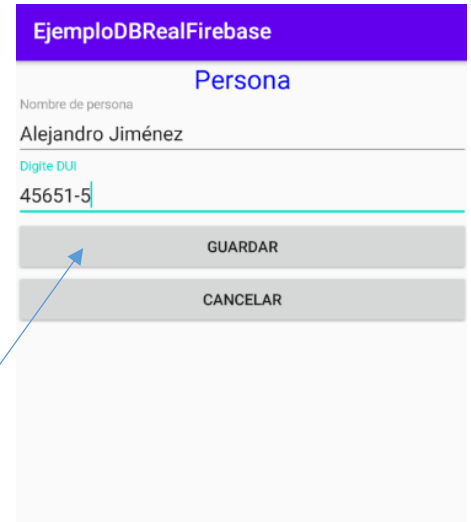
También será posible eliminar un nodo conociendo su identificador (key) y usando el método removeValue()

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

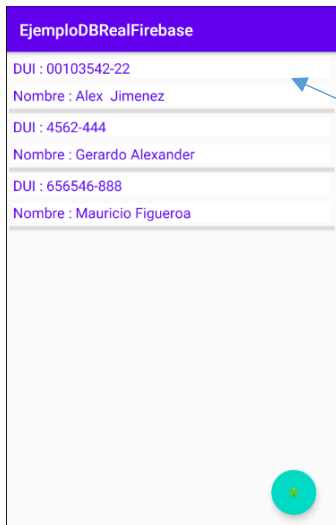
La aplicación debería de tener una apariencia similar:



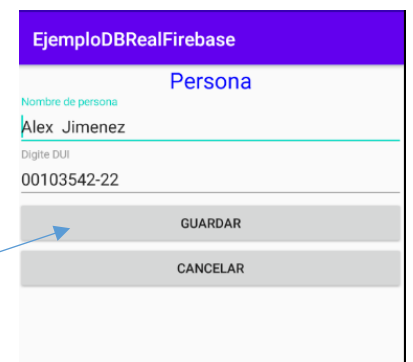
Puedes usar este botón para agregar nuevos registros de personas



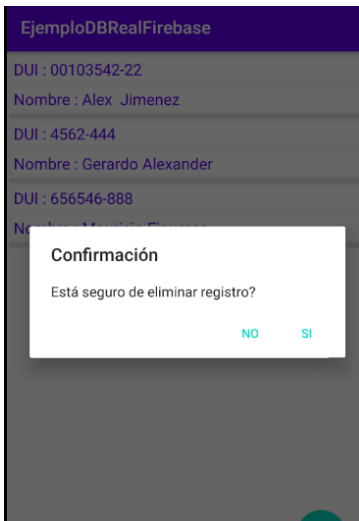
Posteriormente podrás agregar una nueva persona y dar clic en el botón guardar



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

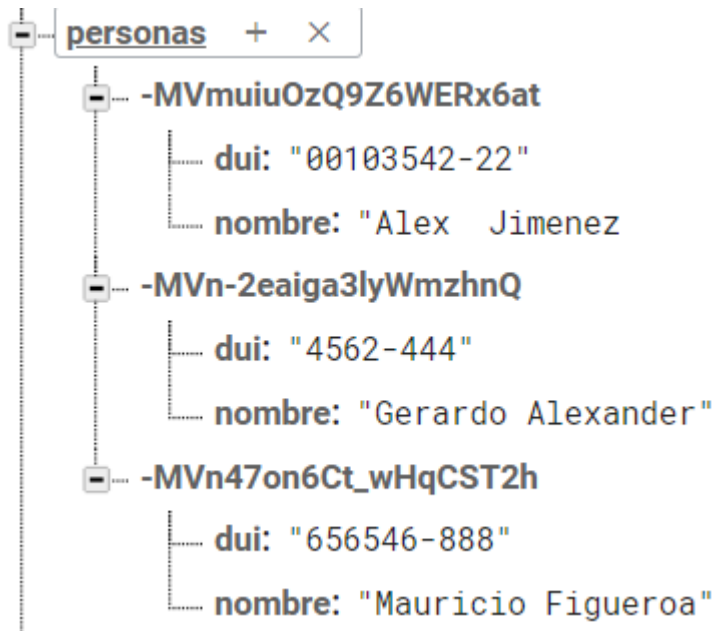


Posteriormente podrás modificar a la persona seleccionada y seleccionar guardar para actualizar los datos



También podrás mantener presionado uno de los registros por mas 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 como 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)



IV. DISCUSION DE RESULTADOS

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

V. BIBLIOGRAFIA

- RealTime Database de Firebase. Recuperado 15 de marzo de 2021, de <https://firebase.google.com/docs/database?authuser=0>