

# PROGRAMACIÓN MULTIMEDIA Y DISPOSITIVOS MÓVILES

## UF2 PAC DESARROLLO

Aníbal Santos Gómez

## **1. Investiga sobre los Content Provider. Explica que son, para qué sirven, cómo se utilizan.**

Un Content Provider son aquellos proveedores que administran el acceso a un conjunto de datos. Encapsulan los datos y proporcionan mecanismos para definir la seguridad de estos.

Es un mecanismo que implementa Android para compartir datos entre aplicaciones. Este mecanismo permite desacoplar la capa de aplicación de la capa de datos.

Es una interfaz estándar que conecta datos en un proceso con código que se ejecuta en otro proceso.

Para acceder a datos en un proveedor de contenido se usa el objeto ContentResolver en el Context de la aplicación, para comunicarte con el proveedor como cliente. El objeto ContentResolver se comunica con el objeto del proveedor, una instancia de una clase que implementa Content Provider. El objeto del proveedor recibe solicitudes de datos de clientes, realiza la acción solicitada y devuelve resultados.

Android incluye proveedores de contenido que administran datos como vídeo, audio, imágenes e información de contacto personal. Entre ellos destacan:

- Browser: posibilita la lectura o modificación de bookmarks, historial de navegación o búsquedas realizadas.
- CallLog: permite el acceso al registro de llamadas, para tanto visualizar el historial como poder modificarlo.
- Settings: proporciona acceso a las preferencias del dispositivo.
- MediaStore: acceso a los recursos multimedia.
- Contacts: Permite el acceso a la lista de contactos, para leer, modificar, o guardar nuevos contactos.

## **2. Investiga sobre los posibles sensores que podemos encontrar en un dispositivo Android. Explica cómo podemos utilizarlos.**

Los sensores son dispositivos que recogen un determinado tipo de información del medio exterior. Todos los dispositivos Android tienen un número de sesiones que nos permite un mejor uso del mismo.

Todos los sensores se manipulan de forma homogénea. Pero no todos los dispositivos disponen de los mismos sensores. Por lo que debemos en primer lugar averiguar los sensores que están disponibles.

Los sensores más importantes y comunes dentro de los diferentes dispositivos con Android como sistema operativo móvil son, el acelerómetro, la gravedad, el giroscopio, el acelerador lineal, la rotación, el sensor de proximidad, la luminosidad, la presión, la temperatura, o la humedad.

Para usar estos sensores utilizaremos constantes:

### **acelerómetro**

TYPE\_ACCELEROMETER

### **campo magnético**

TYPE\_MAGNETIC\_FIELD

### **giroscopio**

TYPE\_GYROSCOPE

### **orientación**

TYPE\_ORIENTATION

### **luz ambiental**

TYPE\_LIGHT

### **proximidad**

TYPE\_PROXIMITY

### **presión atmosférica**

TYPE\_PRESSURE

### **temperatura interna**

TYPE\_TEMPERATURE

### **gravedad**

TYPE\_GRAVITY

**acelerómetro lineal**

TYPE\_LINEAR\_ACCELERATION

**vector de rotación**

TYPE\_ROTATION\_VECTOR

**temperatura ambiental**

TYPE\_AMBIENT\_TEMPERATURE

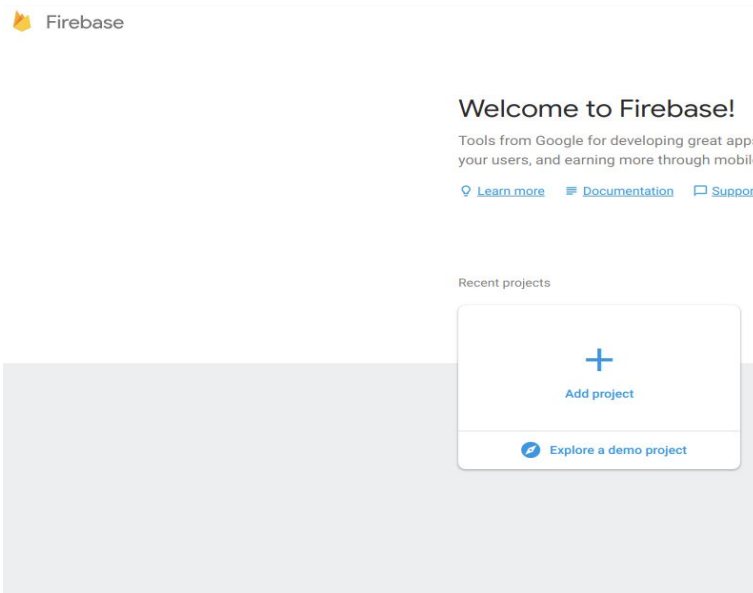
**humedad relativa**

TYPE\_RELATIVE\_HUMIDITY

**3. . Realiza una aplicación muy simple, pero que guarde información en FireBase. (Introduce capturas de pantallas tanto de los pasos seguidos para integrar FireBase en tu proyecto como de los nodos creados en la base de datos). Copiar y pegar el código del archivo .java donde se trabaja con dicha base de datos (comentado).**

He realizado una aplicación toDoList actualizable con los métodos básicos de POST, UPDATE y DELETE (el GET lo hace automáticamente para mostrar cada una de nuestras inserciones. ) A continuación mostraremos como se ha realizado la integración de FireBase con nuestro proyecto Android:

Crearemos un nuevo proyecto en Firebase:



Registraremos el mismo, con el nombre de nuestro paquete de la app.

× **Add Firebase to your Android app**

1

Register app

Android package name ⓘ

com.ansan.todolist

App nickname (optional) ⓘ

Freemium Android App

Debug signing certificate SHA-1 (optional) ⓘ

6f4d87867cb0bc692a57e00e56875e28c8e0274a

Required for Dynamic Links, Invites, and Google Sign-In or phone number support in Auth. Edit SHA-1s in Settings.

Register app

2

Download config file

3

Add Firebase SDK

4

Run your app to verify installation

Nos descargamos el JSON que nos proporciona los servicios de google.

×

Add Firebase to your Android app

✓

Register app

Android package name: com.ansan.todolist

2

Download config file

Instructions for Android Studio below | [Unity](#) [C++](#)

↓

Download google-services.json

Switch to the Project view in Android Studio to see your project root directory.

Move the google-services.json file you just downloaded into your Android app module root directory.

google-services.json

Previous

Next

3

Add Firebase SDK

4

Run your app to verify installation

Project Packages Scratches

MyApplication (~ / Desktop / My

gradle

idea

app

build

libs

src

.gitignore

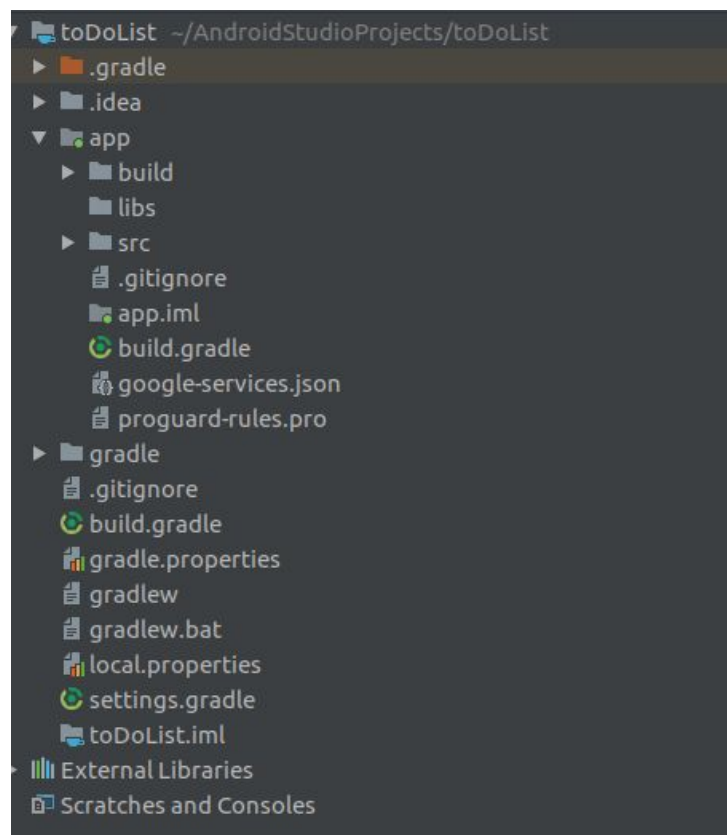
app.iml

build.gradle

google-services.json

proguard-rules.pro

gradle



Añadiremos el SDK de Firebase a nuestro proyecto de forma manual en el build.gradle a nivel de proyecto y de app.

✓

Register app

Android package name: com.ansan.todolist

✓

Download config file

3

Add Firebase SDK

Instructions for Gradle | [Unity](#) [C++](#)

The Google services plugin for [Gradle](#) loads the `google-services.json` file you just downloaded. Modify your build.gradle files to use the plugin.

**Project-level build.gradle** (<project>/build.gradle):

```
buildscript {
    dependencies {
        // Add this line
        classpath 'com.google.gms:google-services:4.0.1'
    }
}
```

**App-level build.gradle** (<project>/<app-module>/build.gradle):

```
dependencies {
    // Add this line
    implementation 'com.google.firebase:firebase-core:16.0.1'
}
...
// Add to the bottom of the file
apply plugin: 'com.google.gms.google-services'
```

Includes Analytics by default ⓘ

Finally, press "Sync now" in the bar that appears in the IDE:

Gradle files have changed since last sync 

Sync now

Previous

Next

4

Run your app to verify installation

```
// Top-level build file where you can add configuration options common to all sub-projects/modules

buildscript {
    repositories {
        google()
        jcenter()
    }
    dependencies {
        classpath 'com.android.tools.build:gradle:3.2.1'
        classpath 'com.google.gms:google-services:4.0.1'

        // NOTE: Do not place your application dependencies here; they belong
        // in the individual module build.gradle files
    }
}

allprojects {
    repositories {
        google()
        jcenter()
    }
}

task clean(type: Delete) {
    delete rootProject.buildDir
}
```

```

apply plugin: 'com.android.application'

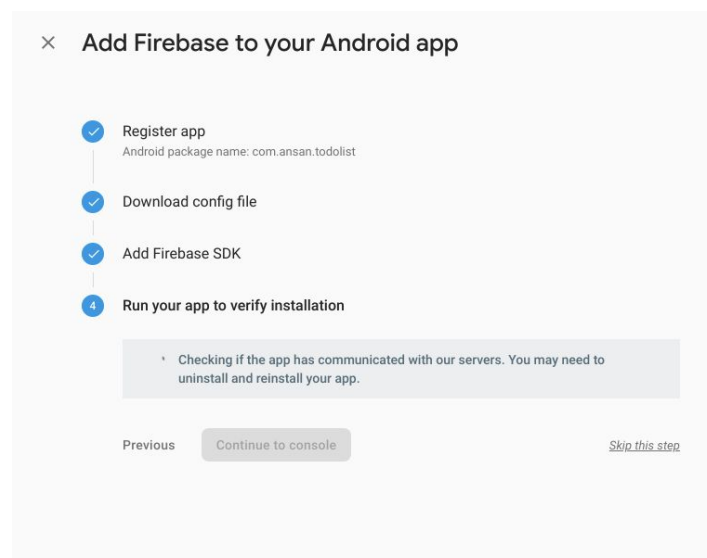
android {
    compileSdkVersion 28
    defaultConfig {
        applicationId "com.ansan.todolist"
        minSdkVersion 19
        targetSdkVersion 28
        versionCode 1
        versionName "1.0"
        testInstrumentationRunner "android.support.test.runner.AndroidJUnitRunner"
    }
    buildTypes {
        release {
            minifyEnabled false
            proguardFiles getDefaultProguardFile('proguard-android.txt'), 'proguard-rules.pro'
        }
    }
}

dependencies {
    implementation fileTree(dir: 'libs', include: ['*.jar'])
    //noinspection GradleCompatible
    implementation 'com.android.support:appcompat-v7:28.+'
    implementation 'com.android.support:recyclerview-v7:28.+'
    implementation 'com.android.support.constraint:constraint-layout:1.1.3'
    testImplementation 'junit:junit:4.12'
    androidTestImplementation 'com.android.support.test:runner:1.0.2'
    androidTestImplementation 'com.android.support.test.espresso:espresso-core:3.0.2'
    implementation 'com.google.firebase:firebase-database:16.0.1'
    implementation 'com.firebaseui:firebase-ui-database:4.2.1'
}

apply plugin: 'com.google.gms.google-services'

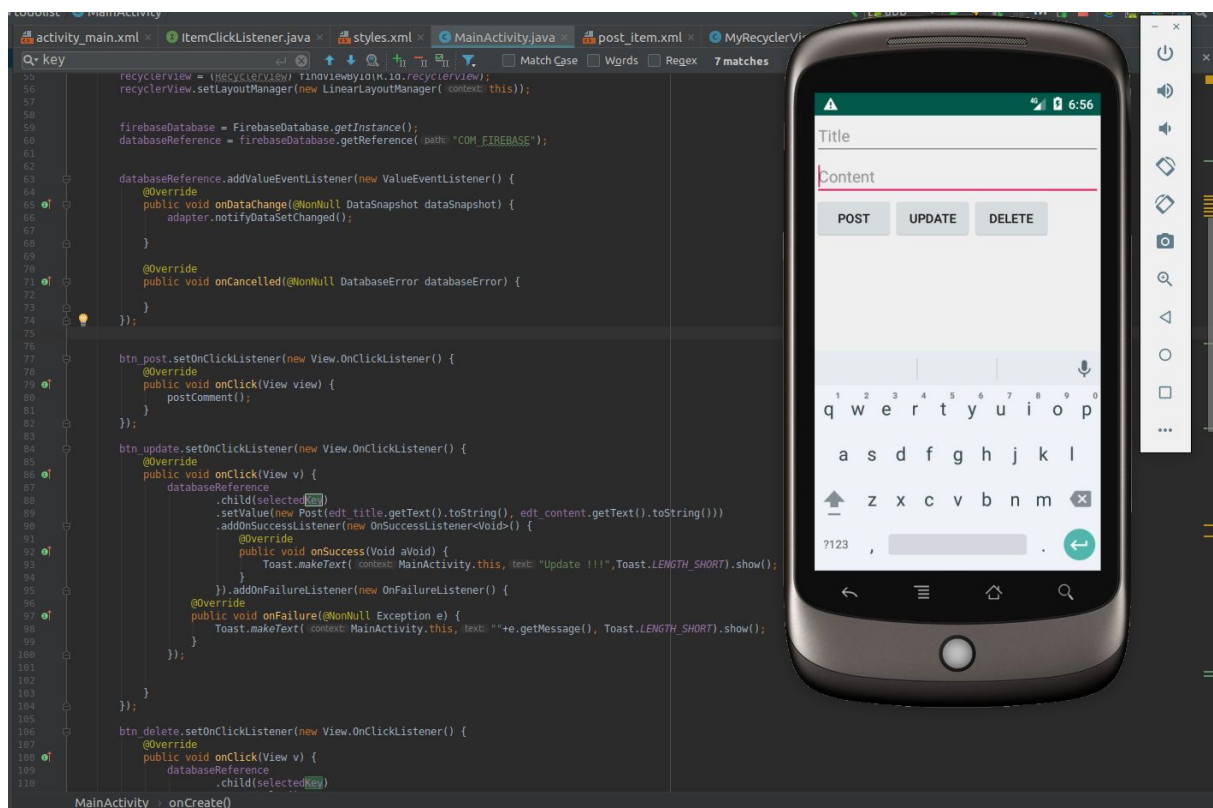
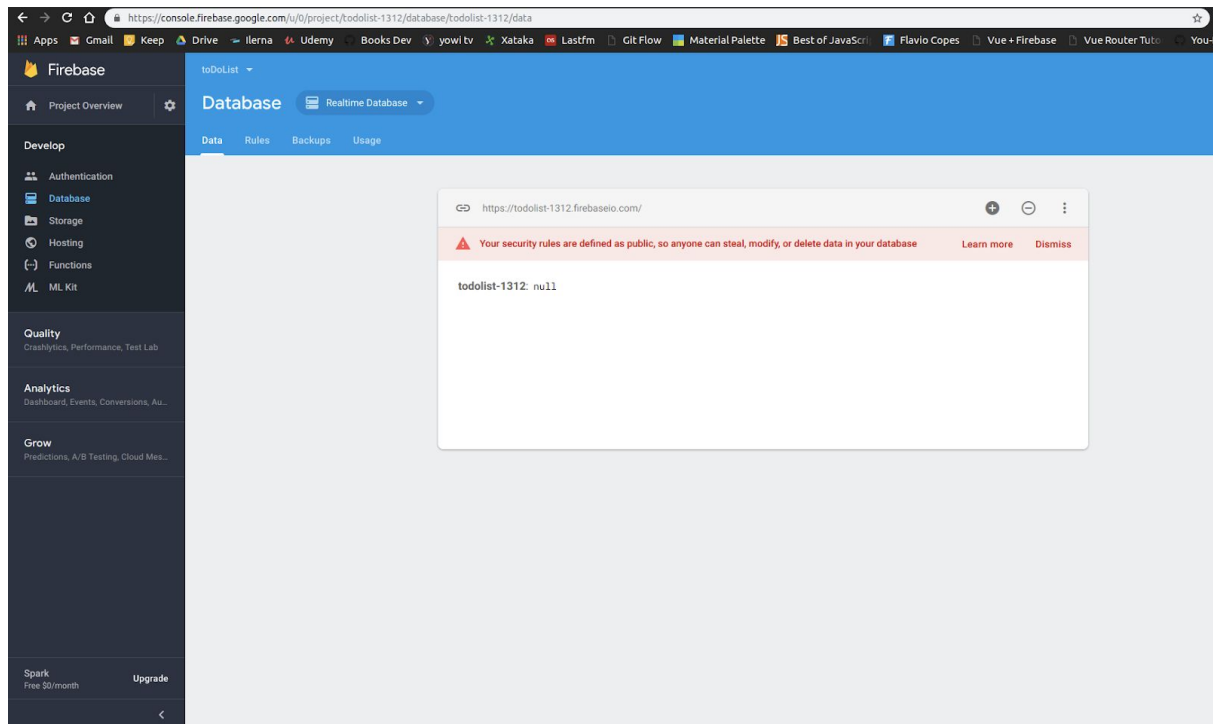
```

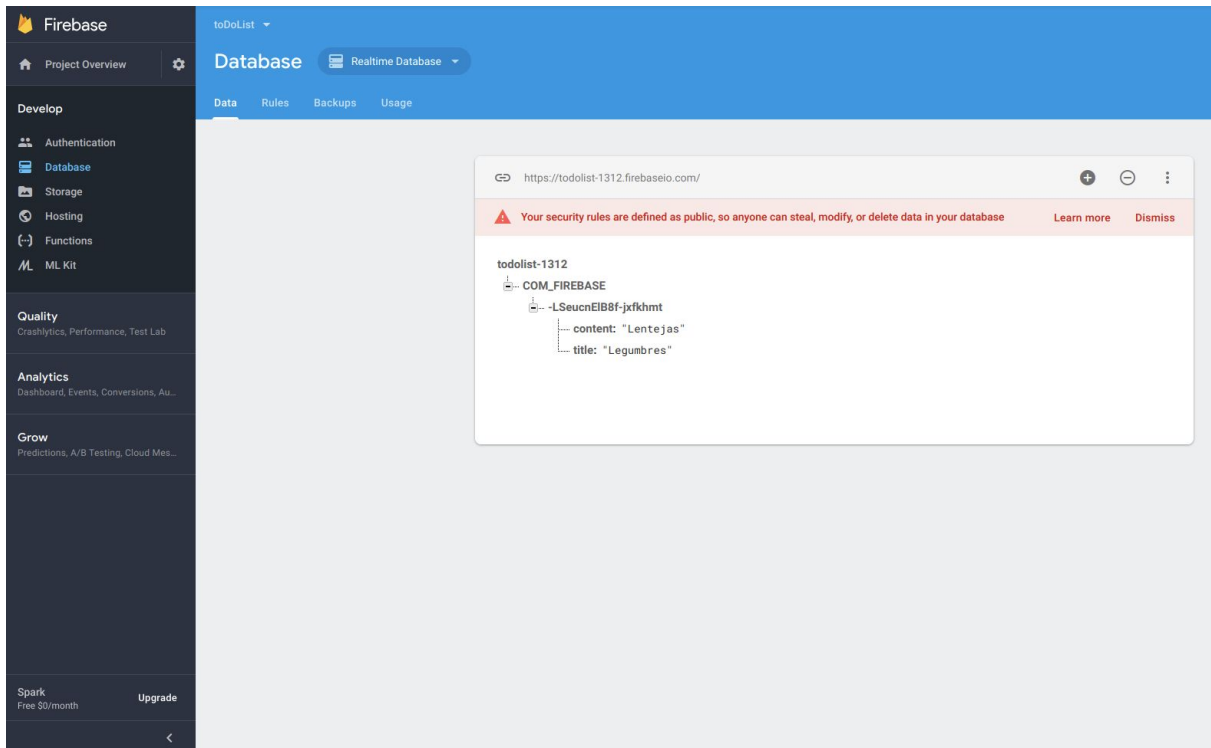
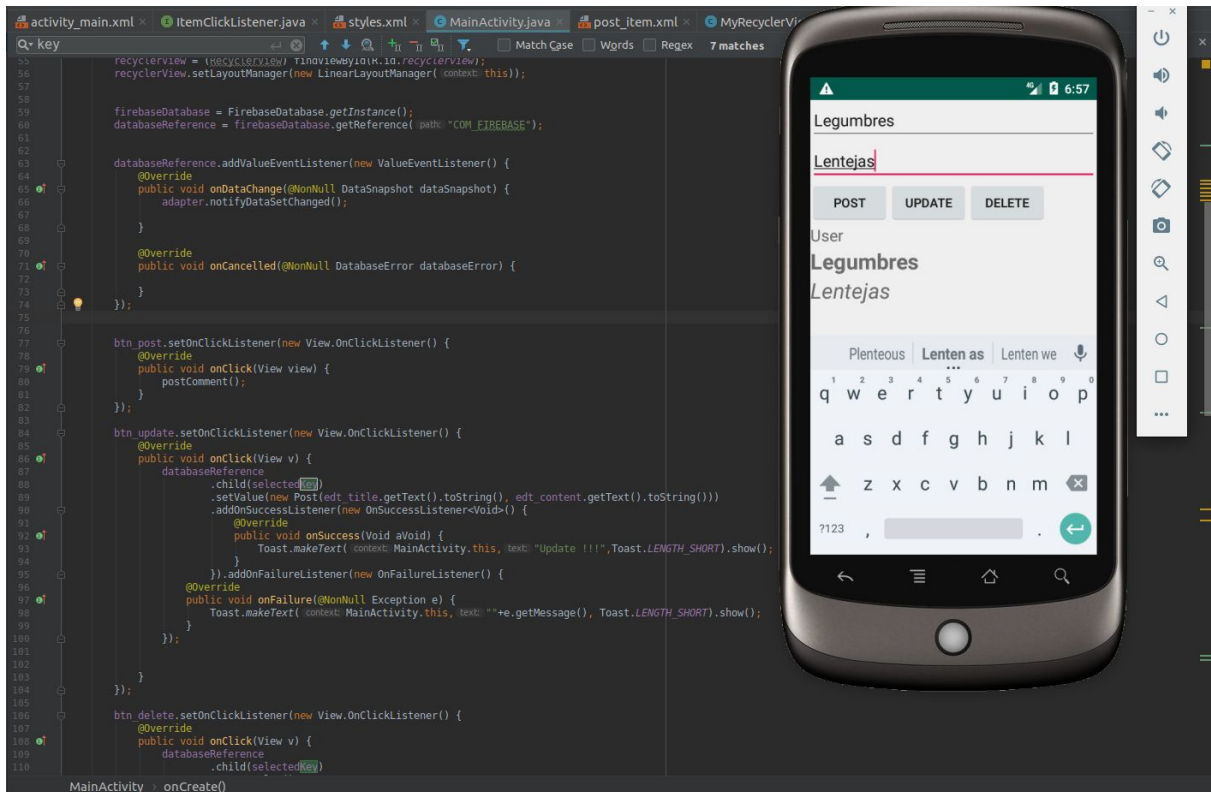
Una vez hecho esto sincronizaremos el gradle y tendremos sincronizada la aplicacion con Firebase, deberemos de tener abierta la aplicacion en AVD poder completar la sincronizacion.

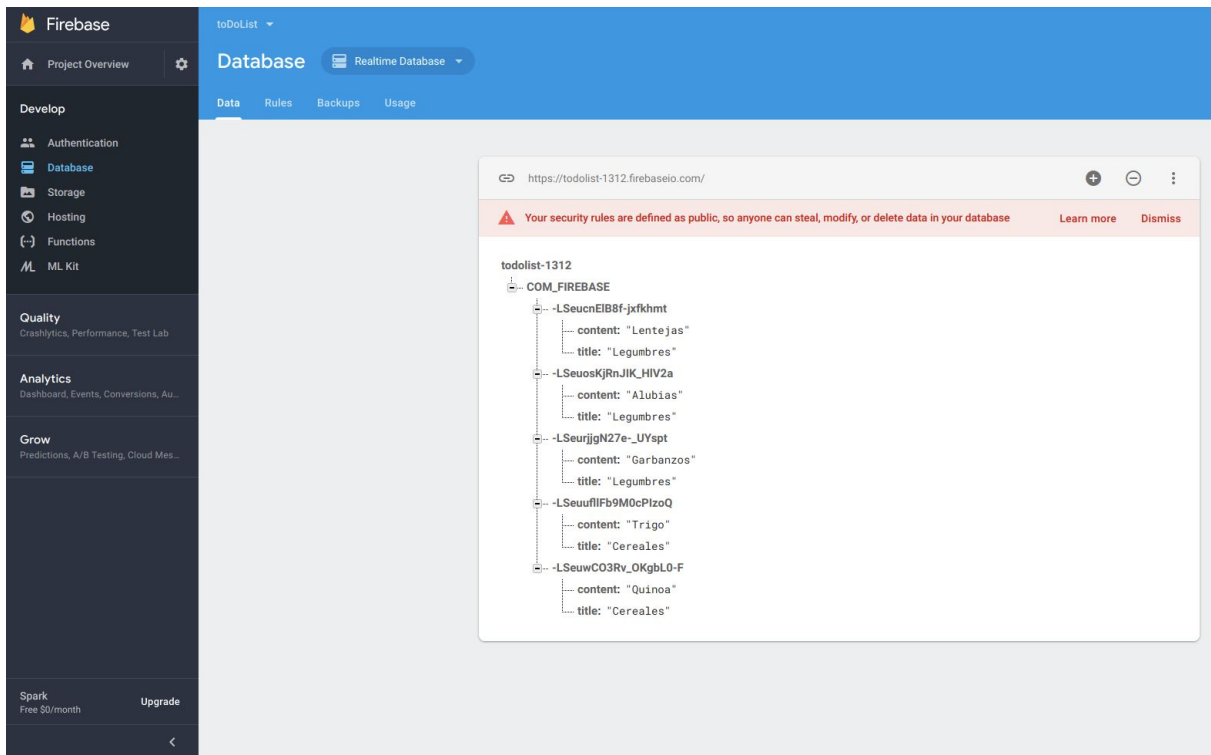
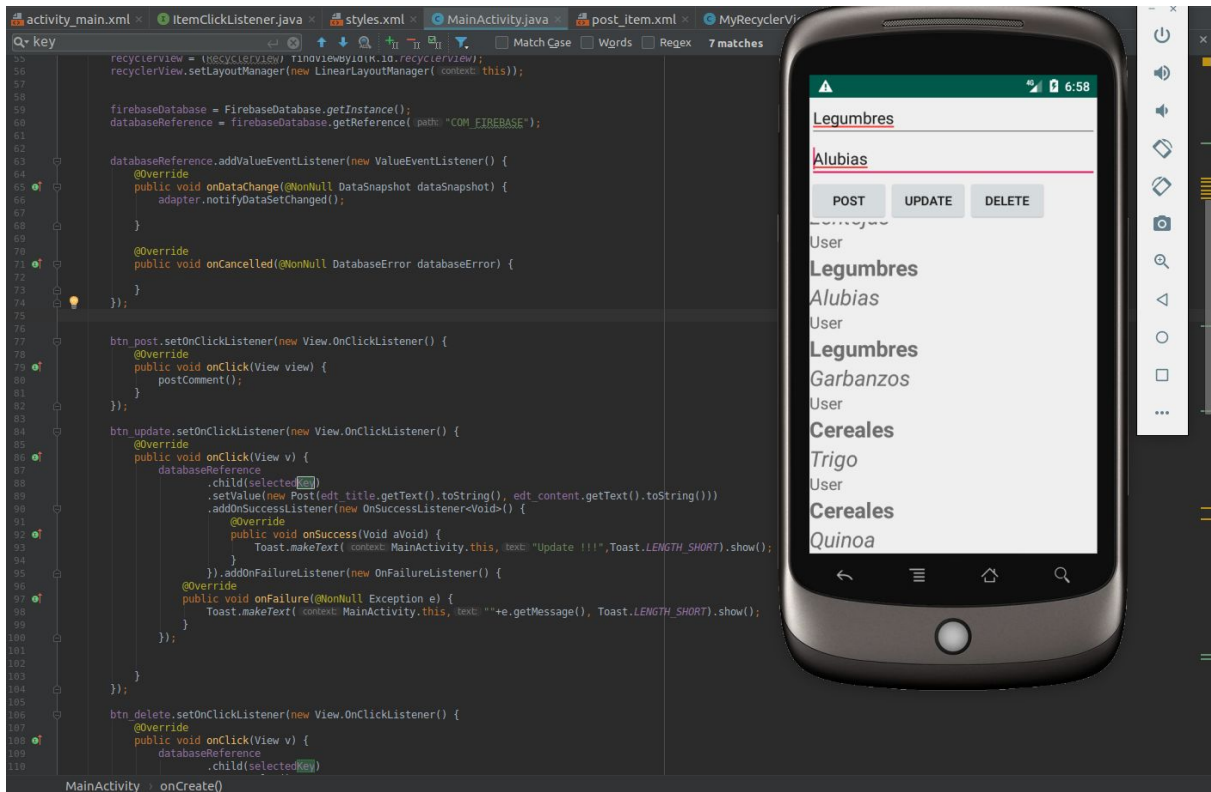




Posteriormente mostraremos el funcionamiento de la aplicación en AVD y la consola vacía de Firebase, y la completaremos introduciendo datos:







Codigo de la aplicacion:

### MainActivity.java:

```
package com.ansan.todolist;

import android.support.annotation.NonNull;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.support.v7.widget.LinearLayoutManager;
import android.support.v7.widget.RecyclerView;
import android.util.Log;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.Button;
import android.widget.EditText;
import android.widget.Toast;

import com.firebase.ui.database.FirebaseRecyclerAdapter;
import com.firebase.ui.database.FirebaseRecyclerOptions;
import com.google.android.gms.tasks.OnFailureListener;
import com.google.android.gms.tasks.OnSuccessListener;
import com.google.firebase.database.DataSnapshot;
import com.google.firebase.database.DatabaseError;
import com.google.firebase.database.DatabaseReference;
import com.google.firebase.database.FirebaseDatabase;
import com.google.firebase.database.ValueEventListener;

public class MainActivity extends AppCompatActivity {

    EditText edt_title, edt_content;
    Button btn_post, btn_update, btn_delete;
    RecyclerView recyclerView;

    //FIREBASE

    FirebaseDatabase firebaseDatabase;
    DatabaseReference databaseReference;

    FirebaseRecyclerOptions<Post> options;
    FirebaseRecyclerAdapter<Post, MyRecyclerViewHolder> adapter;

    Post selectedPost;
    String selectedKey;

    @Override
    protected void onCreate(Bundle savedInstanceState) {

        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
```

```

edt_content = (EditText) findViewById(R.id.edt_content);
edt_title = (EditText) findViewById(R.id.edt_title);
btn_post = (Button) findViewById(R.id.btn_post);
btn_update = (Button) findViewById(R.id.btn_update);
btn_delete = (Button) findViewById(R.id.btn_delete);
recyclerView = (RecyclerView) findViewById(R.id.recyclerview);
recyclerView.setLayoutManager(new LinearLayoutManager(this));

```

```

firebaseDatabase = FirebaseDatabase.getInstance();
databaseReference = firebaseDatabase.getReference("COM_FIREBASE");

```

```

databaseReference.addValueEventListener(new ValueEventListener() {
    @Override
    public void onDataChange(@NonNull DataSnapshot dataSnapshot) {
        adapter.notifyDataSetChanged();
    }
}

```

```

    @Override
    public void onCancelled(@NonNull DatabaseError databaseError) {

    }
});

```

```

btn_post.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        postComment();
    }
});

```

```

btn_update.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        databaseReference
            .child(selectedKey)
            .setValue(new Post(edt_title.getText().toString(), edt_content.getText().toString()))
            .addOnSuccessListener(new OnSuccessListener<Void>() {
                @Override
                public void onSuccess(Void aVoid) {
                    Toast.makeText(MainActivity.this, "Update !!!", Toast.LENGTH_SHORT).show();
                }
            }).addOnFailureListener(new OnFailureListener() {
                @Override
                public void onFailure(@NonNull Exception e) {
                    Toast.makeText(MainActivity.this, ""+e.getMessage(),
                    Toast.LENGTH_SHORT).show();
                }
            });
    }
});

```

```

    }
    });

    btn_delete.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {
            databaseReference
                .child(selectedKey)
                .removeValue()
                .addOnSuccessListener(new OnSuccessListener<Void>() {
                    @Override
                    public void onSuccess(Void aVoid) {
                        Toast.makeText(MainActivity.this, "Deleted !!!", Toast.LENGTH_SHORT).show();
                    }
                }).addOnFailureListener(new OnFailureListener() {
                    @Override
                    public void onFailure(@NonNull Exception e) {
                        Toast.makeText(MainActivity.this, ""+e.getMessage(),
                        Toast.LENGTH_SHORT).show();
                    }
                });
        }
    });

    }
    });

    displayComment();
}

@Override
protected void onStop() {

    if (adapter != null) {

    }
    adapter.startListening();

    super.onStop();
}

private void postComment() {
    String title = edt_title.getText().toString();
    String content = edt_content.getText().toString();

    Post post = new Post(title, content);

    databaseReference.push().setValue(post); // Usamos este metodo para crear un id unico
    por coment

    adapter.notifyDataSetChanged();

}

```

```

private void displayComment() {

    options =
        new FirebaseRecyclerOptions.Builder<Post>()
            .setQuery(databaseReference, Post.class)
            .build();

    adapter =
        new FirebaseRecyclerAdapter<Post, MyRecyclerViewHolder>(options) {
            @Override
            protected void onBindViewHolder(@NonNull MyRecyclerViewHolder holder, int
position, @NonNull final Post model) {
                holder.txt_title.setText(model.getTitle());
                holder.txt_comment.setText(model.getContent());

                holder.setItemClickListener(new ItemClickListener() {
                    @Override
                    public void onClick(View view, int position) {
                        selectedPost = model;
                        selectedKey = getSnapshots().getSnapshot(position).getKey();
                        Log.d("Key Item", "" + selectedKey);

                        //Binding de datos

                        edt_content.setText(model.getContent());
                        edt_title.setText(model.getTitle());
                    }
                });
            }

            @NonNull
            @Override
            public MyRecyclerViewHolder onCreateViewHolder(@NonNull ViewGroup
viewGroup, int i) {
                View itemView = LayoutInflater.from(getBaseContext()).inflate(R.layout.post_item,
viewGroup, false);
                return new MyRecyclerViewHolder(itemView);
            }
        };

    adapter.startListening();
    recyclerView.setAdapter(adapter);
}
}

```

## Post.java

```

package com.ansan.todolist;

public class Post {
    private String title, content;
}

```

```

    public Post() {
    }

    public Post(String title, String content) {
        this.title = title;
        this.content = content;
    }

    public String getTitle() {
        return title;
    }

    public void setTitle(String title) {
        this.title = title;
    }

    public String getContent() {
        return content;
    }

    public void setContent(String content) {
        this.content = content;
    }
}

```

## MyRecyclerViewHolder.java:

```

package com.ansan.todolist;

import android.support.annotation.NonNull;
import android.support.v7.widget.RecyclerView;
import android.view.View;
import android.widget.TextView;

public class MyRecyclerViewHolder extends RecyclerView.ViewHolder implements
View.OnClickListener {

    TextView txt_title, txt_comment;

    ItemClickListener itemClickListener;

    public void setItemClickListener(ItemClickListener itemClickListener) {
        this.itemClickListener = itemClickListener;
    }

    public MyRecyclerViewHolder(@NonNull View itemView) {
        super(itemView);

        txt_comment = (TextView)itemView.findViewById(R.id.txt_content);
        txt_title = (TextView)itemView.findViewById(R.id.txt_title);

        itemView.setOnClickListener(this);
    }
}

```



```
}  
  
@Override  
public void onClick(View view) {  
    itemClickListener.onClick(view, getAdapterPosition());  
}  
}
```

### ItemClickListener.java:

```
package com.ansan.todolist;  
  
import android.view.View;  
  
public interface ItemClickListener {  
    void onClick(View view, int position);  
}
```

