## PROGRAMACIÓN MULTIMEDIA Y DISPOSITIVOS MÓVILES

2° DAM

## FIREBASE

#### **CONTENIDOS DE LA UNIDAD**

**REGISTRO EN FIREBASE** 

SIGN IN

**LOG IN** 

AGREGAR ELEMENTOS



## ¿QUÉ ES FIREBASE?

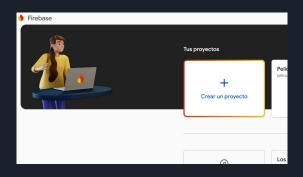
Firebase es una plataforma desarrollada por Google que proporciona una serie de servicios en la nube diseñados para facilitar la creación y gestión de aplicaciones. Es especialmente útil en el desarrollo de aplicaciones móviles y web, ya que permite implementar funcionalidades complejas de manera sencilla y rápida, sin necesidad de configurar servidores ni bases de datos tradicionales.

Algunos de los servicios más destacados de Firebase son:

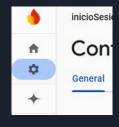
- Autenticación (Authentication): Permite a los usuarios iniciar sesión en la aplicación utilizando correo electrónico, Google, Facebook, o incluso iniciar sesión de forma anónima.
- Base de Datos en Tiempo Real (Realtime Database): Una base de datos NoSQL que sincroniza los datos entre los usuarios en tiempo real.
- Firestore: Una base de datos más avanzada, flexible y escalable que también funciona en tiempo real.
- Almacenamiento (Storage): Permite guardar archivos como imágenes, vídeos o documentos en la nube.
- Notificaciones Push: Envía notificaciones a los usuarios de la aplicación directamente desde la nube.
- Análisis (Analytics): Ofrece herramientas para analizar el uso de la aplicación y el comportamiento de los usuarios.

## VINCULAR ANDROID STUDIO CON FIREBASE

Accedemos a la web de Firebase, accedemos con nuestra cuenta y le damos a *Comenzar* para poder ver nuestros proyectos.



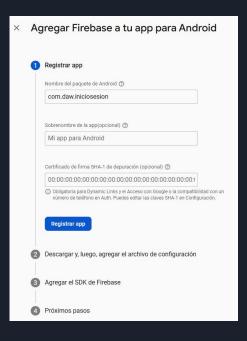
Le ponemos un nombre y seleccionamos una cuenta. Nos dirigimos a *Ajustes* y seleccionamos <u>Android</u>



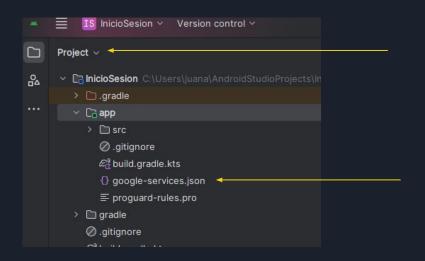


#### VINCULAR ANDROID STUDIO CON FIREBASE

Ahora debemos rellenar una serie de campos, antes debemos crear un proyecto en Android Studio.



Seguimos los pasos y descargamos el archivo .json. El cual pegamos en la carpeta app (desde la vista Project)



#### VINCULAR ANDROID STUDIO CON FIREBASE

Volvemos a poner la vista de proyecto en Android y añadimos la dependencia que se nos indica. La primera la copiamos en build.gradle.kts (*Project: InicioSesion*)

y las demás en build.gradle.kts (Module: app)

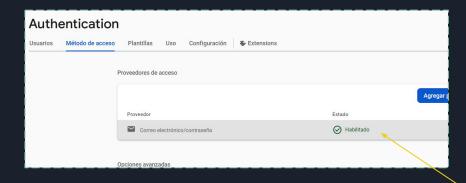
```
plugins {
    alias(libs.plugins.android.application)
    id("com.google.gms.google-services")
}
```

y sincronizamos Gradle

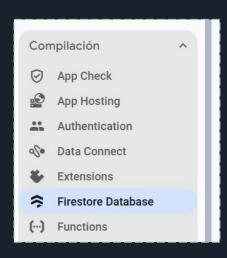
En Manifest hacemos que primero se ejecute Registro:

```
// cactivity
// android:name=".MainActivity"
// android:exported="false" />
// cactivity
// android:name=".Registro"
// android:exported="true">
// android:exported="true">
// catenory android:name="android.intent.action.MAIN" />
// catenory android:name="android intent category LAUNCHER" />
// category android:name="android intent category launcher category launc
```

Tenemos que, desde Authentication, habilitar el registro por correo/contraseña:



Para que la base de datos empiece a ser útil, debemos crearla en Firestore Database:



Y establecer en reglas, escritura y lectura a true:



```
rules_version = '2';

service cloud.firestore {
  match /databases/{database}/documents {
   match /{document=**} {
    allow read, write: if true;
   }
}
}
```

Dejaremos la MainActivity como estaba (modificando el texto) y creamos una nueva (Registro) para el registro:







En la activity de Registrar, enlazamos la parte gráfica/lógica y creamos dos atributos: EditText nombre, correo, pass;
FirebaseFirestore firestore;
FirebaseAuth auth;

@Override
protected void onCreate(Bundle savedInstanceState) {
 super.onCreate(savedInstanceState);
 EdgeToEdge.enable(this);
 setContentView(R.layout.activity\_registro);
 ViewCompat.setOnApplyWindowInsetsListener(findViewById(R.id.main), (v, insets) -> {
 Insets systemBars = insets.getInsets(WindowInsetsCompat.Type.systemBars());
 v.setPadding(systemBars.left, systemBars.top, systemBars.right, systemBars.bottom);
 return insets;
 });

public class Registro extends AppCompatActivity {

nombre= findViewById(R.id.nombre); correo = findViewById(R.id.correo); pass = findViewById(R.id.pass);

auth = FirebaseAuth.getInstance();

firestore = FirebaseFirestore.getInstance();

El método para el botón:

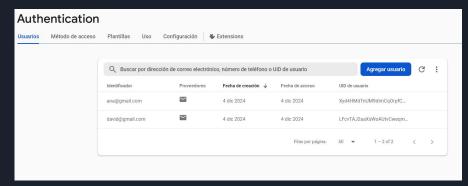
```
public void registrarme(View view){
   String nombreUsuario = nombre.getText().toString().trim();
   String correoUsuario = correo.getText().toString().trim();
   String passUsuario = pass.getText().toString().trim();

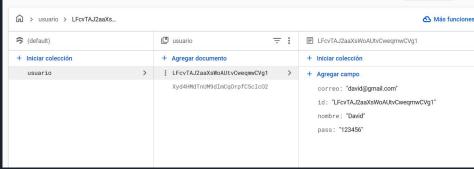
if(nombreUsuario.isEmpty() || correoUsuario.isEmpty() || passUsuario.isEmpty()){
   Toast.makeText(this, "Complete todos los campos", Toast.LENGTH_LONG).show();
}else{
   registroUsuario(nombreUsuario, correoUsuario, passUsuario);
}
```

Y un método que registra y añade a la Firestore:

```
private void registroUsuario(String nombreUsuario, String correoUsuario, String passUsuario) {
 auth.createUserWithEmailAndPassword(correoUsuario, passUsuario).addOnCompleteListener(new
OnCompleteListener<AuthResult>() {
   @Override
   public void onComplete(@NonNull Task<AuthResult> task) {
     String id = auth.getCurrentUser().getUid();
     Map<String,Object> map = new HashMap<>();
     map.put("id", id);
     map.put("nombre", nombreUsuario);
     map.put("correo", correoUsuario);
     map.put("pass", passUsuario);
     firestore.collection("usuario").document(id).set(map).addOnSuccessListener(new OnSuccessListener<Void>() {
       @Override
       public void onSuccess(Void unused) {
         finish():
         startActivity(new Intent(Registro.this, MainActivity.class));
         Toast.makeText(Registro.this, "Usuario registrado", Toast.LENGTH_LONG).show();
     }).addOnFailureListener(new OnFailureListener() {
       @Override
       public void onFailure(@NonNull Exception e) {
         Toast.makeText(Registro.this, "Error al quardar", Toast.LENGTH_LONG).show();
     });
 }).addOnFailureListener(new OnFailureListener() {
   @Override
   public void onFailure(@NonNull Exception e) {
     Toast.makeText(Registro.this, "Error al registrar", Toast.LENGTH_SHORT).show();
```

Comprobamos desde Firebase las Autenticaciones y los registros en la base de datos:





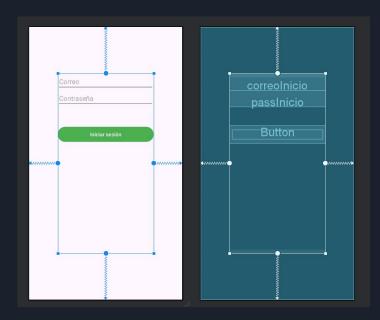
Ahora vamos a crear el Login

Crearemos una nueva activity, Inicio, y desde Manifest decimos que la primera vista, ahora debe ser esa.

```
android:roundIcon="@mipmap/ic_launcher_round"
android:supportsRtl="true"
android:theme="@style/Theme.InicioSesion"
tools:targetApi="31">
<activity
   android:name=".Registro"
   android:exported="true" />
<activity
   android:exported="false" />
<activity
   android:name=".Inicio" -
   android:exported="true">
   <intent-filter>
       <action android:name="android.intent.action.MAIN" />
       <category android:name="android.intent.category.LAUNCHER" />
   </intent-filter>
</activitv≥
```

Es necesario ese orden y que los exported estén a true para que pueda ser accesible desde fuera de la app (como Firebase)

La parte gráfica puede ser así:



En la activity de Inicio:

```
EditText correo, pass;
FirebaseAuth auth:
@Override
protected void onCreate(Bundle savedInstanceState) {
 super.onCreate(savedInstanceState);
 EdgeToEdge.enable(this);
 setContentView(R.layout.activity_inicio);
       [...]
 correo = findViewById(R.id.correoInicio);
 pass = findViewById(R.id.passInicio);
 auth = FirebaseAuth.getInstance();
public void iniciarSesion(View view){
 String correoUsuario = correo.getText().toString().trim();
 String passUsuario = pass.getText().toString().trim();
 if(correoUsuario.isEmpty() || passUsuario.isEmpty()){
   Toast.makeText(this, "Ingrese todos los datos",
Toast. LENGTH_LONG). show();
 }else{
   inicioUsuario(correoUsuario, passUsuario);
```

Y creamos el método para comprobar que existe dicho usuario y que inicia sesión, para llevar a la activity de contenido privado.

```
private void inicioUsuario(String correoUsuario, String passUsuario)
 auth.signInWithEmailAndPassword(correoUsuario,
passUsuario).addOnCompleteListener(new
OnCompleteListener<AuthResult>() {
   @Override
   public void onComplete(@NonNull Task<AuthResult> task) {
     if(task.isSuccessful()){
       finish():
       startActivity(new Intent(Inicio.this, MainActivity.class));
       Toast.makeText(Inicio.this, "Bienvenido/a",
Toast. LENGTH_LONG). show();
     }else{
       Toast.makeText(Inicio.this. "Error".
Toast. LENGTH_LONG). show();
 }).addOnFailureListener(new OnFailureListener() {
   @Override
   public void onFailure(@NonNull Exception e) {
     Toast.makeText(Inicio.this, "Error al iniciar sesión",
Toast. LENGTH_LONG). show();
 });
```