

UNIVERSIDAD DON BOSCO



Tema:

Foro 2.

Materia:

Desarrollo de Software para Móviles.

Titular de Catedra:

Ing. Alexander Alberto Sigüenza Campos.

Integrantes:

Rivas Serrano, Omar Alexander

RS060867

Fecha de entrega:

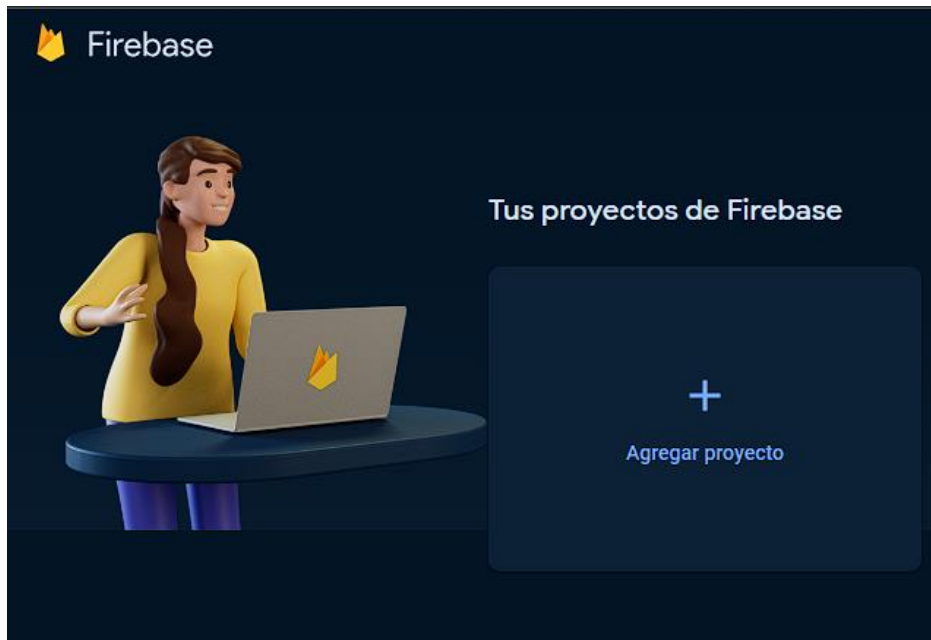
03 de diciembre de 2023

Contenido

| | |
|---|----|
| Configuración Firebase | 3 |
| Creación de Proyecto Kotlin..... | 5 |
| Vinculación de proyecto a firebase..... | 6 |
| Creación de Activity y Autenticación dentro de la Aplicación..... | 13 |
| Pruebas de funcionamiento..... | 18 |

Configuración Firebase

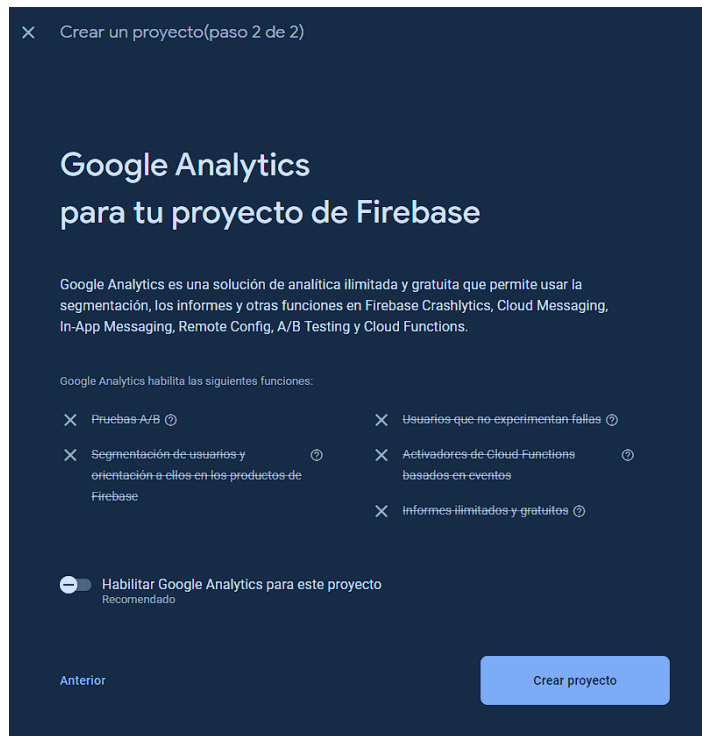
Primero se debe crear un proyecto en el sitio web de firebase:



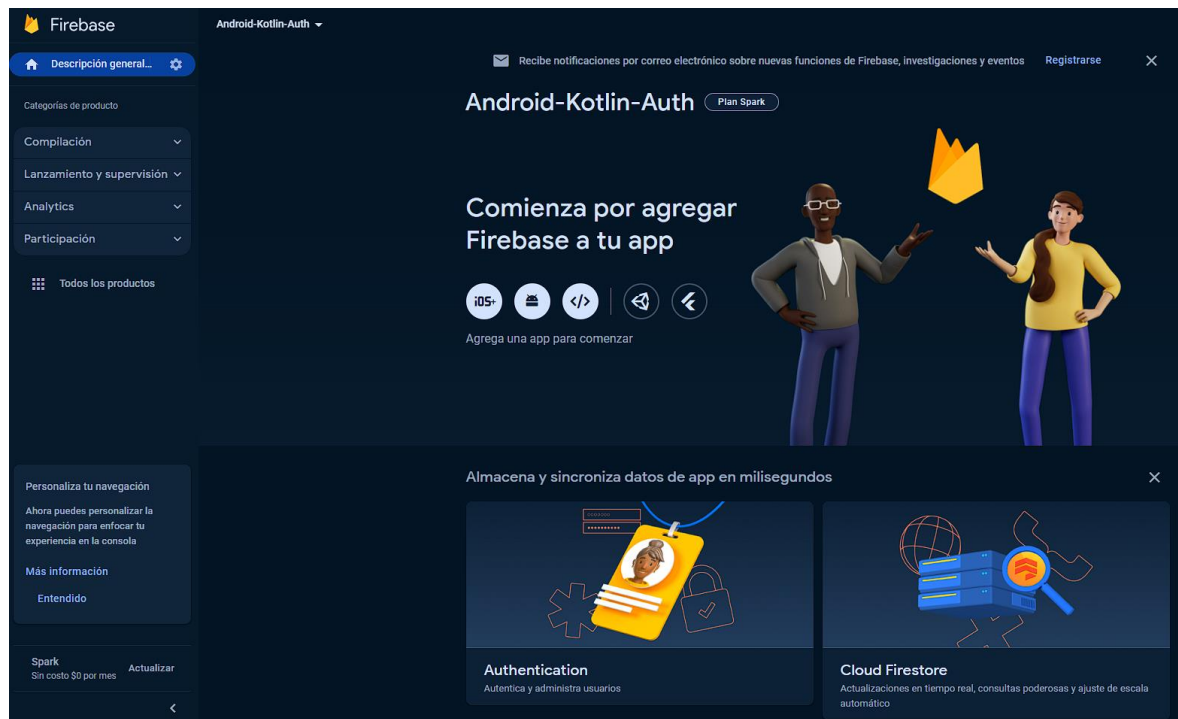
Nos pedirá colocarle un nombre al proyecto:

The image shows the "Crear un proyecto (paso 1 de 3)" screen in the Firebase console. At the top left is a close button (X) and the title "Crear un proyecto(paso 1 de 3)". The main heading is "Comencemos con el nombre de tu proyecto®". Below this is a text input field with the placeholder "Nombre del proyecto" and the text "Android-Kotlin-Auth" entered. Below the input field is a small button with a pencil icon and the text "android-kotlin-auth-95ffa". At the bottom is a blue button labeled "Continuar".

Deshabilitamos la sección de Google Analytics, y le damos en crear proyecto:

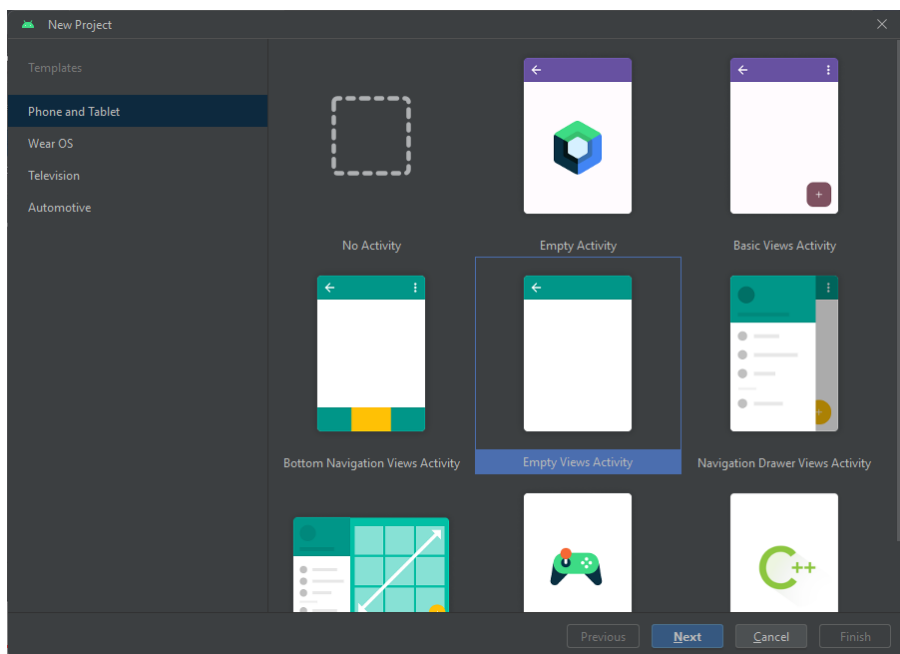


Y listo, tenemos creado nuestro proyecto:

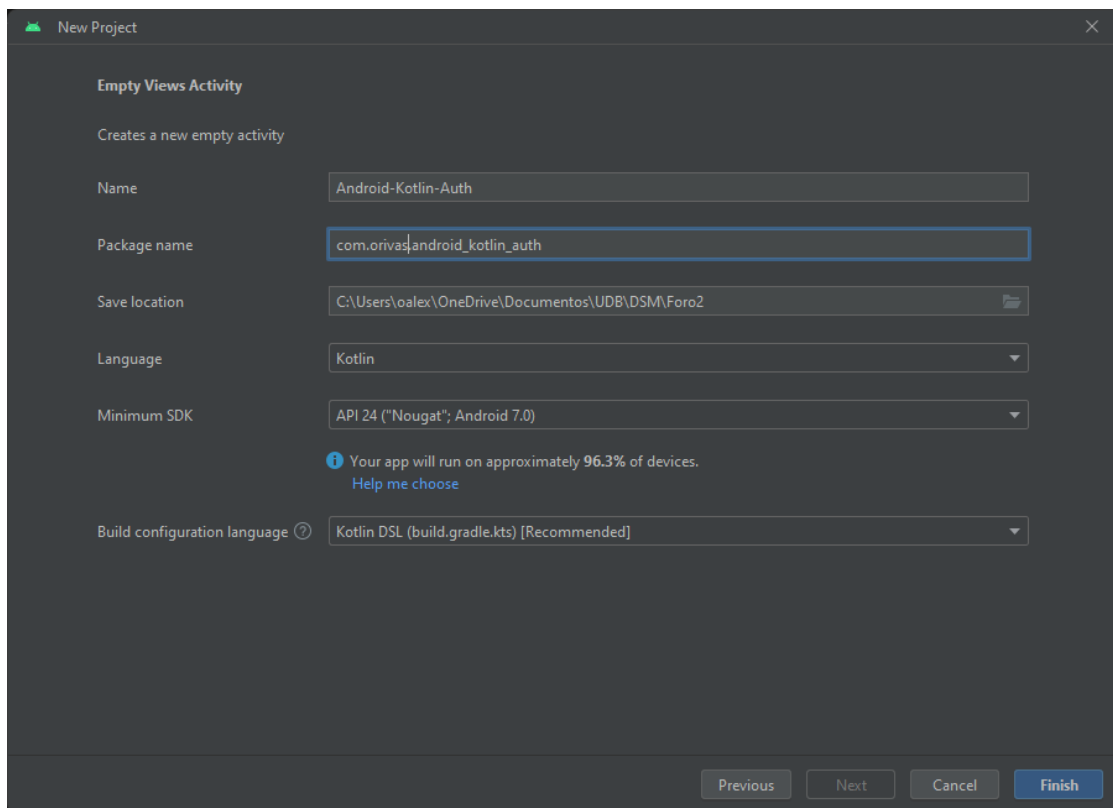


Creación de Proyecto Kotlin

Ahora creamos nuestro proyecto en Android Studio:



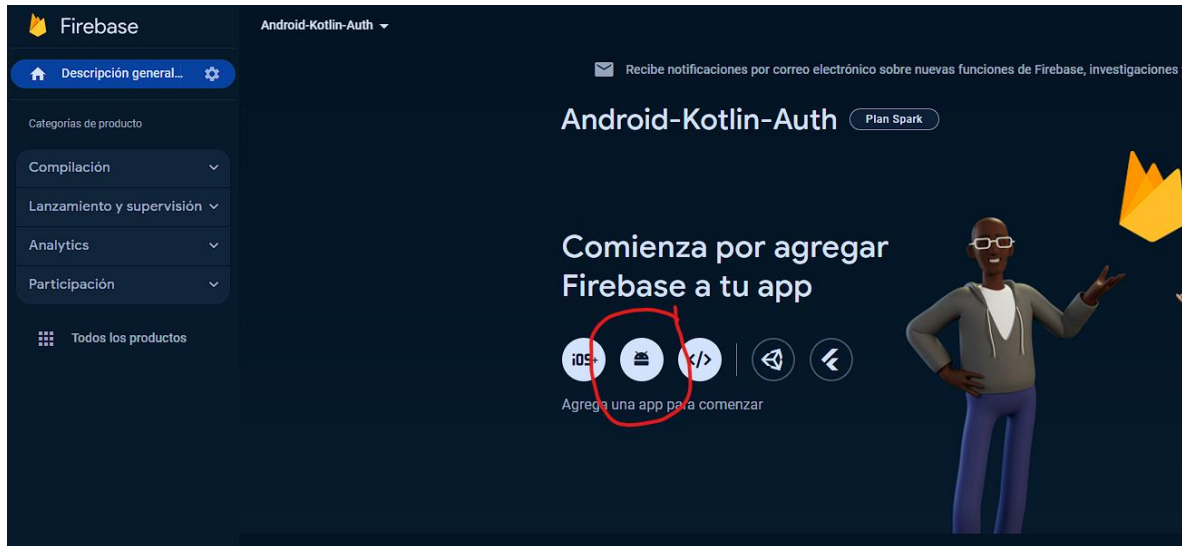
Colocamos el nombre al proyecto, y damos en finalizar:



Vinculación de proyecto a firebase

Una vez creado el proyecto, procederemos a adaptar nuestra aplicación registrándola al nuevo servicio de Firebase:

Damos clic en el icono de Android:



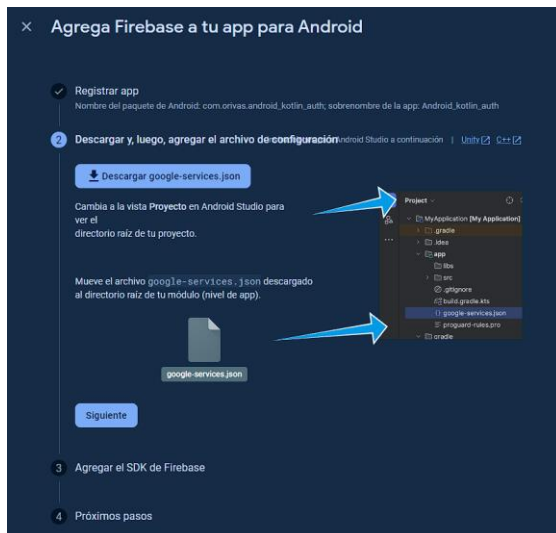
Esto nos pedirá registrar nuestra app:

The image shows a dialog box titled 'Agregar Firebase a tu app para Android'. It has a close button (X) in the top left. The dialog is divided into a list of steps on the left and a form area on the right. The steps are: 1. Registrar app, 2. Descargar y, luego, agregar el archivo de configuración, 3. Agregar el SDK de Firebase, and 4. Próximos pasos. The first step is active. The form area contains three input fields: 'Nombre del paquete de Android' with the value 'com.orivas.android_kotlin_auth', 'Sobrenombre de la app (opcional)' with the value 'Android_kotlin_auth', and 'Certificado de firma SHA-1 de depuración (opcional)' with the value '20:14:C3:3E:46:A8:A2:1B:1E:10:DD:89:0A:AB:E3:AE:3'. Below the third field is a note: 'Obligatoria para Dynamic Links y el Acceso con Google o la asistencia con un número de teléfono en Auth. Puedes editar las claves SHA-1 en Configuración.' At the bottom of the form area is a blue button labeled 'Registrar app'.

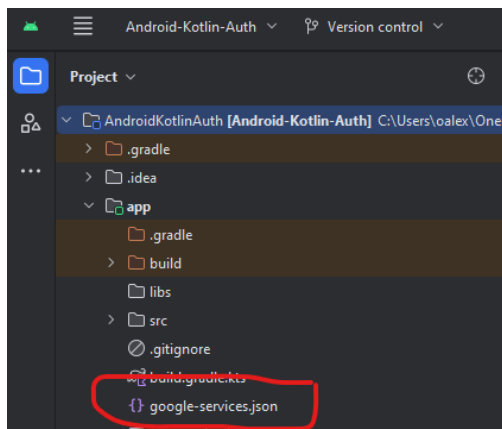
Para generar el certificado de firma SHA 1, lo haremos en consola de la siguiente manera:

```
C:\Windows\system32\cmd.exe
c:\Program Files (x86)\Java\jre1.8.0_361\bin>keytool -list -v -alias androiddebugkey -keystore %USERPROFILE%\android\debug.keystore
Introduzca la contraseña del almacén de claves:
Nombre de Alias: androiddebugkey
Fecha de Creación: 10-29-2023
Tipo de Entrada: PrivateKeyEntry
Longitud de la Cadena de Certificado: 1
Certificado[1]:
Propietario: C=US, O=Android, CN=Android Debug
Emisor: C=US, O=Android, CN=Android Debug
Número de serie: 1
Válido desde: Sun Oct 29 22:17:37 CST 2023 hasta: Tue Oct 21 22:17:37 CST 2053
Huellas digitales del certificado:
  SHA1: 20:14:C3:3E:46:A8:A2:1B:1E:10:DD:89:0A:AB:E3:AE:3F:94:6D:67
  SHA256: 24:DA:67:6B:8E:9C:D1:45:27:B7:A3:41:EA:26:D0:B5:7D:21:37:62:05:35:01:99:4F:17:E3:EA:B0:E5:CA:42
Nombre del algoritmo de firma: SHA256withRSA
Algoritmo de clave pública de asunto: Clave RSA de 2048 bits
Versión: 1
c:\Program Files (x86)\Java\jre1.8.0_361\bin>
```

Una vez registrada la aplicación, nos pedirá descargar un archivo a nuestro proyecto:



Quedando de la siguiente manera:



En el siguiente paso, el sistema pedirá configurar los gradle del proyecto:



Quedando de la siguiente manera:

Project: build.gradle.kts

```
1 // Top-level build file where you can add configuration options common to all sub-projects/modules.  
2 plugins {  
3     id("com.android.application") version "8.1.3" apply false  
4     id("org.jetbrains.kotlin.android") version "1.9.0" apply false  
5     id("com.google.gms.google-services") version "4.4.0" apply false  
6 }
```

Module: build.gradle.kts

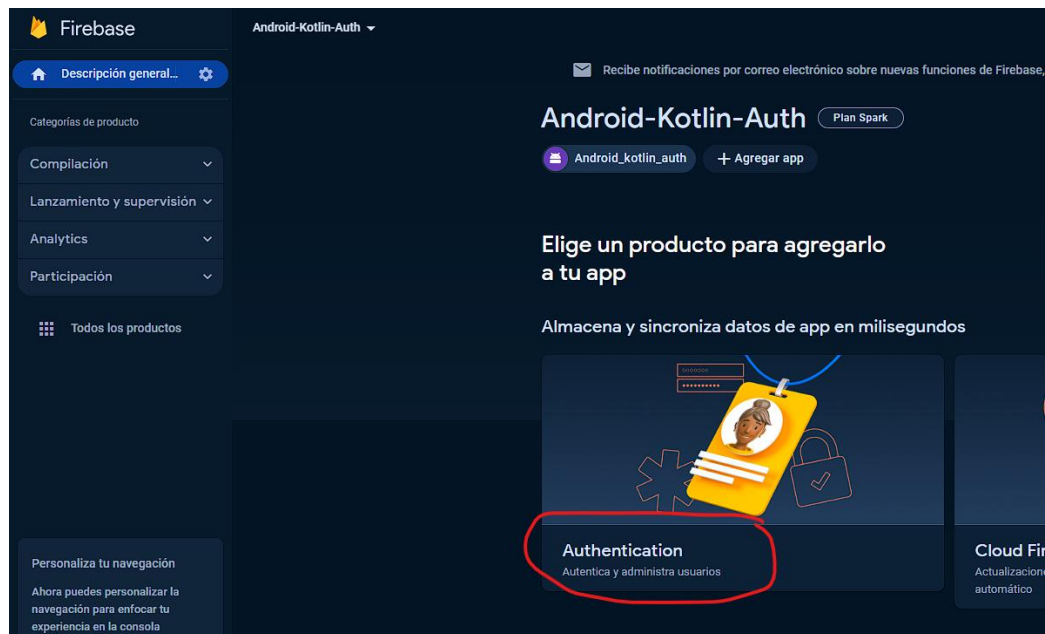
```
1 plugins { this: PluginDependenciesSpecScope  
2     id("com.android.application")  
3     id("org.jetbrains.kotlin.android")  
4     id("com.google.gms.google-services")  
5 }
```

```
39 dependencies { this: DependencyHandlerScope  
40  
41     implementation("androidx.core:core-ktx:1.9.0")  
42     implementation("androidx.appcompat:appcompat:1.6.1")  
43     implementation("com.google.android.material:material:1.10.0")  
44     implementation("androidx.constraintlayout:constraintlayout:2.1.4")  
45     testImplementation("junit:junit:4.13.2")  
46     androidTestImplementation("androidx.test.ext:junit:1.1.5")  
47     androidTestImplementation("androidx.test.espresso:espresso-core:3.5.1")  
48     implementation(platform("com.google.firebase:firebase-bom:32.6.0"))  
49 }
```

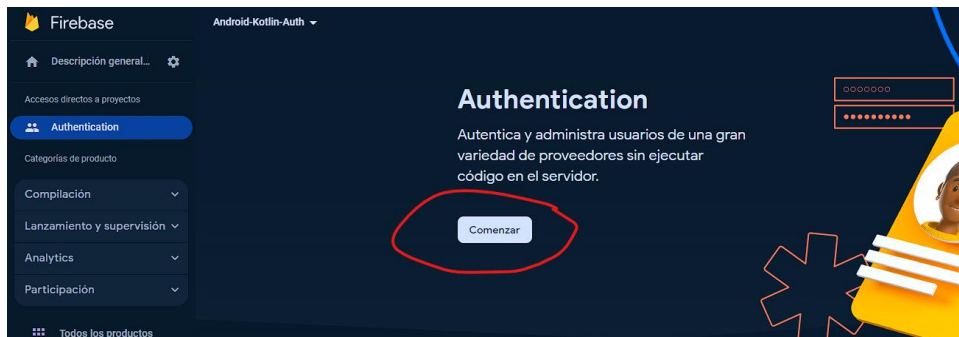

Finalmente, daremos clic en “Ir a la consola”



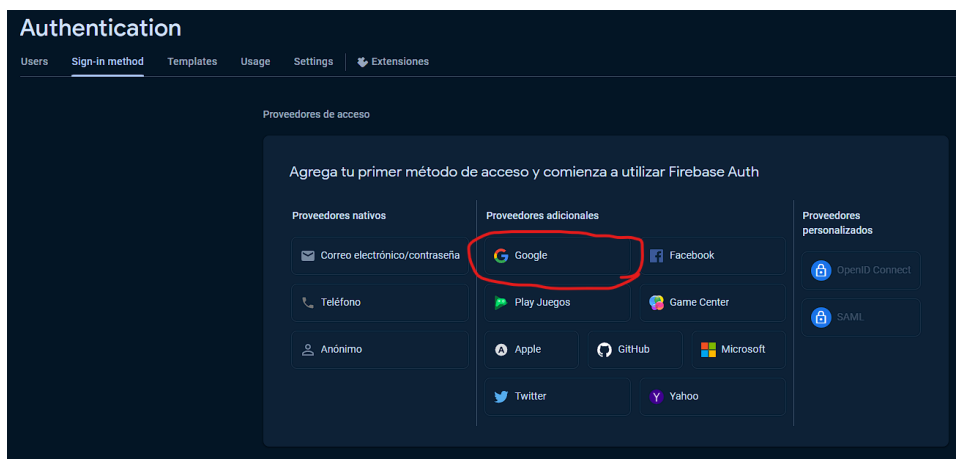
Una vez enlazada la aplicación, procederemos a configurar la autenticación con Google, para ello, daremos clic en la siguiente opción en Firebase:



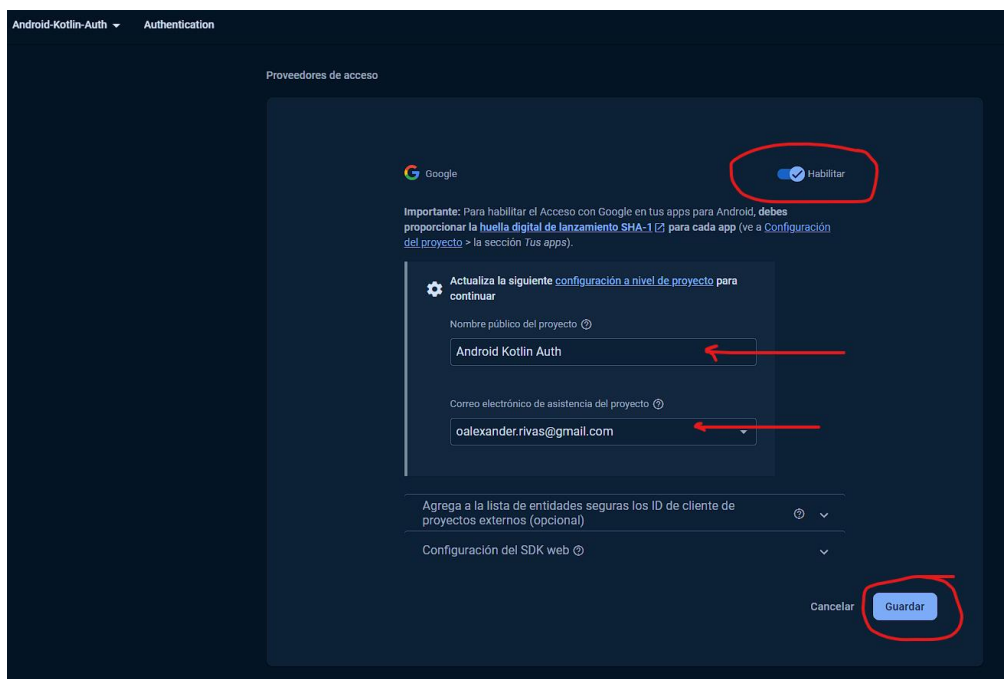
Luego, clic en comenzar:



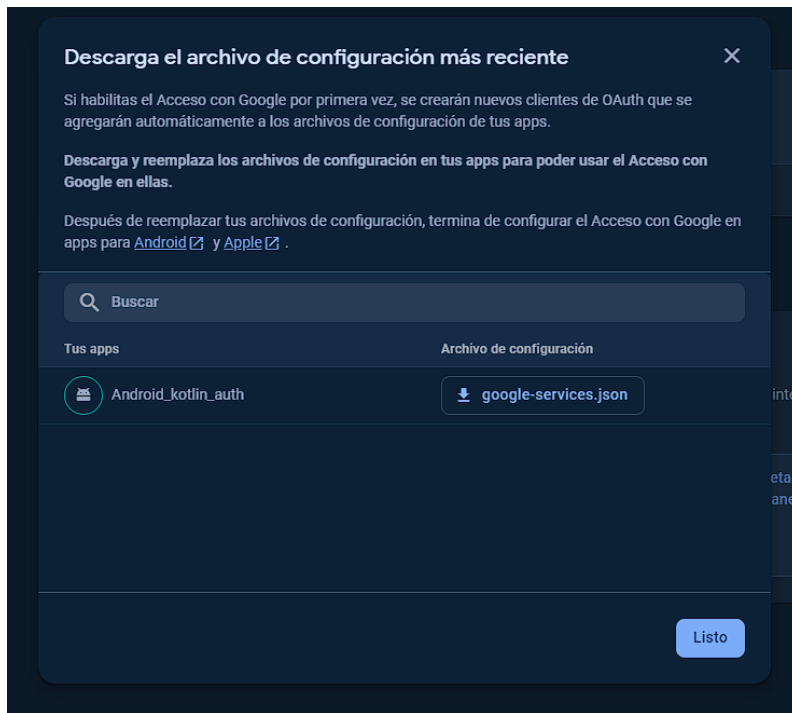
Entre los métodos ofrecidos, seleccionamos Google:



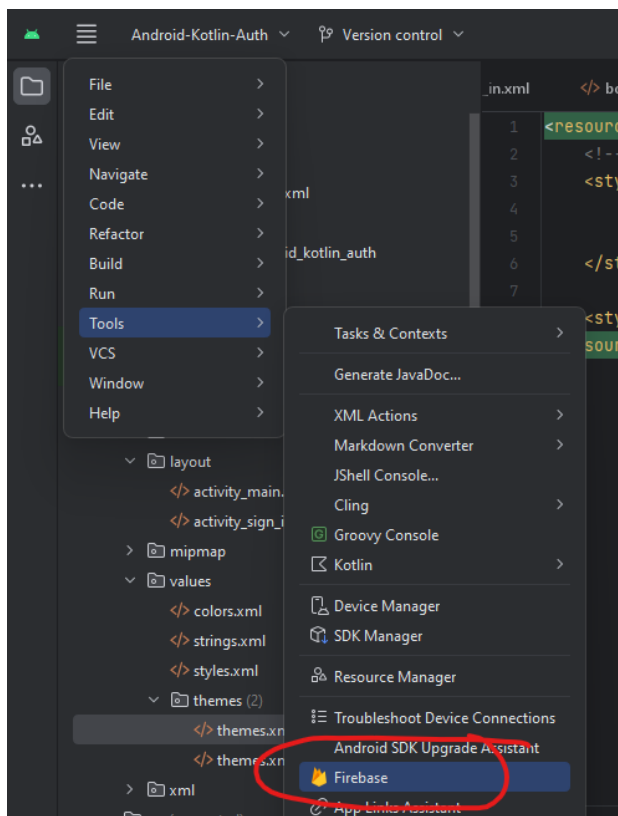
Configuramos las siguientes opciones, y damos clic en Guardar:



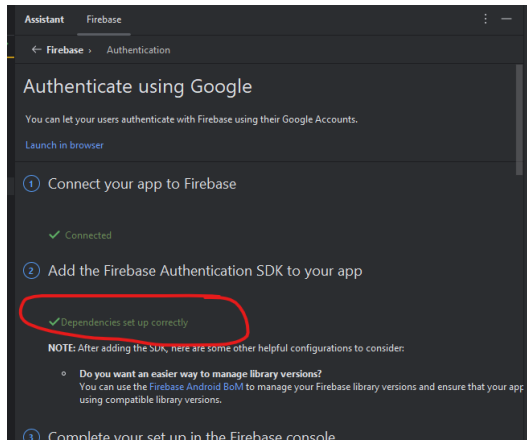
Esto generará nuevos archivos de configuración, que deberemos volver a descargar:



Una vez realizado este paso, procederemos a instalar el SDK de Firebase, esto usando la opción de menú Tools->Firebase:



Luego nos habilitará una ventana, donde podremos instalar el SDK:



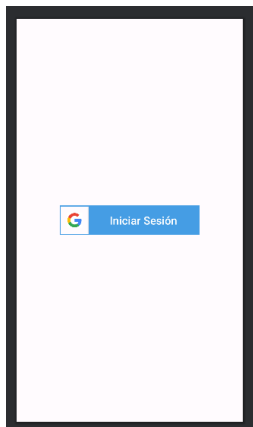
Creación de Activity y Autenticación dentro de la Aplicación

Una vez realizado este paso, procederemos a crear la interfaz de SignInActivity:

Layout:

```
10 <RelativeLayout
11     android:layout_width="match_parent"
12     android:layout_height="match_parent"
13     android:padding="16dp"
14     tools:context=".SignInActivity"
15     tools:layout_editor_absoluteX="-102dp"
16     tools:layout_editor_absoluteY="188dp">
17
18     <LinearLayout
19         android:layout_width="wrap_content"
20         android:layout_height="wrap_content"
21         android:layout_centerInParent="true"
22         android:layout_centerVertical="true"
23         android:layout_marginTop="50dp"
24         android:background="@drawable/border">
25
26         <ImageView
27             android:layout_width="28dp"
28             android:layout_height="28dp"
29             android:layout_alignParentLeft="true"
30             android:layout_centerVertical="true"
31             android:layout_gravity="center"
32             android:layout_marginLeft="11dp"
33             android:layout_marginRight="11dp"
34             android:background="@color/white"
35             android:src="@drawable/fui_ic_google_color_24dp" />
36
37         <androidx.appcompat.widget.AppCompatButton
38             android:id="@+id/signInButton"
39             android:layout_width="200dp"
40             android:layout_height="50dp"
41             android:background="@color/blue"
42             android:text="Iniciar Sesión"
43             android:textAllCaps="false"
44             android:textColor="@color/white"
45             android:textSize="20sp" />
46     </LinearLayout>
47 </RelativeLayout>
```

Vista previa:

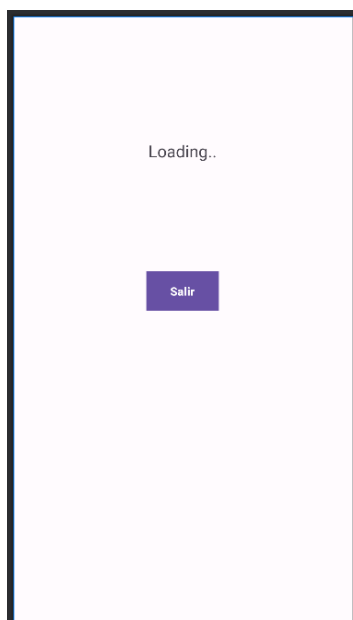


Y ahora la interfaz de MainActivity:

Layout:

```
1  <?xml version="1.0" encoding="utf-8"?>
2  <androidx.constraintlayout.widget.ConstraintLayout xmlns:android="http://schemas.android.com/apk/res/android"
3      xmlns:app="http://schemas.android.com/apk/res-auto"
4      xmlns:tools="http://schemas.android.com/tools"
5      android:layout_width="match_parent"
6      android:layout_height="match_parent"
7      tools:context=".MainActivity">
8
9      <TextView
10         android:id="@+id/name"
11         android:layout_width="wrap_content"
12         android:layout_height="wrap_content"
13         android:text="Loading.."
14         android:textSize="20dp"
15         app:layout_constraintBottom_toBottomOf="parent"
16         app:layout_constraintEnd_toEndOf="parent"
17         app:layout_constraintHorizontal_bias="0.498"
18         app:layout_constraintStart_toStartOf="parent"
19         app:layout_constraintTop_toTopOf="parent"
20         app:layout_constraintVertical_bias="0.212" />
21
22      <Button
23         android:id="@+id/logout_button"
24         android:layout_width="wrap_content"
25         android:layout_height="wrap_content"
26         android:layout_marginTop="132dp"
27         android:text="Salir"
28         android:background="@color/red"
29         android:textColor="@color/white"
30         android:textStyle="bold"
31         app:layout_constraintEnd_toEndOf="parent"
32         app:layout_constraintHorizontal_bias="0.498"
33         app:layout_constraintStart_toStartOf="parent"
34         app:layout_constraintTop_toBottomOf="@+id/name" />
35
36  </androidx.constraintlayout.widget.ConstraintLayout>
```

Vista previa:



Para habilitar el funcionamiento de login, agregaremos el siguiente código en el SignInActivity:

Método onCreate: se inicializa variable de Firebase y se valida si ya se encuentra un usuario autenticado:

```
15 </> class SignInActivity : AppCompatActivity() {
16     companion object {
17         private const val RC_SIGN_IN = 9001
18     }
19
20     private lateinit var auth: FirebaseAuth
21
22     override fun onCreate(savedInstanceState: Bundle?) {
23         super.onCreate(savedInstanceState)
24         setContentView(R.layout.activity_sign_in)
25
26         auth = FirebaseAuth.getInstance()
27
28         val currentUser = auth.currentUser
29
30         if (currentUser != null) {
31             // The user is already signed in, navigate to MainActivity
32             val intent = Intent(packageContext, this, MainActivity::class.java)
33             startActivity(intent)
34             finish()
35         }
36
37         val signInButton = findViewById<Button>(R.id.signInButton)
38         signInButton.setOnClickListener { it: View!
39             signIn()
40         }
41     }
```

Método clic del botón, con los respectivos eventos de respuestas de autenticación:

```
43     private fun signIn() {
44         val gso = GoogleSignInOptions.Builder(GoogleSignInOptions.DEFAULT_SIGN_IN)
45             .requestIdToken(getString(R.string.default_web_client_id))
46             .requestEmail()
47             .build()
48
49         val googleSignInClient = GoogleSignIn.getClient(activity, gso)
50         val signInIntent = googleSignInClient.signInIntent
51         startActivityForResult(signInIntent, RC_SIGN_IN)
52     }
53
54     override fun onActivityResult(requestCode: Int, resultCode: Int, data: Intent?) {
55         super.onActivityResult(requestCode, resultCode, data)
56
57         if (requestCode == RC_SIGN_IN) {
58             val task = GoogleSignIn.getSignedInAccountFromIntent(data)
59             try {
60                 val account = task.getResult(ApiException::class.java)
61                 firebaseAuthWithGoogle(account.idToken!!)
62             } catch (e: ApiException) {
63                 Toast.makeText(context, "Autenticación con Google fallida: ${e.message}", Toast.LENGTH_SHORT).show()
64             }
65         }
66     }
67
68     private fun firebaseAuthWithGoogle(idToken: String) {
69         val credential = GoogleAuthProvider.getCredential(idToken, null)
70         auth.signInWithCredential(credential)
71             .addOnCompleteListener(this) { task ->
72                 if (task.isSuccessful) {
73                     val user = auth.currentUser
74                     Toast.makeText(context, "Autenticación como ${user?.displayName}", Toast.LENGTH_SHORT).show()
75                     startActivity(Intent(packageContext, MainActivity::class.java))
76                     finish()
77                 } else {
78                     Toast.makeText(context, "Autenticación fallida", Toast.LENGTH_SHORT).show()
79                 }
80             }
81     }
82 }
```


Para el MainActivity, agregamos el siguiente código:

Método onCreate: se inicializa variable de Firebase y se valida si ya se encuentra un usuario autenticado:

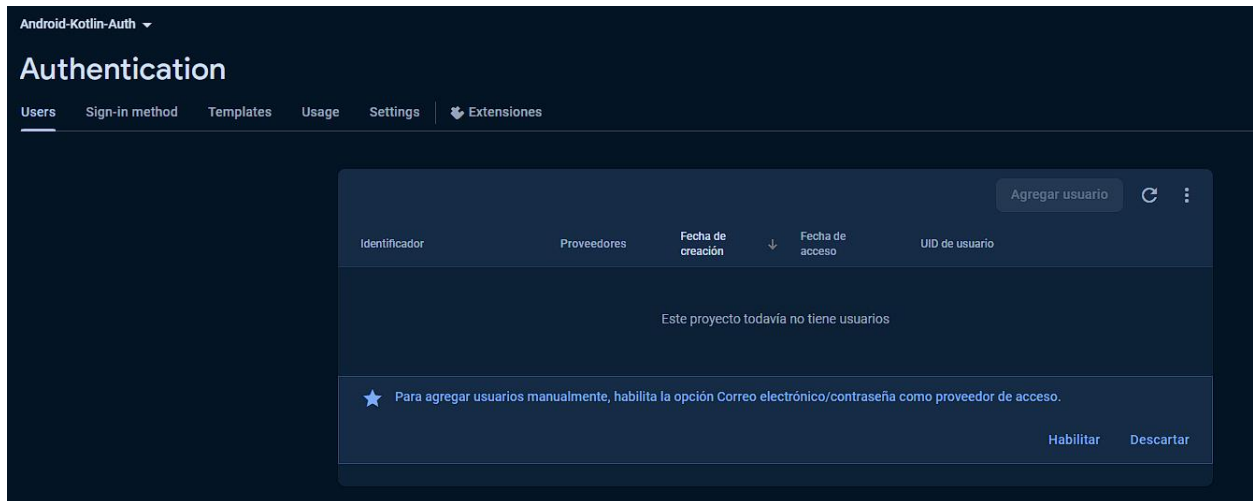
```
16 class MainActivity : AppCompatActivity() {
17     private lateinit var mGoogleSignInClient: GoogleSignInClient
18     private lateinit var mAuth: FirebaseAuth
19
20     override fun onCreate(savedInstanceState: Bundle?) {
21         super.onCreate(savedInstanceState)
22         setContentView(R.layout.activity_main)
23
24         mAuth = FirebaseAuth.getInstance()
25
26         val gso = GoogleSignInOptions.Builder(GoogleSignInOptions.DEFAULT_SIGN_IN)
27             .requestIdToken(getString(R.string.default_web_client_id))
28             .requestEmail()
29             .build()
30
31         mGoogleSignInClient = GoogleSignIn.getClient(this, gso)
32
33         val textView = findViewById<TextView>(R.id.name)
34
35         val auth = FirebaseAuth
36         val user = auth.currentUser
37
38         if (user != null) {
39             val userName = user.displayName
40             textView.text = "Bienvenido, ${user.displayName} \nCorreo: ${user.email}"
41         } else {
42             val intent = Intent(packageContext, SignInActivity::class.java)
43             startActivity(intent)
44             finish()
45         }
46
47         val sign_out_button = findViewById<Button>(R.id.logout_button)
48         sign_out_button.setOnClickListener {
49             signOutAndStartSignInActivity()
50         }
51     }
52 }
```

Método clic del botón cerrar sesión:

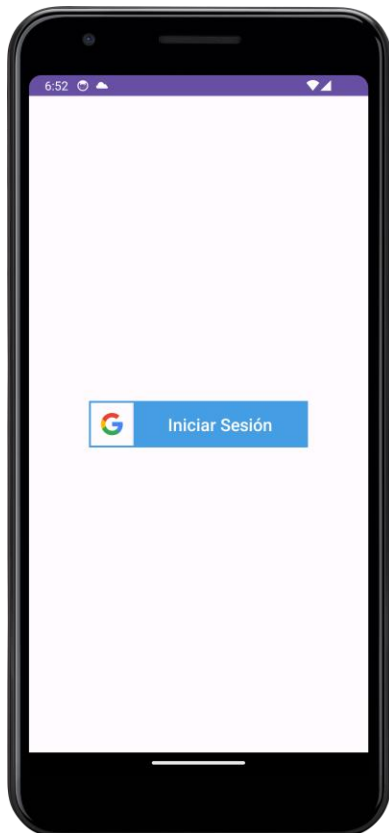
```
52
53     private fun signOutAndStartSignInActivity() {
54         mAuth.signOut()
55         mGoogleSignInClient.signOut().addOnCompleteListener(this) {
56             // Optional: Update UI or show a message to the user
57             val intent = Intent(packageContext, SignInActivity::class.java)
58             startActivity(intent)
59             finish()
60             Toast.makeText(context, "Sesión cerrada correctamente", Toast.LENGTH_SHORT).show()
61         }
62     }
63 }
```

Pruebas de funcionamiento

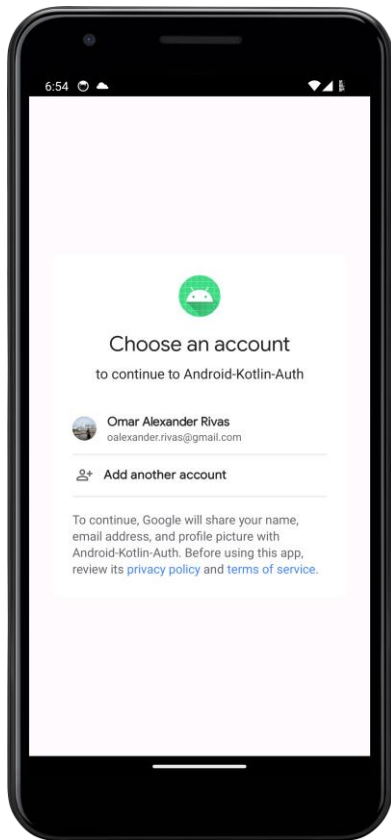
Para ver los usuarios que han iniciado sesión usando el servicio, accedemos desde el sitio web de firebase, donde podemos validar que no existe ni aún un usuario existente:



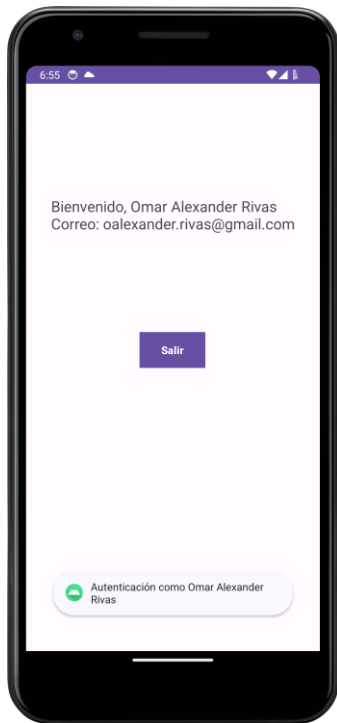
Iniciamos la aplicación:



Al dar clic en “Iniciar Sesión”, nos pedirá con que cuenta existente en el SO queremos iniciar sesión, así como también agregar una nueva cuenta al dispositivo:



Seleccionamos una, y esto nos llevará a la actividad principal, donde se mostrará nuestro nombre y dirección de correo electrónico utilizada para el login:



Al dar clic en cerrar sesión, regresaremos a la actividad de Login:



Podemos verificar en el sitio web de Firebase, que ahora el usuario ya se encuentra registrado:

Android-Kotlin-Auth ▾

Authentication

[Users](#) [Sign-in method](#) [Templates](#) [Usage](#) [Settings](#) | [Extensiones](#)

Agregar usuario

| Identificador | Proveedores | Fecha de creación | ↓ | Fecha de acceso | UID de usuario |
|------------------------------|-------------|-------------------|---|-----------------|------------------------------|
| oalexander.rivas@gmail.co... | | 3 dic 2023 | | 3 dic 2023 | CleavQ8FrRdES4enTVybPENuhAa2 |

Filas por página: 50 ▾ 1 – 1 of 1 < >