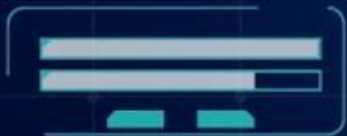


# Curso de Desarrollo Móvil en Android





## Bienvenido alumno

### Sesiones



Completadas



6

Exámenes

5

Tareas

5



José María Cruz Parada  
chema.cruz.p@gmail.com



Resumen de curso



Reglamento



Temario



Sesiones

0



Exámenes

5

Tareas

5



Fundamentos de Java



Fundamentos de Android

Sesión 3



Ciclo de vida del Activity

Sesión 4



Componentes gráficos

Sesión 5



Fragments

Sesión 6



Servicios Web

Sesión 7



Firebase

Sesión 8



Utilidades avanzadas



## Firebase



¿Qué es un Firebase?





## Firestore



### Authentication

Autentica y administra usuarios



## Firestore



### Cloud Firestore

Actualizaciones en tiempo real, consultas poderosas y ajuste de escala automático



## Firestore



### Realtime Database

Almacena y sincroniza datos en tiempo real







## Firestore



### Storage

Almacena y recupera contenido generado por el usuario



## Firebase



### Hosting

Implementa apps web en segundos



## Firestore



### Cloud Messaging

Infraestructura de mensajería push  
multiplataforma entre tus servidores y los  
dispositivos de los usuarios



### Firebase



#### Crashlytics

Haz un seguimiento en tiempo real de los problemas de estabilidad que afectan la calidad de una app, para priorizarlos y corregirlos



## Firestore

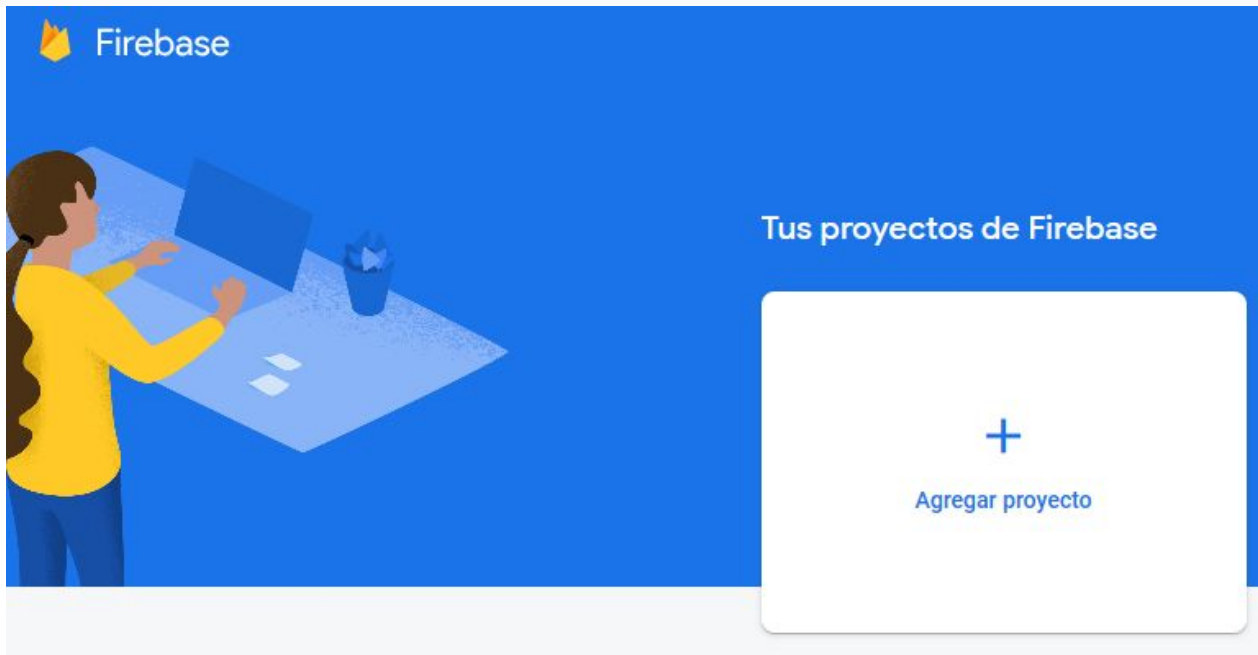


# Instalación





## Firebase






## Firebase

Comencemos con el  
nombre de tu proyecto<sup>?</sup>

Nombre del proyecto

curso-android-ieca

 curso-android-ieca



Confirmo que usaré Firebase exclusivamente para fines relacionados con mi trabajo, empresa, oficio o profesión.

Continuar



## Firebase



curso-android-ieca



Tu proyecto nuevo está listo

Continuar





## Firestore

Comienza por agregar  
Firestore a tu app

iOS+



Agrega una app para comenzar


# Firestore

## Agrega Firebase a tu app para Android

1 Registrar app

Nombre del paquete de Android ⓘ

com.company.appname

Sobrenombre de la app (opcional) 

Mi app para Android

Certificado de firma SHA-1 de depuración (opcional) ⓘ

[illegible]

 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.

Registrar app





## Firestore



New Project

**Empty Views Activity**

Creates a new empty activity

Name

Package name

Save location

Language

Minimum SDK

Your app will run on approximately 95.4% of devices.  
[Help me choose](#)





## Firestore



The screenshot shows an IDE interface. On the left, the Project Structure dialog is open, displaying a tree view of the project structure. The 'app' module is selected, and its 'build.gradle' file is highlighted. The right pane shows the content of the selected 'build.gradle' file, which contains the following code:


```
1 plugins {  
2     id 'com.android.application'  
3     id 'org.jetbrains.kotlin.android'  
4 }  
5  
6 android {  
7     namespace 'com.example.firebasetest'  
8     compileSdk 33  
9 }
```



## Firebase

### 2 Descargar y, luego, agregar el archivo de configuración

Android Studio a continuación | [Unity](#) | [C++](#)

 Descargar google-services.json

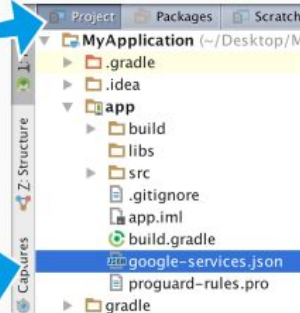
Cambia a la vista **Proyecto** en Android Studio para ver el directorio raíz de tu proyecto.

Mueve el archivo `google-services.json` descargado al directorio raíz de tu módulo (nivel de app).



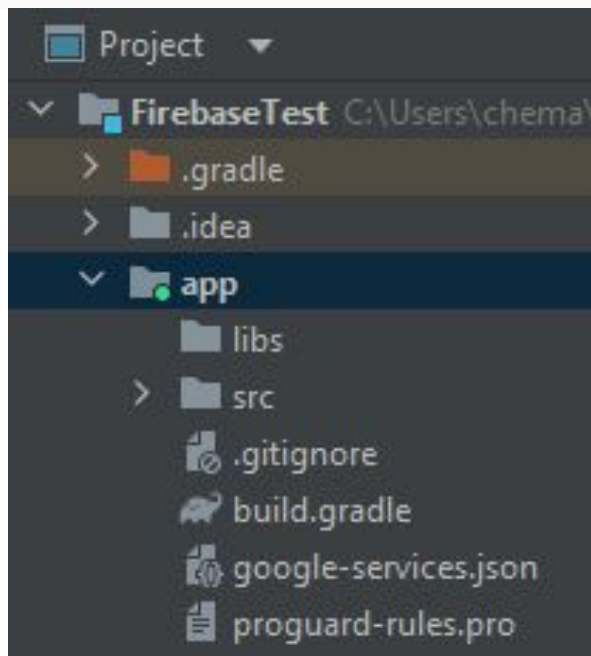
google-services.json

Siguiente



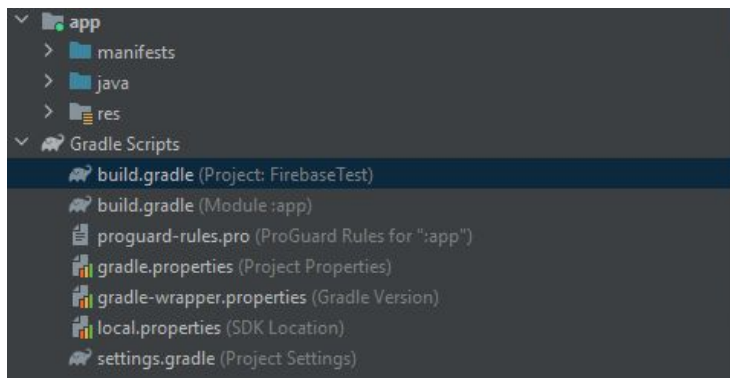


## Firestore





## Firestore

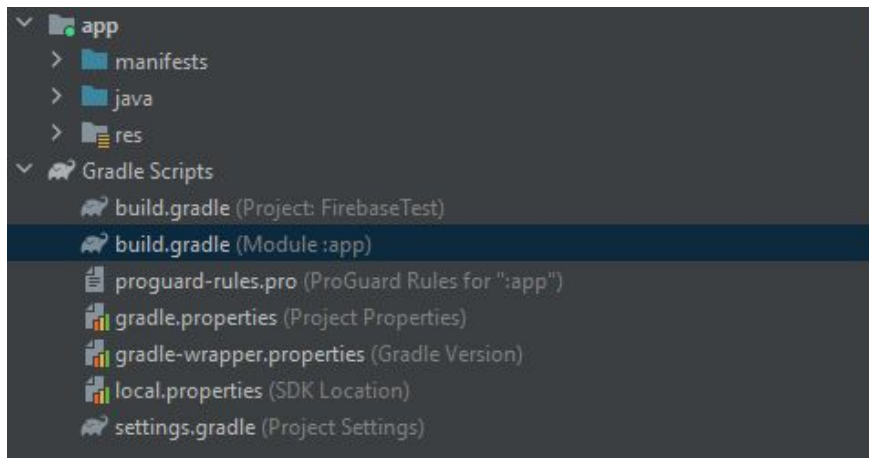


Gradle files have changed since last project sync. A project sync may be necessary for the IDE to work properly.

```
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 '8.0.2' apply false
4     id 'com.android.library' version '8.0.2' apply false
5     id 'org.jetbrains.kotlin.android' version '1.8.20' apply false
6
7     // Agregamos las dependencias de Google
8     id 'com.google.gms.google-services' version '4.3.15' apply false
9 }
```



## Firestore



Gradle files have changed since last project sync. A project sync may be needed.


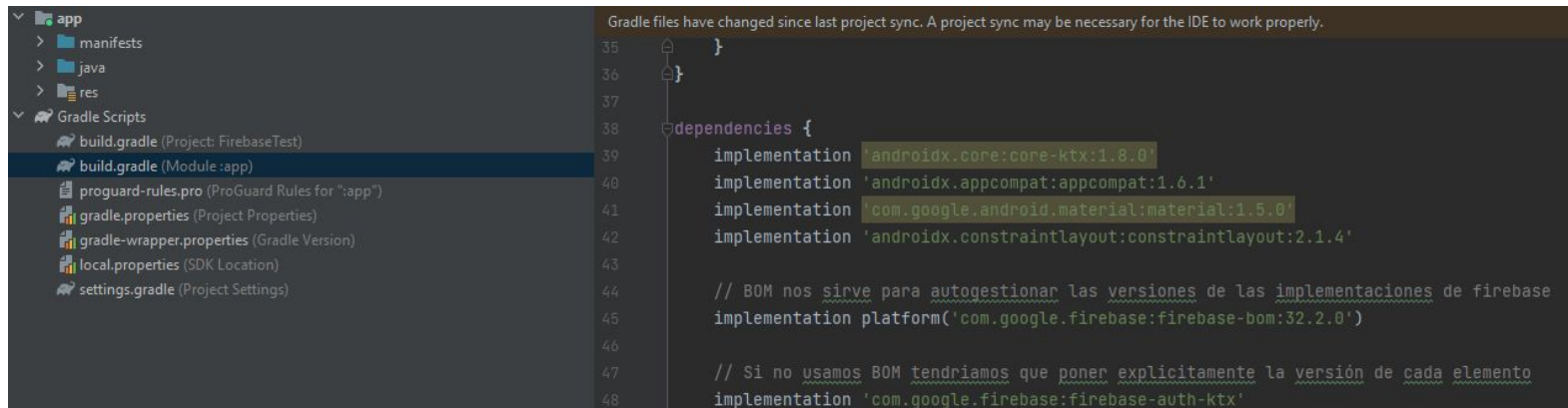

```
1  plugins {  
2      id 'com.android.application'  
3      id 'org.jetbrains.kotlin.android'  
4  
5      // Agregamos los servicios de Google  
6      id 'com.google.gms.google-services'  
7  }  
8  
9  android {  
10     namespace 'com.example.firebasetest'
```







## Firestore



```
35  }
36  }
37
38  dependencies {
39      implementation 'androidx.core:core-ktx:1.8.0'
40      implementation 'androidx.appcompat:appcompat:1.6.1'
41      implementation 'com.google.android.material:material:1.5.0'
42      implementation 'androidx.constraintlayout:constraintlayout:2.1.4'
43
44      // BOM nos sirve para autogestionar las versiones de las implementaciones de firebase
45      implementation platform('com.google.firebase:firebase-bom:32.2.0')
46
47      // Si no usamos BOM tendríamos que poner explícitamente la versión de cada elemento
48      implementation 'com.google.firebase:firebase-auth-ktx'
```



### Firestore



¡Listo!

Asegúrate de revisar la [documentación](#) para conocer los primeros pasos con cada producto de Firebase que quieras usar en tu app.

También puedes explorar las [apps de muestra de Firebase](#).

También puedes ir a console para explorar Firebase

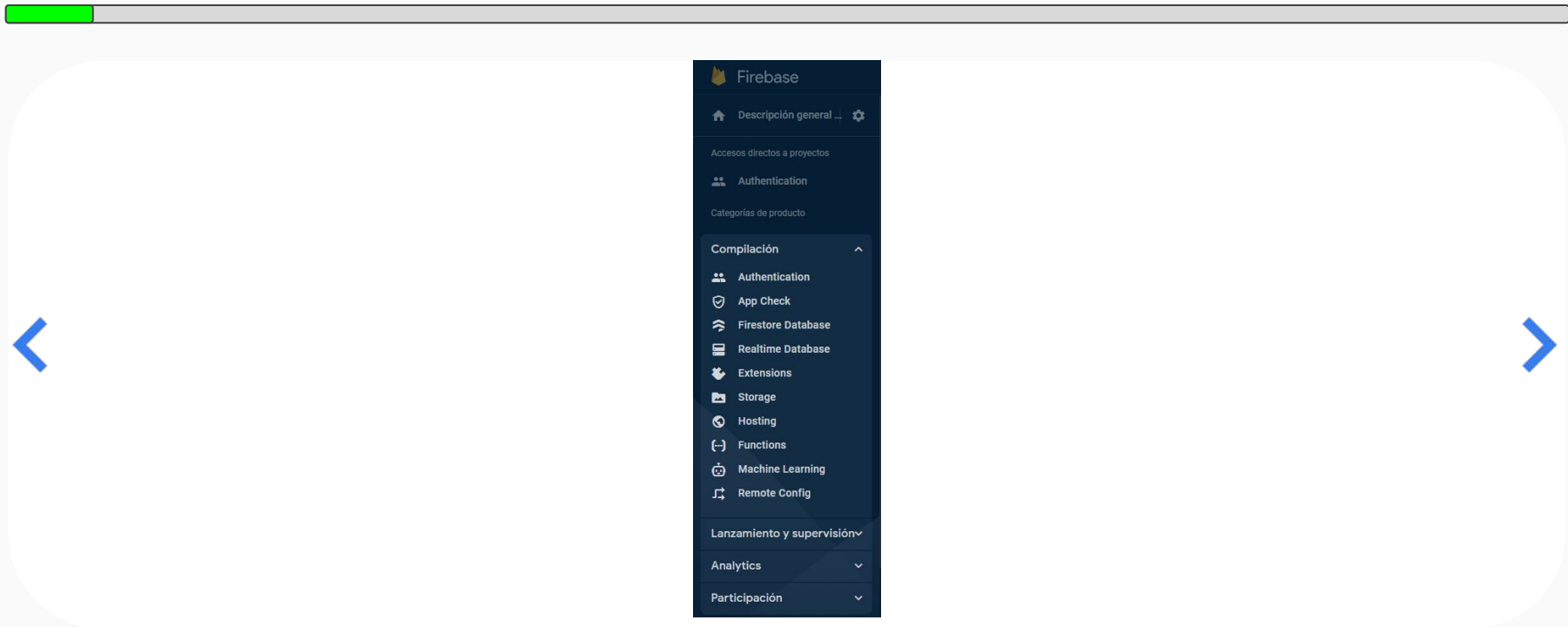
[Anterior](#)

[Ir a la consola](#)





## Firestore





## Firestore

curso-android-ica

?

Configuración de proyecto

General

Cloud Messaging

Integraciones

Cuentas de servicio

Privacidad de los datos

Usuarios y permisos

Tu proyecto

Nombre del proyecto

curso-android-ica

ID del proyecto

curso-android-ica

Número de proyecto

988957629772

Ubicación predeterminada de los recursos de GCP

Aún sin seleccionar

Clave de API web

No hay clave de API web para este proyecto

Entorno

Este parámetro de configuración permite personalizar tu proyecto para las diferentes etapas del ciclo de vida de la app

Tipo de entorno

Sin especificar

Configuración pública

Esta configuración controla las instancias de tu proyecto que se muestran al público

Nombre público

project-988957629772

Correo electrónico de asistencia

No configurada



## Firebase



### Authentication

Autentica y administra usuarios de una gran variedad de proveedores sin ejecutar código en el servidor.

Comenzar





## Firestore




Agrega tu primer método de acceso y comienza a utilizar Firebase Auth

### Proveedores nativos

 Correo electrónico/contraseña


 Teléfono

 Anónimo

### Proveedores adicionales

 Google

 Facebook

 Play Juegos

 Game Center

 Apple

 GitHub

 Microsoft

 Twitter

 Yahoo

### Proveedores personalizados

 OpenID Connect

 SAML





## Firestore



### Proveedores de acceso

Agregar proveedor nuevo

Proveedor

Estado



Correo electrónico/contraseña



Habilitado



Anónimo



Habilitado



# Firestore



<div>Agregar usuario</div>					
Identificador	Proveedores	Fecha de creación	↓	Fecha de acceso	UID de usuario
Este proyecto todavía no tiene usuarios					







## Firestore



Agrega un usuario con correo electrónico y contraseña

Correo electrónico

chema@ieca.edu.mx

Contraseña

1234567890

Cancelar

Agregar usuario



### Firebase

```
// Inicializar FirebaseAuth
auth = FirebaseAuth.getInstance()

val email = "chema@ieca.edu.mx"
val password = "1234567890"

// Iniciar sesión con correo electrónico y contraseña
auth.signInWithEmailAndPassword(email, password)
    .addOnCompleteListener(this) { task ->
        if (task.isSuccessful) {
            // Inicio de sesión exitoso
            val user = auth.currentUser
            // Hacer algo con el usuario autenticado
            Toast.makeText(context, this, text: "Inicio de sesión exitoso", Toast.LENGTH_LONG).show()
        } else {
            // Error en el inicio de sesión
            Toast.makeText(context, this, text: "Error en el inicio de sesión: ${task.exception?.message}", Toast.LENGTH_LONG).show()
        }
    }
}
```



## Firestore



Inicio de sesión exitoso





## Firestore

```
override fun onStart() {
    super.onStart()

    // Verificar si el usuario ya está autenticado
    val currentUser: FirebaseUser? = auth.currentUser
    if (currentUser != null) {
        // El usuario ya está autenticado, realizar alguna acción
    } else {
        // El usuario no está autenticado
        login()
    }
}

fun login() {
    // Iniciar sesión con correo electrónico y contraseña
    auth.signInWithEmailAndPassword(email, password)
        .addOnCompleteListener(this) { task ->
            if (task.isSuccessful) {
                // Inicio de sesión exitoso
                val user = auth.currentUser
                // Hacer algo con el usuario autenticado
                Toast.makeText(context, this, text "Inicio de sesión exitoso", Toast.LENGTH_LONG).show()
            } else {
                // Error en el inicio de sesión
                Toast.makeText(context, this, text "Error en el inicio de sesión: ${task.exception?.message}", Toast.LENGTH_LONG).show()
            }
        }
}
```



### Firebase

```
fun loginAnonymous() {  
    // Iniciar sesión de forma anónima  
    auth.signInAnonymously()  
        .addOnCompleteListener { task ->  
        if (task.isSuccessful) {  
            // Inicio de sesión anónimo exitoso  
            val user = auth.currentUser  
            // Hacer algo con el usuario autenticado  
            Toast.makeText(context, this, text: "Inicio anónimo de sesión exitoso", Toast.LENGTH_LONG).show()  
        } else {  
            // Error en el inicio de sesión anónimo  
            Toast.makeText(context, this, text: "Error en el inicio de sesión anónimo: ${task.exception?.message}", Toast.LENGTH_SHORT).show()  
        }  
    }  
}
```



## Firestore

```
override fun onStart() {
    super.onStart()

    // Verificar si el usuario ya está autenticado
    val currentUser: FirebaseUser? = auth.currentUser

    if (currentUser != null) {
        // El usuario ya está autenticado
        if (currentUser.email != "") {
            Toast.makeText(context, this, text: "Bienvenido " + currentUser.email, Toast.LENGTH_LONG).show()
        } else {
            // El usuario ya está autenticado como anónimo, realizar alguna acción
            Toast.makeText(context, this, text: "Bienvenido anónimo", Toast.LENGTH_LONG).show()
        }
    } else {
        // El usuario no está autenticado
        loginAnonymous()
    }
}
```



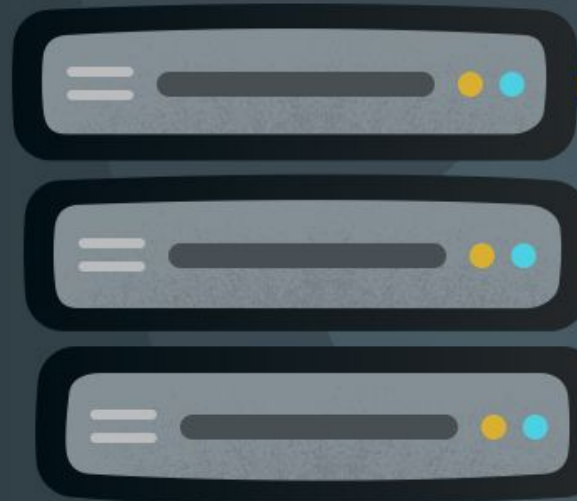
## Firebase



### Realtime Database

Almacena y sincroniza datos en tiempo real

Crear una base de datos





## Firestore

### Configurar base de datos



- 1 Opciones de base de datos — 2 Reglas de seguridad

El parámetro de configuración de la ubicación determina en qué lugar se almacenarán tus datos de Realtime Database.

Ubicación de Realtime Database

Estados Unidos (us-central1)



Cancelar

Siguiente





### Firebase



```
{  
  /* Visit https://firebase.google.com/docs/database/security to learn more about security rules. */  
  "rules": {  
    ".read": true,  
    ".write": false  
  }  
}
```





## Firestore

<https://curso-android-ieca-default-rtdb.firebaseio.com> > seasons

seasons

▼ 0

description: "Fundamentos de Java"

name: "Sesión 1"

▶ 1

▶ 2

▶ 3

▶ 4

▶ 5

▶ 6

▶ 7





## Firebase



```
data class Season(  
    val description: String = "",  
    val name: String = ""  
)
```





## Firebase

```
override fun onCreate(savedInstanceState: Bundle?) {  
    super.onCreate(savedInstanceState)  
    setContentView(R.layout.activity_main)  
  
    // Inicializar FirebaseAuth  
    auth = FirebaseAuth.getInstance()  
  
    // Inicializar FirebaseDatabase  
    database = FirebaseDatabase.getInstance()  
}
```



### Firestore

```
fun getSeasons() {  
    val reference = database.getReference( path: "seasons")  
  
    reference.addListenerForSingleValueEvent(object : ValueEventListener {  
        override fun onDataChange(dataSnapshot: dataSnapshot) {  
            for (seasonSnapshot in dataSnapshot.children) {  
                val season = seasonSnapshot.getValue(Season::class.java)  
                // Hacer algo con cada valor de la lista "seasons"  
                Toast.makeText( context: this@MainActivity, text: season?.name + " : " + season?.description, Toast.LENGTH_LONG).show()  
            }  
        }  
  
        override fun onCancelled(databaseError: DatabaseError) {  
            // Manejar el error  
            println("Error al leer los datos: ${databaseError.message}")  
        }  
    })  
}
```



Sesión 7



## Firestore



Sesión 1 : Fundamentos de Java





## Firebase

```
fun setSeason(seasonID : String) {  
    val reference = database.getReference( path: "seasons")  
  
    // Crea un objeto Sesión  
    val season = Season( description: "Descripción de Prueba", name: "Sesión test",)  
  
    // Escribe los datos en la base de datos  
    reference.child(seasonID).setValue(season).addOnCompleteListener { it: Task<Void!>  
        Toast.makeText( context: this@MainActivity, text: "Se actualizo la base de datos", Toast.LENGTH_LONG).show()  
    }  
}
```



## Firebase



▼ — 8

description: "Descripción de Prueba"

name: "Sesión test"







REST

Retrofit



## Cloud Firestore

Actualizaciones en tiempo real, consultas potentes y ajuste de escala automático

Crear base de datos





REST

Retrofit

## Crear base de datos



Crea reglas de seguridad de  
Cloud Firestore



2 Configura la ubicación de  
Cloud Firestore

La configuración de la ubicación es el lugar donde se almacenarán tus datos de Cloud Firestore.



No podrás cambiar la ubicación después de configurarla. Además, la configuración de ubicación será la de tu bucket predeterminado de Cloud Storage.

[Más información](#)

Ubicación de Cloud Firestore

nam5 (United States)



Si habilitas Cloud Firestore, no podrás usar Cloud Datastore en este proyecto

Cancelar

Habilitar



REST

Retrofit

```
// Inicializar FirebaseFirestore  
firestore = FirebaseFirestore.getInstance()
```



REST

Retrofit

```
fun updateUser(id: String, email: String?) {  
    val user = hashMapOf(  
        "email" to email,  
    )  
    firestore.collection(collectionPath: "users") CollectionReference  
        .document(id) DocumentReference  
        .set(user) Task<Void>  
        .addOnSuccessListener { documentReference ->  
            Toast.makeText(context: this@MainActivity, text: "Se actualizo la base de datos", Toast.LENGTH_LONG).show()  
        }  
        .addOnFailureListener { e ->  
            println("Error al agregar el documento: $e")  
        }  
}
```



REST

Retrofit



Se actualizo la base de datos



REST

Retrofit



> users > PUcqGXo1ubdl6 <span>Más funciones en Google Cloud</span>		
curso-android-ieca	users	PUcqGXo1ubdl6ekQa7xDyUxVER63
+ Iniciar colección	+ Agregar documento	+ Iniciar colección
users >	PUcqGXo1ubdl6ekQa7xDyUxVER63 >	+ Agregar campo
		email: "chema@ieca.edu.mx"



REST

Retrofit

```
service cloud.firestore {  
  match /databases/{database}/documents {  
    match /users/{userId} {  
      allow read, write: if request.auth != null && request.auth.uid == userId;  
    }  
  }  
}
```



REST

Retrofit

+ Agregar campo

email: "chema@ieca.edu.mx"

name: "Chema Cruz"





REST

Retrofit

```
data class User(  
    val email: String = "",  
    val name: String = ""  
)
```



```
fun getUser(id: String){
    firestore.collection( collectionPath: "Users") CollectionReference
        .document(id).get() Task<DocumentSnapshot!>
        .addOnSuccessListener { documentSnapshot ->
            if (documentSnapshot.exists()) {
                val user = documentSnapshot.toObject(User::class.java)
                // Hacer algo con la nota obtenida
                Toast.makeText( context: this@MainActivity, text: "Hola " + user?.name, Toast.LENGTH_LONG).show()
            } else {
                // El documento no existe
                println("El usuario con el ID $id no existe.")
            }
        }
        .addOnFailureListener { e ->
            println("Error al obtener al usuario: $e")
        }
}
```



REST

Retrofit



Hola Chema Cruz





Fundamentos de Java



Fundamentos de Android

Sesión 3



Ciclo de vida del Activity

Sesión 4



Componentes gráficos

Sesión 5



Fragments

Sesión 6



Servicios Web

Sesión 7



Firebase

Sesión 8



Utilidades avanzadas

# Curso de Desarrollo Móvil en Android

