

¿Qué es Firebase?

Firebase de Google es una plataforma en la nube para el desarrollo de aplicaciones web y móvil. Está disponible para distintas plataformas (iOS, Android y web), con lo que es más rápido trabajar en el desarrollo.

Aunque fue creada en 2011 pasó a ser parte de Google en 2014, comenzando como una base de datos en tiempo real. Sin embargo, se añadieron más y más funciones que, en parte, permitieron agrupar los SDK de productos de Google con distintos fines, facilitando su uso.

¿Para qué sirve Firebase?

Su función esencial es hacer más sencilla la creación de tanto aplicaciones webs como móviles y su desarrollo, procurando que el trabajo sea más rápido, pero sin renunciar a la calidad requerida.

Sus herramientas son variadas y de fácil uso, considerando que su agrupación simplifica las tareas de gestión a una misma plataforma. Las finalidades de las mismas se pueden dividir en cuatro grupos: desarrollo, crecimiento, monetización y análisis. Es especialmente interesante para que los desarrolladores no necesiten dedicarle tanto tiempo al backend, tanto en cuestiones de desarrollo como de mantenimiento.

Tal vez estés buscando en internet: tutorial Firebase, manual firebase o dev firebase, pero la mejor forma de entender en qué consiste Firebase es hacernos una idea de las herramientas que ofrece y a qué están destinadas, algo que veremos a continuación.

Funciones de Firebase

Firebase dispone de diferentes funcionalidades, que se pueden dividir básicamente en 3 grupos: Desarrollo (Develop), Crecimiento (Grow) y Monetización (Earn), a los que hay que sumar la Analítica (Analytics).

Desarrollo

El primer grupo de funciones es conocido como Desarrollo o Develop en Firebase. Como su nombre indica, incluye los servicios necesarios para el desarrollo de un proyecto de aplicación móvil o web. Estos contribuyen a que el proceso sea más rápido, puesto que se dejan determinadas actividades a mano de Firebase, mientras que otras permiten optimizar diversos aspectos para conseguir la calidad deseada.

REALTIME DATABASE

Una de las herramientas más destacadas y esenciales de Firebase son las bases de datos en tiempo real. Estas se alojan en la nube, son No SQL y almacenan los datos como JSON. Permiten alojar y disponer de los datos e información de la aplicación en tiempo real, manteniéndolos actualizados aunque el usuario no realice ninguna acción.

Firebase envía automáticamente eventos a las aplicaciones cuando los datos cambian, almacenando los datos nuevos en el disco. Aunque no hubiera conexión por parte de un usuario, sus datos estarían disponibles para el resto y los cambios realizados se sincronizarían una vez restablecida la conexión.

AUTENTICACIÓN DE USUARIOS

La identificación de los usuarios de una app es necesaria en la mayoría de los casos si estos quieren acceder a todas sus características.

Firebase ofrece un sistema de autenticación que permite tanto el registro propiamente dicho (mediante email y contraseña) como el acceso utilizando perfiles de otras plataformas externas (por ejemplo, de Facebook, Google o Twitter), una alternativa muy cómoda para usuarios reacios a completar el proceso.

Así, este tipo de tareas se ven simplificadas, considerando también que desde aquí se gestionan los accesos y se consigue una mayor seguridad y protección de los datos. Se debe mencionar que Firebase puede guardar en la nube los datos de inicio de sesión con total seguridad, evitando que una persona tenga que identificarse cada vez que abra la aplicación.

ALMACENAMIENTO EN LA NUBE

Firebase cuenta con un sistema de almacenamiento, donde los desarrolladores pueden guardar los ficheros de sus aplicaciones (y vinculándolos con referencias a un árbol de ficheros para mejorar el rendimiento de la app) y sincronizarlos. Al igual que la mayoría de herramientas de Firebase, es personalizable mediante determinadas reglas.

Este almacenamiento es de gran ayuda para tratar archivos de los usuarios (por ejemplo, fotografías que hayan subido), que se pueden servir de forma más rápida y fácil. También hace la descarga de referencias a ficheros más segura.

Ventajas y desventajas

Como se puede concluir a partir de las funcionalidades ofrecidas, esta herramienta presenta numerosos beneficios para los desarrolladores que lo utilicen. Aunque son muchos más, algunos de ellos se recopilan brevemente. Te explicamos a continuación cuáles son los beneficios de Firebase:

Muy recomendable para aplicaciones que necesiten compartir datos en tiempo real.

Sus funcionalidades, además de ser variadas, se complementan muy bien y se pueden gestionar de forma sencilla desde un único panel. Además, no es necesario usar todas estas opciones para la aplicación, pudiendo elegir solo aquellas que más nos interesen.

Facilita el envío de notificaciones: son muy sencillas de implementar y gestionar, además de ser extremadamente útiles para mantener la atención de los usuarios.

Permite la monetización: desde el propio Firebase se puede agregar publicidad a la app, permitiendo fácilmente rentabilizarla obteniendo el ROI.

Engloba Analytics: especializado en determinadas métricas de aplicaciones móviles e integrado en el panel central de Firebase con un funcionamiento muy intuitivo. Esencial para tomar decisiones en distintas fases del proceso.

Google ofrece numerosos documentos y tutoriales a modo introductorio e informativo (con gran profundidad) para que sumergirse en Firebase sea mucho más fácil.

Soporte gratuito vía email, sin importar si el desarrollador utiliza la versión gratuita o de pago.

Escalabilidad: los inicios son gratuitos, pero permite ir adaptándose a las necesidades de la aplicación con diferentes planes de pago.

Ofrece seguridad al usuario: con los certificados SSL.

Permite a los desarrolladores restarle atención al backend y a las infraestructuras complejas para centrarse completamente en otros aspectos.

La desventaja más comúnmente mencionada es el precio. Se ha hablado de la escalabilidad de Firebase, donde el inicio con el plan Spark es gratuito. Sin embargo, tiene limitaciones (principalmente de número de usuarios simultáneos y de espacio de almacenamiento), por lo que puede resultar necesario contratar una versión de pago. Los planes de pago son Flame (25 dólares al mes) y Blaze (se paga según el consumo).

Así que como puedes ver, puedes encontrar Firebase gratis, pero si necesitas trabajar con ello de manera profesional, verás que tienes que buscar cuál es el precio de Firebase. Depende de ti invertir más o menos en un proyecto.

Conclusiones

En definitiva, Firebase es una plataforma muy completa y con grandes y numerosos beneficios, que terminan por dejar en segundo plano las pocas desventajas que presenta.

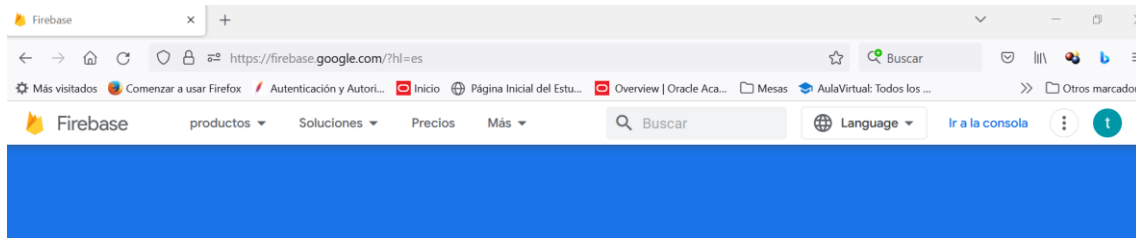
Es especialmente interesante en los inicios de un proyecto, puesto que sus funciones están pensadas para acompañar a este proceso (desarrollo, crecimiento y monetización) y ofrece prácticamente todo lo necesario para “dar el salto”. Es muy relevante, de igual modo, que permita a los desarrolladores centrar sus esfuerzos determinados aspectos, como puede ser el frontend, y dejar otros como el backend en segundo plano. Las diversas herramientas para fomentar el crecimiento son también uno de sus puntos fuertes.

Como hemos mencionado, uno de los principales inconvenientes podría ser la necesidad de pago. Sin embargo, para los proyectos que se encuentren en sus primeras etapas, la versión gratuita de Firebase es más que suficiente. Así, cuando sea necesario actualizar a la versión de pago o abandonar la plataforma para basarse en medios propios, en la mayoría de los casos será consecuencia del crecimiento del proyecto. Como esta evolución suele ir asociada a una rentabilidad, en principio no debería suponer un freno para comenzar con Firebase.

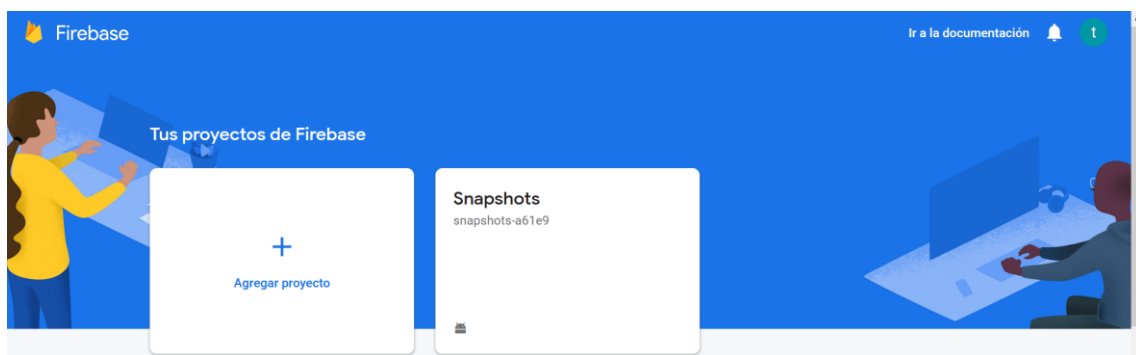
Crear un proyecto en Firebase

Lo primero tenemos que acceder a Firebase con una cuenta de Gmail

<https://firebase.google.com/?hl=es>

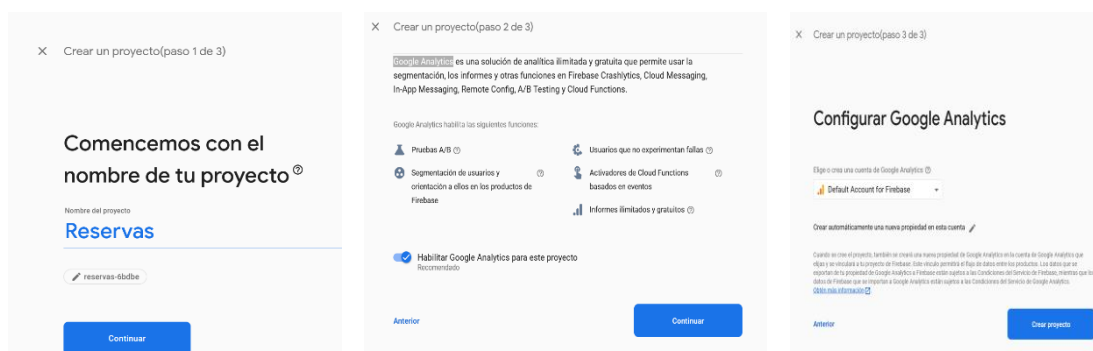


Una vez que estamos en Firebase pulsamos en Ir a la consola y se mostraran todos los proyectos que tenemos y nos permitirá añadir proyecto.

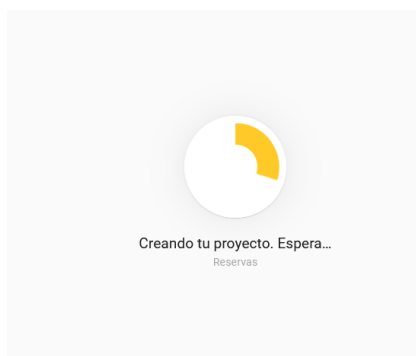


Creación de un proyecto

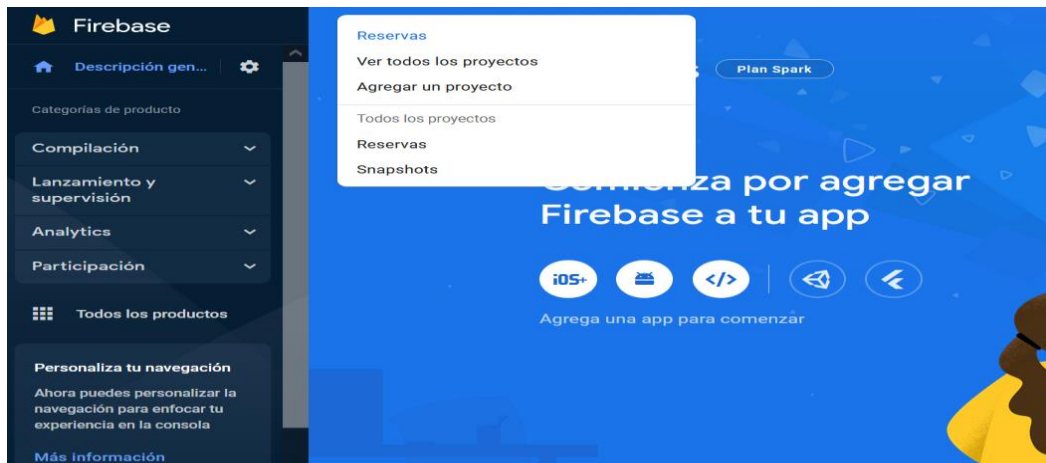
El proyecto se crea en tres pasos que son:



Y empieza la creación del proyecto



Una vez creado el proyecto nos aparece el proyecto creado y si pulsamos arriba a la izquierda podremos ver todos los proyectos que tenemos creado

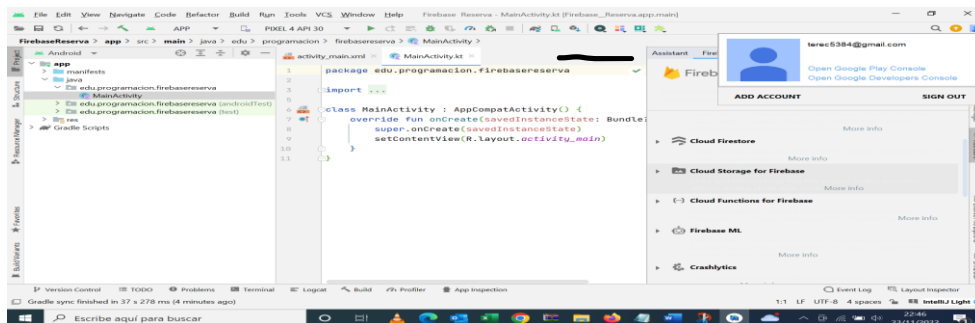


Unir proyecto a la APP

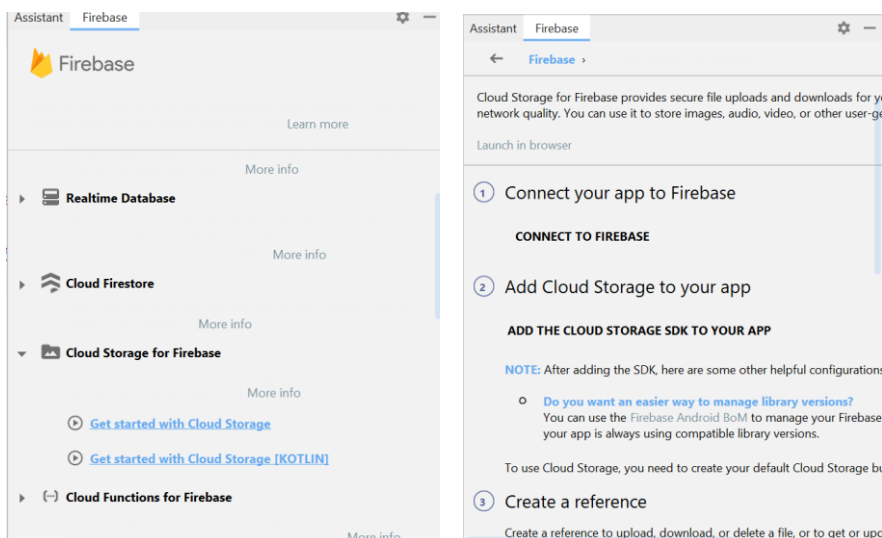
El siguiente paso es unir el proyecto con la App que tenemos creadas en Android Studio. Para ellos nos vamos a android studio.

En el menú Opción Tools -> Firebase

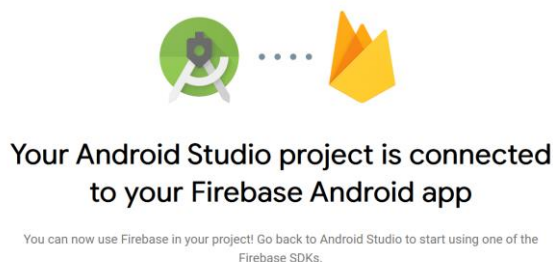
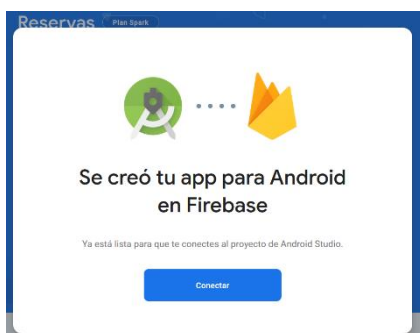
Y ya con el asistente vamos a agregar la dependencia de que nos interese utilizar de Firebase, pero antes deberemos de comprobar que estamos logeados con el usuario de Firebase (esquina superior izquierda)



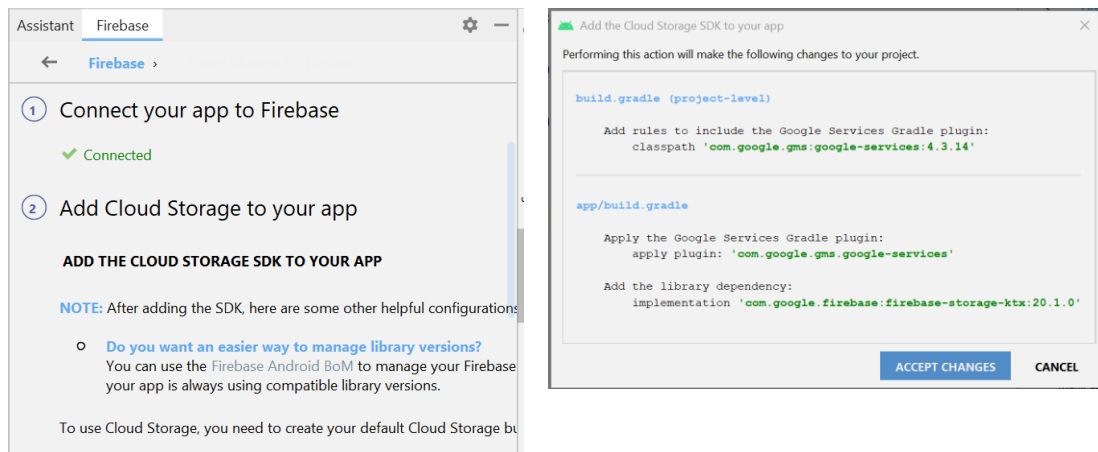
En nuestro caso una vez accedemos al asistente de Firebase iremos a la opción Cloud Storage for Firebase ya que es el servicio que queremos utilizar:



Para terminar el proceso de conexión se nos pide que seleccionemos el proyecto con el que queremos conectarnos y al finalizar nos indicara que estamos conectados



Una vez que comprobamos que estamos conectado podemos pulsar en ADD CLOUD STORAGE...



Comprobamos las dependencias que va a añadir y pulsamos Accept Changes

Una vez que se instalan las dependencias nos indicaran si se han instalado correctamente.

Vamos a comprobar donde están instaladas y que ha instalado

En build.gradle Module ha instalado la dependencia firebase-storage y un nuevo plugin

```
dependencies {
    implementation 'androidx.core:core-ktx:1.7.0'
    implementation 'androidx.appcompat:appcompat:1.5.1'
    implementation 'com.google.android.material:material:1.7.0'
    implementation 'androidx.constraintlayout:constraintlayout:2.1.4'
    implementation 'com.google.firebase:firebase-storage-ktx:20.1.0'
    testImplementation 'junit:junit:4.13.2'
    androidTestImplementation 'androidx.test.ext:junit:1.1.4'
    androidTestImplementation 'androidx.test.espresso:espresso-core:3.5.0'
}

plugins {
    id 'com.android.application'
    id 'org.jetbrains.kotlin.android'
    id 'com.google.gms.google-services'
}
```

En build.gradle Project se instala del servicio

```
buildscript {
    dependencies {
        classpath 'com.google.gms:google-services:4.3.14'
    }
}

// Top-level build file where you can add configuration options common to all sub-projects/modules.
plugins {
    id 'com.android.application' version '7.2.1' apply false
    id 'com.android.library' version '7.2.1' apply false
    id 'org.jetbrains.kotlin.android' version '1.7.10' apply false
}

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

Este fichero se modifica y debe quedar de la siguiente forma:

```
plugins {
    id 'com.android.application' version '7.2.1' apply false
    id 'com.android.library' version '7.2.1' apply false
    id 'org.jetbrains.kotlin.android' version '1.7.10' apply false
    id 'com.google.gms.google-services' version '4.3.14' apply false
}

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


Puedes consultar en la siguiente url el proceso de conexión a firebase:

<https://developer.android.com/studio/write/firebase?hl=es-419>

Añadir la librería Firebase UI

Esta librería va a facilitar el uso de los servicios de Firebase para que sea más sencillo implementar las funciones dentro de la APP.

Accedemos a la url: <https://firebase.google.com/docs/auth/android/firebaseui>

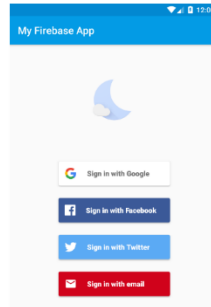
Agregue fácilmente el inicio de sesión a su aplicación de Android con FirebaseUI 

[Enviar comentarios](#)

Y accedemos a FirebaseUI

FirebaseUI es una biblioteca construida sobre el SDK de autenticación de Firebase que proporciona flujos de IU directos para usar en su aplicación. FirebaseUI proporciona los siguientes beneficios:

- **Múltiples proveedores** : flujos de inicio de sesión para correo electrónico/contraseña, enlace de correo electrónico, autenticación telefónica, inicio de sesión de Google, inicio de sesión de Facebook, inicio de sesión de Twitter e inicio de sesión de GitHub.
- **Gestión de cuentas** : flujos para gestionar las tareas de gestión de cuentas, como la creación de cuentas y el restablecimiento de contraseñas.



```
dependencies {  
    // FirebaseUI for Firebase Realtime Database  
    implementation 'com.firebaseui:firebase-ui-database:8.0.2'  
  
    // FirebaseUI for Cloud Firestore  
    implementation 'com.firebaseui:firebase-ui-firestore:8.0.2'  
  
    // FirebaseUI for Firebase Auth  
    implementation 'com.firebaseui:firebase-ui-auth:8.0.2'  
  
    // FirebaseUI for Cloud Storage  
    implementation 'com.firebaseui:firebase-ui-storage:8.0.2'  
}
```



Seleccionamos las dependencias de los servicios que vamos a utilizar.

Una vez que copiamos las dependencias que hay que instalar, el fichero build.gradle modulo debe quedar de la siguiente forma:

```
// FirebaseUI for Firebase Realtime Database  
implementation 'com.firebaseui:firebase-ui-database:8.0.2'  
  
implementation 'com.google.firebase:firebase-storage-ktx:20.1.0'  
  
// FirebaseUI for Cloud Firestore  
//implementation 'com.firebaseui:firebase-ui-firestore:8.0.2'  
  
// FirebaseUI for Firebase Auth  
implementation 'com.firebaseui:firebase-ui-auth:8.0.2'
```

La dependencia del storage es muy específica y lo correcto es coger la dependencia que nos muestra en la siguiente página: <https://firebase.google.com/docs/storage/android/start>

En el **archivo Gradle de tu módulo (nivel de aplicación)** (generalmente `<project>/<app-module>/build.gradle`), agrega la dependencia para la biblioteca de Android de Cloud Storage. Recomendamos usar [Firebase Android BoM](#) para controlar el control de versiones de la biblioteca.



El fichero quedaría de la siguiente forma:

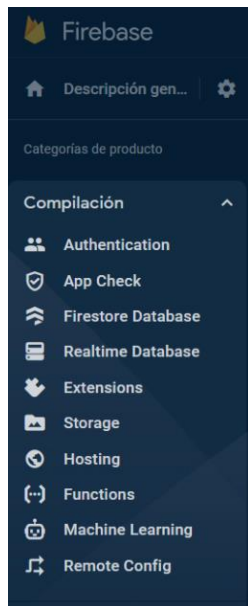
```
// FirebaseUI for Firebase Realtime Database  
implementation 'com.firebaseui:firebase-ui-database:8.0.2'  
  
// Import the BoM for the Firebase platform  
implementation platform('com.google.firebase:firebase-bom:31.1.0')  
  
// Add the dependency for the Cloud Storage library  
// When using the BoM, you don't specify versions in Firebase library dependencies  
implementation 'com.google.firebase:firebase-storage-ktx'  
  
// FirebaseUI for Firebase Auth  
implementation 'com.firebaseui:firebase-ui-auth:8.0.2'
```

Con este cambio nos aseguramos de que todas las versiones están sincronizadas, solo nos tendremos de comprobar la versión de firebase-bom para tener todo actualizado.

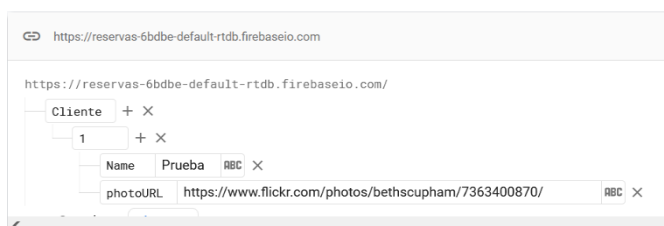
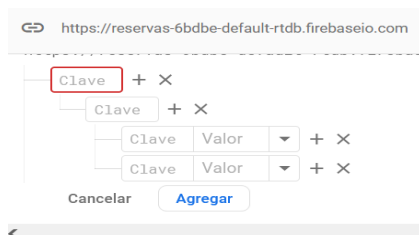
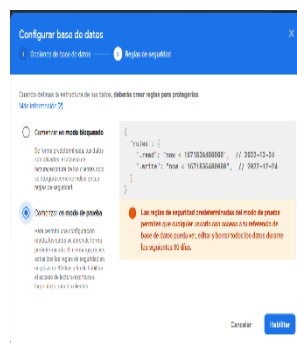
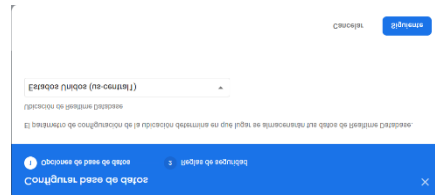
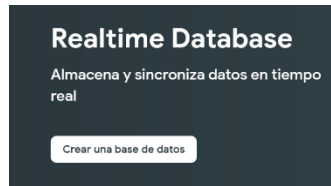
Para terminar el proceso deberemos de instalar las dependencias que necesitemos para realizar la APP.

Configurar Firebase

Para configurar los servicios de Firebase hay que seleccionar el proyecto y acceder a la opción de compilación, se desplegarán los servicios para que podamos indicar el servicio a configurar.



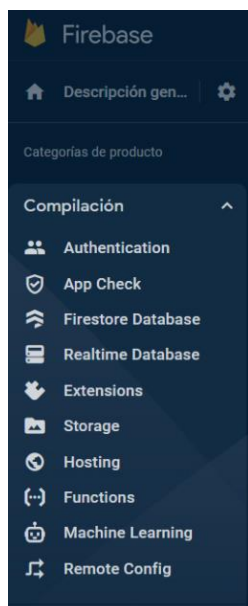
Configurar Realtime Database



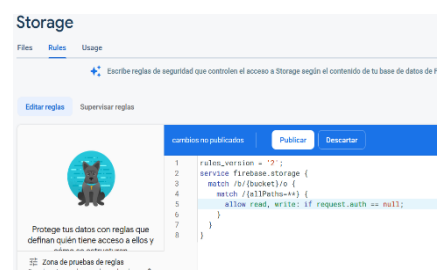
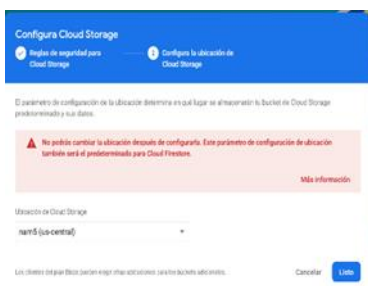
Código Kotlin para acceder a la base de datos.

FirebaseDatabase.getInstance().reference.child("cliente")

Cliente es el nodo de la base de datos.

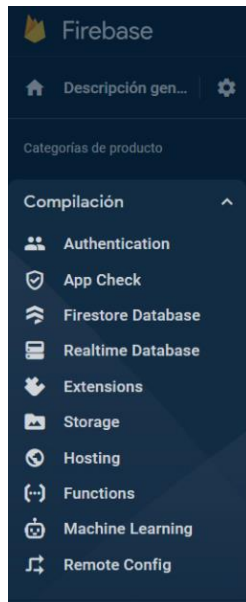


Configurar el Storage



Código kotlin para acceder a Storage

```
mStorageReference = FirebaseStorage.getInstance().reference  
val storageReference = mStorageReference.child("clientes").child("my_photo")
```



Authentication

