MINISTERIO DE TECNOLOGÍAS DE LA INFORMACIÓN Y LAS COMUNICACIONES

Unidad 5

Firebase

¿Qué es Firebase?

Se trata de una plataforma móvil creada por Google, cuya principal función es desarrollar y facilitar la creación de apps de elevada calidad de una forma rápida, con el fin de que se pueda aumentar la base de usuarios y ganar más dinero.

La plataforma está subida en la nube y está disponible para diferentes plataformas como iOS, Android y web.

Contiene diversas funciones para que cualquier desarrollador pueda combinar y adaptar la plataforma a medida de sus necesidades.

Principales características de Firebase

- **Desarrollo**: Firebase permite la creación de mejores apps, minimizando el tiempo de optimización y desarrollo, mediante diferentes funciones, entre las que destacan la detección de errores y de testeo, que supone poder dar un salto de calidad a la app.
- **Analítica**: Tener un control máximo del rendimiento de la app mediante métricas analíticas, todo desde un único panel y de forma gratuita, es una de las ventajas que ofrece Firebase respecto a la analítica web. Facilita la toma de decisiones basadas y fundamentadas en datos reales.
- **Poder de crecimiento**: Permite gestionar de manera fácil todos los usuarios de las aplicaciones, con el añadido de que se pueden captar nuevos usuarios, mediante invitaciones o notificaciones.
- **Rapidez**: Implementar Firebase puede ser fácil y rápido, gracias a su API que es muy intuitiva, sostenida en un solo SDK.
- **Agilidad**: Firebase ofrece apps multiplataforma con una APIs integradas a SDK individuales para iOS, Android y Javascript, de tal forma que se puede gestionar diferentes apps sin necesidad de salir de la propia plataforma.

Productos

Compilación

Acelera el desarrollo de apps con una infraestructura de backend completamente administrada

Ver todos los productos de compilación

- Cloud Firestore
- Authentication
- Extensions

Lanzamiento y supervisión

Realiza lanzamientos con confianza y supervisa el rendimiento y la estabilidad

Ver todos los productos de supervisión y lanzamiento

- Crashlytics
- Performance Monitoring
- Google Analytics

Interactúa

Aumenta la participación de los usuarios con estadísticas enriquecidas, pruebas A/B y campañas de mensajes

Ver todos los productos de participación

- Remote Config
- Cloud Messaging
- A/B testing

Compilación

Crea apps potentes. Inicia tu backend sin administrar servidores. Escala sin interrupciones para brindar asistencia a millones de usuarios con las bases de datos de Firebase, la infraestructura de aprendizaje automático, las soluciones de hosting y almacenamiento y Cloud Functions.

Lanzamiento y supervisión

Mejora la calidad de tus apps en menos tiempo y con menos esfuerzo. Simplifica las pruebas, la clasificación y la solución de problemas. Lanza cuidadosamente las funciones y supervisa su adopción. Identifica, prioriza y corrige los problemas de estabilidad y rendimiento desde el principio.

Participación

Aumenta la participación de los usuarios con estadísticas enriquecidas, pruebas A/B y campañas de mensajes. Comprende a los usuarios para mejorar la asistencia y retenerlos. Ejecuta experimentos para probar ideas y descubrir estadísticas nuevas. Personaliza tu app para diferentes segmentos de usuarios.

Precios

La plataforma dispone de 2 planes de precio, diversificados por las necesidades que tenga el usuario.

El plan "**Spark**" es gratuito, pero presenta varias limitaciones, sobre todo en el espacio de almacenamiento y las conexiones simultáneas, que puede provocar que en muchos casos no sea suficiente para un desarrollador.

El Plan "**Blaze**" su precio varía en función del consumo que se haga de la plataforma.

Productos	Sin costo Plan Spark Límites amplios para comenzar	Prepago Plan Blaze Calcula los precios de las apps a gran escala ✓ Se incluye el uso sin costo del plan Spark*
A/B Testing		Sin costo
Analytics		Sin costo
App Distribution		Sin costo
App Indexing		Sin costo
Autenticación		
Autenticación telefónica: EE.UU., India y Canadá	10,000 por mes	$0.01 por verificación
Autenticación telefónica: Todos los demás países	10,000 por mes	$0.06 por verificación
Otros servicios de autenticación	✓	✓
Con Identity Platform		
Usuarios activos por mes	50,000 por mes	Sin costo hasta 50,000 MA Luego, se aplican los precios de Goc
Usuarios activos por mes: SAML/OIDC	50 por mes	Sin costo hasta 50 MAU Luego, se aplican los precios de Goc

https://firebase.google.com/pricing?hl=es-419

MINISTERIO DE TECNOLOGÍAS DE LA INFORMACIÓN Y LAS COMUNICACIONES

Firebase en aplicación Flutter

Paso 1 : Instalar herramientas de línea de comando

1. Si aún no lo ha hecho, instale Firebase CLI .
2. Inicie sesión en Firebase usando su cuenta de Google ejecutando el siguiente comando:

   ```
   firebase login
   ```

3. Instale FlutterFire CLI ejecutando el siguiente comando desde cualquier directorio:

   ```
   dart pub global activate flutterfire_cli
   ```

Paso 2 : Configurar aplicación para usar Firebase

Use la CLI de FlutterFire para configurar sus aplicaciones de Flutter para conectarse a Firebase.

Desde el directorio de su proyecto Flutter, ejecute el siguiente comando para iniciar el flujo de trabajo de configuración de la aplicación:

```
flutterfire configure
```

Paso 3 : Inicializar Firebase en su aplicación

1. Desde el directorio de su proyecto Flutter, ejecute el siguiente comando para instalar el complemento principal:

```
flutter pub add firebase_core
```

2. En su archivo lib/main.dart , importe el complemento principal de Firebase y el archivo de configuración que generó anteriormente:

```
import 'package:firebase_core/firebase_core.dart';
import 'firebase_options.dart';
```

3. También en su archivo lib/main.dart , inicialice Firebase usando el objeto DefaultFirebaseOptions exportado por el archivo de configuración:

```
await Firebase.initializeApp(
    options: DefaultFirebaseOptions.currentPlatform,
);
```

Paso 4 : Agregar complementos de Firebase

1. Desde el directorio de tu proyecto Flutter, ejecuta el siguiente comando:

   ```
   flutter pub add PLUGIN_NAME
   ```

2. Desde el directorio de tu proyecto Flutter, ejecuta el siguiente comando:

   ```
   flutterfire configure
   ```

3. Una vez completado, reconstruya su proyecto Flutter:

   ```
   flutter run
   ```

Complementos

Accede a Firebase en su aplicación Flutter a través de los diversos complementos de Firebase Flutter, uno para cada producto de Firebase (por ejemplo: Cloud Firestore, Authentication, Analytics, etc.).

Dado que Flutter es un marco multiplataforma, cada complemento de Firebase es aplicable para Apple, Android y plataformas web.

Por lo tanto, si agrega cualquier complemento de Firebase a su aplicación Flutter, las versiones de Apple, Android y web de su aplicación lo utilizarán.

Producto	Nombre del complemento	iOS	Androide	Web	Otra manzana (macOS, etc)
Analítica	`firebase_analytics`	✓	✓	✓	beta
Comprobación de la aplicación	`firebase_app_check`	✓	✓	✓	beta
Autenticación	`firebase_auth`	✓	✓	✓	beta
Tienda de fuego en la nube	`cloud_firestore`	✓	✓	✓	beta
Funciones en la nube	`cloud_functions`	✓	✓	✓	beta
Mensajería en la nube	`firebase_messaging`	✓	✓	✓	beta
Almacenamiento en la nube	`firebase_storage`	✓	✓	✓	beta
Crashlytics	`firebase_crashlytics`	✓	✓		beta
Enlaces dinámicos	`firebase_dynamic_links`	✓	✓		
Mensajería en la aplicación	`firebase_in_app_messaging`	✓	✓		
Instalaciones de base de fuego	`firebase_app_installations`	✓	✓	✓	beta
Descargador de modelos ML	`firebase_ml_model_downloader`	✓	✓		beta
Supervisión del rendimiento	`firebase_performance`	✓	✓	✓	
Base de datos en tiempo real	`firebase_database`	✓	✓	✓	beta
Configuración remota	`firebase_remote_config`	✓	✓	✓	beta

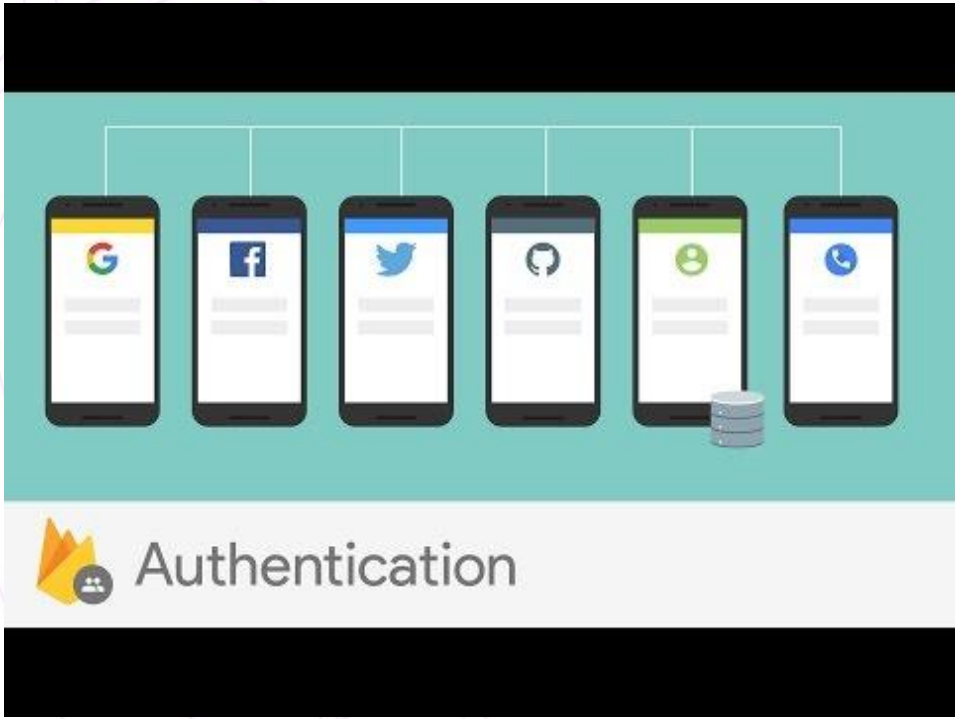
https://firebase.google.com/docs/flutter/setup?hl=es-419&platform=android#available-plugins

Firebase Authentication

Authentication

La mayoría de las aplicaciones necesitan conocer la identidad de un usuario. Conocer la identidad de un usuario permite que una aplicación guarde de forma segura los datos del usuario en la nube y brinde la misma experiencia personalizada en todos los dispositivos del usuario.

Firebase Authentication proporciona servicios de backend, SDK fáciles de usar y bibliotecas de interfaz de usuario listas para usar para autenticar a los usuarios en su aplicación. Admite la autenticación mediante contraseñas, números de teléfono, proveedores de identidad federados populares como Google, Facebook y Twitter, y más.



Agrega Firebase Authentication a tu aplicación

1. Desde la raíz de su proyecto Flutter, ejecute el siguiente comando para instalar el complemento:

   ```
   flutter pub add firebase_auth
   ```

2. Una vez completada, reconstruya su aplicación Flutter:

   ```
   flutter run
   ```

3. Importe el complemento en su código Dart:

   ```
   import 'package:firebase_auth/firebase_auth.dart';
   ```

https://firebase.google.com/docs/auth/flutter/start?hl=es-419

Obtener un perfil de usuario

Para obtener la información del perfil de un usuario, utilice las propiedades de User .

Hay tres formas de obtener un objeto User que represente al usuario actual:

- Las `authStateChanges` , `idTokenChanges` y `userChanges` : sus oyentes recibirán el User actual, o null si no se autentica ningún usuario
- El objeto `UserCredential` devuelto por los métodos de autenticación (`signIn` -): el objeto `UserCredential` tiene una propiedad de `user` con el `User` actual
- La propiedad `currentUser` de la instancia de `FirebaseAuth` : si está seguro de que el usuario ha iniciado sesión actualmente, puede acceder al `User` desde la propiedad `currentUser`

```
FirebaseAuth.instance
  .authStateChanges()
  .listen((User? user) {
    if (user != null) {
      print(user.uid);
    }
  });
```

```
final userCredential = await
  FirebaseAuth.instance.signInWithCredential(credential);
final user = userCredential.user;
print(user?.uid);
```

```
if (FirebaseAuth.instance.currentUser != null) {
  print(FirebaseAuth.instance.currentUser?.uid);
}
```

Autenticarse usando cuentas con contraseña

Habilitar inicio de sesión con correo electrónico/contraseña:

- En la sección **Authentication** de la consola Firebase, abra la página **Método de inicio de sesión**.
- En la página Método de inicio de sesión , habilite el método de inicio de sesión con **correo electrónico/contraseña** y haga clic en **Guardar**.

Crear una cuenta basada en contraseña

```
try {
  final credential =
    await FirebaseAuth.instance.createUserWithEmailAndPassword(
      email: emailAddress,
      password: password,
    );
} on FirebaseAuthException catch (e) {
  if (e.code == 'weak-password') {
    print('The password provided is too weak.');
  } else if (e.code == 'email-already-in-use') {
    print('The account already exists for that email.');
  }
} catch (e) {
  print(e);
}
```

https://firebase.google.com/docs/auth/flutter/password-auth?hl=es-419

Autenticarse usando cuentas con contraseña

Inicie sesión

```
try {
  final credential =
    await FirebaseAuth.instance
      .signInWithEmailAndPassword(
        email: emailAddress,
        password: password
      );
} on FirebaseAuthException catch (e) {
  if (e.code == 'user-not-found') {
    print('No user found for that email.');
  } else if (e.code == 'wrong-password') {
    print('Wrong password provided for that user.');
  }
}
```

Cerrar sesión

```
await FirebaseAuth.instance.signOut();
```

https://firebase.google.com/docs/auth/flutter/password-auth?hl=es-419

Identidad federada e inicio de sesión social

Google

Instale el complemento oficial de `google_sign_in`.

```
import 'package:google_sign_in/google_sign_in.dart';

Future<UserCredential> signInWithGoogle() async {
  // Trigger the authentication flow
  var googleUser = await GoogleSignIn().signIn();

  // Obtain the auth details from the request
  var googleAuth = await googleUser?.authentication;

  // Create a new credential
  var credential = GoogleAuthProvider.credential(
    accessToken: googleAuth?.accessToken,
    idToken: googleAuth?.idToken,
  );

  // Once signed in, return the UserCredential
  return
    await FirebaseAuth.instance.signInWithCredential(credential);
}
```

Facebook

Instale el complemento oficial de `flutter_facebook_auth`.

```
import 'package:google_sign_in/google_sign_in.dart';

import 'package:flutter_facebook_auth/flutter_facebook_auth.dart';

Future<UserCredential> signInWithFacebook() async {
  // Trigger the sign-in flow
  var loginResult = await FacebookAuth.instance.login();

  // Create a credential from the access token
  var facebookAuthCredential =
FacebookAuthProvider.credential(loginResult.accessToken.token);

  // Once signed in, return the UserCredential
  return
FirebaseAuth.instance.signInWithCredential(facebookAuthCredential);
}
```

https://firebase.google.com/docs/auth/flutter/federated-auth?hl=es-419

Firebase Firestore

Firestore

Cloud Firestore es una base de datos flexible y escalable para el desarrollo móvil, web y de servidor de Firebase y Google Cloud.

Al igual que Firebase Realtime Database, mantiene sus datos sincronizados entre las aplicaciones de los clientes a través de escuchas en tiempo real y ofrece soporte sin conexión para dispositivos móviles y web para que pueda crear aplicaciones receptivas que funcionen independientemente de la latencia de la red o la conectividad a Internet.

Cloud Firestore también ofrece una integración perfecta con otros productos de Firebase y Google Cloud, incluidas Cloud Functions.

Modelo de datos de Cloud Firestore

Cloud Firestore es una base de datos NoSQL orientada a documentos. A diferencia de una base de datos SQL, no hay tablas ni filas. En su lugar, almacena datos en **documentos** , que están organizados en **colecciones** .

Cada documento contiene un conjunto de pares **clave-valor**.

Los documentos pueden contener subcolecciones y objetos anidados, los cuales pueden incluir campos primitivos como cadenas u objetos complejos como listas.

Las colecciones y los documentos se crean implícitamente en Cloud Firestore. Simplemente asigne datos a un documento dentro de una colección. Si la colección o el documento no existe, Cloud Firestore lo crea.

https://firebase.google.com/docs/firestore/data-model?hl=es-419

Crear una base de datos de Cloud Firestore

- Vaya a la sección Cloud Firestore de Firebase console. Siga el flujo de trabajo de creación de la base de datos.
- Seleccione un modo de inicio para sus reglas de seguridad de Cloud Firestore:
  - Modo de prueba
  - Modo bloqueado
- Desde la raíz de su proyecto Flutter, ejecute el siguiente comando para instalar el complemento.

```
flutter pub add cloud_firestore
```

- Inicializa una instancia de Cloud Firestore

```
db = FirebaseFirestore.instance;
```

Proteja sus datos

```
// Uso a usuarios logueados
service cloud.firestore {
  match /databases/{database}/documents {
    match /{document=**} {
      allow read, write: if request.auth != null;
    }
  }
}
```

```
// Modo de prueba, NO PARA PRODUCCIÓN
service cloud.firestore {
  match /databases/{database}/documents {
    match /{document=**} {
      allow read, write: if true;
    }
  }
}
```

Agregar datos a Cloud Firestore

Hay varias formas de escribir datos en Cloud Firestore:

- Establecer los datos de un documento dentro de una colección, especificando explícitamente un identificador de documento.
- Agregar un nuevo documento a una colección. En este caso, Cloud Firestore genera automáticamente el identificador del documento.
- Cree un documento vacío con un identificador generado automáticamente y asígnele datos más tarde.

```
final city = <String, String>{
  "name": "Los Angeles",
  "state": "CA",
  "country": "USA"
};

db.collection("cities")
  .doc("LA")
  .set(city)
  .onError((e, _) =>
     print("Error writing document: $e"));
```

https://firebase.google.com/docs/firestore/manage-data/add-data?hl=es-419#dart

Agregar datos - Objetos personalizados

```
class City {
  final String? name;
  final String? state;
  final String? country;
  final bool? capital;
  final int? population;
  final List<String>? regions;

  City({
    this.name,
    this.state,
    this.country,
    this.capital,
    this.population,
    this.regions,
  });
  . . .
```

```
. . .
factory City.fromFirestore(
  DocumentSnapshot<Map<String, dynamic>> snapshot,
  SnapshotOptions? options) {
  final data = snapshot.data();
  return City(
    name: data?['name'],
    state: data?['state'],
    country: data?['country'],
    capital: data?['capital'],
    population: data?['population'],
    regions:  data?['regions'] is Iterable
      ? List.from(data?['regions']) : null,
  );
}
. . .
```

Agregar datos - Objetos personalizados

```
  . . .
  Map<String, dynamic> toFirestore() {
    return {
      if (name != null) "name": name,
      if (state != null) "state": state,
      if (country != null) "country": country,
      if (capital != null) "capital": capital,
      if (population != null) "population":
                                    population,
      if (regions != null) "regions": regions,
    };
  }
}
```

```
final city = City(
  name: "Los Angeles",
  state: "CA",
  country: "USA",
  capital: false,
  population: 5000000,
  regions: ["west_coast", "socal"],
);
final docRef = db.collection("cities")
    .withConverter(
      fromFirestore: City.fromFirestore,
      toFirestore: (City city, options)
                        => city.toFirestore(),
    )
    .doc("LA");
await docRef.set(city);
```

Obtener datos de Cloud Firestore

Hay tres formas de recuperar datos almacenados en Cloud Firestore.

Cualquiera de estos métodos se puede utilizar con documentos, colecciones de documentos o los resultados de consultas:

- Llame a un método para obtener los datos una vez.
- Configure un detector para recibir eventos de cambio de datos.
- Carga masiva de datos de instantáneas de Firestore desde una fuente externa a través de paquetes de datos.

```
db.collection("cities")
  .doc("SF");
  .get()
  .then((doc) {
    final data = doc.data() as Map<String,
dynamic>;
    // ...
  },
  onError: (e) => print("Error getting document:
$e"),
);

db.collection("cities")
  .where("capital", isEqualTo: true) // Filtro
  .get()
  .then((res) => print("Successfully completed"),
    onError: (e) => print("Error completing: $e"),
  );
```

https://firebase.google.com/docs/firestore/query-data/get-data?hl=es-419