

Firestore Basics

📅 Date @April 25, 2023 → May 2, 2023

Que es Firebase?

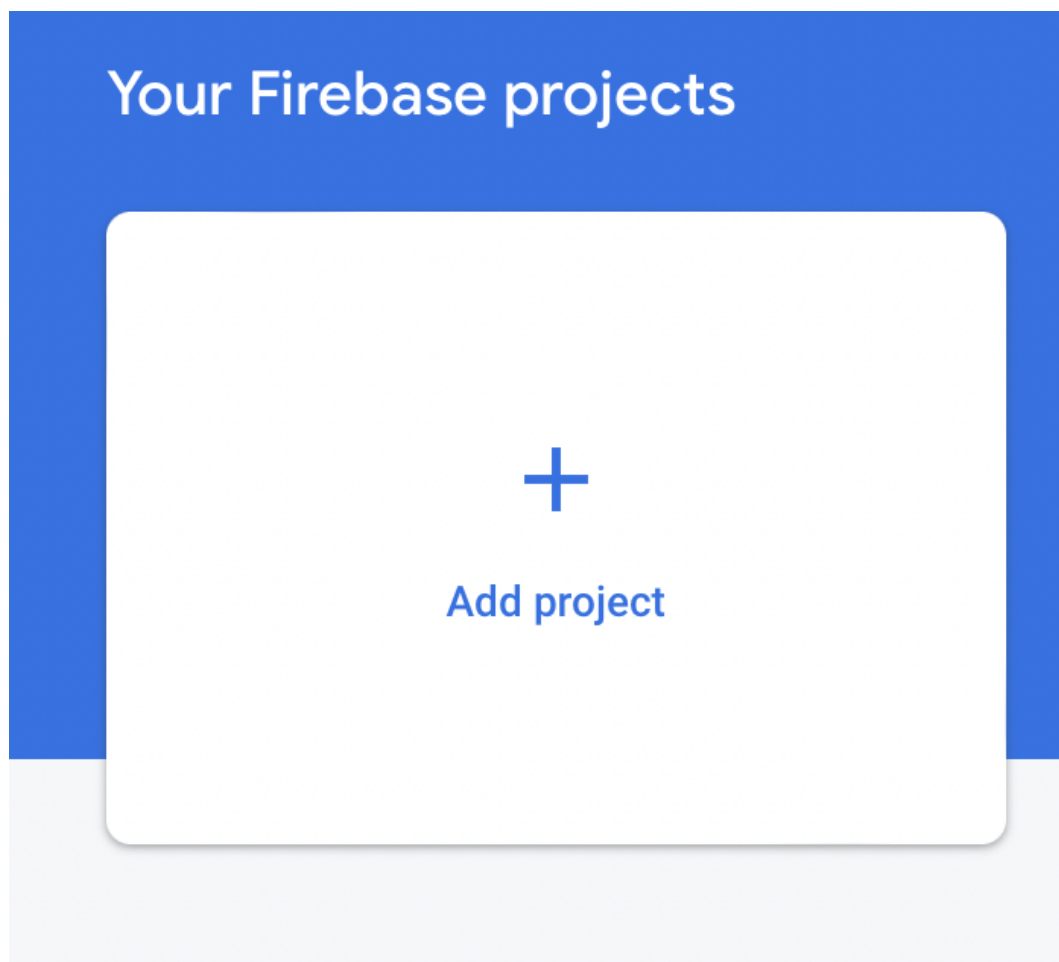
Un grupo de herramientas de backend, propiedad de Google que nos brindan servicios de bases de datos, authentication, hosting y mas que podemos integrar con multiples lenguajes de programación como IOS, Android, Node, Javascript.

Como lo usamos?

Firebase funciona a base de proyectos, para crear uno debemos tener una cuenta google en firebase e ingresamos a la pagina de Firebase Console (<https://console.firebase.google.com>)

Crear nuevo proyecto.


- Click en crear nuevo proyecto.




- Debemos configurar nuestro proyecto con un nombre y decidir si queremos tener google analytics (depende de uds, yo lo dejo activado).
- Despues le damos una cuenta para los google analytics, seleccionamos la default.

Configure Google Analytics

Choose or create a Google Analytics account ?

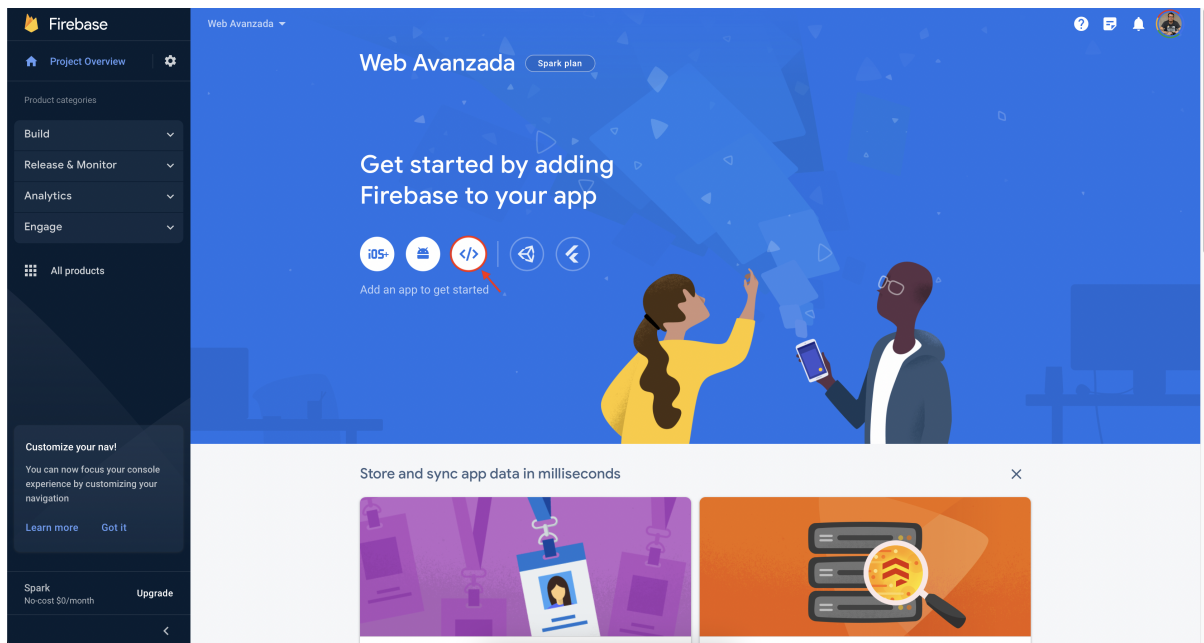
 Default Account for Firebase ▼

Automatically create a new property in this account ✎

Upon project creation, a new Google Analytics property will be created in your chosen Google Analytics account and linked to your Firebase project. This link will enable data flow between the products. Data exported from your Google Analytics property into Firebase is subject to the Firebase terms of service, while Firebase data imported into Google Analytics is subject to the Google Analytics terms of service. [Learn more](#) .

[Previous](#) [Create project](#)

- Una vez terminado el proceso de creación de nuestro proyecto nos muestra la consola de la app, aqui podemos encontrar todos los servicios que tenemos disponibles para usar en nuestro proyecto de react.
- Para empezar debemos dar click en el boton de web para empezar el registro de nuestra app y el setup en react.



- Registramos un nombre para la app y le damos click a setup firebase hosting, esto nos servira mas adelante para poder tener nuestra app live, damos click en register app.

1

Register app

App nickname ?

Web Avanzada

☒

Also set up **Firestore** for this app. [Learn more](#)

Hosting can also be set up later. There is no cost to get started anytime.

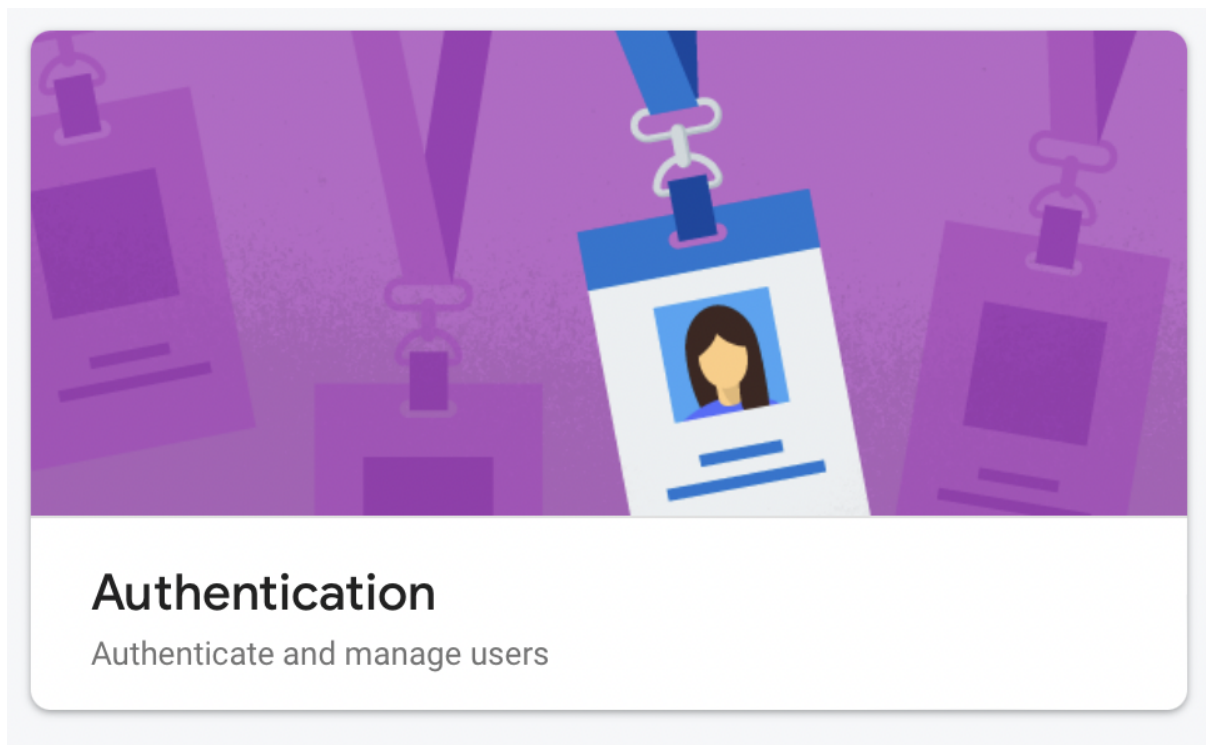
web-avanzada-24977 (No deploys yet)

Register app

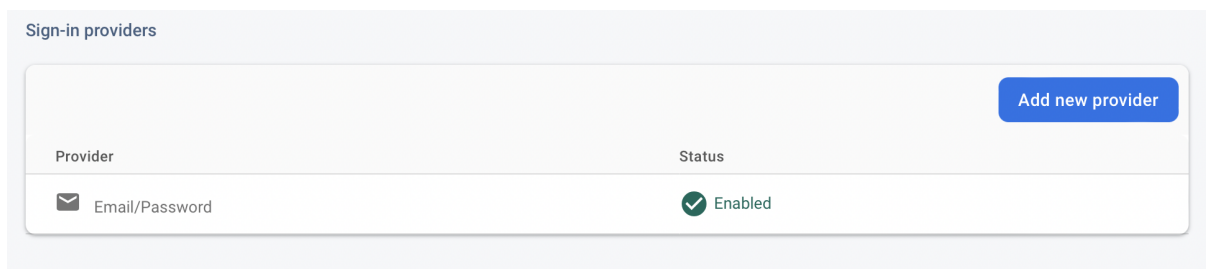
- una vez tenemos registrada la app tenemos una pantalla que nos indica correr `npm install firebase` para instalar el SDK y nos muestra el código que necesitamos para inicializarla en nuestra app, esto es único para cada proyecto, copiándolo y agregándolo un archivo en `src/config/firebase.js`
 - Esta información es privada, deberíamos tener los keys en un `.env` pero por ahora usemos esto así.
- Como más adelante vamos a hostear nuestro proyecto en Firebase necesitamos el CLI así que debemos correr `npm install -g firebase-tools` para instalarlo de forma global en nuestro computador.
 - ignoremos el último paso **"Deploy to Firebase Hosting"** pues todavía no vamos a hacer hosting.

Sign In.

- En la página de la consola vamos a usar la opción de authentication y damos click en Get started.



- Elegimos el provider que queremos, en nuestro caso Email y password, y damos click en enable.



- En nuestra app de react, en el archivo de configuracion de firebase tenemos que importar `getAuth` de `firebase/auth` y lo inicializamos usando `export const auth = getAuth(app)`, esta constante la exportamos debido a que la vamos a usar en toda la app.
- en nuestro componente de sign in debemos usar el siguiente codigo

```
import { auth } from "../../config/firebase";
import { createUserWithEmailAndPassword } from "firebase/auth";
import { useState } from "react";

function SignIn() {
  const [email, setEmail] = useState("");
  const [password, setPassword] = useState("");

  const handleSignIn = async () => {
    console.log("handleSignIn");
    try {
      await createUserWithEmailAndPassword(auth, email, password);
    } catch (error) {
      console.error(error);
    }
  }
  return (
    <>
      <input
        placeholder='email@email.com'
        onChange={(e) => setEmail(e.target.value)}
      />
      <input
        placeholder='Password'
        type='password'
        onChange={(e) => setPassword(e.target.value)}
      />
      <button onClick={handleSignIn}>Sign In</button>
    </>
  )
}

export default SignIn;
```

Sign Out

- para hacer logout de nuestra app es mucho mas sencillo, solo debemos importar el metodo `signOut` y usarlo, entregandole el objeto auth que hemos creado al inicializar nuestro authentication.

Log In

- en nuestro componente de Log in debemos usar el siguiente codigo

```
import { useState } from "react";
import { auth } from "../../config/firebase";
import { signInWithEmailAndPassword } from "firebase/auth";

function LogIn() {
  const [email, setEmail] = useState("");
  const [password, setPassword] = useState("");

  const handleLogIn = async () => {
    console.log("handleLogIn");
```

```

    try {
      await signInWithEmailAndPassword(auth, email, password);
    } catch (error) {
      console.error(error);
    }
  }
}
return (
  <>
    <h3>Already have an account?</h3>
    <input
      placeholder='email@email.com'
      onChange={(e) => setEmail(e.target.value)}
    />
    <input
      placeholder='Password'
      type='password'
      onChange={(e) => setPassword(e.target.value)}
    />
    <button onClick={handleLogIn}>Log In</button>
  </>
)
}

export default LogIn

```

Firestore - DB

- Es una base de datos no relacional, como mongodb... lo contrario a SQL.
 - Guardamos las cosas en documentos y estructuras mas libres que en las db relacionales.
- Para crear una DB vamos a la seccion de firestore en nuestra consola de firebase y damos click en “create database”
- podemos iniciarla en modo produccion o modo test, lo haremos en modo produccion para poder ver las reglas de la base de datos, damos click en “next”.

Create database [X]

1 Secure rules for Cloud Firestore — 2 Set Cloud Firestore location

After you define your data structure, **you will need to write rules to secure your data.**
[Learn more](#)

☒ **Start in production mode**
 Your data is private by default. Client read/write access will only be granted as specified by your security rules.

☐ **Start in test mode**
 Your data is open by default to enable quick setup. However, you must update your security rules within 30 days to enable long-term client read/write access.

```
rules_version = '2';
service cloud.firestore {
  match /databases/{database}/documents {
    match /{document=**} {
      allow read, write: if false;
    }
  }
}
```

i All third party reads and writes will be denied

Enabling Cloud Firestore will prevent you from using Cloud Datastore with this project

Cancel **Next**

- Seleccionamos la region mas cercana a nosotros, en nuestro caso USA y damos click en enable.

Create database [X]

✓ Secure rules for Cloud Firestore — 2 Set Cloud Firestore location

Your location setting is where your Cloud Firestore data will be stored.

⚠ After you set this location, you cannot change it later. Also, this location setting will be the location for your default Cloud Storage bucket.
[Learn more](#)

Cloud Firestore location

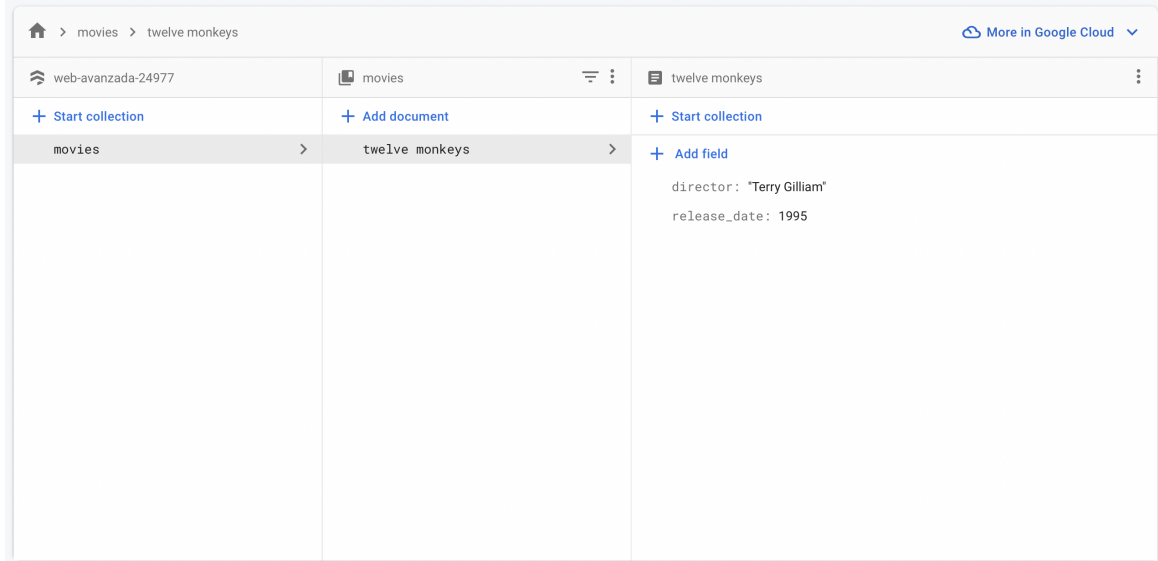
nam5 (United States) ▼

Enabling Cloud Firestore will prevent you from using Cloud Datastore with this project

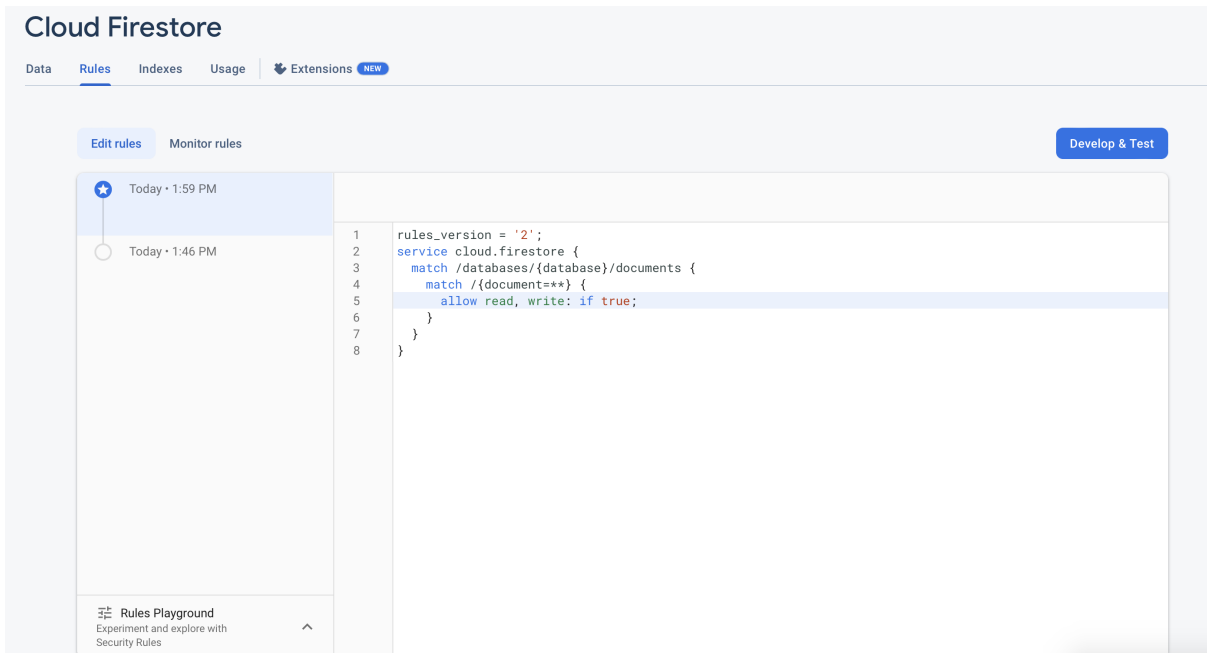
Cancel **Enable**

- damos click en start collection (similar a una tabla en SQL), cada una representa una entidad de datos... usuarios, libros, peliculas y asi.

- nos pide que ingresemos un documento (equivalente a un row en una tabla), podemos ingresar cualquier cosa que represente lo que vamos a tener en el documento.
 - debe existir minimo un documento.



- vamos a reglas y por ahora solo diremos que todo el mundo puede acceder a la db, despues vamos a cambiar esto.



- En el componente que vamos a usar para visualizar nuestros datos usamos el siguiente codigo.

```
const moviesCollectionRef = collection(db, "movies"); // hacemos una referencia a la coleccion que necesitamos

const getMovies = async () => {
```



```
try {
  const data = await getDocs(moviesCollectionRef);
  const formattedData = data.docs.map((doc) => ({
    ...doc.data(), //metodo incluido para obtener los datos
    id: doc.id // tenemos que sacar el id del documento, en nuestro caso tambien es el nombre o titulo de la pelicula.
  }));
  setMovies(formattedData); // seteamos las peliculas en nuestro estado
} catch (error) {
  console.error(error);
}
}
```