

# Configuración de Firestore Module



## Configuración de Firestore Module en Angular

En esta lección, aprenderemos cómo configurar el módulo de **Firestore** en nuestro proyecto de Angular. Es importante recordar que los valores deben ser los de tu propia aplicación de **Firestore**, ya que los datos de este proyecto se eliminarán al finalizar el mismo.

### Configuración del Proyecto en Firebase

The screenshot shows the Firebase Project Configuration interface. The left sidebar lists various services: Descripción general, IA generativa, Build with Gemini (NUEVO), Accesos directos a proyectos, and Firestore Database (highlighted by a yellow box labeled 2). The main navigation bar has tabs for control-clientes, Descripción general, Configuración del proyecto (highlighted by a yellow box labeled 1), Usuarios y permisos, Uso y facturación, and Extensiones. The 'Configuración del proyecto' tab is active, showing a warning about protecting resources from abuse. Below this, the 'Firestore Database' section is expanded, showing the database structure: a collection 'clientes' with document '1', and a subcollection 'configuracion' with a field 'permitirRegistro: false'. There are also buttons for 'Iniciar colección' and 'Agregar documento'.

Baja en la pantalla de configuración de tu proyecto en **Firebase** y copia la configuración de tu proyecto.

The screenshot shows the 'Configuración del SDK' section in the Firebase Project Configuration. It provides instructions for installing the Firebase SDK using npm or CDN. It includes a command line: '\$ npm install firebase'. Below it, there's sample code for initializing the Firebase web app using the 'Config' option:

```
// Import the functions you need from the SDKs you need
import { initializeApp } from "firebase/app";
// TODO: Add SDKs for Firebase products that you want to use
// https://firebase.google.com/docs/web/setup#available-libraries

// Your web app's Firebase configuration
const firebaseConfig = {
  apiKey: "AIzaSyBLmNX6efPWPswa0cIpnnZx9Qcyjp_gqI4",
  authDomain: "control-clientes-e627b.firebaseioapp.com",
  projectId: "control-clientes-e627b",
  storageBucket: "control-clientes-e627b.appspot.com",
  messagingSenderId: "820310380226",
  appId: "1:820310380226:web:9dd15c12f760276252755b"
};

// Initialize Firebase
const app = initializeApp(firebaseConfig);
```

Below the code, there's a note: 'Nota: Esta opción utiliza el SDK de JavaScript modular, que proporciona un tamaño reducido del SDK.' and a link: 'Obtén más información sobre Firebase para la Web: [primeros pasos](#), [referencia de la API del SDK web](#) y [muestras](#)'.

Ejecutamos el siguiente comando para que podamos interactuar con firebase:

```
npm install firebase @angular/fire --save
```

En nuestro proyecto Angular, con las últimas versiones, **no se crea por defecto la carpeta de environments** con sus archivos correspondientes. Para crearlos, necesitamos ejecutar el siguiente comando:

```
ng g environments
```

Configuración del Archivo `environment.development.ts`:

```
export const environment = {
  production: false,
  firebaseConfig: {
    apiKey: "AIzaSyBLmNX6efPWPswa0cIpmpNZx9Qcyjp_gqI4",
    authDomain: "control-clientes-e627b.firebaseio.com",
    projectId: "control-clientes-e627b",
    storageBucket: "control-clientes-e627b.appspot.com",
    messagingSenderId: "820310380226",
    appId: "1:820310380226:web:9dd15c12f760276252755b"
  }
};
```

Este archivo corresponde al entorno de desarrollo. Aquí debes reemplazar los valores de `firebaseConfig` por los de tu propio proyecto de Firebase.

Configuración del Archivo `environment.ts`:

```
export const environment = {
  production: true,
  firebaseConfig: {
    apiKey: "AIzaSyBLmNX6efPWPswa0cIpmpNZx9Qcyjp_gqI4",
    authDomain: "control-clientes-e627b.firebaseio.com",
    projectId: "control-clientes-e627b",
    storageBucket: "control-clientes-e627b.appspot.com",
    messagingSenderId: "820310380226",
    appId: "1:820310380226:web:9dd15c12f760276252755b"
  }
};
```

Este archivo corresponde al entorno de producción. Asegúrate de mantener las configuraciones adecuadas para este entorno.

## Modificación del Archivo `main.ts`

Dado que estamos utilizando **standalone components** en nuestro proyecto de Angular, debemos modificar el archivo `main.ts` para agregar la configuración de **Firestore** y conectarnos a la base de datos. También aprovechamos para agregar el proveedor de **HTTP** que usaremos más adelante para realizar peticiones a la base de **Cloud Firestore**.

### Código del archivo `main.ts`:

```
import { bootstrapApplication } from '@angular/platform-browser';
import { provideHttpClient } from '@angular/common/http';
import { appConfig } from './app/app.config';
import { AppComponent } from './app/app.component';
import { environment } from './environments/environment';

// Firebase imports para aplicaciones standalone
import { provideFirebaseApp, initializeApp } from '@angular/fire/app';
import { provideFirestore, getFirestore } from '@angular/fire/firestore';
import { provideAuth, getAuth } from '@angular/fire/auth';
import { provideStorage, getStorage } from '@angular/fire/storage';

bootstrapApplication(AppComponent, {
  ...appConfig,
  providers: [
    provideHttpClient(),
    provideFirebaseApp(() => initializeApp(environment.firebaseioConfig)), // Inicializa Firebase
    provideFirestore(() => getFirestore()), // Inicializa Firestore
    provideAuth(() => getAuth()), // Inicializa Auth
    provideStorage(() => getStorage()), // Inicializa Storage
    ...appConfig.providers // Desestructuramos los providers del appConfig si es necesario
  ]
}).catch((err) => console.error(err));
```

### Explicación:

- `provideFirebaseApp(() => initializeApp(environment.firebaseioConfig))`: Inicializa la configuración de Firebase con los valores del archivo `environment`.
- `provideFirestore(() => getFirestore())`: Proveedor de Firestore que permite la conexión con la base de datos.
- `provideAuth(() => getAuth())`: Proveedor de autenticación para manejar usuarios y sesiones.
- `provideStorage(() => getStorage())`: Proveedor de almacenamiento para trabajar con archivos en Firebase.

- `provideHttpClient()`: Añade el cliente HTTP que usaremos para hacer peticiones a Firebase y otras APIs.

## Resultado Final

Con estos cambios, ya tienes la configuración de **Firestore** y otros servicios de Firebase integrados en tu aplicación Angular. Recuerda que, si encuentras algún problema, puedes descargar el proyecto con la configuración completa hasta esta lección.

¡Saludos!

Ing. Ubaldo Acosta

Fundador de [GlobalMentoring.com.mx](http://www.globalmentoring.com.mx)