

LABORATORIO

IV

2023

Clase 04

Firestore

Instalación

1. Tener Firebase instalado (npm i --g firebase-tools --save y firebase login)
2. ng add @angular/fire
3. Seguir los pasos y no olvidarse de marcar la opción de Firestore Database.

Tenemos dos formas de usarlo. Firestore o Angular Fire.

Firestore

Lo primero es importar:

```
import { Firestore } from '@angular/fire/firestore';
```

E inyectar el servicio en el constructor:

```
constructor(private firestore: Firestore) {}
```

Esta forma utiliza métodos que se importan por separado. Dejo algunos acá:

```
import { addDoc, collection, Firestore, getDoc, getDocs, updateDoc } from  
'@angular/fire/firestore';  
/*Nota: Es el mismo import que hay arriba, el Firestore está ahí en el medio*/
```

Una vez realizado esto, podemos crear funciones como guardar:

```
guardar() {  
  const col = collection(this.firestore, 'usuarios');  
  addDoc(col, { Nombre: 'Fernando', Apellido: 'Fernandez', edad: 30 });  
}
```

Traer (con actualización automática al detectar algún cambio en la colección):

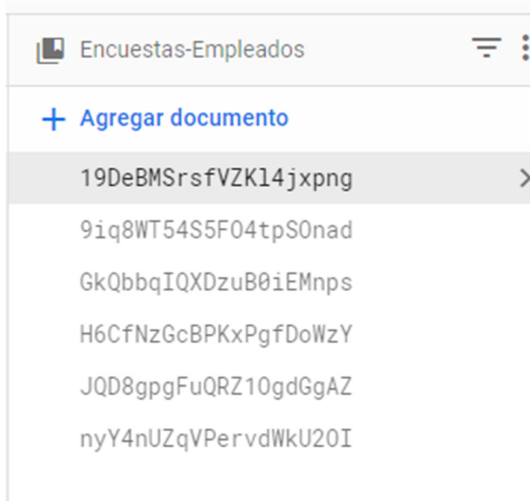
```
traer() {  
  const col = collection(this.firestore, 'usuarios');  
  const observable = collectionData(col);  
  
  observable.subscribe((respuesta) => {  
    console.log(respuesta); // Array de usuarios  
  });  
}  
  
/*recordar que esta función debe ser llamada una única vez,  
ya que el subscribe se va a encargar de traer los datos todas las veces necesarias.  
Como consejo, probar lo que pasa al llamar la función traer muchas veces  
y hacer cambios en la bd*/
```



Update y delete.

```
update(id: string) {  
  const col = collection(this.firestore, 'usuarios');  
  const documento = doc(col, id); //obtenemos referencia del documento  
  updateDoc(documento, {  
    /* Nueva Info. Solo cambia lo que cambiemos */  
  });  
}  
  
delete(id: string) {  
  const col = collection(this.firestore, 'usuarios');  
  const documento = doc(col, id); //obtenemos referencia del documento  
  deleteDoc(documento);  
}
```

El parametro id se refiere a esta id:



Una forma para acceder fácilmente a esto es guardar esa id al momento de crear el documento:

```
guardar() {  
  const documentoNuevo = doc(this.coleccionUsuarios);  
  const nuevoId = documentoNuevo.id;  
  
  setDoc(documentoNuevo, {  
    id: nuevoId,  
    nombre: 'Franco',  
    apellido: 'Lippi',  
    edad: 22,  
  });  
}
```

Nota: Esta es solo una forma, pueden encontrar más.

Aclaración Firestore

Si está esta versión primero en este PDF es solamente porque está mejor documentada. De Angular Fire hay muy poco en comparación. Con ambas maneras pueden hacer lo mismo. La elección depende de ustedes.

Angular Fire

La primera es importar:

```
import { AngularFirestore } from '@angular/fire/compat/firestore';
```

E inyectar el servicio en el constructor:

```
constructor(private firestore: AngularFirestore) {}
```

Una vez realizado esto, podemos crear funciones como guardar:

```
guardar() {  
  const col = this.firestore.collection('usuarios');  
  col.add({ Nombre: 'Fernando', Apellido: 'Fernandez', edad: 30 });  
}
```



Traer (con actualización automática al detectar algún cambio en la colección):

```
traer() {
  const col = this.firestore.collection('usuarios');
  col.valueChanges().subscribe((next: any) => {
    console.log(next);
  });
}
```

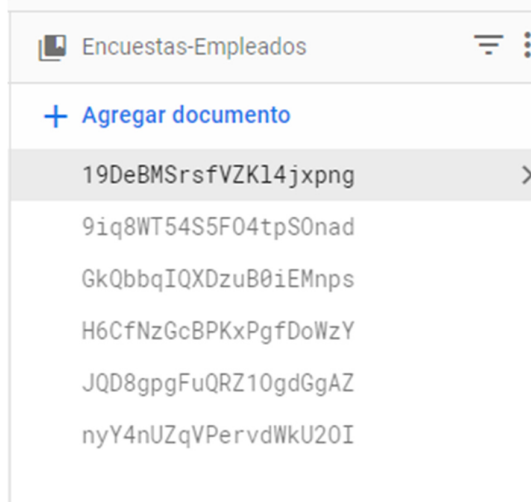
/*recordar que esta función debe ser llamada una única vez, ya que el subscribe se va a encargar de traer los datos todas las veces necesarias. Como consejo, probar lo que pasa al llamar la función traer muchas veces y hacer cambios en la bd*/

Update y delete

```
update(id: string) {
  //obtenemos referencia del documento
  const documento = this.firestore.doc('usuarios/' + id);
  documento.update({
    // Nueva Info. Solo cambia lo que cambiemos
  });
}

delete(id: string) {
  //obtenemos referencia del documento
  const documento = this.firestore.doc('usuarios/' + id);
  documento.delete();
}
```

El parametro id se refiere a esta id:



Una forma para acceder fácilmente a esto es guardar esa id al momento de crear el documento:

```
guardar() {  
  const col = this.firestore.collection('usuarios');  
  // Con solo la coleccion como parametro, crea un documento  
  const documento = this.firestore.doc('usuarios');  
  // Setteamos la información  
  documento.set({  
    id: documento.ref.id,  
    Nombre: 'Fernando',  
    Apellido: 'Fernandez',  
    edad: 30,  
  });  
}
```

Nota: Esta es solo una forma, pueden encontrar más.



Errores comunes (si tienen algún otro pueden comentarlo)

● Error: `Interface 'DatabaseSnapshotExists<T>' incorrectly extends interface 'DataSnapshot'`
Solución: `npm install firebase@9.9.1 --save --save-exact`

● Error: `NullInjectionError`
Solución: Agregar al **AppModule** en la parte de providers:

```
providers: [{ provide: FIREBASE_OPTIONS, useValue: environment.firebase }],
```

También realizar arriba en el mismo archivo el siguiente import:

```
import { FIREBASE_OPTIONS } from '@angular/fire/compat';
```

● Error: No encuentra la carpeta environments durante la instalación.
Solución: `ng generate environments`

Ante cualquier problema no duden en consultar.
Si tienen otro error que tuvieron y quieren agregarlo al documento pueden mandarlo también.

