



Firebase

LABORATORIO – FIREBASE

Firebase: <https://firebase.google.com>

Servicios de Google en la nube para App y Web Servless!

LABORATORIO – FIREBASE

Cloud Firestore

Anteriormente: Firebase Realtime Database.

Actual: Firebase Cloud Firestore (Realtime + funciones más completas de BBDD).

LABORATORIO – FIREBASE

Uso de Cloud Firestore (BBDD en la nube):

PASO 1: Crear proyecto en FIREBASE:

- **Usar una cuenta de GMAIL. No funciona con @uc3m.es.**
- Crear proyecto.
- Añadir Firebase a tu aplicación → Database → Crear colección.

LABORATORIO – FIREBASE

Cloud Firestore - reglas de acceso

```
rules_version = '2';

service cloud.firestore {

    match /databases/{database}/documents {

        match /{document=**} {

            allow read;if true;

            allow write;if true;

        } } }
```

<https://firebase.google.com/docs/firestore/security/rules-structure>

LABORATORIO – FIREBASE

Cloud Firestore - obtener datos de conexión

Aplicación → Settings→ Firebase SDK snippet → Configuración:

```
const firebaseConfig = {  
  
  apiKey: "AlzaSyD0bPAtlsk6f2R0WVQOF3PSdKID--cfINw",  
  
  authDomain: "uc3m-b84c8.firebaseio.com",  
  
  databaseURL: "https://uc3m-b84c8.firebaseio.com",  
  
  projectId: "uc3m-b84c8",  
  
  storageBucket: "uc3m-b84c8.appspot.com",  
  
  ... };
```

LABORATORIO – FIREBASE

Cloud Firestore - instalar firebase en angular

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

LABORATORIO – FIREBASE

Cloud Firestore - configurar environments

src/app/environments/environment.ts

```
export const environment = {  
  production: false,  
  firebase: {  
    apiKey: "AIzaSyD0bPATlsk6f2R0WVQOF3PSdKID--cfINw",  
    authDomain: "uc3m-b84c8.firebaseio.com",  
    databaseURL: "https://uc3m-b84c8.firebaseio.com",  
    projectId: "uc3m-b84c8",  
    storageBucket: "uc3m-b84c8.appspot.com",  
    messagingSenderId: "1020394909731",  
    appId: "1:1020394909731:web:571c311af4f57df292c196",  
    measurementId: "G-G3TQ8461TT"  } };
```

LABORATORIO – FIREBASE

Cloud Firestore - importar en nuestro módulo

src/app/app.module.ts

```
import { AngularFireModule } from '@angular/fire';
import { AngularFirestoreModule } from '@angular/fire/firestore';
import { environment } from '../environments/environment';
```

Imports:[...

```
    AngularFireModule.initializeApp(environment.firebaseio),
```

```
    AngularFirestoreModule
```

```
]
```

LABORATORIO – FIREBASE

Cloud Firestore - crear servicio

ng g service services/firestore/firestore

```
import { Injectable } from '@angular/core';
import { AngularFirestore, AngularFirestoreCollection } from '@angular/fire/firestore';
import { Observable } from 'rxjs';
import { IContacto, Contacto } from '../../contacto';

@Injectable({
  providedIn: 'root'
})
```

LABORATORIO – FIREBASE

Cloud Firestore - crear métodos en servicio

services/firestore/firestore - Constructor

```
export class FirestoreService {  
  private afs:AngularFirestoreCollection<Contacto>;  
  constructor(private firestore: AngularFirestore) {  
    this.afs=this.firestore.collection('contactos');  
  }  
}
```

LABORATORIO – FIREBASE

Cloud Firestore - crear métodos en servicio

services/firestore/firestore – GET

```
public getContactos():Observable<Contacto[]>
{
    return this.afs.valueChanges();
}
```

LABORATORIO – FIREBASE

Cloud Firestore - crear métodos en servicio

services/firestore/firestore – READ

```
public getContacto(id: string):Promise<Contacto>
{
  return this.afs.doc(id).get().toPromise().then(r=>{
    //Si quisieras forzar que se ejecute constructor de Contacto:
    //return new Contacto(r.data() as IContacto);
    return r.data() as Contacto;
  });
}
```

LABORATORIO – FIREBASE

Cloud Firestore - crear métodos en servicio

services/firestore/firestore – UPDATE y DELETE

```
public updateContacto(data:Contacto)
{
    return this.afs.doc(data.id).set(data);
}

public removeContacto(id: string)
{
    return this.afs.doc(id).delete();
}
```

LABORATORIO – FIREBASE

Cloud Firestore - crear métodos en servicio

services/firestore/firestore – INSERT

```
public createContacto(data: IContacto):Promise<string>
{
    //return this.firebaseio.collection<IContacto>('contactos').add({... data}).then(r=>{
    //    return r.id;
    //}

    data.id=this.firebaseio.createId();
    return this.afs.doc(data.id).set({... data}).then(r=>{
        return data.id;
    });
}
```

LABORATORIO – FIREBASE

Cloud Firestore - crear métodos en servicio

services/firestore/firestore – Consultas OrderBy

```
public getContactosSorted():Observable<Contacto[]>
{
    Return this.firestore.collection<Contacto>('contactos',ref=>ref.orderBy('nombre')).valueChanges();
}
```

LABORATORIO – FIREBASE

Cloud Firestore - crear métodos en servicio

services/firestore/firestore – Consultas Where

```
public getContactosFiltered():Observable<Contacto[]>
{
    Return this.firestore.collection<Contacto>('contactos',ref=>ref.where('nombre','==','Ricardo')).valueChanges();
}
```

LABORATORIO – FIREBASE

Cloud Firestore - crear métodos en servicio

services/firestore/firestore – Consultas Filtro+Order

```
public getContactosCombined():Observable<Contacto[]>
{
    Return this.firestore.collection<Contacto>('contactos',ref=>ref.where('nombre','>','O').orderBy('nombre')).valueChanges();
}
```

LABORATORIO – FIREBASE

Cloud Firestore - uso en componentes

Listado.component.ts - imports

```
import { Component, OnInit, OnDestroy } from '@angular/core';
import { FirestoreService } from '../services/firestore/firestore.service'
import { Subscription } from 'rxjs';
```

LABORATORIO – FIREBASE

Cloud Firestore - uso en componentes

Listado.component.ts – Ciclo de vida

```
export class ListadoComponent implements OnInit,OnDestroy {  
  public contactos:Contacto[];  
  public s_contactos:Subscription;  
  
  constructor(private firestoreService: FirestoreService)  
  {  
    this.contactos=[];  
  }  
}
```

LABORATORIO – FIREBASE

Cloud Firestore - uso en componentes

Listado.component.ts – Ciclo de vida

```
ngOnInit()  
{  
    this.s_contactos=this.firestoreService.getContactos().subscribe(data=>{  
        this.contactos=data;  
    });  
}  
  
ngOnDestroy()  
{  
    this.s_contactos.unsubscribe();  
}
```

LABORATORIO – FIREBASE

Cloud Firestore - uso en componentes

Listado.component.ts – operaciones CRUD

```
public eliminar(id){  
    this.firestoreService.removeContacto(id).then(r=>{  
        Console.log('Contacto eliminado!');  
    });  
}
```

LABORATORIO – FIREBASE

Práctica Directorio II

Incorporar persistencia de datos
a la práctica Directorio I