TECNOLÓGICO DE ESTUDIOS SUPERIORES DE ECATEPEC

PROFESORA: Griselda Cortes Barrera

MATERIA: Base de datos para Dispositivos Móviles.

GRUPO: 5801

Nombre:

Arce Baeña Alam Jael

Hernández Rodríguez Paula

Robert Garibay Víctor Dalith

Vázquez Ríos Luis Ángel

Equipo titans

Proyecto enviar correos

Manual de instalación

Requisitos tener instalado angular, node, firebase y un editor texto puede ser visual code o webstorm

Primero creamos un proyecto en angular para esto primero creamos una carpeta en el escritorio y abrimos la consola después ingresamos en la consola la carpeta que creamos y ingresamos el comando ng new nombredelproyecto(aquí ponemos el nombre que quieramos)

```
PS C:\Users\luisv\OneDrive\Escritorio\corre> ng new form-reactive
Node.js version v17.7.0 detected.
Odd numbered Node.js versions will not enter LTS status and should not be used for prod
uction. For more information, please see https://nodejs.org/en/about/releases/.
? Would you like to add Angular routing? Yes
? Which stylesheet format would you like to use? CSS
CREATE form-reactive/angular.json (3081 bytes)
CREATE form-reactive/package.json (1076 bytes)
CREATE form-reactive/README.md (1058 bytes)
CREATE form-reactive/tsconfig.json (863 bytes)
CREATE form-reactive/.editorconfig (274 bytes)
CREATE form-reactive/.gitignore (548 bytes)
CREATE form-reactive/.browserslistrc (600 bytes)
CREATE form-reactive/karma.conf.js (1430 bytes)
CREATE form-reactive/tsconfig.app.json (287 bytes)
CREATE form-reactive/tsconfig.spec.json (333 bytes)
CREATE form-reactive/.vscode/extensions.json (130 bytes)
CREATE form-reactive/.vscode/launch.json (474 bytes)
CREATE form-reactive/.vscode/tasks.json (938 bytes)
CREATE form-reactive/src/favicon.ico (948 bytes)
CREATE form-reactive/src/index.html (298 bytes)
CREATE form-reactive/src/main.ts (372 bytes)
CREATE form-reactive/src/polyfills.ts (2338 bytes)
CREATE form-reactive/src/styles.css (80 bytes)
CREATE form-reactive/src/test.ts (745 bytes)
CREATE form-reactive/src/assets/.gitkeep (0 bytes)
CREATE form-reactive/src/environments/environment.prod.ts (51 bytes)
CREATE form-reactive/src/environments/environment.ts (658 bytes)
CREATE form-reactive/src/app/app-routing.module.ts (245 bytes)
CREATE form-reactive/src/app/app.module.ts (393 bytes)
CREATE form-reactive/src/app/app.component.html (23364 bytes)
CREATE form-reactive/src/app/app.component.spec.ts (1094 bytes)
CREATE form-reactive/src/app/app.component.ts (217 bytes)
CREATE form-reactive/src/app/app.component.css (0 bytes)
✓ Packages installed successfully.
warning: LF will be replaced by CRLF in .browserslistrc.
The file will have its original line endings in your working directory
```

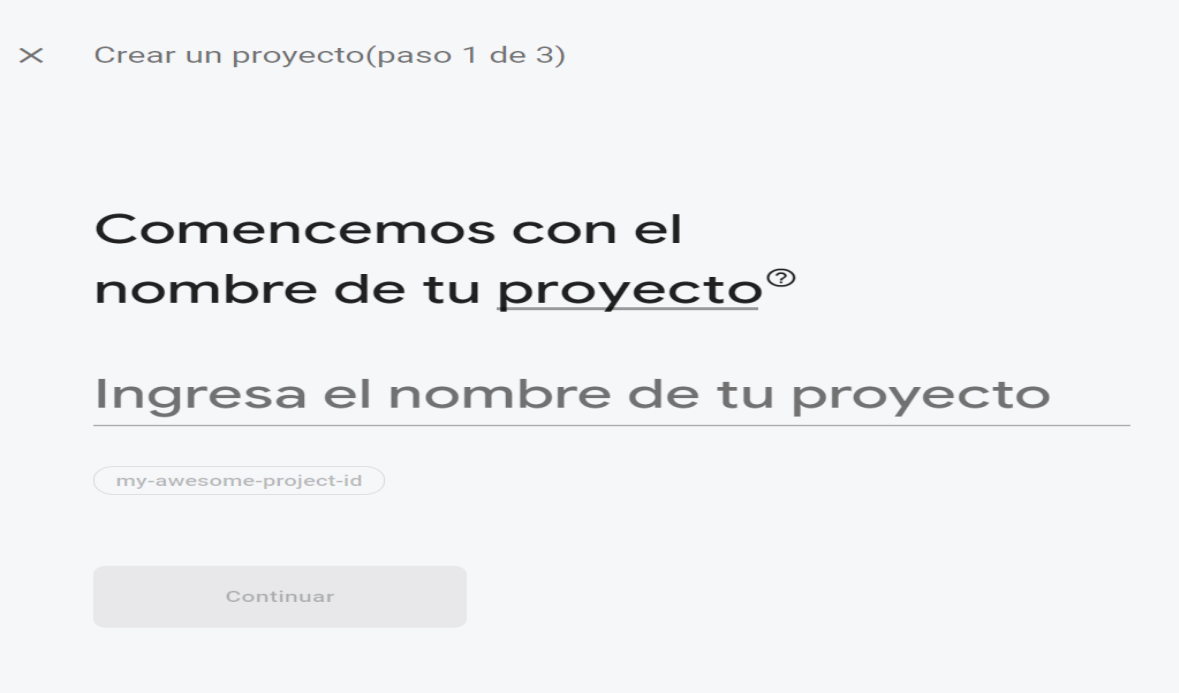
Después creamos la carpeta de componentes/ contact ng g c components/contact

```
PS C:\Users\luisv\OneDrive\Escritorio\corre> cd form-reactive
PS C:\Users\luisv\OneDrive\Escritorio\corre\form-reactive> ng g c components/contact
Node.js version v17.7.0 detected.
Odd numbered Node.js versions will not enter LTS status and should not be used for prod
uction. For more information, please see https://nodejs.org/en/about/releases/.
CREATE src/app/components/contact/contact.component.html (22 bytes)
CREATE src/app/components/contact/contact.component.spec.ts (633 bytes)
CREATE src/app/components/contact/contact.component.ts (279 bytes)
CREATE src/app/components/contact/contact.component.css (0 bytes)
UPDATE src/app/app.module.ts (490 bytes)
PS C:\Users\luisv\OneDrive\Escritorio\corre\form-reactive>
```

Crear proyecto en firebase

Ingresamos a https://console.firebase.google.com/ nos registramos

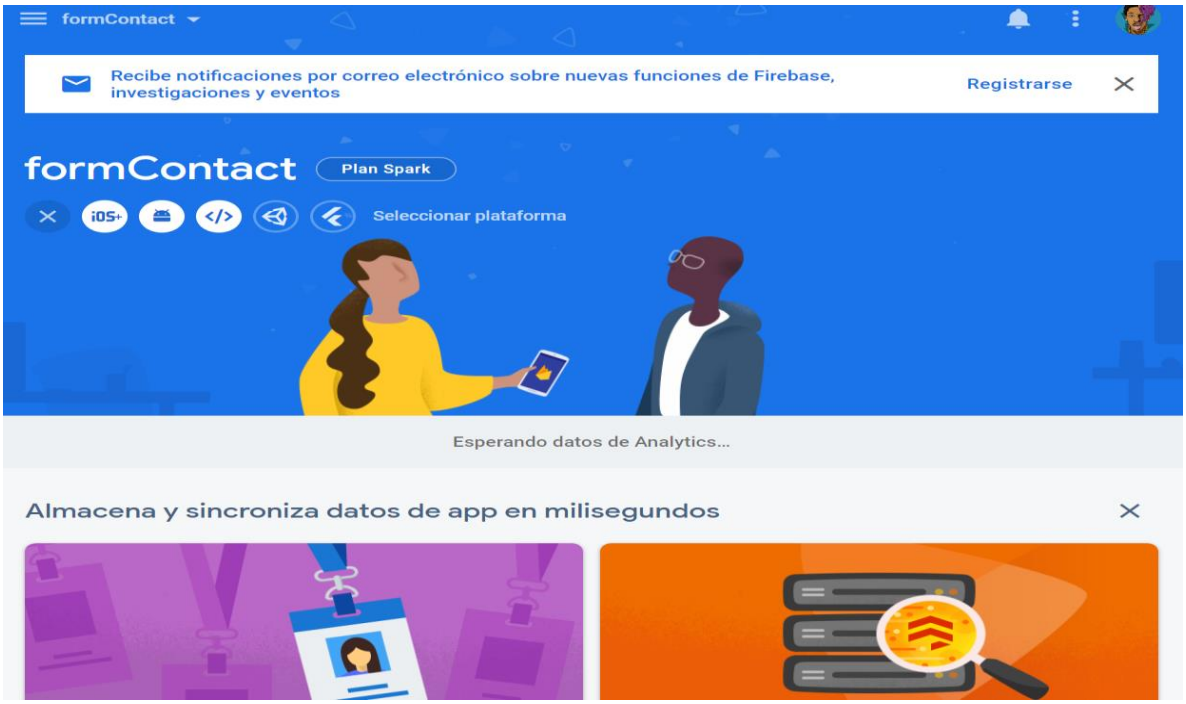
Depues creamos un proyecto le ponemos el nombre que quieramos le damos continuar



Otra vez continuar

Le damos en default y crear proyecto

Después nos mandara a esta ventana



Le damos clic en </> que es la consola

Y después le ponemos el nombre que queramos y le damos en registrar app

Después no abrirá otra ventana en la cual copiaremos el código subrayado y lo pegamos en nuestro proyecto de angular

Luego, inicializa Firebase y comienza a usar los SDK de los productos que quieres usar.

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

// Your web app's Firebase configuration
// For Firebase JS SDK v7.20.0 and later, measurementId is o
const firebaseConfig = {
  apiKey: "AIzaSyC9Cr7lWsNZsrm1Eu5saeNVJGuDIJ6Nq84",
  authDomain: "formcontact-b50ac.firebaseapp.com",
  projectId: "formcontact-b50ac",
  storageBucket: "formcontact-b50ac.appspot.com",
  messagingSenderId: "704646355298",
  appId: "1:704646355298:web:2f30f826c013eb22b2f43b",
  measurementId: "G-9H2H0L9YKE"
};

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

Nota: Esta opción utiliza el SDK de JavaScript modular, que proporciona un tamaño reducido del SDK.

En visual code o webstorn abrimos la carpeta enviroments y abrimos el archivo enviaronments.ts y pegaremos el código en production: false y agregamos una , y pegamos el código y borraremos const y cambiamos = por : y borramos el ; del antepenúltimo llave y le damos guardar

```ts
// This file can be replaced during build by using the `
// `ng build` replaces `environment.ts` with `environmer
// The list of file replacements can be found in `angula

export const environment = {
  production: false,
  firebaseConfig : {
    apiKey: "AIzaSyC9Cr7lWsNZsrm1Eu5saeNVJGuDIJ6Nq84",
    authDomain: "formcontact-b50ac.firebaseapp.com",
    projectId: "formcontact-b50ac",
    storageBucket: "formcontact-b50ac.appspot.com",
    messagingSenderId: "704646355298",
    appId: "1:704646355298:web:2f30f826c013eb22b2f43b",
    measurementId: "G-9H2H0L9YKE"
  }
};

/*
 * For easier debugging in development mode, you can imp
 * to ignore zone related error stack frames such as `zo
 *
 * This import should be commented out in production moc
 * on performance if an error is thrown.
 */
// import 'zone.js/plugins/zone-error';  // Included wit
```

En la terminal de visual code instalaremos firebase en nuestro proyecto con la siguiente línea de código npm i firebase @angular/fire

```
PS C:\Users\luisv\OneDrive\Escritorio\corre\form-reactive> npm i firebase @angular/fire

added 118 packages, removed 1 package, and audited 1040 packages in 33s

107 packages are looking for funding
  run `npm fund` for details

found 0 vulnerabilities
PS C:\Users\luisv\OneDrive\Escritorio\corre\form-reactive>
```

Después empezaremos a importar módulos nos vamos al archivo app.module.ts

Agregaremos cuatro modules angularfiremodule, angularforestoremodule, enviroment y reactiveformsmodule

```
import { NgModule } from '@angular/core';
import { BrowserModule } from '@angular/platform-browser';
import {ReactiveFormsModule} from "@angular/forms";
import { AppComponent } from './app.component';
import { ContactComponent } from './components/contact/contact.component';
import {AngularFireModule} from '@angular/fire/compat';
import {AngularFirestoreModule} from '@angular/fire/compat/firestore';
import {environment} from "../environments/environment";
```

```
imports: [
  BrowserModule,
  AngularFireModule.initializeApp(environment.firebaseConfig),
  AngularFirestoreModule,
  ReactiveFormsModule
]
```

Ahora creamos service para conectar a backed firebase, con el comando ng g s service/data-db

```
PS C:\Users\luisv\WebstormProjects\form-reactive> ng g s services/data-db
Node.js version v17.7.0 detected.
Odd numbered Node.js versions will not enter LTS status and should not be used for production. For
re information, please see https://nodejs.org/en/about/releases/.
CREATE src/app/services/data-db.service.spec.ts (358 bytes)
CREATE src/app/services/data-db.service.ts (135 bytes)
PS C:\Users\luisv\WebstormProjects\form-reactive>
```

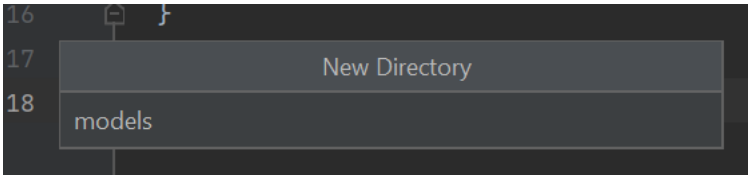
Ahora nos vamos ala carpeta creada y abrimos data-db. Service.ts y aquí importaremos el angularfirestore, angularfirestorecollection, observable y después declararemos el contactcollection y en el constructor ingresamos el angularfirestore

```
import { Injectable } from '@angular/core';
import {AngularFirestore, AngularFirestoreCollection} from "@angular/fire/compat/firestore";
import  {Observable} from "rxjs";

@Injectable({
  providedIn: 'root'
})
export class DataDbService {
private contactCollection: AngularFirestoreCollection<any>;
  constructor(private  afs: AngularFirestore) {
    this.contactCollection = afs.collection<any>( path: 'contacts');

  }
  saveMesasage (newContact: any): void{
    this.contactCollection.add(newContact);
  }
```

Después en la carpeta app creamos una carpeta nueva llamada models y en dicha carpeta crearemos un nuevo fichero message.interface.ts



En este fichero crearemos la interfaz llevara email nombre y mensaje

```
export interface MessageI {
  email: string;
  name: string;
  message: string;

}
```

Después volvemos a data-db y cambiamos los any por messagei y lo importamos

```
import { MessageI} from "../models/message.interface";

@Injectable({
  providedIn: 'root'
})
export class DataDbService {
private contactCollection: AngularFirestoreCollection<MessageI>;
  constructor(private  afs: AngularFirestore) {
    this.contactCollection = afs.collection<MessageI>( path: 'contacts')

  }
  saveMesasage (newContact: MessageI): void{
    this.contactCollection.add(newContact);
  }
}
```

Ahora en nuestro app.module.ts importaremos nuestro services y lo agregamos en providers

```
import  {DataDbService} from "./services/data-db.service";

@NgModule({
  declarations: [
    AppComponent,
    ContactComponent
  ],
  imports: [
    BrowserModule,
    AngularFireModule.initializeApp(environment.firebaseConfig),
    AngularFirestoreModule,
    ReactiveFormsModule
  ],
  providers: [DataDbService],
  bootstrap: [AppComponent]
```

Ahora abrimos contact.component.ts y importamos datadbservice , formcontrol, formgroup,

Después de importar crearemos el grupo de formularios que vamos a ocupar

```
import  {DataDbService} from "../../services/data-db.service";
import {FormControl, FormGroup} from "@angular/forms";

@Component({
  selector: 'contactForm',
  templateUrl: './contact.component.html',
  styleUrls: ['./contact.component.css']
})
export class ContactComponent implements OnInit {
 createFormGroup () {
   return new FormGroup( controls: {
     email : new FormControl( formState: ''),
     name: new FormControl( formState: ''),
     message: new FormControl( formState: '')
   });
```

Crearemos contactform y en constructor agregamos services después creamos un método para limpiar los formularios y para salvar el formulario

```
contacForm: FormGroup;
 constructor(private dbData: DataDbService) {
   this.contacForm = this.createFormGroup();
 }

 ngOnInit(): void {
 }
onResetForm(){
 this.contacForm.reset();
}
onSaveForm(){
    console.log('Saved');
}
```

Ahora trabajaremos con el Contac.html y en Contac.css aquí depende del estilo que le quieras dar a tu pantalla para redactar les mostrare como mi equipo lo hicimos

```
<div class="form-container" xmlns="http://www.w3.org/1999/html">
  <h1>redactar correo</h1>
  <form [formGroup]="contacForm"(ngSubmit)="onSaveForm()">
    <input type="text" formControlName="name" placeholder="ingresa tu nombre">
    <input type="text" formControlName="email" placeholder="ingresa tu email">
    <textarea formControlName="message" placeholder="escriba su mensaje"></textarea>
    <button type="submit" class="btn-send">Send </button>
  </form>
</div>
```

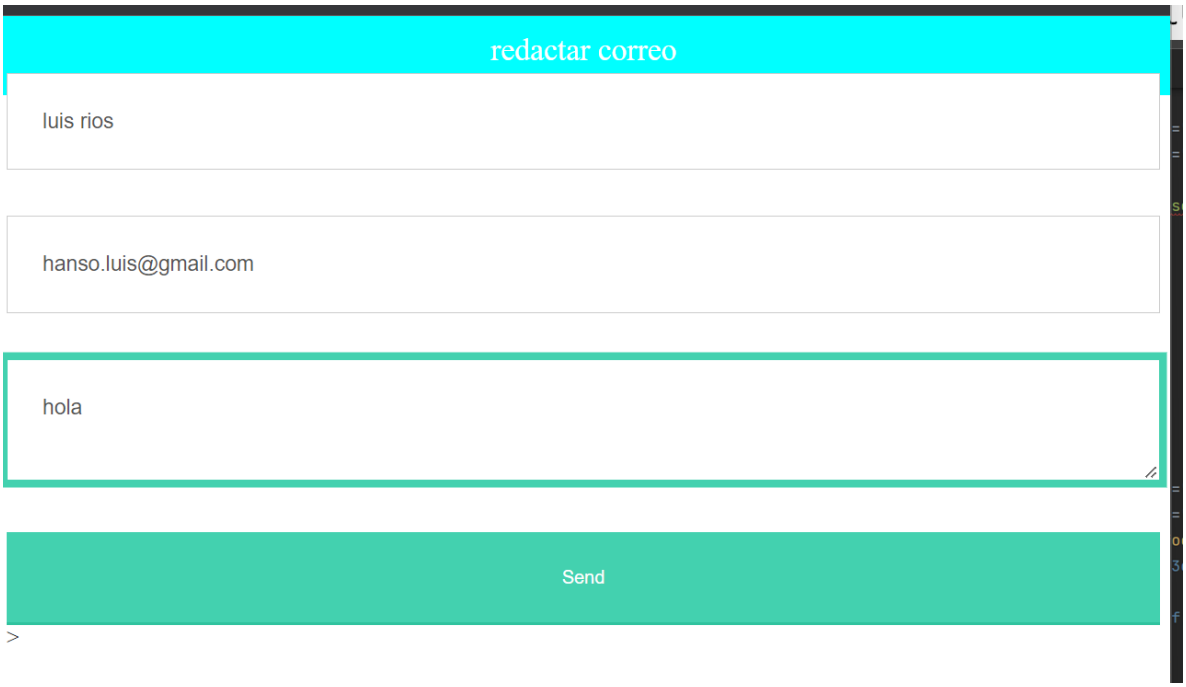
contact.component.html | contact.component.css

```css
.form-container h1{
  background: aqua;
  padding: 20px 0;
  font-size: 140%;
  font-weight: 300;
  text-align: center;
  color: white;
  margin: -16px;
}
.form-container input[type="text"],
.form-container input[type="email"],
.form-container textarea{
  transition: all 0.30s ease-in-out;
  outline: none;
  box-sizing: border-box;
  width: 100%;
  background: #fff;
  margin-bottom: 4%;
  border: 1px solid #ccc;
  padding: 3%;
  color: #555;
  font: 95%Arial;
}
.form-container input[type="text"]:focus,
.form-container input[type="email"]:focus,
.form-container textarea:focus{
  box-shadow: 0 0 0 5px #43d1af;
  padding: 3%;
  border: 1px solid #43d1af;
}
.form-container .btn-send{
```

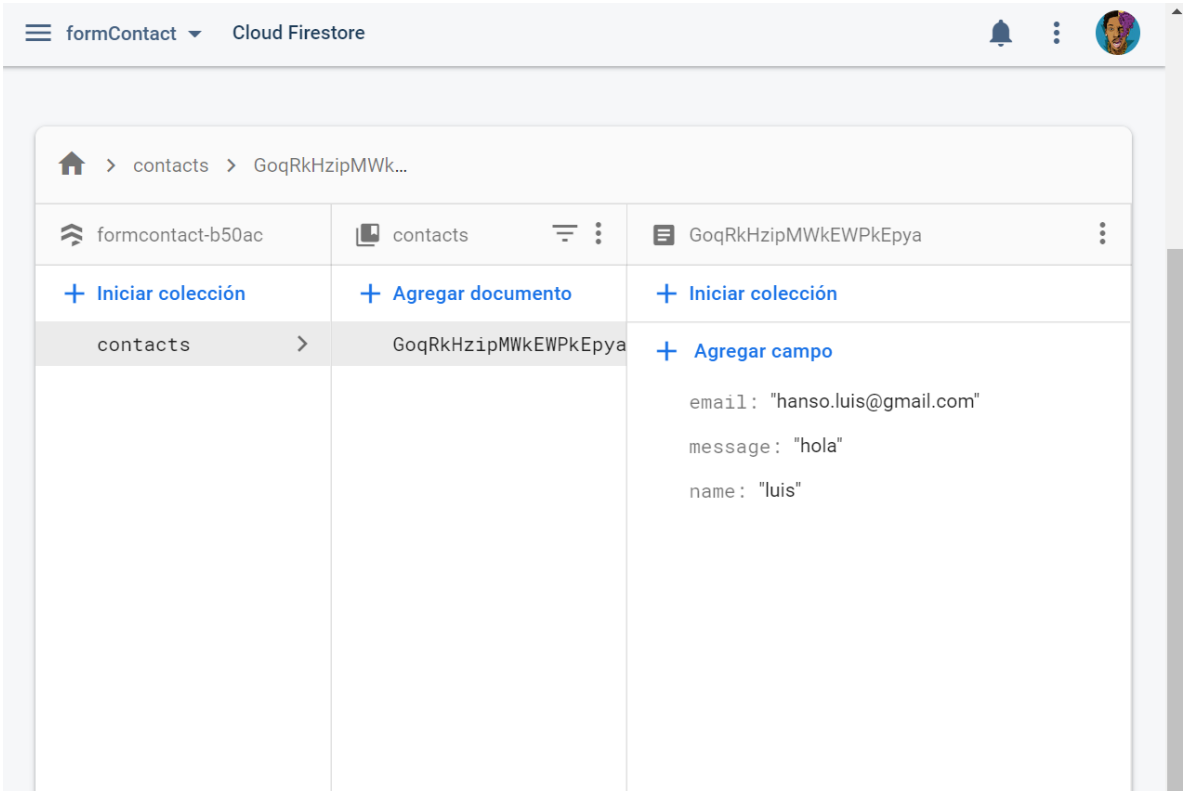
```css
.form-container .btn-send{
  box-sizing: border-box;
  width: 100%;
  padding: 3%;
  background: #43d1af;
  border-bottom: 2px solid #30c29e;
  border-top-style: none;
  border-right-style: none;
  border-left-style: none;
  color: #fff;
}
```

Después de diseñar nuestra interfaz probaremos nuestra aplicación de angular

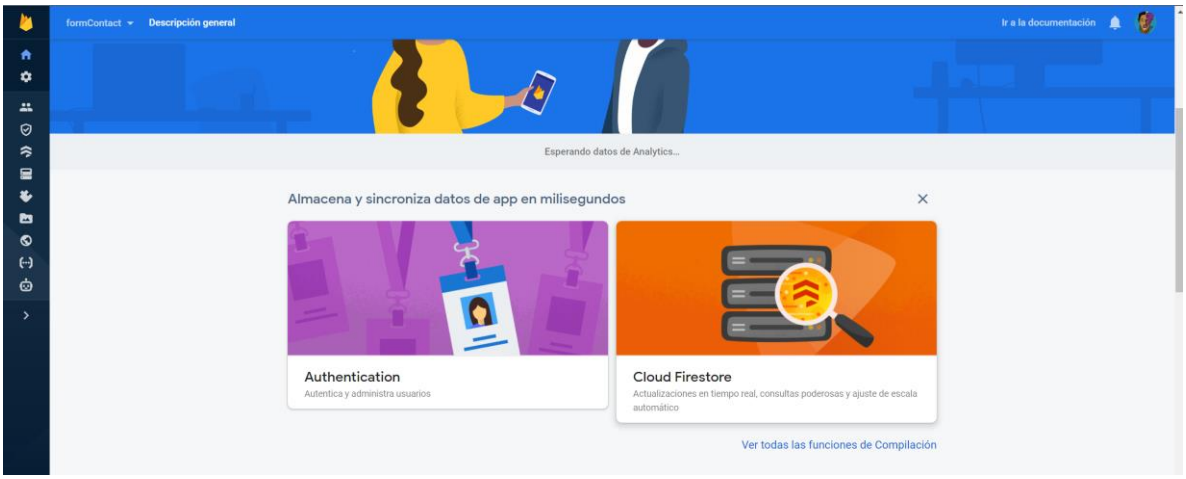
El ejemplo



Y en la base de datos de firebase podemos observar el correo



Para activar la base de datos de firebase tenemos que ir en la pantalla principal a cloud firestore y creamos la base de datos



Conclusión

Usamos firebase para hacer las pruebas y ver si nuestro código funcionaba correctamente pero si lo quisieran poner en producción tendrían que cambiar firebase por un servidor smtp, que es protocolo simple de transferencia de correo (SMTP) es un protocolo TCP/IP que se utiliza para enviar y recibir correo electrónico. Normalmente se utiliza con POP3 o con el protocolo de acceso a mensajes de Internet (IMAP) para guardar mensajes en un buzón del servidor y descargarlos periódicamente del servidor para el usuario.

El establecimiento y mantenimiento de un servidor SMTP propio suele ser una opción para aquellas empresas preocupadas por la seguridad y el control del proceso, así como por la fiabilidad del servicio, pues en ocasiones desconocen los estándares que pueden ofrecer los proveedores externos.

Sin embargo, disponer de un servidor SMTP propio es una tarea que puede rápidamente convertirse en algo costoso y que consuma mucho tiempo. El uso de tecnología propia implica destinar recursos a su gestión y mantenimiento, y no siempre se puede contar con los mayores avances o las mejores funciones.