

# Desarrollo de una aplicación que permite realizar funciones CRUD utilizando Ionic & Firebase.

Jessica Amoguimba, Gabriel Ibujés, Escuela Politécnica Nacional (EPN), Quito - Ecuador

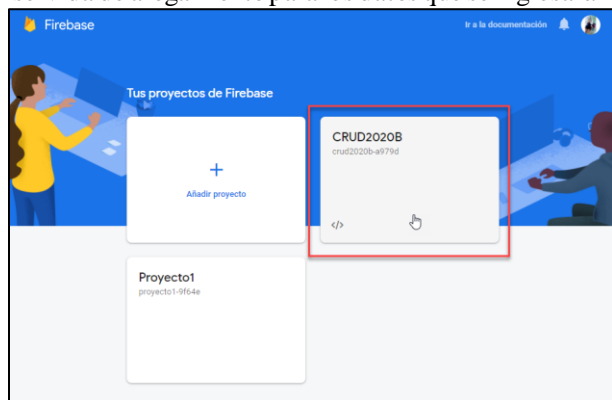
**Resumen** – En este documento se muestra la el desarrollo de una aplicación en este caso una agenda, utilizando un SGBD no relacional “Firebase”, Ionic con Angular y el IDE Visual Studio Code, para luego ser transformada en .apk y ser ejecutado desde cualquier dispositivo en el que se instale, además este recopilará los datos ingresados y serán almacenados en tiempo real en Firebase.

## I. INTRODUCCIÓN

La presente detalla paso a paso cómo se realizó el proceso de esta aplicación, su implementación en Firebase y luego como se realiza la recopilación de datos en la misma mediante esta aplicación.

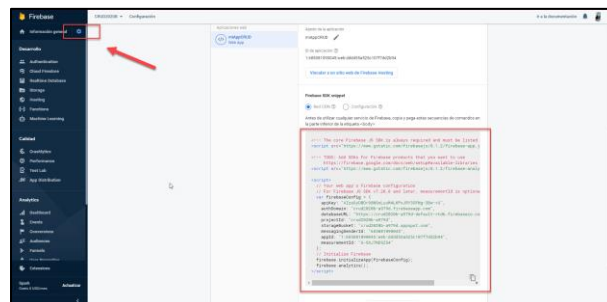
## II. DESARROLLO DEL PROYECTO

Para empezar, vamos a crear un proyecto en FIREBASE el cual servirá de alojamiento para los datos que se ingresarán.



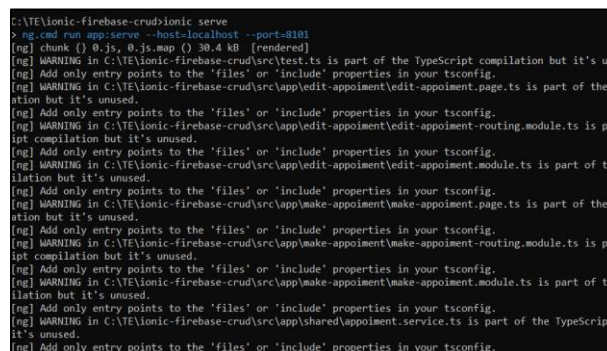
**Ilustración 1 Creación de proyecto en Firebase**

Acto seguido nos dirigimos a la opción Configuración para poder obtener las credenciales que serán de utilidad para establecer la conexión con la aplicación. Ve en la Ilustración 2.



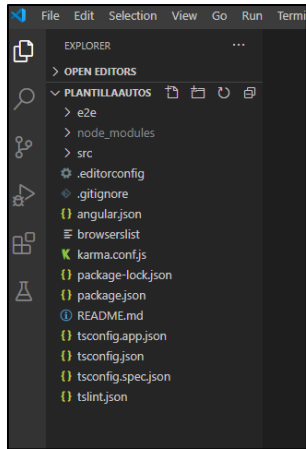
**Ilustración 2 Credenciales Firebase**

Para la creación del proyecto en Ionic ejecutamos desde una carpeta en la terminal lo siguiente 'ionic start ionic-firebase-crud blank --type=angular' esperamos unos minutos para que se cree todo el proyecto y en cuanto termine se debe ejecutar otro código 'ionic serve' el cual sirve para poner en marcha el proyecto así se puede ir trabajando sobre este con la ayuda de Visual Studio Code.



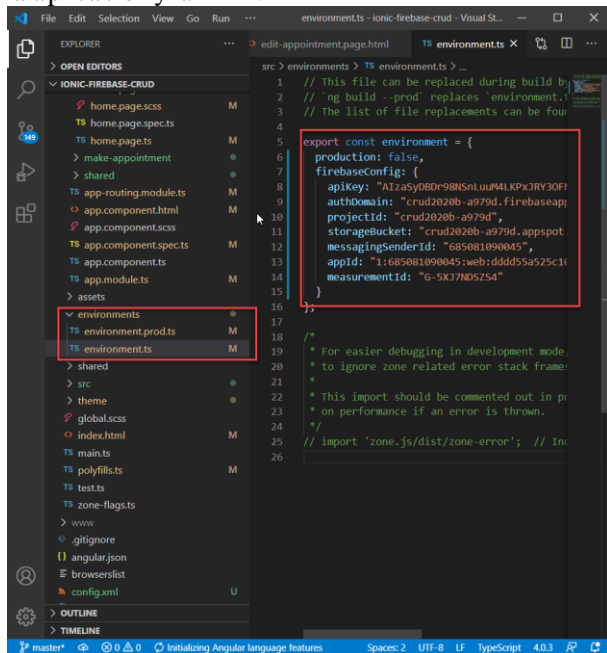
**Ilustración 3 Ejecución del proyecto**

A continuación, se observa las carpetas del proyecto en Visual Studio Code, en las cuales se trabajará, este proyecto.



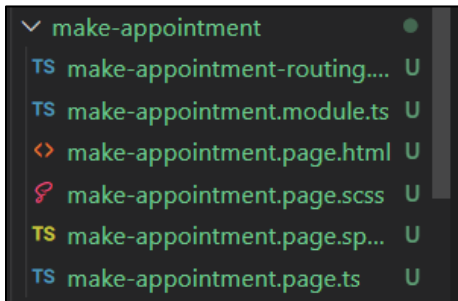
**Ilustración 4 Proyecto en Visual Studio Code**

Luego de eso empezamos con la conexión de Firebase en 'environments', esto permitirá que se establezca conexión entre la aplicación y la BDD.

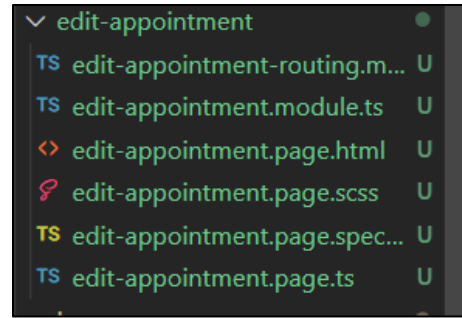


**Ilustración 5 Conexión con credenciales Firebase**

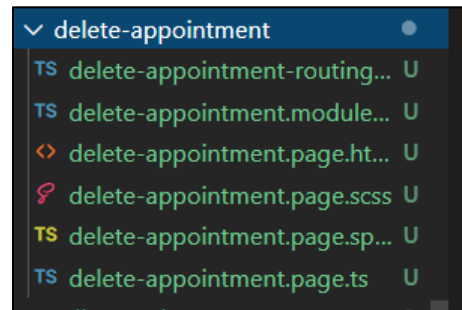
Luego de que configuramos el proyecto con las credenciales para que se comunique, creamos appointments make, delete, edit, estas permitirán el manejo de los datos para realizar el CRUD.



**Ilustración 6 Appointment Make**

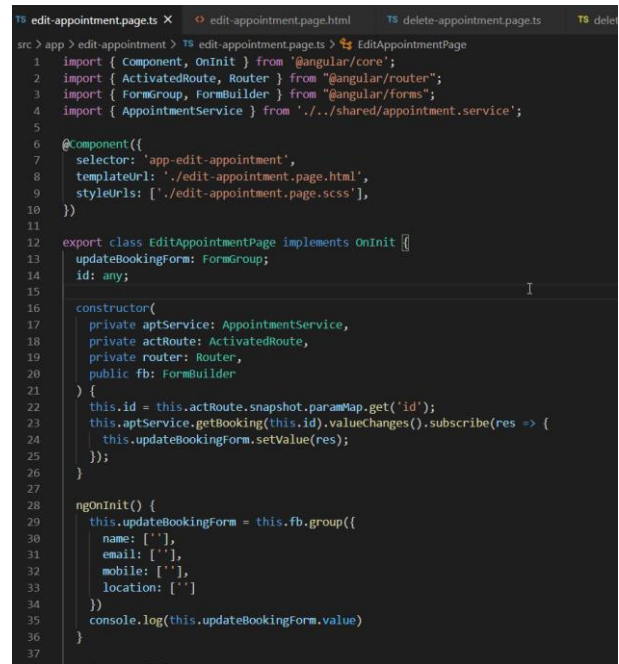


**Ilustración 7 Appointment Edit**



**Ilustración 8 Appointment Make**

Dentro de estas se codifican de acuerdo al método de cada uno de los CRUD tal y como se ve en la Ilustración 9 y la Ilustración 10.



**Ilustración 9 Métodos en appointment Edit**

### Ilustración 10 Métodos en appointment Make

[illegible]

## Ilustración 12 Código para Home

The screenshot shows the Visual Studio Code interface. On the left, the 'EXPLORER' sidebar displays the project structure. The file 'appointment.service.ts' is selected under the 'src' directory. The main editor window shows the code for the 'AppointmentService' class. The code includes imports for 'AppDataSource' and 'Appointment', and defines four methods: 'createBooking', 'getBooking', 'updateBooking', and 'deleteBooking'. Each method interacts with the database using 'this.db'.

```

import { AppDataSource } from './app.datasource';
import { Appointment } from './appointment.entity';

export class AppointmentService {
    // Create
    createBooking(apt: Appointment) {
        return this.bookingListRef.push({
            name: apt.name,
            email: apt.email,
            mobile: apt.mobile,
            location: apt.location
        });
    }

    // Get Single
    getBooking(id: string) {
        this.bookingRef = this.db.object('/appo');
        return this.bookingRef;
    }

    // Get List
    getBookingList() {
        this.bookingListRef = this.db.list('/app');
        return this.bookingListRef;
    }

    // Update
    updateBooking(id, apt: Appointment) {
        return this.bookingRef.update({
            name: apt.name,
            email: apt.email,
            mobile: apt.mobile,
            location: apt.location
        });
    }

    // Delete
    deleteBooking(id: string) {
        this.bookingRef = this.db.object('/appo');
        this.bookingRef.remove();
    }
}

```

## Ilustración 11 Creación funciones CRUD

## Registros

Nombre

Email

Celular

Dirección

AGENDAR

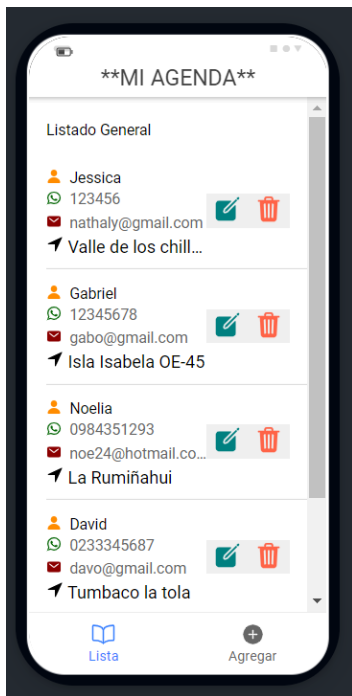
Lista

Agregar

**Ilustración 13 Página Registros**

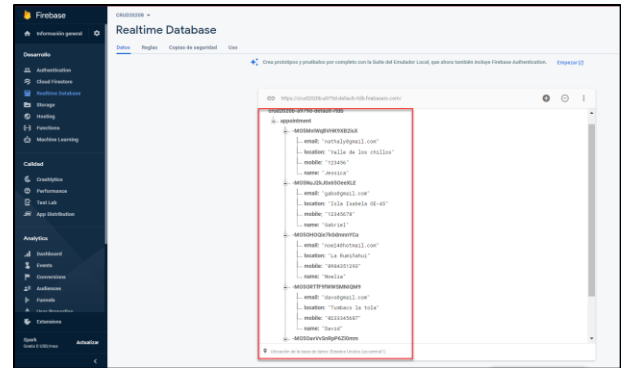


**Ilustración 14** Página Editar registro



**Ilustración 15** Página Inicial

A medida que se ingresen datos a la agenda estos son registrados en Firebase como se observa en la Ilustración 16.

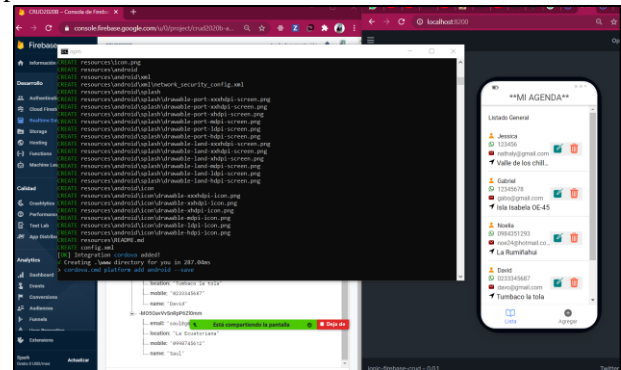


**Ilustración 16** Registros en Firebase

Una vez que todo se encuentre funcionando bien procedemos a realizar la .apk de la aplicación. Para ello detenemos la ejecución del proyecto y colocamos el comando:

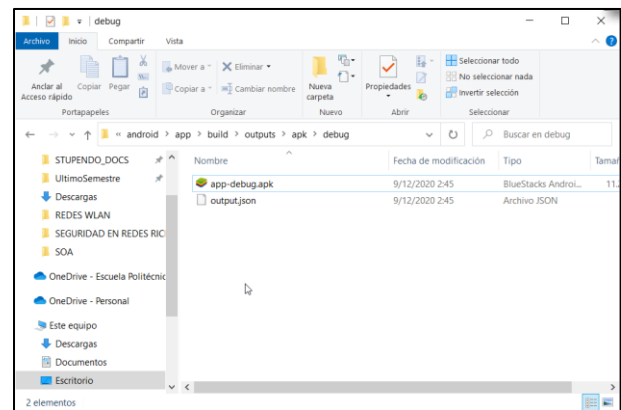
```
C:\TE\ionic-firebase-crud>ionic cordova build android
```

Se debe esperar unos minutos mientras se transfiere la .apk.



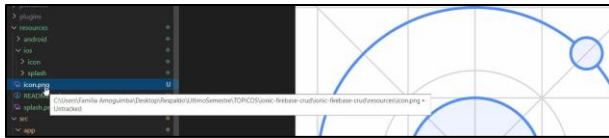
**Ilustración 17** Ejecución del código

Una vez terminada la ejecución nos dirigimos a la siguiente ruta  
 'ionic-firebase-crud\platforms\android\app\build\outputs\apk\debug\app-debug.apk'



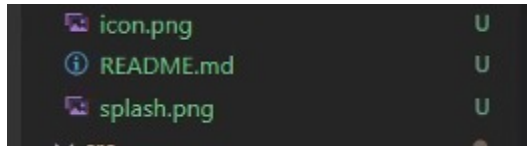
**Ilustración 18** Ruta donde se aloja la apk

Para finalizar agregamos splash screen y un icono personalizado a nuestra apk. Nos dirigimos a la carpeta 'resources' y abrimos la imagen icon.png con Paint para modificarlo.



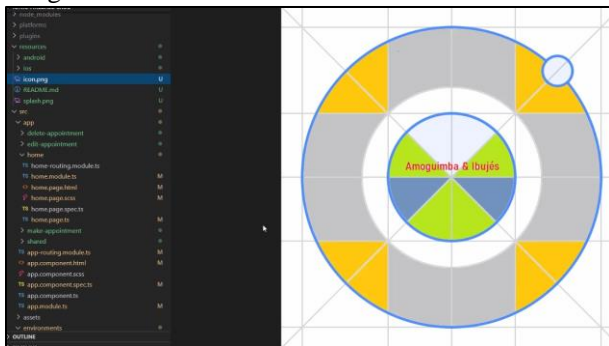
**Ilustración 19 Imagen icon.png**

De igual manera con la imagen splash.png

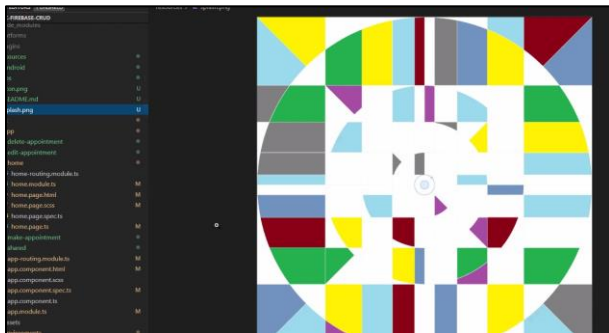


**Ilustración 20 Imagen splash.png**

Una vez personalizados volvemos a ejecutar la aplicación para que se guarden los cambios.



**Ilustración 21 Imagen personalizada para icon**



**Ilustración 22 Imagen personalizada para splash screen**

Para finalizar ejecutamos nuevamente el código:

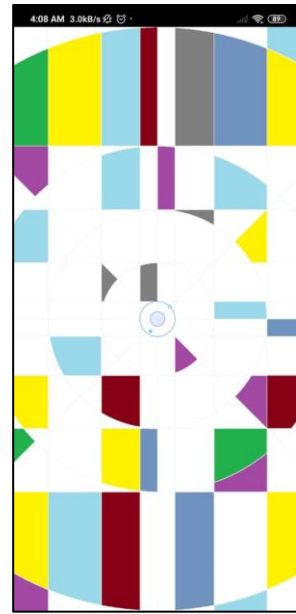
```
C:\TE\ionic-firebase-crud>ionic cordova build android
```

Y para verificar que la aplicación esta funcional totalmente la instalamos en un dispositivo celular.



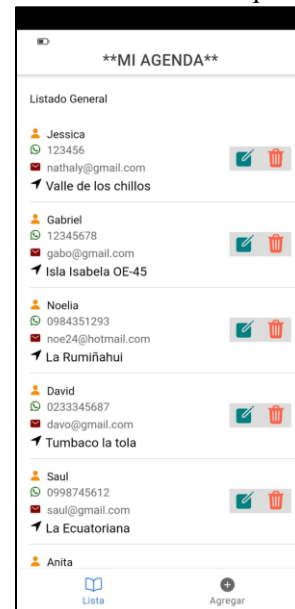
**Ilustración 23 Instalación de la app en el celular**

Al iniciar la aplicación se abre el splash screen que lo personalizamos junto con el icono de la aplicación.



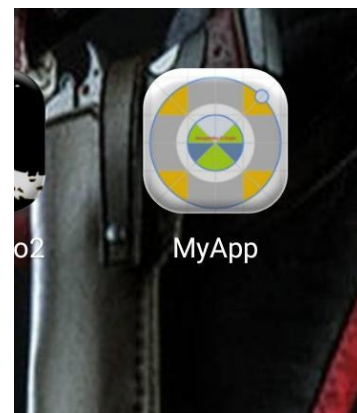
**Ilustración 24 Pantalla con splash screen**

La pantalla de inicio con los contactos que están el Firebase.



**Ilustración 25 Registros ingresados guardados en Firebase**

Y en la pantalla del celular se observa nuestro icono distintivo.



**Ilustración 26 Icono de la aplicación**



### III. CODIGO FUENTE

#### **Enlace de GitHub:**

<https://github.com/JESSICAAMOGUIMBA/TOPICOS-ESPECIALES-PRUEBA1-AMOGUIMBA-IBUJES>

### IV. CONCLUSIONES

La creación de la aplicación utilizando Ionic y Firebase resulta interesante ya que con código no tan difícil se puede establecer una comunicación entre la BDD y la aplicación otorgando a la vez almacenar los datos en tiempo real.

Al instalar en el celular e ingresar más datos a la agenda se almacenan correctamente en Firebase.

### V. REFERENCIAS

#### **BDD FIREBASE:**

<https://console.firebase.google.com/u/0/project/crud2020b-a979d/settings/general/web:NjQyOGY4ZTkxZTMxNS00NWVxLTUjNTMtODYwNGZiNzcyZjRk?hl=es>

<https://console.firebase.google.com/u/0/project/crud2020b-a979d/settings/general/web:NjQyOGY4ZTkxZTMxNS00NWVxLTUjNTMtODYwNGZiNzcyZjRk?hl=es>

**ICONOS:** <https://ionicons.com/>

#### **VIDEOS DE AYUDA:**

<https://www.youtube.com/watch?v=REGMMe2FYKA>

<https://www.youtube.com/watch?v=AMCvLAUTGX4>

**VIDEO DE EXPLICACIÓN:** <https://youtu.be/F0nWLMaI4GM>