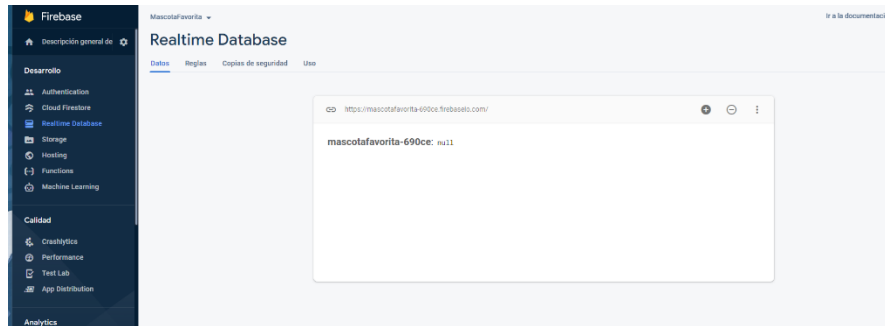


Notificaciones-Tarea

- a) Para el presente trabajo se creó un proyecto nuevo en firebase llamado MascotaFavorita que es donde guardare los registros de usuarios.



- b) Una vez hecho eso modifique el index de mi servidor Heroku para que enviara y retornara los atributos de id, token del dispositivo y nombre de usuario a mi base de datos en tiempo real de Firebase.

```
111
112 .post('/'+ tokenDevicesURI, function(req, res){
113     const token = req.body.token
114     const nombre = req.body.nombre
115
116
117     const db = firebase.database()
118     const tokenDevice = db.ref(tokenDevicesURI).push()
119     tokenDevice.set({
120         token : token,
121         nombre : nombre
122     })
123
124     const path = tokenDevice.toString();
125     const pathSplit = path.split( tokenDevicesURI + '/')
126     const idAutoGenerado = pathSplit[1]
127
128     const respuesta = generarRespuestaAToken(db, idAutoGenerado)
129     res.setHeader('Content-Type', 'application/json')
130     res.send(JSON.stringify(respuesta))
131
132 })
```

```
/////////Mascota-Favorita/////////
const firebase = require('firebase-admin')
const serviceAccount = require('./mascotafavorita-690ce-firebase-adminsdk-bmepc-3fe98f099f.json')

firebase.initializeApp({
  credential: firebase.credential.cert(serviceAccount),
  databaseURL: 'https://mascotafavorita-690ce.firebaseio.com'
})

const tokenDevicesURI = 'token-device'

function generarRespuestaAToken(db, idAutoGenerado){
  var respuesta = {}
  var usuario = {}
  var ref = db.ref('token-device')
  ref.on('child_added', function(childSnapshot, prevChildKey){
    usuario = childSnapshot.val()
    respuesta = {
      id: idAutoGenerado,
      token: usuario.token,
      nombre : usuario.nombre
    }
  })
  return respuesta;
}
```

- c) En mi proyecto de Android cree el siguiente EndPoint que recibe los atributos de token y nombre.

```
package com.romychsa.favorito_mascota.Notificaciones;

import com.romychsa.favorito_mascota.Notificaciones.RestAPI.model.UsuarioResponse;

import retrofit2.Call;
import retrofit2.http.Field;
import retrofit2.http.FormUrlEncoded;
import retrofit2.http.POST;

public interface EndPoints {
    @FormUrlEncoded
    @POST(ConstantsRestAPI.KEY_POST_ID_TOKEN)
    Call<UsuarioResponse> registrarUsuario(@Field("token") String token, @Field("nombre") String nombre);
}
```

- d) Los métodos de LanzaNotificación() y enviarRegistro() están planteados de la siguiente forma respectivamente.

```
public void lanzarNotificacion(){
    FirebaseInstanceId.getInstance().getInstanceId()
        .addOnCompleteListener(new OnCompleteListener<InstanceIdResult>() {
            @Override
            public void onComplete(@NonNull Task<InstanceIdResult> task) {
                if (!task.isSuccessful()) {
                    Log.w(TAG, "getInstanceId failed", task.getException());
                    return;
                }
                // Get new Instance ID token
                String token = task.getResult().getToken();

                // Log and toast
                String msg = "InstanceID Token: {token}";
                Log.d(TAG, msg);
                Toast.makeText(context: MainActivity.this, msg, Toast.LENGTH_SHORT).show();
                enviarRegistro(token);
            }
        });
}

private void enviarRegistro(String token){
    Mascota mascota = mascotas.get(0);
    username = mascota.getNombre();

    Log.d(tag: "TOKEN", token);
    Log.d(tag: "USER", username);

    RestApiAdapter restApiAdapter = new RestApiAdapter();
    EndPoints endPoints = restApiAdapter.establecerConexionRestAPI();
    Call<UsuarioResponse> usuarioResponseCall = endPoints.registrarUsuario(token, username);

    usuarioResponseCall.enqueue(new Callback<UsuarioResponse>() {
        @Override
        public void onResponse(Call<UsuarioResponse> call, Response<UsuarioResponse> response) {
            UsuarioResponse usuarioResponse = response.body();
            Log.d(tag: "ID_FIREBASE", usuarioResponse.getId());
            Log.d(tag: "TOKEN_FIREBASE", usuarioResponse.getToken());
            Log.d(tag: "USER_NAME", usuarioResponse.getNombre());
        }

        @Override
        public void onFailure(Call<UsuarioResponse> call, Throwable t) {
        }
    });
}
```

- e) Respuesta de la aplicación

