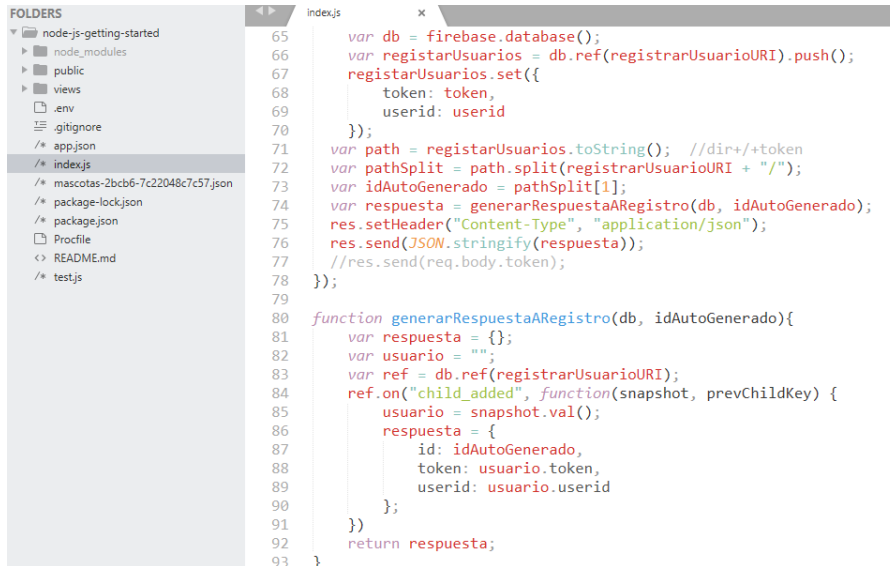
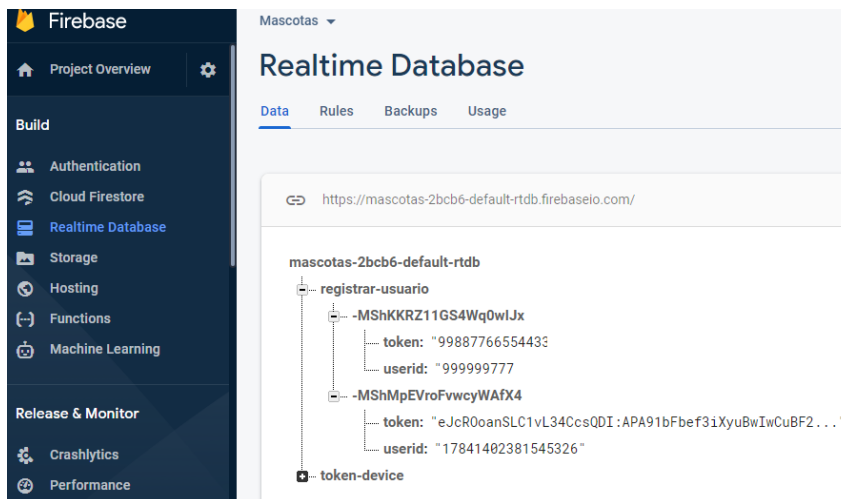


Servidor

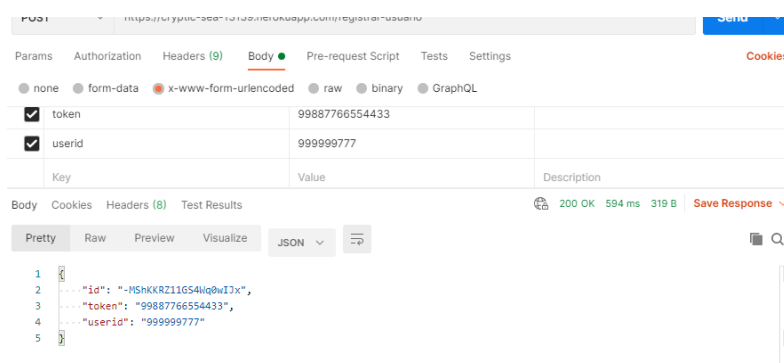
Se uso Heroku con Node.js y se configuro el directorio “registrar-usuario” para recibir el token y el id del usuario y se almacenan en Firebase



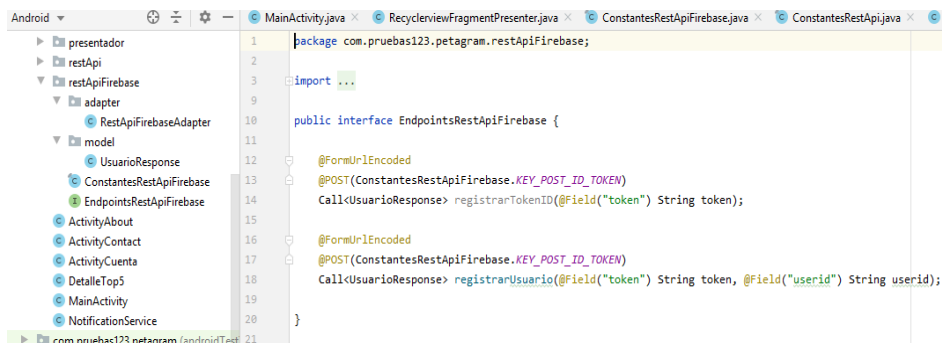
```
65 var db = firebase.database();
66 var registrarUsuarios = db.ref(registrarUsuarioURI).push();
67 registrarUsuarios.set({
68   token: token,
69   userid: userid
70 });
71 var path = registrarUsuarios.toString(); //dir+/+token
72 var pathSplit = path.split(registrarUsuarioURI + "/");
73 var idAutoGenerado = pathSplit[1];
74 var respuesta = generarRespuestaARegistro(db, idAutoGenerado);
75 res.setHeader("Content-Type", "application/json");
76 res.send(JSON.stringify(respuesta));
77 //res.send(req.body.token);
78 });
79
80 function generarRespuestaARegistro(db, idAutoGenerado){
81   var respuesta = {};
82   var usuario = "";
83   var ref = db.ref(registrarUsuarioURI);
84   ref.on("child_added", function(snapshot, prevChildKey) {
85     usuario = snapshot.val();
86     respuesta = {
87       id: idAutoGenerado,
88       token: usuario.token,
89       userid: usuario.userid
90     };
91   });
92   return respuesta;
93 }
```



Con Postman se realizaron las pruebas



En la aplicación se genero el Endpoint “registrarusuario” usando Retrofit



```
1 package com.pruebas123.petagram.restApiFirebase;
2
3 import ...
4
5 public interface EndpointsRestApiFirebase {
6
7     @FormUrlEncoded
8     @POST(ConstantsRestApiFirebase.KEY_POST_ID_TOKEN)
9     Call<UsuarioResponse> registrarTokenID(@Field("token") String token);
10
11     @FormUrlEncoded
12     @POST(ConstantsRestApiFirebase.KEY_POST_ID_TOKEN)
13     Call<UsuarioResponse> registrarUsuario(@Field("token") String token, @Field("userid") String userid);
14 }
15
16
17
18
19
20
21
```

