

Modifique el archivo index.js del ejercicio anterior para establecer el método GET para endpoint “//notificacion/:id” el cual devuelve un ID, un ID de dispositivo y la ID de una foto de instagram. A continuación se ve el archivo index.js

```
index.js
1  const express = require('express')
2  const path = require('path')
3  const PORT = process.env.PORT || 5000
4
5  var bodyParser = require("body-parser")
6
7  var admin = require("firebase-admin");
8
9  var serviceAccount = require("./petagramc-firebase-adminsdk-dmyf7-9c4b728331.json");
10
11  var tokenDevicesURI = "registrar-usuario";
12
13  var FCM = require('fcm-push');
14
15  //inicializa base de datos
16  admin.initializeApp({
17    credential: admin.credential.cert(serviceAccount),
18    databaseURL: "https://petagramc.firebaseio.com"
19  });
20
21  express()
22    .use(express.static(path.join(__dirname, 'public')))
23    .set('views', path.join(__dirname, 'views'))
24    .set('view engine', 'ejs')
25    //https://protected-fortress-77222.herokuapp.com
26    // /android
27    .get('/android', (req, res) => res.render('pages/index'))
28    .use(bodyParser.json()) //Soporte para codificar json
29    .use(bodyParser.urlencoded({extended: true})) //Soporte para decodificar las url
30
31  //POST
32  //https://protected-fortress-77222.herokuapp.com
33  // /registrar-usuario
34  .post("/"+tokenDevicesURI, (req, res) =>
35  {
36    var Database = admin.database();
37
38    var id_dispositivo = req.body.id_dispositivo;
39    var id_usuario = req.body.id_usuario;
40    var id_foto_instagram = req.body.id_foto_instagram;
41
42    var tokenDevices = Database.ref(tokenDevicesURI).push();
43
44    //modelo a insertar en firebase
45    tokenDevices.set({
46      id_dispositivo: id_dispositivo,
47      id_usuario: id_usuario,
48      id_foto_instagram: id_foto_instagram
49    });
50
51    var path = tokenDevices.toString();
52    var pathSplit = path.split(tokenDevicesURI + "/");
53    var idAutoGenerado = pathSplit[1];
54
55    var respuesta = generarRespuesta(Database, idAutoGenerado);
56    res.setHeader("Content-Type", "application/json");
57
58    res.send(JSON.stringify(respuesta));
59  })
```

En la siguiente imagen se puede observar el método GET el cual devuelve la respuesta antes mencionada en formato Json, además, realiza una llamada a la función enviarNotificación, la cual se puede observar en la imagen que se encuentra a continuación de esta, esta función utiliza FCM para generar que el servidor envíe una notificación al dispositivo que tiene configurada la base de datos, además, se configura el título de la notificación, el mensaje, el icono, el color y el sonido con el cual llegara la notificación. Como se utiliza un solo dispositivo para hacer las pruebas, la notificación generada por el servidor se envía al mismo dispositivo que da like a la foto.

```
59 //Get
60 //https://protected-fortress-77222.herokuapp.com
61 .get("/notificacion/:id", (req, res) =>
62 {
63     var id = req.params.id;
64
65     //var db = firebase.database();
66     //var ref = db.ref("registrar-usuario/" + id);
67     var db = admin.database();
68     var ref = db.ref("registrar-usuario/" + id);
69
70     var usuario = "";
71     var respuesta = {};
72
73     ref.on("value", function(snapshot) {
74         console.log(snapshot.val());
75         usuario = snapshot.val();
76         var mensaje = "Dieron me gusta a tu foto";
77         enviarNotificaion(usuario.id_dispositivo, mensaje);
78         respuesta = {
79             id: id,
80             id_dispositivo: usuario.id_dispositivo,
81             id_foto_instagram: usuario.id_foto_instagram
82         };
83         res.send(JSON.stringify(respuesta));
84     }, function (errorObject) {
85         console.log("The read failed: " + errorObject.code);
86         respuesta = {
87             id: "",
88             id_dispositivo: "",
89             id_foto_instagram: ""
90         };
91         res.send(JSON.stringify(respuesta));
92     });
93 }
94 }
```

```

94     })
95     .listen(PORT, () => console.log(`Listening on ${ PORT }`))
96
97
98
99     function enviarNotificaion(tokenDestinatario, mensaje) {
100         var serverKey = 'AAAAcQOKk8:APA91bGALQxon7LL9ZtexpBoz1OFvR1Gg-jwMCjmhyIMXasGwuePpt8RAIVt
101         var fcm = new FCM(serverKey);
102
103         var message = {
104             to: tokenDestinatario, // required
105             collapse_key: '',
106             data: {},
107             notification: {
108                 title: 'Notificacion desde Servidor',
109                 body: mensaje,
110                 icon: "huesoblanco",
111                 sound: "default",
112                 color: "#00BCD4"
113             }
114         };
115
116         fcm.send(message, function(err, response){
117             if (err) {
118                 console.log("Something has gone wrong!");
119             } else {
120                 console.log("Successfully sent with response: ", response);
121             }
122         });
123     }
124
125
126     function generarRespuesta(Database, idAutoGenerado){
127         var respuesta = {}; //JSON

```

```

function generarRespuesta(Database, idAutoGenerado){
    var respuesta = {}; //JSON
    var usuario = "";
    var ref = Database.ref(tokenDevicesURI);
    ref.on("child_added", function(snapshot, prevChildKey){
        usuario = snapshot.val();
        respuesta = {
            id: idAutoGenerado,
            id_dispositivo: usuario.id_dispositivo,
            id_usuario: usuario.id_usuario,
            id_foto_instagram: usuario.id_foto_instagram
        }
    })
    return respuesta;
}

```

En la aplicación de Android Studio se implementó el siguiente endpoint

```

@GET(ConstantesRestApi.KEY_NOTIFICACION)
Call<UsuarioResponse> notificacion (@Path("id") String id);

```

Cuando se da un like a una foto, se llama primero al método POST para guardar en la base de datos los datos de que foto recibió el like y el ID del dispositivo, posteriormente se llama a la función notificaciónFirebase para que el servidor envíe la notificación correspondiente.

```
private void notificaciónFirebase (String Id, String Id_dispositivo) {
    UsuarioResponse usuarioResponse = new UsuarioResponse(Id, Id_dispositivo, id_usuario: "", id_foto_instagram: "");
    RestApiAdapter restApiAdapter = new RestApiAdapter();
    Endpoints endpoints = restApiAdapter.establecerConexionRestApi();
    Call<UsuarioResponse> usuarioResponseCall = endpoints.notificacion(usuarioResponse.getId());
    usuarioResponseCall.enqueue(new Callback<UsuarioResponse>() {
        @Override
        public void onResponse(Call<UsuarioResponse> call, Response<UsuarioResponse> response) {
            UsuarioResponse usuarioResponse1 = response.body();
            Log.d( tag: "ID_FIREBASE_NOTIF", usuarioResponse1.getId());
        }

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

        }
    });
}
```