

# COURSERA

## SEMANA 3

### TAREA 2

#### CREANDO, RECIBIENDO Y ACTUANDO PUSH NOTIFICATIONS

- Se utilizó como servidor Heroku y está desarrollado en Node.js.
- Se agregaron las librerías:
  - body-parser para el procesamiento del request.
  - firebase-admin para la conexión con firebase.
  - fcm-push para el envío de notificaciones.
- Se configuró cada una de las librerías para su respectivo uso.
- En el caso de la librería de firebase-admin se añadió un archivo de configuración generado en la consola de administración de firebase para su acceso.
- Se definió un EndPoint post **registrar-like-foto** que recibe como parámetros:
  - id\_dispositivo: Token único por dispositivo dado por firebase.
  - id\_usuario\_instagram: Id del usuario de Instagram.
  - id\_foto\_instagram: Id de la foto de Instagram.
- Se procesa el request para obtener los parámetros y almacenarlos en variables respectivamente.
- Se obtiene la base de datos del firebase y se hace referencia a **like\_foto\_instagram** usando el método push asignando el resultado a un objeto.
- Luego se setea a ese objeto los respectivos campos del request que fueron almacenados en variables.
- Se realiza un proceso de obtención del id generado cuando se realizó el insert.
- Se crea una nueva función la cual recibe como parámetros la base de datos de firebase y el id generado y mediante un callback generado cuando se realiza el insert en la base de datos de tiempo real obtenemos ese mismo como la respuesta y lo retorna.
- La respuesta del servidor tomara el resultado de la función y se procede a realizar un cast de tipo JSON a string.



```
1 var express = require('express');
2 var app = express();
3
4 app.set('port', (process.env.PORT || 5000));
5
6 var bodyParser = require('body-parser');
7 app.use(bodyParser.json());
8 app.use(bodyParser.urlencoded({ extended: true }));
9
10 var firebaseAdmin = require("firebase-admin");
11 var serviceAccount = require(__dirname + '/Petagram-28676a2475ee.json');
12 firebaseAdmin.initializeApp({
13   credential: firebaseAdmin.credential.cert(serviceAccount),
14   databaseURL: "https://petagram-d2bf9.firebaseio.com"
15 });
16
17 var FCM = require('fcm-push');
18
19 app.use(express.static(__dirname + '/public'));
20
21 // views is directory for all template files
22 app.set('views', __dirname + '/views');
23 app.set('view engine', 'ejs');
24
25 app.get('/android', function(request, response) {
26   response.render('pages/index')
27 });
28
```

```
61
62 var registrarLikeFoto = "registrar-like-foto";
63 app.post('/', registrarLikeFoto, function(request, response) {
64   var id_dispositivo = request.body.id_dispositivo;
65   var id_usuario_instagram = request.body.id_usuario_instagram;
66   var id_foto_instagram = request.body.id_foto_instagram;
67   var db = firebaseAdmin.database();
68   var likeFotoInstagram = db.ref("like_foto_instagram").push();
69   likeFotoInstagram.set({
70     id_dispositivo: id_dispositivo,
71     id_usuario_instagram: id_usuario_instagram,
72     id_foto_instagram: id_foto_instagram
73   });
74   var path = likeFotoInstagram.toString();
75   var pathSplit = path.split("like_foto_instagram/");
76   var idAutoGenerado = pathSplit[1];
77   var respuesta = generarRespuestaRegistrarLikeFoto(db, idAutoGenerado);
78   response.setHeader("Content-Type", "application/json");
79   response.send(JSON.stringify(respuesta));
80 });
81
82 function generarRespuestaRegistrarLikeFoto(db, idAutoGenerado) {
83   var respuesta = {};
84   var usuario = "";
85   var ref = db.ref("like_foto_instagram");
86   ref.on("child_added", function(snapshot, prevChildKey) {
87     usuario = snapshot.val();
88     respuesta = {
89       id: idAutoGenerado,
90       idDevice: usuario.id_dispositivo,
91       idUserInstagram: usuario.id_usuario_instagram,
92       idPhotoInstagram: usuario.id_foto_instagram
93     };
94   });
95   return respuesta;
96 }
97
```

- Luego se procedió a configurar el EndPoint `get /like-foto/:id/:usuario_instagram` con los siguientes parámetros:
  - `id`: Id generado en la base de datos de firebase.
  - `usuario_instagram`: Usuario que da like en las fotos.
- Se realiza el proceso de obtener los parámetros y almacenarlos en variables respectivamente.
- Se obtiene la base de datos del firebase.
- Se hace referencia a `usuario_instagram` (Tarea 1.- Configurando mi primer EndPoint) concatenando el `id` que se envió como parámetro.
- Se declaran dos variables:
  - `usuario`: Obtendrá el resultado de la referencia obtenida anteriormente.
  - `respuesta`: Respuesta del EndPoint.
- Mediante un callback obtenemos la respuesta de la referencia en snapshot donde almacenaremos en una de las variables mencionadas anteriormente.
- Creamos una variable para el mensaje de la notificación.
- Se crea una nueva función la cual recibe como parámetros el id del dispositivo y el mensaje personalizado para el envío de la notificación.
- Mediante la librería `fcm-push` generamos un envío de notificación. Aquí colocamos la llave del servidor que nos provee en la configuración de la consola de firebase, el id del dispositivo y el mensaje.
- El EndPoint genera como respuesta un objeto con el id de la base de datos, id del dispositivo y el id del usuario de Instagram.

```
97
98 app.get("/like-foto/:id/:usuario_instagram", function(request, response){
99   var id = request.params.id;
100   var usuario_instagram = request.params.usuario_instagram;
101   var db = firebaseAdmin.database();
102   var ref = db.ref("usuario_instagram/" + id);
103   var usuario = "";
104   var respuesta = {};
105   ref.on("value", function(snapshot) {
106     console.log(snapshot.val());
107     usuario = snapshot.val();
108     var mensaje = usuario_instagram + " dio like a una de tus fotos.";
109     enviarNotificacion(usuario.id_dispositivo, mensaje);
110     respuesta = {
111       id: id,
112       idDevice: usuario.id_dispositivo,
113       idUserInstagram: usuario.id_usuario_instagram,
114     };
115     response.send(JSON.stringify(respuesta));
116   }, function(errorObject) {
117     console.log("The read failed: " + errorObject.code);
118     respuesta = {
119       id: "",
120       idDevice: "",
121       idUserInstagram: ""
122     };
123     response.send(JSON.stringify(respuesta));
124   });
125 });
```



```
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155

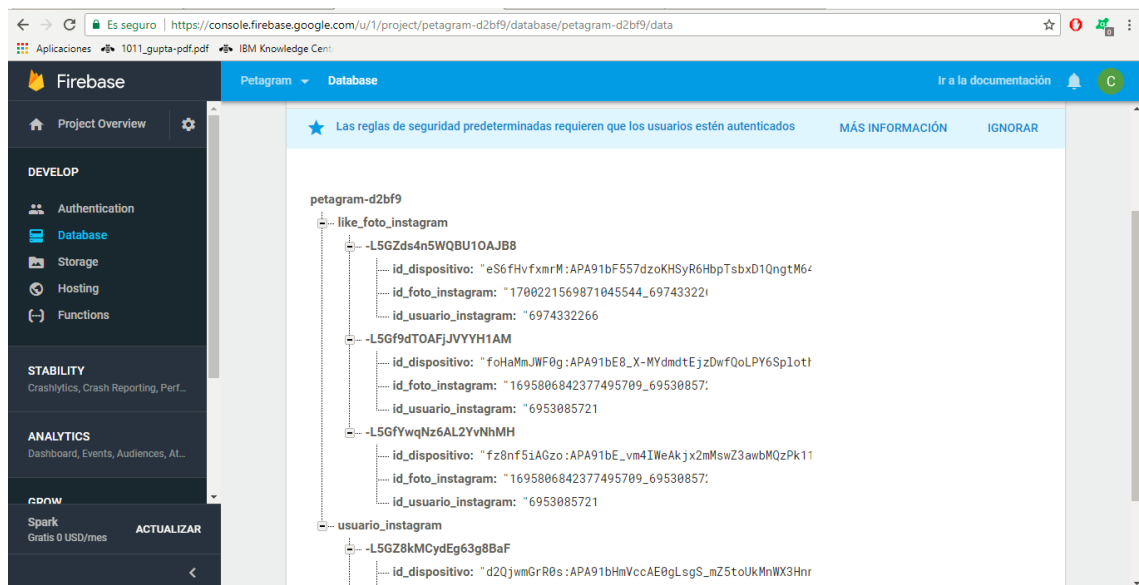
function enviarNotificacion(id_dispositivo, mensaje) {
  var serverKey = 'AIzaSyD3s1HKwCG84AYl2yrrOZAda5DI3ianA4A';
  var fcm = new FCM(serverKey);
  var message = {
    to: id_dispositivo,
    collapse_key: '',
    data: {},
    notification: {
      title: 'Notificacion desde Servidor',
      body: mensaje,
      icon: "notification",
      sound: "default",
      color: "#00BCD4"
    }
  };
};

fcm.send(message, function(err, response){
  if (err) {
    console.log("Something has gone wrong!");
  }
  else {
    console.log("Successfully sent with response: ", response);
  }
});

app.listen(app.get('port'), function() {
  console.log('Node app is running on port', app.get('port'));
});
```

- Luego de realizar la correcta configuración de nuestros EndPoints se procedió a configurar nuestro proyecto Android Petagram para que a través de Retrofit se registre en la base de datos de firebase y se envíen las notificaciones.

## FIREBASE



Es seguro | <https://console.firebase.google.com/u/1/project/petagram-d2bf9/database/petagram-d2bf9/data>

Aplicaciones 1011\_gupta-pdf.pdf IBM Knowledge Cent

**Firestore** Petagram Database Ir a la documentación

Project Overview

**DEVELOP**

- Authentication
- Database**
- Storage
- Hosting
- Functions

**STABILITY**  
Crashlytics, Crash Reporting, Perf...

**ANALYTICS**  
Dashboard, Events, Audiences, At...

**PRICING**  
Spark  
Gratis 0 USD/mes **ACTUALIZAR**

```

-LSGZds4n5WQBU1OAJB8
  id_dispositivo: "eS6fHvfXmrM:APA91bF557dzoKHSyR6HbpTsbxD1QngtM64"
  id_foto_instagram: "1708221569871045544_697433226"
  id_usuario_instagram: "6974332266"
-LSGf9dTOAFJVVYH1AM
  id_dispositivo: "foHaMmJWF0g:APA91bE8_X-MYdmdtEjzDwfQoLPY6Sp1ott"
  id_foto_instagram: "1695806842377495709_695308571"
  id_usuario_instagram: "6953085721"
-LSGfYwqNz6AL2YvNhMH
  id_dispositivo: "fz8nf5iAGzo:APA91bE_vm4IWeAkjx2mMswZ3awbWQzPk11"
  id_foto_instagram: "1695806842377495709_695308571"
  id_usuario_instagram: "6953085721"
-usuario_instagram
  -LSGZ8kMCydEg63g8BaF
    id_dispositivo: "d2QjwmGrR0s:APA91bHmVccAE0gLsgS_mZ5toUkMnWX3Hnr"
    id_usuario_instagram: "6974332266"
  -LSGZSf6ULIr6yv0CRo2
    id_dispositivo: "eS6fHvfXmrM:APA91bF557dzoKHSyR6HbpTsbxD1QngtM64"
    id_usuario_instagram: "6953085721"

```

## PANTALLAZOS DE LA APLICACIÓN

