

# Petagram

## Creando, Recibiendo y Actuando Push Notifications

---

Continuando con los pasos indicados en el video del curso, creo dos nuevos Endpoints en Heroku con Node.js siguiendo los pasos del video de Coursera. Hago la notación que actualmente en Noviembre del 2020, el ApiRest de Instagram, no permite dar like directamente a la cuenta de Instagram, salvo que se sigan los pasos de formalizar y registrar la app, según los requerimientos de FACEBOOK. Por lo que solo quedando registrado en la base de datos creada en Firebase, los datos con la estructura requerida, like a la foto seleccionada, el id de la cuenta de instagram y el Id del dispositivo, también se crea el Endpoint notificación de mensaje enviado desde CM de Google Firebase.

Endpoint que Registra el Like de la Foto:

```
var tokenLikeFotoURI = 'registrar-like'
```

```
app.post('/', + tokenLikeFotoURI, (req, res) =>{
  var id_dispositivo = req.body.id_dispositivo
  var id_usuario_instagram = req.body.id_usuario_instagram
  var id_foto = req.body.id_foto
  var db = firebase.database()

  var tokenLikeFoto = db.ref(tokenLikeFotoURI).push()
  tokenLikeFoto.set({
    id_dispositivo: id_dispositivo,
    id_usuario_instagram: id_usuario_instagram,
    id_foto: id_foto
  })

  var pathRegistro = tokenLikeFoto.toString()
  var idReg = pathRegistro.split(tokenLikeFotoURI+'/')[1]

  var respuesta = generarRespuestaALikeFoto(db,idReg)
  res.setHeader('Content-Type','application/json')
  res.send(JSON.stringify(respuesta))
} )
```

```
function generarRespuestaALikeFoto(db,idReg){
  var respuesta = {}
  //var usuario= ''
  var registro= ''
  var ref = db.ref('registrar-like')
  ref.on('child_added',function(snapshot, prevChildKey){
    registro = snapshot.val()
    respuesta = {
      id: idReg,
      id_dispositivo: registro.id_dispositivo,
      id_usuario_instagram: registro.id_usuario_instagram,
      id_foto: registro.id_foto
    }
  })
  return respuesta
}
```

Configurando la App, para hacer el registro del like de la foto:

```
public static final String KEY_POST_LIKE_FOTO = "registrar-like/";
```

Recibe los parámetros: id\_dispositivo, id\_usuario\_instagram, id foto

Método de la Interfaz implementada:

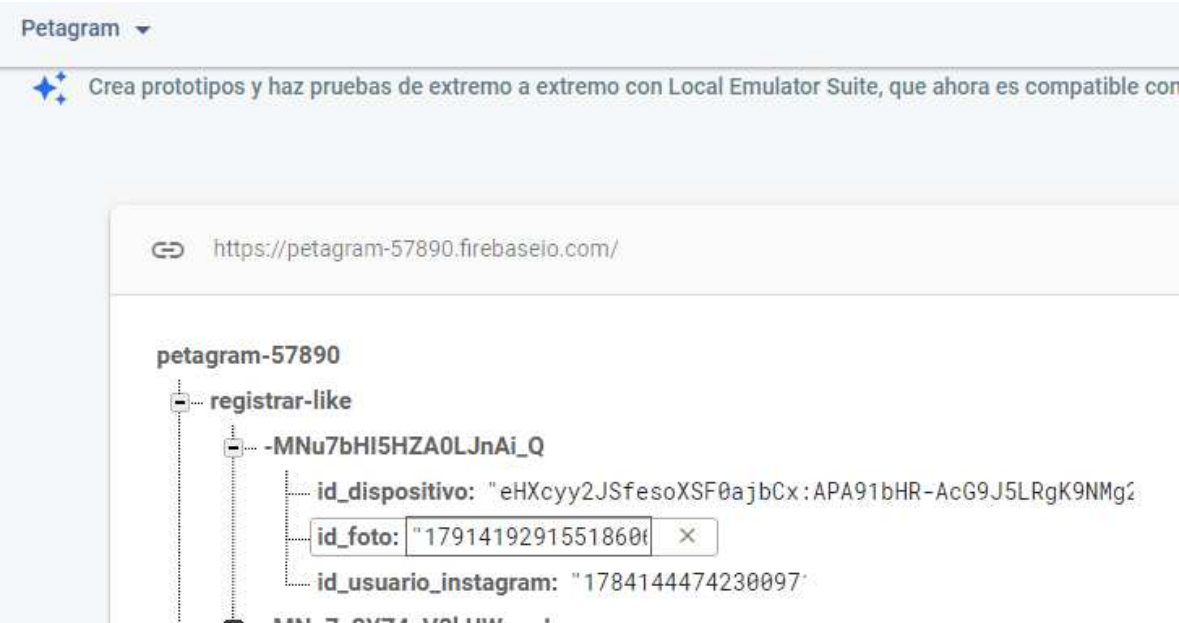
```
@FormUrlEncoded
@POST(ConstantsRestApi.KEY_POST_LIKE_FOTO)
Call<LikeUsuarioResponse> registrarLikeFoto(@Field("id_dispositivo") String id_dispositivo,
                                             @Field("id_usuario_instagram") String id_usuario_instagram,
                                             @Field("id_foto") String id_foto);
```

Para registrar los datos en la base de datos de Firebase:

```
private void EnviarLike(String id_apiRest) {
    Log.d(TAG,id_apiRest);
    RestApiAdapter restApiAdapter = new RestApiAdapter();
    IEndpoinApi iEndpoinApi = restApiAdapter.establecerConexionRestMiApi();
    Call<LikeUsuarioResponse> usuarioResponseCall = iEndpoinApi.registrarLikeFoto(ConstantsRestApi.gId_Dispositivo
    usuarioResponseCall.enqueue(new Callback<LikeUsuarioResponse>() {
        @Override
        public void onResponse(Call<LikeUsuarioResponse> call, Response<LikeUsuarioResponse> response) {
            LikeUsuarioResponse likeUsuarioResponse = response.body();
            Log.d("ID_FIREBASE",likeUsuarioResponse.getId());
            Log.d("Id_dispositivo",likeUsuarioResponse.getId_dispositivo());
            Log.d("Id_usuario_instagram",likeUsuarioResponse.getId_usuario_instagram());
            Toast.makeText(activity,"Diste Like",Toast.LENGTH_LONG).show();
            NotificarLike(likeUsuarioResponse.getId());
        }

        @Override
        public void onFailure(Call<LikeUsuarioResponse> call, Throwable t) {
            //Toast.makeText(this,"Error Notificación",Toast.LENGTH_SHORT).show();
            Log.e("EnviarLike",t.getStackTrace().toString());
        }
    });
}
```

Registro en la base de datos Firebase de Google.



El Método al realizar efectivamente el registro en la base de datos, se pasa al método de Enviar la Notificación al Dispositivo registrado:

NOTA: la prueba se hizo solo con un dispositivo que se tenía disponible. Siguiendo la lógica de lo explicado, el código funciona con otro dispositivo.

Endpoint que hace la Notificación el Like de la Foto al dispositivo vía FCM:

```
var tokenNotificarLikeFotoURI = 'notificar-like/:id'
```

```
app.get('/' + tokenNotificarLikeFotoURI, (req, res) =>{
  var id_firebase = req.params.id
  var db = firebase.database()
  var ref = db.ref('registrar-like/'+id_firebase)
  var usuario = ''
  var respuesta = {}

  ref.on('value',function(snapshot){
    console.log(snapshot.val())
    usuario = snapshot.val()
    var mensaje = 'Te dieron Like'
    enviarNotificacion(usuario.id_dispositivo,mensaje)
    respuesta = {
      id: id_firebase,
      id_dispositivo: usuario.id_dispositivo,
      id_usuario_instagram: usuario.id_usuario_instagram,
      id_foto: usuario.id_foto
    }
    res.send(JSON.stringify(respuesta))

  }, function(errorObject){
    console.log('Error en Like:'+errorObject.code)
    respuesta = {
      id: 'Error',
      id_dispositivo: 'Error',
      id_usuario_instagram: 'Error',
      id_foto: 'Error'
    }
    res.send(JSON.stringify(respuesta))

  })
})
```

```
function enviarNotificacion(id_dispositivo_destino, mensaje){
  var serverKey = 'AAAAcENPGsI:APA91bGhgHhLKjuD6RFUvY4Tr_DSqxJhBAUvntZE5
  var fcm = new FCM(serverKey);

  var message = {
    to: id_dispositivo_destino,
    collapse_key: '',
    data: {
    },
    notification: {
      title: 'Notificación de Petagram App',
      body: mensaje,
      icon: 'hueso',
      sound: 'default',
      color: '#66DDE3'
    }
  };

  fcm.send(message, function(err, response){
    if (err) {
      console.log("¡Algo salió mal!");
    } else {
      console.log("Enviado con éxito con respuesta: ", response);
    }
  });
}
```

Configurando la App, para hacer el envío de la notificación del like de la foto:

```
public static final String KEY_POST_NOTIFICAR_LIKE_FOTO = "notificar-like/{id}";
```

Recibe los parámetros: d foto

Método de la Interfaz implementada:

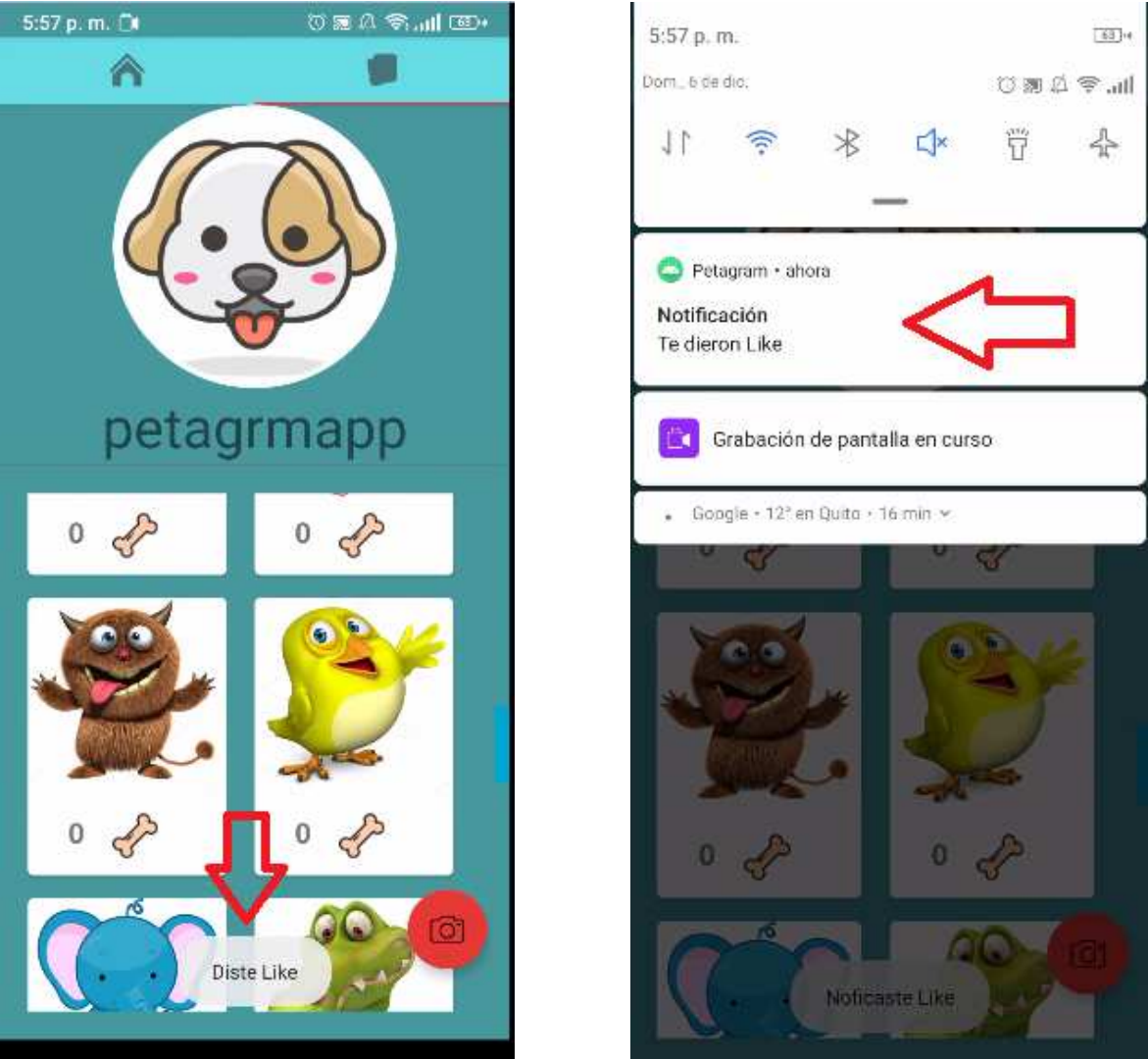
```
@GET(ConstantsRestApi.KEY_POST_NOTIFICAR_LIKE_FOTO)
Call<LikeUsuarioResponse> notificarLikeFoto(@Path(value = "id", encoded = true) String id);
```

Para enviar la notificación al dispositivo:

```
private void NotificarLike(String id) {
    Log.d(TAG,id);
    RestApiAdapter restApiAdapter = new RestApiAdapter();
    IEndpointsApi iEndpointsApi = restApiAdapter.establecerConexionRestMiApi();
    Call<LikeUsuarioResponse> usuarioResponseCall = iEndpointsApi.notificarLikeFoto(id);
    usuarioResponseCall.enqueue(new Callback<LikeUsuarioResponse>() {
        @Override
        public void onResponse(Call<LikeUsuarioResponse> call, Response<LikeUsuarioResponse> response) {
            LikeUsuarioResponse likeUsuarioResponse = response.body();
            Log.d("ID_FIREBASE",likeUsuarioResponse.getId());
            Log.d("Id_dispositivo",likeUsuarioResponse.getId_dispositivo());
            Log.d("Id_usuario_instagram",likeUsuarioResponse.getId_usuario_instagram());
            Toast.makeText(activity,"Noficaste Like",Toast.LENGTH_LONG).show();
        }

        @Override
        public void onFailure(Call<LikeUsuarioResponse> call, Throwable t) {
            //Toast.makeText(this,"Error Notificación",Toast.LENGTH_SHORT).show();
            Log.e("LikeUsuarioResponse",t.getStackTrace().toString());
        }
    });
}
```

Efecto de Marcado de Like y Envío de Notificación



Video del funcionamiento de la App: [https://youtu.be/Pfw-e\\_8D2WQ](https://youtu.be/Pfw-e_8D2WQ)