

Con la ayuda de Heroku implemente una aplicación Node.js y modifique el archivo index.js para establecer el modelo de la respuesta del endpoint “/registrar-usuario” el cual devuelve un ID, un ID de dispositivo y un ID de usuario. A continuación se ve un fragmento del archivo index.js donde se establece la respuesta al endpoint.

```
29     var id_dispositivo = req.body.id_dispositivo;
30     var id_usuario = req.body.id_usuario;
31
32     var tokenDevices = Database.ref(tokenDevicesURI).push();
33
34     tokenDevices.set({
35         id_dispositivo: id_dispositivo,
36         id_usuario: id_usuario
37     });
38
39     var path = tokenDevices.toString();
40     var pathSplit = path.split(tokenDevicesURI + "/");
41     var idAutoGenerado = pathSplit[1];
42
43     var respuesta = generarRespuesta(Database, idAutoGenerado);
44     res.setHeader("Content-Type", "application/json");
45
46     res.send(JSON.stringify(respuesta));
47 }
48 .listen(PORT, () => console.log(`Listening on ${ PORT }`))
49
50
51 function generarRespuesta(Database, idAutoGenerado){
52     var respuesta = {}; //JSON
53     var usuario = "";
54     var ref = Database.ref(tokenDevicesURI);
55     ref.on("child_added", function(snapshot, prevChildKey){
56         usuario = snapshot.val();
57         respuesta = {
58             id: idAutoGenerado,
59             id_dispositivo: usuario.id_dispositivo,
60             id_usuario: usuario.id_usuario
61         }
62     })
63
64     return respuesta;
65 }
```

En mi aplicación de Android Studio defino el endpoint, donde establezco que se va a recibir un id_dispositivo y un id_usuario, los dos de tipo String, tal como se lo definio en el archivo index.js anteriormente.

```
package com.marcelolissa.cardview.restApiFirebase;

import ...

public interface Endpoints {

    @FormUrlEncoded
    @POST(ConstantsRestApi.KEY_POST_ID_TOKEN)
    Call<UsuarioResponse> registrarUsuario(@Field("id_dispositivo") String id_dispositivo, @Field("id_usuario") String id_usuario);

}
```

Cuando se presione sobre el menú Recibir Notificaciones se recibe el token del dispositivo y carga en la base de datos el token del dispositivo junto con el id de usuario que corresponde a la cuenta de Instagram desde la cual se están cargando los datos en la aplicación.

```
case R.id.menNotificaciones:
    String token = FirebaseInstanceId.getInstance().getToken();
    Log.d(TAG, token);
    enviarIdtokenIdUsuario(token);
    break;
```

```
private void enviarIdtokenIdUsuario(String token){
    String USER_ID = "17841444055836399";
    RestApiAdapter restApiAdapter = new RestApiAdapter();
    Endpoints endpoints = restApiAdapter.establishConexionRestApi();
    Call<UsuarioResponse> usuarioResponseCall = endpoints.registrarUsuario(token, USER_ID);

    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: "ID_DISPOSITIVO", usuarioResponse.getId_dispositivo());
            Log.d(tag: "ID_USUARIO", usuarioResponse.getId_usuario());
        }

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

        }
    });
}
```