

Semana 3: Configurando mi propio Endpoint en mi Servidor

URL de proyecto Heroku: <https://guarded-springs-46050.herokuapp.com/>

URI: [registrar-usuario/](#)

Plataforma: [Node.js](#)

Implementacion de Endpoint utilizando NodeJS:

```
var insertarUsuarioURI = 'registrar-usuario';
app.post('/'+insertarUsuarioURI,function(request, response){
  var disp      = request.body.dispositivo;
  var user      = request.body.user;
  var db        = firebase.database();
  var inserUser = db.ref(insertarUsuarioURI).push();
  inserUser.set({
    id_dispositivo:disp,
    id_usuario_instagram:user
  });

  var path      = inserUser.toString();
  var pathSplit = path.split(insertarUsuarioURI);
  var id        = pathSplit[1];
  response.send('Usuario registrado');
});
```

Pruebas:

1. Se realizó una prueba unitaria del servicio antes de implementar la llamada al servicio en android:

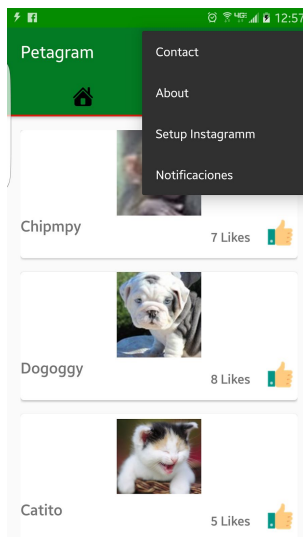
The screenshot shows a web browser with two tabs. The active tab is displaying a REST client interface for testing a POST endpoint. The URL bar shows `https://guarded-springs-46050.herokuapp.com/registrar-usuario`. The interface includes tabs for **POST**, **Params**, **Send**, and **Save**. Below these, there are tabs for **Authorization**, **Headers**, **Body** (selected), **Pre-request Script**, and **Tests**. The **Body** tab shows the request body in **x-www-form-urlencoded** format. It contains two form fields: **dispositivo** with the value `27896` and **user** with the value `lkjksjdd`. Below the form fields, there is a **key** and **value** section. The **Body** tab also shows a **Generate Code** button. Below the **Body** tab, there are tabs for **Body**, **Cookies**, **Headers (8)**, and **Tests**. The **Body** tab shows the response body in **HTML** format, displaying `Usuario registrado`. The status bar at the bottom shows **Status: 200 OK** and **Time: 244 ms**.

Descripcion:

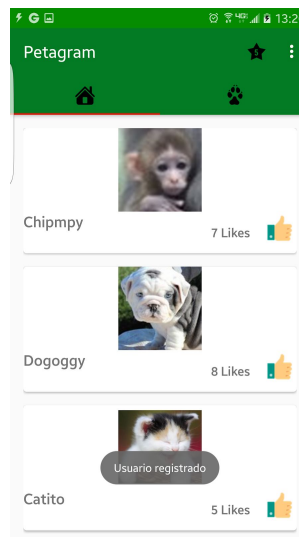
1. Se realizó la implementación del Endpoint de Instragram para obtener el id de usuario utilizado en la sesión anterior.
<https://github.com/flogog/Mascotas/blob/master/CourseraEndpointsS1C4.pdf>
2. Una vez obtenido el id de usuario de instagram se obtiene el id de dispositivo de Firebase:
 - a. `String dispositivoID = FirebaseInstanceId.getInstance().getToken();`
3. Ya que se cuenta con la información que se desea guardar en la base de datos, se realiza la llamada al Endpoint generado:
 - a. `@FormUrlEncoded`
 - b. `@POST(ConstantsRestAPI.HEROKU_POST_USER)`
 - c. `Call<String> registerHerokuUser(@Field("dispositivo") String dispositivo, @Field("user") String user);`

Pantallas:

1. Se da clic en Notificaciones:



2. Se envia la notificacion



3. Los datos se guardan en la base:

```
mascotas-4192a
├── registrar-usuario
│   ├── -KNSq4mmKJC7dL9kk334
│   ├── -KNSq6PJm_4628JPe4j
│   ├── -KNSq7vahzBS7AklM0Z
│   ├── -KNSrAJeUHFNoqpwgRK
│   └── -KNSm8Efr96cqmISWBT
│       ├── id_dispositivo: "fTIOQ4vs1WI:APA91bG"
│       └── id_usuario_instagram: "1367162880"
```