

CONFIGURANDO MI PROPIO ENDPOINT EN MI SERVIDOR

Siguiendo con el desarrollo de la app Petagram, se añadió el Endpoint para tomar datos de la aplicación y guardarlos permanentemente en una base de datos en tiempo real mediante el servicio Firebase. A continuación una breve descripción de lo realizado:

1. Servidor: el servidor se creó de acuerdo a los videos del curso usando el servicio de Heroku con lenguaje Node.js trabajando desde un PC Windows, así que algunos comandos varían un poco. En esta sección se configura el servidor para que tome el ID de Instagram que se ha definido previamente en las actividades anteriores y lo envíe a la base de datos en tiempo real junto con el token autogenerado en cada dispositivo, así:

```
var tokenDevicesURI = "token-device";
app.post('/' + tokenDevicesURI, function(request, response) {
    var token      = request.body.token;
    var animal     = request.body.animal;
    var db = firebase.database();
    var tokenDevices = db.ref(tokenDevicesURI).push();
    tokenDevices.set({
        token: token,
        animal: animal
    });

    var path = tokenDevices.toString();
    var pathSplit = path.split(tokenDevicesURI + "/")
    var idAutoGenerado = pathSplit[1];

    var respuesta = generarRespuestaAToken(db, idAutoGenerado);
    response.setHeader("Content-Type", "application/json");
    response.send(JSON.stringify(respuesta));
});

function generarRespuestaAToken(db, idAutoGenerado) {
    var respuesta = {};
    var usuario = "";
    var ref = db.ref("token-device");
    ref.on("child_added", function(snapshot, prevChildKey) {
        usuario = snapshot.val();
        respuesta = {
            id: idAutoGenerado,
            token: usuario.token,
            animal: usuario.animal
        };
    });
    return respuesta;
}
```

2. Ahora en la aplicación, se incluye un nuevo ítem del menú como se solicita y se incluye el código para que cuando se seleccione, la aplicación envíe el ID de Instagram y el token. Los pasos son prácticamente iguales que los que se muestra en los videos de la clase. El ID de instagram se encuentra guardado previamente en una clase con todas las constantes requeridas, es este valor el que se envía cuando es requerido.

```
Toast.makeText(this, "Datos de usuario enviados" , Toast.LENGTH_SHORT).show();
String token = FirebaseInstanceId.getInstance().getToken();
enviarTokenRegistro(token);
```

```
private void enviarTokenRegistro(String token) {
    Log.d("TOKEN", token);
    RestApiAdapter restApiAdapter = new RestApiAdapter();
    Endpoints endpoints = restApiAdapter.establecerConexionRestAPI();
    Call<UsuarioResponse> usuarioResponseCall = endpoints.registrarTokenID(token,
    ConstantesRestApi.ID_INSTAGRAM);

    usuarioResponseCall.enqueue(new Callback<UsuarioResponse>() {
        @Override
        public void onResponse(Call<UsuarioResponse> call, Response<UsuarioResponse>
        response) {
            UsuarioResponse usuarioResponse = response.body();
            Log.d("ID_FIREBASE", usuarioResponse.getId());
            Log.d("TOKEN_FIREBASE", usuarioResponse.getToken());
        }

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

Screenshots:

