

1. EndPoint registrar usuario

La tecnología que se utilizó para implementar el EndPoint registrar usuario fue NodeJS ya que facilita la creación de este tipo de servicios de manera practica y dinámica.

El servicio que se implementó es el siguiente:

```
//POST Funcion encargada de registrar
// el token de los dispositivos junto con la cuenta instagram.
//https://mascotasapp2016.herokuapp.com/registrar-usuario
//idDispositivo y idUsuarioInstagram
var registrarUsuarioURI = "registrar-usuario"
app.post("/" + registrarUsuarioURI, function(request, response){

    var idDispositivo = request.body.idDispositivo;
    var idUsuarioInstagram = request.body.idUsuarioInstagram;

    var db = firebase.database();
    var registrarUsuario = db.ref(registrarUsuarioURI).push();

    registrarUsuario.set({
        id_dispositivo : idDispositivo,
        id_usuario_instagram: idUsuarioInstagram
    });

    //https://mascotasapp2016.herokuapp.com/registrar-usuario/-KJlTaQQPwP-ssImryV1
    //se obtiene el id del json que contiene la parametrizacion del device
    var path = registrarUsuario.toString();
    var pathSplit = path.split(registrarUsuarioURI + "/")
    var idAutoGenerado = pathSplit[1];

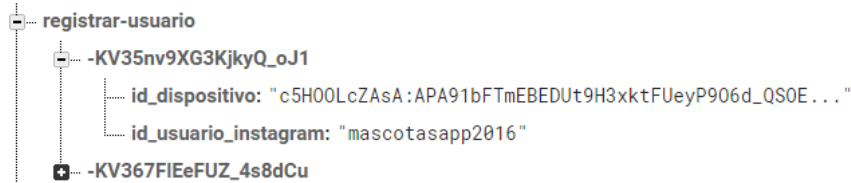
    var respuesta = generarRespuestaServicioRegistro(db, idAutoGenerado);
    response.setHeader("Content-Type", "application/json");
    response.send(JSON.stringify(respuesta));

})
```

Explicación: El objetivo principal de la función consiste en registrar el dispositivo y la cuenta Instagram configurada en la aplicación Mascotas en la base de datos FireBase. Este recibe como parámetros de entrada el id del dispositivo y el id usuario Instagram. Una vez obtiene estos valores el servicio se conecta a firebase y los registra.

A continuación se evidencia el registro del mismo.

mascotasapp2016-7210c



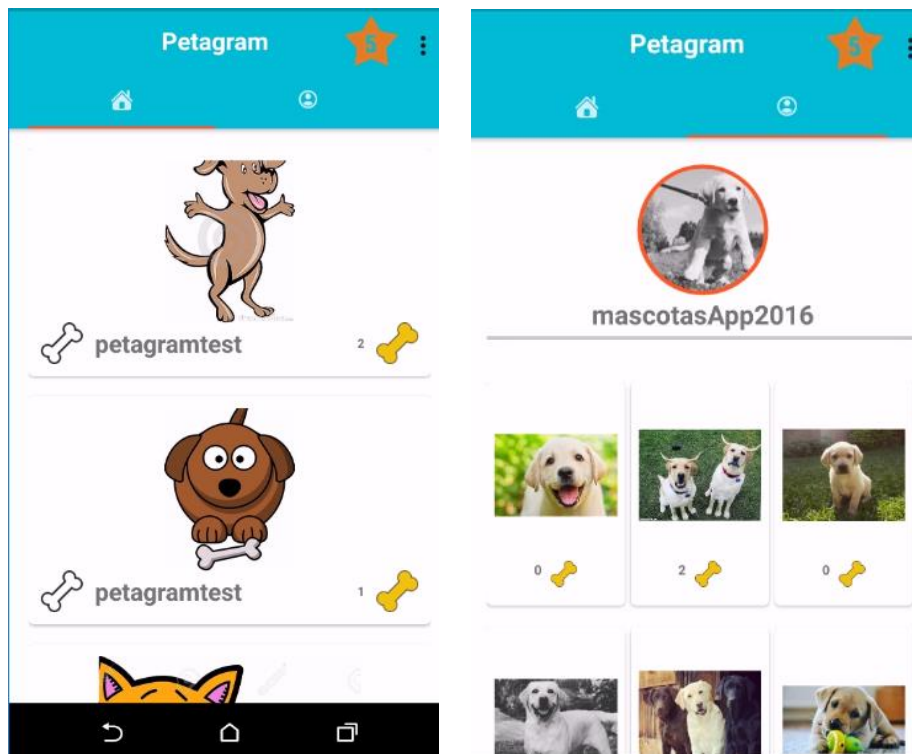
Al finalizar el registro, el servicio genera una respuesta retornando el id del registro Firebase, el id del dispositivo y el id usuario Instagram.

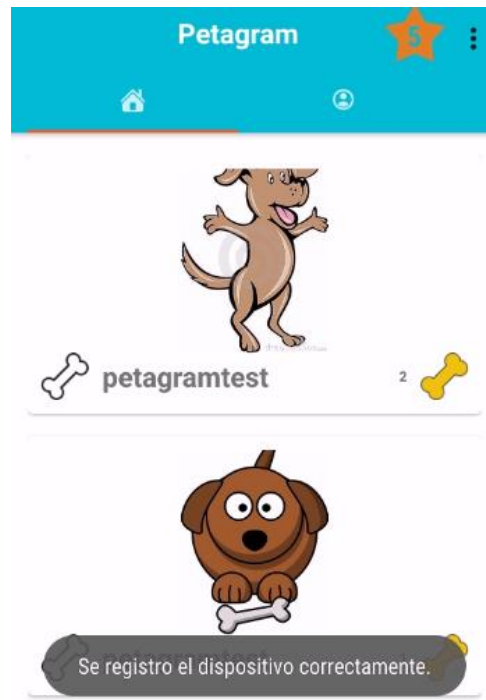
La aplicación Mascotas obtiene la respuesta del servicio registrar usuario y la registra en el log a modo informativo. Adicionalmente, si el proceso fue exitoso notifica al usuario que el dispositivo se registro correctamente o caso contrario informa que se presentó algún error.

```
usuarioDeviceResponseCall.enqueue(new Callback<UsuarioDeviceResponse>() {
    @Override
    public void onResponse(Call<UsuarioDeviceResponse> call, Response<UsuarioDeviceResponse> response) {
        UsuarioDeviceResponse usuarioDeviceResponse = response.body();
        Log.d("ID_FIREBASE", usuarioDeviceResponse.getId());
        Log.d("IDDISPOSITIVO_FIREBASE", usuarioDeviceResponse.getIdDispositivo());
        Log.d("IDUSU_INSTA_FIREBASE", usuarioDeviceResponse.getIdUsuarioInstagram());
        Toast.makeText(context, "Se registro el dispositivo correctamente.", Toast.LENGTH_SHORT).show();
    }

    @Override
    public void onFailure(Call<UsuarioDeviceResponse> call, Throwable t) {
        Toast.makeText(context, "Ocurrió un error en la conexión.", Toast.LENGTH_SHORT).show();
        Log.e("REGISTRAR_USUARIO", "Ocurrió un error al registrar el usuario. " + t.toString());
    }
});
```

2. Evidencias Funcionamiento





mascotasapp2016-7210c

registrar-usuario

-KV7odYYk Jr3idimaFPe

id_dispositivo: "fTUxet50vA8:APA91bGLnb2X_7eqFs1oJH4J3qEs_j8fIWS..."
id_usuario_instagram: "mascotasapp2016"