

En la clase de Main Activity se encuentra un switch que determina la acción que realiza la aplicación, en este caso si en el menú de opciones elegimos la opción de Recibir Notificaciones , se activa el método de enviarToken().

```
@Override
public boolean onOptionsItemSelected(MenuItem item) {

    switch(item.getItemId())
    {
        case R.id.mContacto:
            Intent intentContacto = new Intent(getApplicationContext() , FormularioActivity.class);
            startActivity(intentContacto);
            break;
        case R.id.mFavoritos:
            Intent intentFavorito = new Intent(getApplicationContext() , MascotasFavoritas.class);
            startActivity(intentFavorito);
            break;
        case R.id.mAcercaDe:
            Intent intentAcercaDe = new Intent(getApplicationContext() , AcercaDeActivity.class);
            startActivity(intentAcercaDe);
            break;
        case R.id.mConfigurarCuenta:
            Intent intentConfigurarCuenta = new Intent(getApplicationContext() , ConfigurarCuentaActivity.class);
            startActivity(intentConfigurarCuenta);
            break;
        case R.id.mRecibirNotificaciones:
            //NO GENERA INTENT, SOLO GENERA LOS ENVIOS A GCM
            enviarToken();
            break;
    }
}
```

Aquí esta el funcionamiento del método de enviarToken que a su vez activa el método de enviarTokenRegistro.

```
public void enviarToken(){
    String token = FirebaseInstanceId.getInstance().getToken();
    enviarTokenRegistro(token);
}

private void enviarTokenRegistro(String token){
    Log.d("TOKEN", token);
    RestApiAdapter restApiAdapter = new RestApiAdapter();
    EndpointsApi endpointsApi = restApiAdapter.establecerConexionRestApi();
    Call<UsuarioResponse> usuarioResponseCall = endpointsApi.registrarTokenID(token);

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

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

        }
    });
}
```

Estos son los endpoints que se usaron.

```
.ic interface EndpointsApi {

    @GET(ConstantsRestApi.URL_GET_RECENT_MEDIA_USER)
    Call<MascotaResponse> getRecentMedia();

    @GET("users/search?access_token=6216568161.c002059.032bb9dc8fe84b62ad3e36566348d490")
    Call<MascotaResponse> getUser(@Query("q") String username2);

    @GET(ConstantsRestApi.KEY_USERS + "{idUsuario}" + ConstantsRestApi.KEY_GET_RECENT_MEDIA)
    Call<MascotaResponse> getMediaUser(@Path("idUsuario") String idUsuario2);

    @FormUrlEncoded
    @POST(ConstantsRestApi.KEY_POST_ID_TOKEN)
    Call<UsuarioResponse> registrarTokenID(@Field("token")String token);

    @FormUrlEncoded
    @POST(ConstantsRestApi.KEY_POST_REGISTRO_USUARIO)
    Call<UsuarioResponse> registrarUsuario(@Field("id_dispositivo")String id_dispositivo , @Field("id_usuario_instagram"
```

Este es el documento de Index.Json que muestra la configuración de los Webservices:

```
//Post
//https://gentle-hollows-16128.herokuapp.com/token-device
//token
var registroUsuario = "token-device";
app.post("/", registroUsuario, function(request, response){
  var idDispositivo = request.body.token;
  var idUsuario = request.body.userID;

  var db = firebase.database();
  var tokenDevices = db.ref(registroUsuario).push();
  tokenDevices.set({
    idDispositivo : idDispositivo,
    idUsuario: idUsuario
  });

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

  var respuesta = generarRespuestaAToken(db, idDispositivo, idUsuario);
  response.setHeader("Content-Type", "application/json");

  response.send(JSON.stringify(respuesta));
});

function generarRespuestaAToken(db, tokenDevice, usuarioInstagram){
  var respuesta = {};
  var usuario = "";
  var ref = db.ref("token-device");
  ref.on("child-added", function(snapshot, prevChildKey){
    usuario = snapshot.val();
    respuesta = {
      InstagramID: usuarioInstagram,
      token: tokenDevice
    };
  });
  return respuesta;
}
```