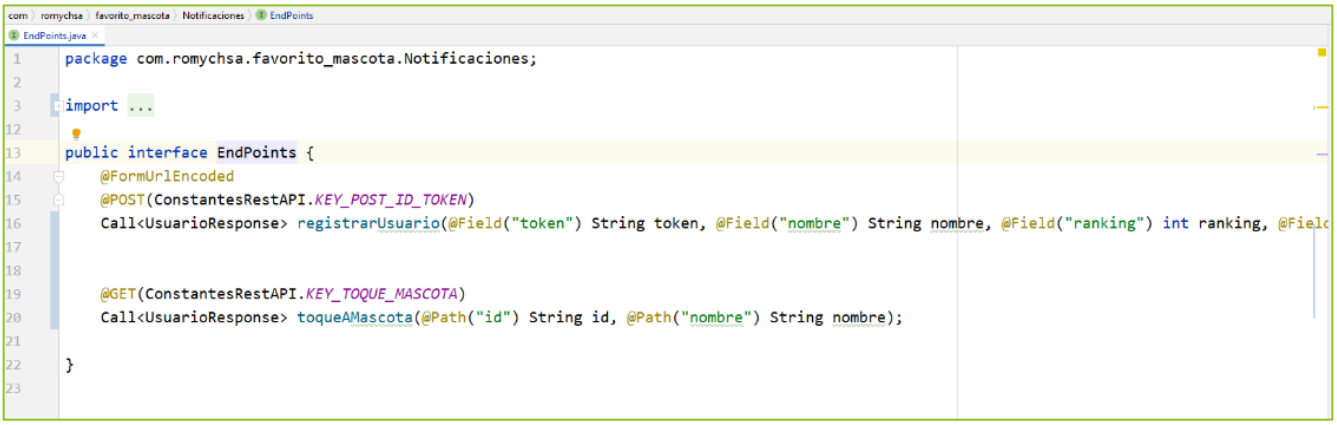
Notificaciones Aplicación “Mascota Favorita”

1) EndPoint

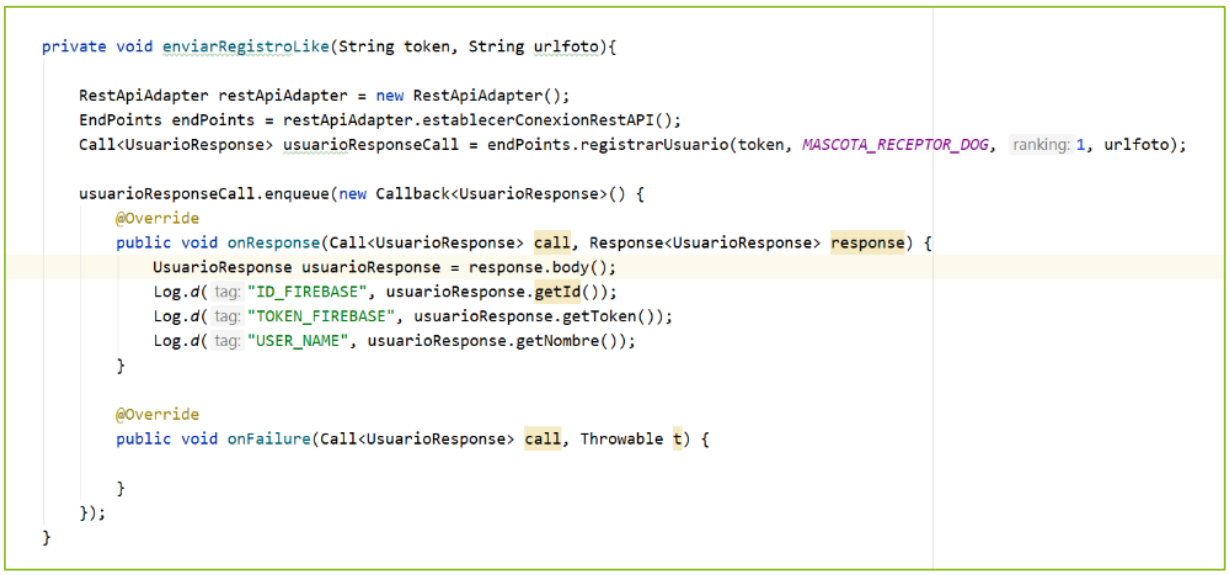
- Modifique el endpoint “registrarUusario” para que registre en Firebase el token, nombre del usuario, el número de likes y la URL de la foto.
- Agregue el endpoint “toqueMascota” que envía el id del dispositivo del receptor y nombre de usuario del emisor del like.



```java
package com.romychsa.favorito_mascota.Notificaciones;

import ...

public interface EndPoints {
    @FormUrlEncoded
    @POST(ConstantesRestAPI.KEY_POST_ID_TOKEN)
    Call<UsuarioResponse> registrarUsuario(@Field("token") String token, @Field("nombre") String nombre, @Field("ranking") int ranking, @Field

    @GET(ConstantesRestAPI.KEY_TOQUE_MASCOTA)
    Call<UsuarioResponse> toqueAMascota(@Path("id") String id, @Path("nombre") String nombre);

}
```

2) MainActivity

- Se modifico el método “enviarRegistroLike()” para que envíe adicionalmente el like y la URL de la foto likeada, así como el nombre del usuario al que se le dio like.



```java
private void enviarRegistroLike(String token, String urlfoto){

    RestApiAdapter restApiAdapter = new RestApiAdapter();
    EndPoints endPoints = restApiAdapter.establecerConexionRestAPI();
    Call<UsuarioResponse> usuarioResponseCall = endPoints.registrarUsuario(token, MASCOTA_RECEPTOR_DOG,  ranking: 1, urlfoto);

    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: "TOKEN_FIREBASE", usuarioResponse.getToken());
            Log.d( tag: "USER_NAME", usuarioResponse.getNombre());
        }

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

        }
    });
}
```

- Se modifico el método “lanzarNotificacion()” para que reciba un String “urlfoto” y lo envie junto con el token generado al método “enviarRegistroLike()”.

```
public void lanzarNotificacion(String urlfoto){
    FirebaseInstanceId.getInstance().getInstanceId()
            .addOnCompleteListener(new OnCompleteListener<InstanceIdResult>() {
                @Override
                public void onComplete(@NonNull Task<InstanceIdResult> task) {
                    if (!task.isSuccessful()) {
                        Log.w(TAG,  msg: "getInstanceId failed", task.getException());
                        return;
                    }
                    // Get new Instance ID token
                    String token = task.getResult().getToken();

                    // Log and toast

                    String msg = getString(R.string.msg_token_fmt, token);
                    Log.d(TAG, msg);
                    Toast.makeText(MainActivity.this, msg, Toast.LENGTH_SHORT).show();


                    enviarRegistroLike(token, urlfoto);

                }
            });
}
```

- Se crea el método “toqueDOG(View v)” donde el emisor de la notificación será el gato (instagram: estrella.chsa)y el receptor de la notificación es el perro (instagram: anmy.dev).

```
public void toqueDOG(View v){
    Log.d( tag: "TOQUE_MASCOTA",  msg: "true");
    final UsuarioResponse usuarioResponse = new UsuarioResponse(ID_DISPOSITIVO_DOG,  token: "123", MASCOTA_RECEPTOR_DOG);
    RestApiAdapter restApiAdapter = new RestApiAdapter();
    EndPoints endPoints = restApiAdapter.establecerConexionRestAPI();
    Call<UsuarioResponse> usuarioResponseCall = endPoints.toqueAMascota(usuarioResponse.getId(), MASCOTA_EMISOR_CAT);
    usuarioResponseCall.enqueue(new Callback<UsuarioResponse>() {
        @Override
        public void onResponse(Call<UsuarioResponse> call, Response<UsuarioResponse> response) {
            UsuarioResponse usuarioResponse1 = response.body();
            Log.d( tag: "ID_FIREBASE", usuarioResponse1.getId());
            Log.d( tag: "TOKEN_FIREBASE", usuarioResponse1.getToken());
            Log.d( tag: "ANIMAL_FIREBASE", usuarioResponse1.getNombre());
        }

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

        }
    });

}
```

3) MascotaAdaptador

- En el método “onBindViewHolder()” aplicaque un método onclick para la imagen icono de un hueso. La acción que realizará será aumentar el like en 1 y pasará la url de la foto likeada al método “lanzarNotificacion()” de MainActivity, así como que iniciara el método “toqueDOG()”.

```java
@Override
public void onBindViewHolder(@NonNull final MascotaViewHolder mascotaViewHolder, int position) {
    //Se agarra un arraylist de objetos mascota y de cada uno se saca la foto y nombre.
    final Mascota mascota = mascotas.get(position);
    Picasso.get () Picasso
            .load (mascota.getFoto()) RequestCreator
            .placeholder (R.drawable.ic_action_dog)
            .error(R.drawable.ic_action_dog)
            .into(mascotaViewHolder.imgMascota);
    mascotaViewHolder.tvNombre.setText(mascota.getNombre());
    mascotaViewHolder.tvRanking.setText(String.valueOf(mascota.getRanking()));
    MainActivity.mascotas = mascotas;
    mascotaViewHolder.btnHuesoblanco.setOnClickListener(new View.OnClickListener() {

        @Override
        public void onClick(View v) {

            Toast.makeText(activity, text: "Me gusta " + mascota.getNombre(), Toast.LENGTH_SHORT).show();
            int ranking = mascota.getRanking() +1;
            String urlfoto = mascota.getFoto();
            mascotaViewHolder.tvRanking.setText(String.valueOf(ranking));

            MainActivity mainActivity = new MainActivity();
            mainActivity.lanzarNotificacion(urlfoto);
            mainActivity.toqueDOG(v);


        }
    });
```

4) Layout Cardview_mascotas

- Se crea el ImageButton.

```xml
<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:orientation="horizontal"
    >

    <ImageButton
        android:id="@+id/btnHuesoblanco"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:background="@mipmap/huesoblanco"
        android:layout_gravity="center"

        />
```

5) Imagen de los dispositivos