PROYECTO #2 DE LA SEMANA 3 PARA CURSO 4 DESARROLLO DE APLICACIONES VANZADAS CON ANDROID UNIVERSIDAD NACIONAL AUTONOMA DE MEXICO

A continuación la implementación del POST para registrar un like hacia Instagram, pero también guardando el like en la base de datos de Firebase y enviando la notificación al dispositivo ya registrado.

Como explicación, los dispositivos que se "emparejan", tienen un usuario principal de Instagram y otro usuario para el Timeline. En el timeline es donde se puede dar like. La idea es que en el dispositivo1 la cuenta principal sea del usuario_instagram1 y la del timeline la del usuario_instagram2. En el dispositivo2 la cuenta principal es la del usuario_instagram2 y la del timeline es la del usuario_instagram1.

Cuando en el dispositivo1 se de like a una foto del timeline (perteneciente al usuario_instagram2), se registra el like pero se envía una notificación al dispositivo2 (que tiene como cuenta principal la cuenta usuario_instagram2. Los mismo sucede (pero a la inversa), en el dispositivo2.

A continuación el código del POST en Android (uno para Instagram y otro para Firebase).

LIKE EN INSTAGRAM.

En la interface se agrega:

```
//para el POST del LIKE
@FormUrlEncoded
@POST(ConstantesRestApi.URL_POST_LIKE)
Call<LikeResponse> postLikeInstagram(@Path("media-id") String
id_media_instagram,@Field("access_token") String access_token);
```

Se agregaron estas constantes:

```
public static final String META                          = "meta";
public static final String CODIGO                        = "code";
public static final String TIPO_ERROR                    = "error_type";
public static final String MENSAJE_ERROR                 = "error_message";
public static final int    CODIGO_OK                     = 200;
```

Y como deserializador se revisa el código que se regresa, si es diferente de 200 hubo error.

```
public class LikeDeserializador implements JsonDeserializer<LikeResponse> {
    @Override
    public LikeResponse deserialize(JsonElement json, Type typeOfT,
JsonDeserializationContext context) throws JsonParseException {
        Gson gson = new Gson();
        int codigo;
        String tipo_error=null;
        String mensaje_error=null;
        LikeResponse likeResponse = gson.fromJson(json,LikeResponse.class);
        JsonObject likeResponseData = json.getAsJsonObject();
        JsonObject likecodigo       =
likeResponseData.getAsJsonObject(JsonKeys.META);
        codigo = likecodigo.get(JsonKeys.CODIGO).getAsInt();
```

```
        if (codigo!=JsonKeys.CODIGO_OK) {
            tipo_error    = likecodigo.get(JsonKeys.TIPO_ERROR).getAsString();
            mensaje_error = likecodigo.get(JsonKeys.MENSAJE_ERROR).getAsString();
        }

        LikeResponse respuestaLike = new LikeResponse();
        respuestaLike.setCodigo(codigo);
        respuestaLike.setTipo_error(tipo_error);
        respuestaLike.setMensaje_error(mensaje_error);
        return respuestaLike;
    }
}
```

A continuación el código cuando se da clic al botón de like en la visualización del TimeLine:

PRIMERA SECCION: POST REGISTRAR LIKE EN INSTAGRAM: Se revisa que el código que regresa Instagram sea el 200, si no es ese código o no se revise respuesta se genera un error.

```
mediaInstagramViewHolder.btLike05.setOnClickListener(new View.OnClickListener(){
    @Override
    public void onClick(View v) {
        Toast.makeText(activity,"Has dado like a "+publicacion.getMedia_id()+"
del usuario "+publicacion.getId(),Toast.LENGTH_SHORT).show();
        //aqui va el RESTFUL POST de like hacia instagram
        RestApiAdapter restApiAdapter = new RestApiAdapter();
        Gson gsonLikeMedia = restApiAdapter.construyeGsonDeserializadorDarLike();
        IEndpointsApi endpointsApi =
restApiAdapter.establecerConexionRestApiInstagram(gsonLikeMedia);
        final Call<LikeResponse> likeResponseCall =
endpointsApi.postLikeInstagram(publicacion.getMedia_id(),
ConstantesRestApi.ACCESS_TOKEN);

        likeResponseCall.enqueue(new Callback<LikeResponse>() {
            @Override
            public void onResponse(Call<LikeResponse> call,
Response<LikeResponse> response) {
                LikeResponse informacionrespuestalike = response.body();
                if (informacionrespuestalike!=null) {
                    if (informacionrespuestalike.getCodigo()!=
JsonKeys.CODIGO_OK) {
                        Log.e("ERROR_DE_LIKE","Codigo:
"+informacionrespuestalike.getCodigo());
                        Log.e("TIPO_ERROR
LIKE",informacionrespuestalike.getTipo_error());

Log.e("MENSAJE_ERROR_LIKE",informacionrespuestalike.getMensaje_error());
```

SEGUNDA SECCION: POST REGISTRAR LIKE EN FIREBASE Y ENVIAR NOTIFICACION: Se revisa que el código que regresa Instagram sea el 200, si es ese código se llama el POST hacia Heroku para registrar en la base de datos el like y mandar la notificación.

```
                    } else {
                        //aqui debo de insertar el like en la base de datos de
firebase y programar la notificacion.
                        // 1) Mandamos con un POST a Heroku, en la respuesta
viene el id_dispositivo que recibira
                        //    la notificacion desde FCM por el like dado a su
media (en caso este registrado algun dispositivo de ese id instagram).

                        //Implementacion:
                        RestApiHerokuAdapter restApiHerokuAdapter = new
RestApiHerokuAdapter();
                        IEndpointsHeroku iEndpointsHeroku =
restApiHerokuAdapter.establecerConexionRestAPIHeroku();
                        Call<RegistraHerokuLikeResponse>
registraHerokuLikeResponseCall =
iEndpointsHeroku.registrarLike(publicacion.getId(),
                                publicacion.getMedia_id(),
MainActivity.idUsuarioInstagram);

                        registraHerokuLikeResponseCall.enqueue(new
Callback<RegistraHerokuLikeResponse>() {

                            @Override
                            public void
onResponse(Call<RegistraHerokuLikeResponse> call,
Response<RegistraHerokuLikeResponse> response) {
                                RegistraHerokuLikeResponse
registraHerokuLikeResponse = response.body();
                                Log.d("id
FIREBASE",registraHerokuLikeResponse.getId());
                                Log.d("dispositivo
FIREBASE",registraHerokuLikeResponse.getId_dispositivo()+"");
                                Log.d("dueño
media",registraHerokuLikeResponse.getId_owner_instagram());
                                Log.d("ID
media",registraHerokuLikeResponse.getId_media_instagram());
                                Log.d("ID usuario
LIKE",registraHerokuLikeResponse.getId_sender_instagram());
                            }

                            @Override

                            public void
onFailure(Call<RegistraHerokuLikeResponse> call, Throwable t) {
                                Log.e("ERROR",t.toString());
                            }

                        });
                    }

                } else { //el cuerpo de la respuesta venia vacio (like a
instagram).
                    if (response.errorBody()!= null) {
                        Log.e("Error en Like",response.errorBody().toString());
                    }
```

```
                }
            }

            @Override
            public void onFailure(Call<LikeResponse> call, Throwable t) {
                Log.e("FALLO LA CONEXION2", t.toString());
            }
        });

    }

});

}
```

A continuación veremos en detalle el código de este POST en la aplicación en Heroku.

PRIMERA SECCION: LIBRERIAS NECESARIAS: Se modifica el uso de firebase a firebase-admin para evitar mensajes de advertencia que salían en la bitácora de Heroku por usar solamente firebase. También se incluye la librería de FCM que es la que envía la notificación.

```
var firebase = require("firebase-admin");
var serviceAccount = require("./PetragramRCENodejs-
6812688ab89d.json");

firebase.initializeApp({
  credential: firebase.credential.cert(serviceAccount),
databaseURL: "https://petragramrcenodejs.firebaseio.com"
});


var FCM = require('fcm-push');  //habilitar el FireBase Cloud
Messagging push notification.
```

```
//POST para registrar un like desde el TimeLine de la aplicacion
(manda id owner, id media y id sender).
// el owner es el id de instagram del dueÃ±o de la cuenta de
instagram donde pertenece la media.
// el id media es el identificador de isntagram para la media a la
que se le dio like en el TimeLine.
// el id sender es el id de instagram definido como principal en
la aplicacion.
// este id owner debe estar registrado en registrar-usuario para
poder enviar notificacion.

//https://whispering-cliffs-37590.herokuapp.com/registrar-like
//id_owner_instagram, id_media_instagram,id_sender_instagram
```

```
//IMPORTANTE:
// dentro de este codigo se hace una llamada a un GET de Firebase
que regresa el nodo completo
// de registrar-usuario. A traves de parser de JSON se revisa si
existe un dispositivo registrado
// para el id_owner_instagram.
// en caso de existir se regresa este valor en la respuesta del
POST en el campo id_dispositivo.


var registrarlikeURI = "registrar-like";
app.post("/" + registrarlikeURI, function(request,response) {
     var id_owner_instagram     = request.body.id_owner_instagram;
     var id_media_instagram     = request.body.id_media_instagram;
     var id_sender_instagram    =
request.body.id_sender_instagram;
```

SEGUNDA SECCION: OBTENER EL IDENTIFICADOR DEL DISPOSITIVO CUYA CUENTA PRINCIPAL ESTA ASOCIADA AL ID DE USUARIO DE INSTAGRAM EN EL TIMELINE: revisando documentación de Firebase se encontró que haciendo un GET a la ruta de un nodo en la base de datos y finalizando la llamada con .json, se puede obtener el JSON de la entrada en la base de datos. Obtenemos todos los dispositivos registrados y buscamos el ID del usuario al que queremos enviar notificación, si encontramos dicho ID, obtenemos el ID del dispositivo registrado.

```
     //tratando de obtener registrar-usuario con un GET finalizado
en .json desde aca.
     var URL_usuarios =
"https://petragramrcenodejs.firebaseio.com/registrar-
usuario.json";
     //'use strict';
     var registrados = require('request');
     var id_dispositivo_recuperado= null;
     var get_finalizado = 0;
     registrados.get({
     url: URL_usuarios,
     json: true,
     headers: {'User-Agent': 'request'}
     }, (err, res, data) => {
     if (err) {
               console.log('Error:', err);
     } else if (res.statusCode !== 200) {
               console.log('Status:', res.statusCode);
     } else {
               // data is already parsed as JSON:
          console.log("dentro de sin error ni codigo != 200");
               //console.log("Body: ",data); //aca venia el Json
               console.log("vamos a recorrer cada nodo que
regreso el GET");
               for (var dato_actual in data) {
```

```
                        var id_dispositivo_actual =
data[dato_actual]["id_dispositivo"];
                        var id_usuario_instagram_actual =
data[dato_actual]["id_usuario_instagram"]
                        console.log(id_dispositivo_actual);
                        console.log(id_usuario_instagram_actual);
                        if (!id_dispositivo_recuperado ||
id_dispositivo_recuperado.length===0) {
                              console.log("Vamos a comparar
"+id_usuario_instagram_actual+" con "+id_owner_instagram);
                              if
(id_usuario_instagram_actual===id_owner_instagram) {
                                    console.log("Eran iguales");
                                    id_dispositivo_recuperado =
id_dispositivo_actual;
                              }
                        }
                  }
      }
      console.log("registrados.");
      get_finalizado = 1;
      //insertando en FireBase el Like.
            var db = firebase.database();
            console.log("Vamos a tener acceso a "+db.ref());
            //while (!id_dispositivo_recuperado &&
get_finalizado===0) { //esperar a tener un valor o salir GET.

            //}
            console.log("Vengo con
dispositivo"+id_dispositivo_recuperado+" Y
get_finalizado"+get_finalizado);
            var registro = db.ref(registrarlikeURI).push();
            console.log("A obtener la llave");
            llave = registro.key;
            console.log("Antes del set de registrar-like");
            registro.set({
                  id_owner_instagram  : id_owner_instagram,
                  id_media_instagram  : id_media_instagram,
                  id_sender_instagram : id_sender_instagram
            });
            console.log("Se empujo la llave "+llave);
            //enviando respuesta
            var respuesta = {};
            var usuario = "";
            var ref1 = db.ref(registrarlikeURI);
            ref1.on("child_added", function(snapshot, prevChildKey)
{
                  usuario = snapshot.val();
                  respuesta = {
                        id: llave,
                        id_dispositivo : id_dispositivo_recuperado,
```

```
                        id_owner_instagram :
usuario.id_owner_instagram,
                        id_media_instagram :
usuario.id_media_instagram,
                        id_sender_instagram : id_sender_instagram
                    };

            });
            var mensaje = "el usuario con id "+id_sender_instagram+"
dio like a tu media con id "+id_media_instagram;
            enviarNotificaion(id_dispositivo_recuperado, mensaje);

            response.setHeader("Content-Type", "application/json");
            response.send(JSON.stringify(respuesta));

      });

      }
);
```

TERCERA SECCION: ENVIAR NOTIFICACION: Si en la sección anterior obtenemos un ID de dispositivo enviamos la notificación al mismo.

```
function enviarNotificaion(tokenDestinatario, mensaje) {
      var serverKey =
'AAAABX3oZWQ:APA91bFwTmwRQJmDVPlRvra3IBrfcbtWZGW1EbvLpSYHSsy1rQXke
U0bGCk7tlDBJoiE_Yg5nBjo2PTtNApQ899joOu8Kz-
PioDpZB_9gp_aL8m0kUtl1D72ZThzu1S_vB94qwElnu0k';
      var fcm = new FCM(serverKey);
      var message = {
          to: tokenDestinatario, // required
          collapse_key: '',
          data: {},
          notification: {
              title: 'Notificacion desde Servidor',
              body: mensaje,
              icon: "notificacion",
              sound: "default",
              color: "#00BCD4"
          }
      };

      fcm.send(message, function(err, response){
          if (err) {
              console.log("Something has gone wrong!");
          } else {
              console.log("Successfully sent with response: ",
response);
          }
      });
```

```
}
```

Fue complicado el tema de revisar si en la base de datos ya existía un dispositivo asociado al ID de usuario de Instagram a cuya media dimos like. Pero consultando documentación se logró. La idea de hacerlo de esta manera era que no tuviéramos que poner esto directamente en el código (hardcodeado). Cabe mencionar que si existen varios dispositivos asociados al mismo ID de usuario Instagram, se toma el primero que se encuentre.