

PROYECTO #1 DE LA SEMANA 3 PARA CURSO 4
DESARROLLO DE APLICACIONES VANZADAS CON ANDROID
UNIVERSIDAD NACIONAL AUTONOMA DE MEXICO

Para la primera parte se configuró el servicio en FireBase y se utilizó Node js para implementar una aplicación en Heroku. A continuación la implementación del POST registrar-usuario que además graba el identificado de dispositivo y el identificador de la cuenta de Instagram de la aplicación desarrollada.

```
var registrarUsuarioURI = "registrar-usuario";
app.post("/" + registrarUsuarioURI, function(request,response) {
    var id_dispositivo      = request.body.id_dispositivo;
    var id_usuario_instagram    = request.body.id_usuario_instagram;
    var db = firebase.database();
    var registro = db.ref(registrarUsuarioURI).push();
    registro.set({
        id_dispositivo : id_dispositivo,
        id_usuario_instagram : id_usuario_instagram
    });

    var path = registro.toString();
    var pathSplit = path.split(registrarUsuarioURI + "/");
    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(registrarUsuarioURI);
    ref.on("child_added", function(snapshot, prevChildKey) {
        usuario = snapshot.val();
        respuesta = {
            id: idAutoGenerado,
            id_dispositivo : usuario.id_dispositivo,
            id_usuario_instagram : usuario.id_usuario_instagram
        };
    });

    });
```

```

        return respuesta;
    }

```

Este RESTAPI devuelve un identificador que es autogenerado por Firebase y también los identificadores de dispositivo y de usuario de Instagram enviados al método POST.

En la parte de Android, se definieron las siguientes constantes:

```

//PARA LA PARTE DE HEROKU.
public static final String ROOT_URL_HEROKU = "https://whispering-cliffs-37590.herokuapp.com/";
public static final String KEY_POST_REGISTRA_USUARIO = "registrar-usuario/";

```

Se definió la interface IEndpointsHeroku, que contiene la definición para la llamada al método POST implementado vía Node.js.

```

public interface IEndpointsHeroku {
    @FormUrlEncoded
    @POST(ConstantsRestApi.KEY_POST_REGISTRA_USUARIO)
    Call<RegistraUsuarioResponse>
    registrarUsuario(@Field("id_dispositivo") String id_dispositivo,
    @Field("id_usuario_instagram") String id_usuario_instagram);
}

```

Para los datos a recibir se definió la clase RegistraUsuarioResponse, el código a continuación:

```

public class RegistraUsuarioResponse {

    private String id;

    private String id_dispositivo;
    private String id_usuario_instagram;

    public RegistraUsuarioResponse(String id, String id_dispositivo,
    String id_usuario_instagram) {
        this.id = id;
        this.id_dispositivo = id_dispositivo;
        this.id_usuario_instagram = id_usuario_instagram;
    }

    public RegistraUsuarioResponse() {
    }

    public String getId() {
        return id;
    }
}

```

```

    public void setId(String id) {
        this.id = id;
    }

    public String getId_dispositivo() {
        return id_dispositivo;
    }

    public void setId_dispositivo(String id_dispositivo) {
        this.id_dispositivo = id_dispositivo;
    }

    public String getId_usuario_instagram() {
        return id_usuario_instagram;
    }

    public void setId_usuario_instagram(String id_usuario_instagram) {
        this.id_usuario_instagram = id_usuario_instagram;
    }
}

```

Se implementó un adaptador del RestApi, el código a continuación (este es el que se encarga de establecer la conexión):

```

public class RestApiHerokuAdapter {
    public IEndpointsHeroku establecerConexionRestAPIHeroku() {
        Retrofit retrofit = new Retrofit.Builder()
            .baseUrl(ConstantsRestApi.ROOT_URL_HEROKU)
            .addConverterFactory(GsonConverterFactory.create())
            .build();

        return retrofit.create(IEndpointsHeroku.class);
    }
}

```

Finalmente, desde la opción de menú de “Recibir Notificaciones”, se hace la llamada a un procedimiento que implementa la llamada al método POST dejando el registro del dispositivo y la cuenta Instagram principal en la base de datos de Firebase.

A continuación el código del procedimiento:

```

case R.id.mRecibeNotificaciones:
    String token1 = FirebaseInstanceId.getInstance().getToken();
    Log.d("TOKEN", token1);
    enviarTokenRegistro("En Recibe notif.", token1);
    insertaRegistroFirebase(token1, cuentaInstagram);
    break;

```

```

private void enviarTokenRegistro(String donde,String token) {

    Log.d(donde,token);
    Toast.makeText(this,"Se obtuvo el siguiente Id de dispositivo:
"+token,Toast.LENGTH_LONG).show();
}

private void insertaRegistroFirebase(String id_dispositivo,String
id_usuario_instagram) {

    RestApiHerokuAdapter restApiHerokuAdapter = new
RestApiHerokuAdapter();
    IEndpointsHeroku iEndpointsHeroku =
restApiHerokuAdapter.establecerConexionRestAPIHeroku();
    Call<RegistraUsuarioResponse> registraUsuarioResponseCall =
iEndpointsHeroku.registrarUsuario(id_dispositivo,id_usuario_instagram);

    registraUsuarioResponseCall.enqueue(new
Callback<RegistraUsuarioResponse>() {

        @Override
        public void onResponse(Call<RegistraUsuarioResponse> call,
Response<RegistraUsuarioResponse> response) {
            RegistraUsuarioResponse registraUsuarioResponse =
response.body();
            Log.d("ID_FIREBASE_DB", registraUsuarioResponse.getId());
            Log.d("ID_DISPOSITIVO",
registraUsuarioResponse.getId_dispositivo());
            Log.d("ID_USUARIO_INSTAGRAM",
registraUsuarioResponse.getId_usuario_instagram());
        }

        @Override

        public void onFailure(Call<RegistraUsuarioResponse> call,
Throwable t) {
        }

    });
}

```