

# **TECNOLOGÍA SUPERIOR EN DESARROLLO DE SOFTWARE**



## **DESARROLLO DE APLICACIONES MÓVILES**

### **DOCENTE:**

Ing. Patricio Pacheco, Mgtr.

### **INTEGRANTES:**

-Adrian Guerrero

### **CURSO:**

M4B

### **TEMA:**

GUIA PRÁCTICA 5



## GUÍA PRÁCTICA

### 1. Datos Generales

<b>Carrera:</b>	<b>Tecnología Superior en Desarrollo de Software</b>
<b>Período académico:</b>	<b>Diciembre 2021 – Abril 2022</b>
<b>Asignatura:</b>	<b>Desarrollo de Aplicaciones Móviles</b>
<b>Unidad N°:</b>	<b>4 Consumiendo Web Services, Acceso a Datos, Sincronización de Información</b>
<b>Tema:</b>	<b>Consumiendo WebServices</b>
<b>Ciclo-Paralelo:</b>	<b>M4B</b>
<b>Fecha de inicio de la Unidad:</b>	<b>15/02/2022</b>
<b>Fecha de fin de la Unidad</b>	<b>16/03/2022</b>
<b>Práctica N°:</b>	<b>4</b>
<b>Horas:</b>	<b>12</b>
<b>Docente:</b>	<b>Ing. Patricio Pacheco</b>

### 1. Contenido

#### 1.1 Introducción

Esta guía práctica está centrada en cómo consumir una API o servicios web desde una aplicación Android.

Si por ejemplo tienes una base de datos, pero no tienes una API creada. Entonces primero deberías definir una API.

Una API es un intermediario entre una base de datos y una aplicación móvil

#### 1.2 Objetivo de la Guía

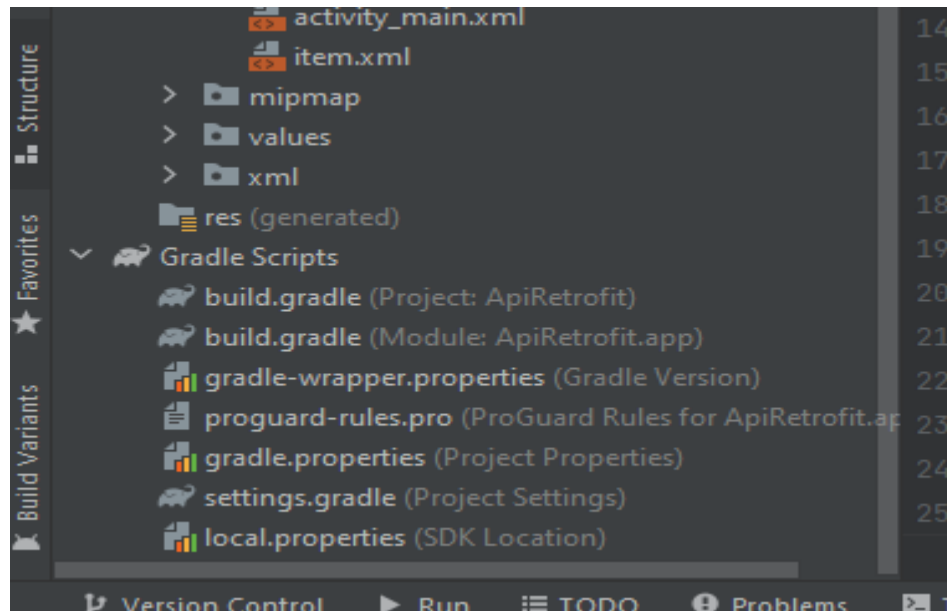
Consumir una API (servicios web) utilizando el IDE de Desarrollo de Android Studio con la librería Retrofit y procesar la respuesta JSON obtenida.

#### 1.3 Materiales, herramientas, equipos y software

- Equipos de computación,
- Android Studio
- Internet,
- Material Guía (Talleres, ejercicios prácticos).

## 1. Añadir la librería de Retrofit a nuestro proyecto.

En este caso usaremos el método más común y recomendado: añadiremos Retrofit vía Gradle.



Eso significa que debemos ir a nuestro archivo build.gradle y añadir las siguientes líneas:

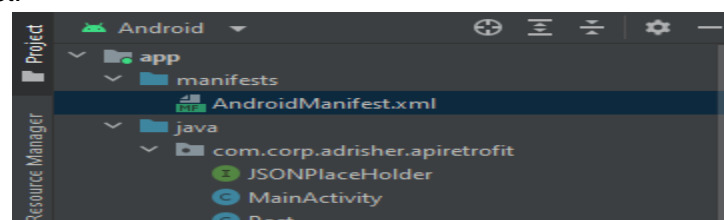
```
dependencies {  
    implementation 'androidx.appcompat:appcompat:1.4.2'  
    implementation 'com.google.android.material:material:1.6.1'  
    implementation 'androidx.constraintlayout:constraintlayout:2.1.4'  
    implementation 'com.squareup.retrofit2:retrofit:2.4.0'  
    implementation 'com.squareup.retrofit2:converter-gson:2.4.0'  
    testImplementation 'junit:junit:4.13.2'  
    androidTestImplementation 'androidx.test.ext:junit:1.1.3'  
    androidTestImplementation 'androidx.test.espresso:espresso-core:3.4.0'  
}
```

Dentro del archivo debes agregar 2 dependencias. Una para Retrofit y otra para GSON.

## 2. Solicitar Permisos

Antes de empezar a configurar Retrofit en nuestro proyecto, es importante que nuestra aplicación se pueda conectar a internet.

Para solicitar este permiso debemos añadir la siguiente línea a nuestro archivo manifest.



```
JSONPlaceholder.java × activity_main.xml × build.gradle (app) × AndroidManifest.xml × MainActivity.java ×
1 <?xml version="1.0" encoding="utf-8"?>
2 <manifest xmlns:android="http://schemas.android.com/apk/res/android"
3   xmlns:tools="http://schemas.android.com/tools"
4   package="com.corp.adrisher.apiretrofit">
5
6   <uses-permission android:name="android.permission.INTERNET"/>
7
8   <application
9     android:allowBackup="true"
10    android:dataExtractionRules="@xml/data_extraction_rules"
11    android:fullBackupContent="@xml/backup_rules"
12    android:icon="@mipmap/ic_launcher"
13    android:label="ApiRetrofit"
14    android:roundIcon="@mipmap/ic_launcher_round"
15    android:supportsRtl="true"
```

### 3. Crear Una clase y una interfaz

```
Project
└─ app
   └─ manifests
      └─ AndroidManifest.xml
   └─ java
      └─ com.corp.adrisher.apiretrofit
         ├── JSONPlaceholder
         ├── MainActivity
         ├── Post
         └── PostAdapter
      └─ com.corp.adrisher.apiretrofit (androidTest)
      └─ com.corp.adrisher.apiretrofit (test)
   └─ java (generated)
   └─ res
      ├── drawable
      ├── layout
      │   ├── activity_main.xml
      │   └── item.xml
      ├── mipmap
      ├── values
      └── xml
   └─ res (generated)
   └─ Gradle Scripts
      ├── build.gradle (Project: ApiRetrofit)
      ├── build.gradle (Module: ApiRetrofit.app)
      ├── gradle-wrapper.properties (Gradle Version)
      └── proguard-rules.pro (ProGuard Rules for ApiRetrofit.app)

Post.java × PostAdapter.java ×
1 package com.corp.adrisher.apiretrofit;
2
3 public class Post {
4
5     private String userId, id, title, body;
6
7     public String getUserId() { return userId; }
8
9     public String getId() { return id; }
10
11    public String getTitle() { return title; }
12
13    public String getBody() { return body; }
14
15 }
16
17
18
19
20
21
22
23
```

```
Post.java × JSONPlaceholder.java × PostAdapter.java ×
1 package com.corp.adrisher.apiretrofit;
2
3 import java.util.List;
4 import retrofit2.Call;
5 import retrofit2.http.GET;
6
7 public interface JSONPlaceholder {
8     @GET("posts")
9     Call<List<Post>> getPost();
10
11 }
12
```

```
Post.java × JSONPlaceholder.java × MainActivity.java × PostAdapter.java ×
1 package com.corp.adrisher.apiretrofit;
2
3 import androidx.appcompat.app.AppCompatActivity;
4 import androidx.recyclerview.widget.LinearLayoutManager;
5 import androidx.recyclerview.widget.RecyclerView;
6 import android.os.Bundle;
7 import android.widget.Toast;
8 import java.util.List;
9 import retrofit2.Call;
10 import retrofit2.Callback;
11 import retrofit2.Response;
12 import retrofit2.Retrofit;
13 import retrofit2.converter.gson.GsonConverterFactory;
14
15 public class MainActivity extends AppCompatActivity {
16
17     private RecyclerView recyclerView;
18     @Override
19     protected void onCreate(Bundle savedInstanceState) {
20         super.onCreate(savedInstanceState);
21         setContentView(R.layout.activity_main);
22
23         recyclerView = findViewById(R.id.recyclerList);
24         recyclerView.setHasFixedSize(true);
25         recyclerView.setLayoutManager(new LinearLayoutManager(context: this));
26
27     }
28 }
```

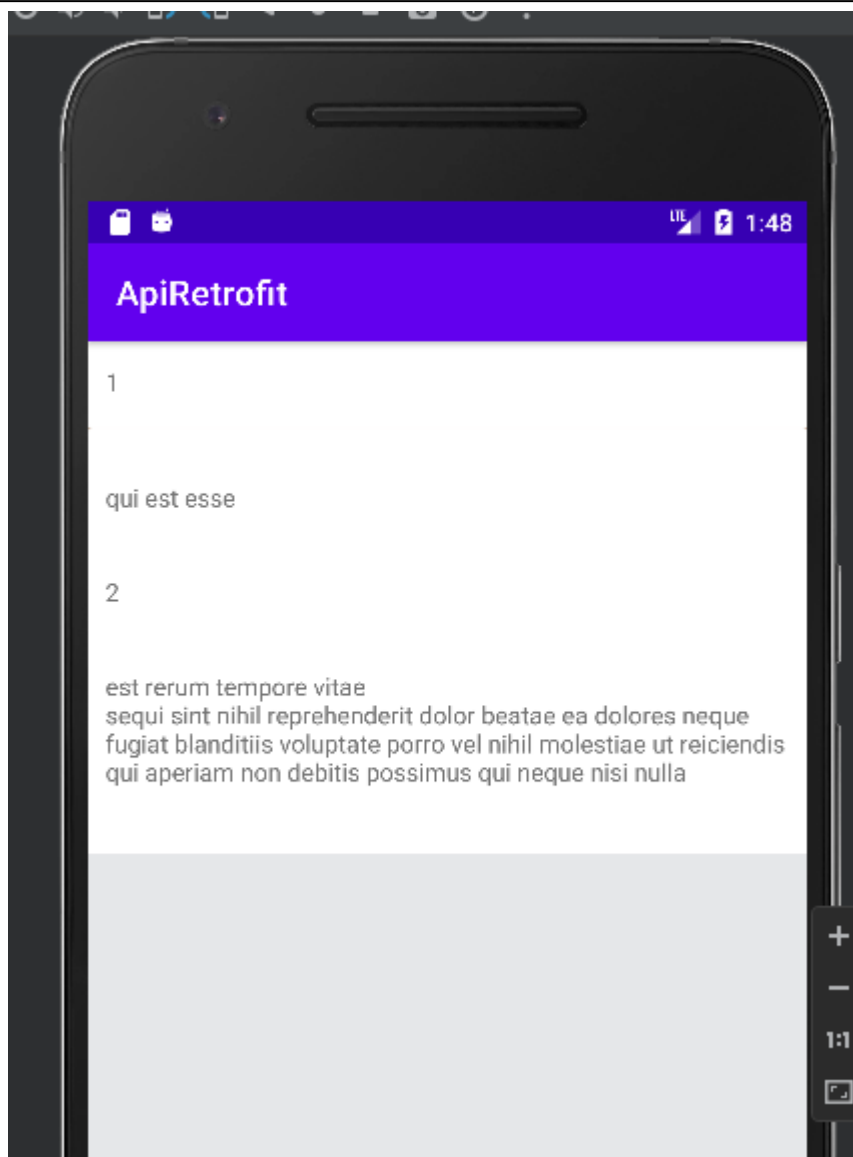
```
Post.java x JSONPlaceholder.java x MainActivity.java x PostAdapter.java x
24     recyclerView.setHasFixedSize(true);
25     recyclerView.setLayoutManager(new LinearLayoutManager(context, this));
26
27
28     Retrofit retrofit = new Retrofit.Builder()
29         .baseUrl("https://jsonplaceholder.typicode.com")
30         .addConverterFactory(GsonConverterFactory.create())
31         .build();
32
33     JSONPlaceholder jsonPlaceholder = retrofit.create(JSONPlaceholder.class);
34     Call<List<Post>> call = jsonPlaceholder.getPost();
35     call.enqueue(new Callback<List<Post>>() {
36         @Override
37         public void onResponse(Call<List<Post>> call, Response<List<Post>> response) {
38             if(!response.isSuccessful()){
39                 Toast.makeText(context, MainActivity.this, response.code(), Toast.LENGTH_LONG).show();
40                 return;
41             }
42             List<Post> postList = response.body();
43             PostAdapter postAdapter = new PostAdapter(context, MainActivity.this, postList);
44             recyclerView.setAdapter(postAdapter);
45         }
46     });
47
```

```
38         if(!response.isSuccessful()){
39             Toast.makeText(context, MainActivity.this, response.code(), Toast.LENGTH_LONG).show();
40             return;
41         }
42         List<Post> postList = response.body();
43         PostAdapter postAdapter = new PostAdapter(context, MainActivity.this, postList);
44         recyclerView.setAdapter(postAdapter);
45     }
46
47     @Override
48     public void onFailure(Call<List<Post>> call, Throwable t) {
49         Toast.makeText(context, MainActivity.this, t.getMessage(), Toast.LENGTH_LONG).show();
50     }
51 }
52
53 }
54
55 }
56
```

< > ↺ 🔒 jsonplaceholder.typicode.com/posts



```
[
  {
    "userId": 1,
    "id": 1,
    "title": "sunt aut facere repellat provident occaecati excepturi optio reprehenderit",
    "body": "quia et suscipit\nsuscipit recusandae consequuntur expedita et cum\nreprehenderit molestiae ut ut quas totam\nnostrum rerum est autem sunt rem eveniet architecto"
  },
  {
    "userId": 1,
    "id": 2,
    "title": "qui est esse",
    "body": "est rerum tempore vitae\nsequi sint nihil reprehenderit dolor beatae ea dolores neque\nfugiat blanditiis voluptate porro vel nihil molestiae ut reiciendis\nqui aperiam non debitis possimus qui neque nisi nulla"
  },
  {
    "userId": 1,
    "id": 3,
    "title": "ea molestias quasi exercitationem repellat qui ipsa sit aut",
    "body": "et iusto sed quo iure\nvoluptatem occaecati omnis eligendi aut ad\nvoluptatem doloribus vel accusantium quis pariatur\nmolestiae porro eius odio et labore et velit aut"
  },
  {
    "userId": 1,
    "id": 4,
    "title": "eum et est occaecati",
    "body": "ullam et saepe reiciendis voluptatem adipisci\nsit amet autem assumenda provident rerum culpa\nquis hic commodi nesciunt rem tenetur doloremque ipsam iure\nquis sunt voluptatem rerum illo velit"
  },
  {
    "userId": 1,
    "id": 5,
    "title": "nesciunt quas odio",
    "body": "repudiandae veniam quaerat sunt sed\nnalias aut fugiat sit autem sed est\nvoluptatem omnis possimus esse voluptatibus quis\nest aut tenetur dolor neque"
  },
  {
    "userId": 1,
    "id": 6,
    "title": "dolorem eum magni eos aperiam quia",
    "body": "ut aspernatur corporis harum nihil quis provident sequi\nmollitia nobis aliquid molestiae\nperspiciatis et ea nemo ab reprehenderit accusantium quas\nvoluptate dolores velit et doloremque molestiae"
  },
  {
    "userId": 1,
    "id": 7
```



Enlace al Repositorio de Git : <https://github.com/Adrisher/ApiRestRetrofit.git>