Peticiones Web

Cliente / Servidor

- **Cliente**:
  - **Navegador**: Chrome, Firefox, IE…
  - **Aplicaciones**: móvil, tableta, PC, domótica…
  - **Aplicaciones backend**
- **Servidor**:
  - Páginas web
  - Ficheros
  - Correo electrónico
  - Base de datos

API de servicio web

- **Formato**:
  - XML, JSON…
- **Sintaxis**:
  - Métodos, parámetros y tipos de dato
- **Acción**:
  - GET, POST, PUT, DELETE…
- **Autenticación**:
  - Nombre, contraseña, token
- **Tipos**:
  - REST, SOAP…

Peticiones en Android

- **AndroidHTTPClient**: Sin soporte
- **HTTPURLConnection**: Código de bajo nivel
- **OkHttp**:
  - Código de alto nivel
  - Cola de conexiones
  - Cacheo
- **Volley**:
  - Resuelve problemas de AsyncTask
  - Peticiones en paralelo
  - Poca documentación
- **Retrofit**:
  - Abstracción con interfaces
  - Transformación a JSON automática

Estado de conexión

- En AndroidManifest:

```
<uses-permission android:name="android.permission.ACCESS_NETWORK_STATE" />
```

- En Activity o Fragment:

```
ConnectivityManager cm = (ConnectivityManager) getSystemService(Context.CONNECTIVITY_SERVICE);

NetworkInfo networkInfo = cm.getActiveNetworkInfo();

if (networkInfo != null){
   // Hay conexión
} else {
   // No hay conexión
}
```

Tipo de conexión (**API <= 23**)

- A partir de **NetworkInfo**:

```
switch (networkInfo.getType()){
    case ConnectivityManager.TYPE_MOBILE:
        // Datos móviles
        break;
    case ConnectivityManager.TYPE_WIFI:
        // Red Wifi
        break;
    case ConnectivityManager.TYPE_BLUETOOTH:
        // Bluetooth
        break;
    case ConnectivityManager.TYPE_ETHERNET:
        // Cable de red
        break;
}
```

Tipo de conexión (**API > 23**)

- A partir de **NetworkCapabilities**:

```
NetworkCapabilities nc = cm.getNetworkCapabilities(cm.getActiveNetwork());
if (nc.hasTransport(NetworkCapabilities.TRANSPORT_CELLULAR)) {
        // Datos móviles
} else if (nc.hasTransport(NetworkCapabilities.TRANSPORT_WIFI)) {
        // Red Wifi
} else if (nc.hasTransport(NetworkCapabilities.TRANSPORT_BLUETOOTH)) {
        // Bluetooth
} else if (nc.hasTransport(NetworkCapabilities.TRANSPORT_ETHERNET)) {
        // Cable de red
}
```

Peticiones con Retrofit

- En AndroidManifest:

```
<uses-permission android:name="android.permission.INTERNET" />
```

- Añadir dependencia en **gradle (Module: app)**:

```
implementation 'com.squareup.retrofit2:retrofit:2.5.0'
implementation 'com.squareup.retrofit2:converter-scalars:2.5.0'
implementation 'com.squareup.retrofit2:converter-gson:2.5.0'
```

Definición de API de llamadas

- Crear interfaz:

```
public interface DatosApi {

   @GET(“datos.json")
   Call<ArrayList<Dato>> obtenerDatos();

   @FormUrlEncoded
   @POST(“login.php")
   Call<String> login(@Field("user") String usuario, @Field("pass") String password);

}
```

- Crear clase de respuesta:

```
public class Dato {

   String dato;
   int otroDato;
   boolean otroDatoMas
}
```

Preparación de petición

- En Activity o Fragment:

```
Retrofit retrofit = new Retrofit.Builder()
        .baseUrl("http://www.host.com/ruta/" )
        .addConverterFactory(ScalarsConverterFactory.create())
        .addConverterFactory(GsonConverterFactory.create())
        .build();
DatosApi api = retrofit.create(DatosApi.class);
```

Llamada de petición GET

- En Activity o Fragment:

```
Call<ArrayList<Dato>> llamada = api.obtenerDatos();
llamada.enqueue(new Callback<ArrayList<Dato>>() {
    @Override
    public void onResponse(Call<ArrayList<Dato>> call,
Response<ArrayList<Dato>> response) {
        ArrayList<Dato> datos = response.body();
        // Usar datos
}

    @Override
    public void onFailure(Call<ArrayList<Dato>> call, Throwable t) {
        // Mostrar error al usuario
    }
});
```

Llamada de petición POST

- En Activity o Fragment:

```
Call<String> llamada = api.login("pepe", "1234");
llamada.enqueue(new Callback<String>() {
    @Override
    public void onResponse(Call<String> call, Response<String> response) {
        String resultado = response.body();
        // Usar datos
}

    @Override
    public void onFailure(Call<String> call, Throwable t) {
        // Mostrar error al usuario
    }
}););
```

Aceptar peticiones sin cifrar

```xml
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.example.alumnos.peticionesweb">

    <uses-permission android:name="android.permission.INTERNET"/>

    <application
        android:allowBackup="true"
        android:icon="@mipmap/ic_launcher"
        android:label="PeticionesWeb"
        android:roundIcon="@mipmap/ic_launcher_round"
        android:supportsRtl="true"
        android:usesCleartextTraffic="true"
        android:theme="@style/AppTheme">
        <activity android:name=".MainActivity">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />

                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
    </application>
```