



## EL LORO HUASTEKO

### Órgano de Divulgación Científica y Tecnológica del Instituto Tecnológico Superior de Pánuco

Memorias del 1er Seminario de Investigación Multidisciplinario Interinstitucional 2019  
Instituto Tecnológico Superior de Pánuco  
Instituto Tecnológico Superior de Ébano

---

## Manejo del protocolo HTTP en aplicaciones nativas de Android

Moreno Pérez Héctor Hugo  
Ramírez Vázquez Juan Carlos  
Vázquez Elorza Fortino  
Email autor correspondiente:  
Área de participación:

*Instituto Tecnológico Superior de Pánuco  
Instituto Tecnológico Superior de Pánuco  
Instituto Tecnológico Superior de Pánuco  
hecktor@itspanuco.edu.mx  
Sistema Computacionales e Informática*

### RESUMEN

El constante crecimiento de los teléfonos inteligentes y la existencia de miles de aplicaciones ha conducido a las empresas a orientar sus productos al sector de la movilidad. En este entorno de evolución existe una lucha constante entre fabricantes de software y hardware, uno de los gladiadores más fuertes es Google con su sistema operativo Android el cual se encuentra en el 90% de los teléfonos de todo el planeta.

Android ha sido diseñado para permitir a los usuarios crear distintas aplicaciones aprovechando las distintas características y utilidades de los teléfonos, siendo la conectividad hacia el Internet un parámetro vital para explotar al máximo los aplicativos. (Amaro, 2014)

Este artículo centra su atención en la conectividad de las aplicaciones Android con un entorno local y externo manipulando el protocolo HTTP a través de la librería Volley, permitiendo a los desarrolladores de software hacer el envío y recepción de datos más simple.

**Palabras claves:** Internet, android, http, redes, Volley.

### ABSTRACT

The constant growth of smartphones and the existence of thousands of applications has led companies to target their products to the mobility sector. In this environment of evolution there is a constant struggle between software and hardware manufacturers, one of the strongest gladiators is Google with its Android operating system which is found in 90% of phones worldwide.

Android has been designed to allow users to create different applications taking advantage of the different features and utilities of the phones, with connectivity to the Internet being a vital parameter to fully exploit the applications.

This article focuses on the connectivity of Android applications with a local and external environment by manipulating the HTTP protocol through the Volley library, allowing software developers to make sending and receiving data simpler.

**Key words:** Internet, android, http, network, Volley.

## INTRODUCCIÓN

Nos encontramos en una época en que todas las actividades cotidianas se traducen al medio digital. La llegada del teléfono inteligente y consecuentemente los aplicativos móviles han provocado que la sociedad realice una gran cantidad de tareas, solo presionando un botón de una pantalla. (Nolazo, 2010)

Los Smartphone actualmente cuentan con grandes prestaciones tanto en hardware como en software, el sistema central que predomina el mercado es Android, pertenece a la empresa Google y brinda la posibilidad de crear aplicativos con el Lenguaje de Programación Java.

Java contribuye a las apps de manera significativa proporcionándoles el poder de la conectividad a Internet, que hoy en día es esencial para poder compartir información en tiempo real, esto se logra programando rutinas utilizando el protocolo HTTP.

HTTP cuenta con los métodos GET Y POST, para el intercambio de Información entre un cliente y un servidor, estas tareas se realizan administrando hilos de control y tareas de parsing .

En esta investigación se describe la librería Vólley que realiza las funciones mencionadas anteriormente de una manera más transparente.

## METODOLOGÍA

En el presente apartado se indicara como se llevó acabo el manejo de la librería Volley en el ambiente de trabajo de Android studio 3.5

### Fase 1 Inclusión de la librería al entorno de trabajo

En esta etapa el desarrollador tiene que ubicar en su estructura de trabajo el apartado de **GladeScripts** y abrir el archivo **build.grade (Module:app)**, colocarse en su segmento **dependencies**, añadiendo la siguiente línea de código **implementation 'com.android.volley:volley:1.0.0'** ,el IDE detecta el cambio e inicia la descarga de los archivos para el manejo de la librería.

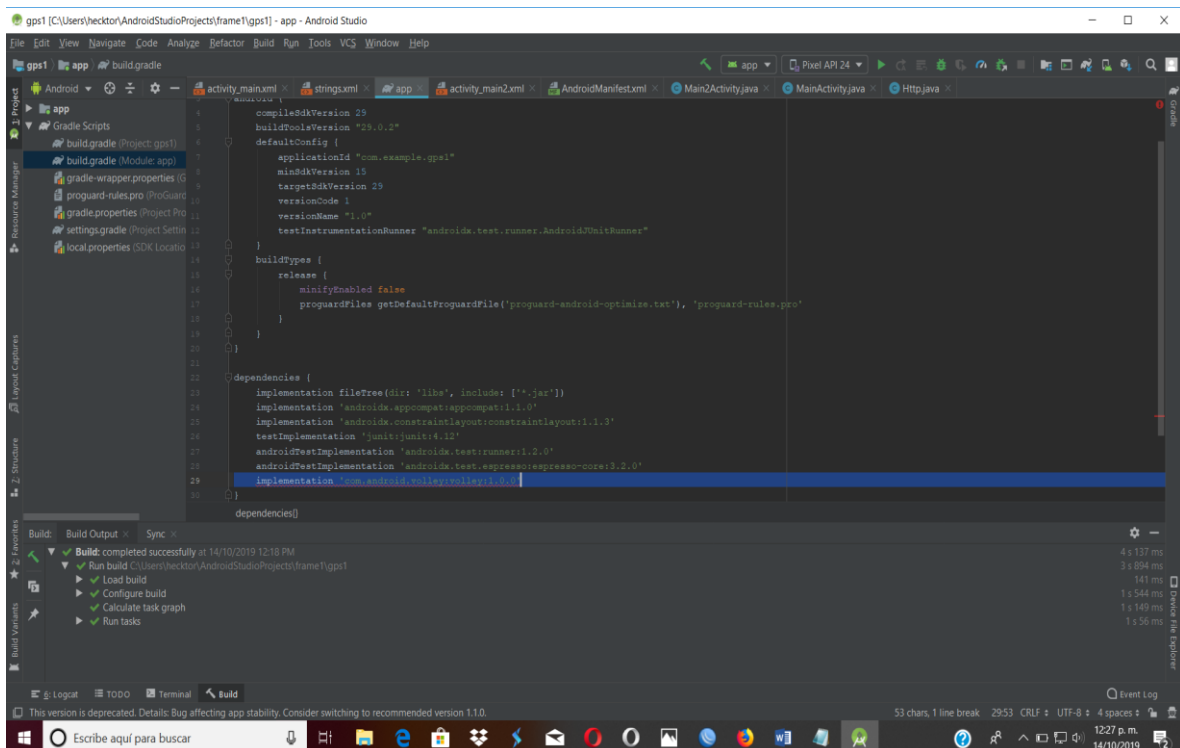


Fig 1. Entorno de Desarrollo Android Studio

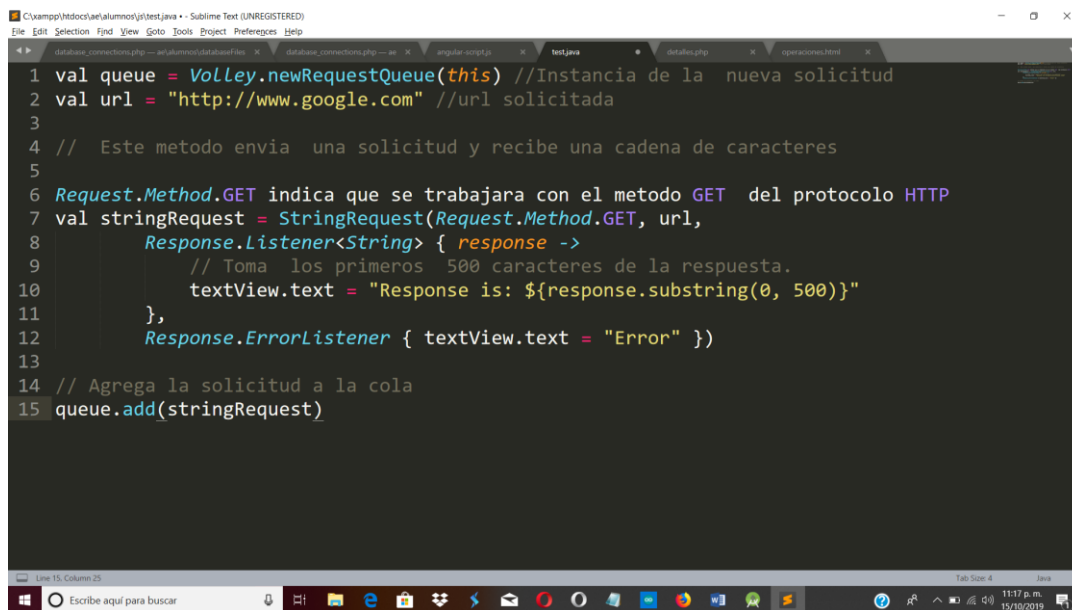
## Fase 2 Agregando permisos a la aplicación

En el archivo AndroidManifest.xml se incluye el permiso para acceso a Internet :

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

## Fase 3 Uso de los objetos

Tras incluir la llamada a la librería "import com.android.volley.\*", se inicia con la creación de los objetos y métodos pertenecientes a la librería esto se describe en la Fig 2 (developer,2019)



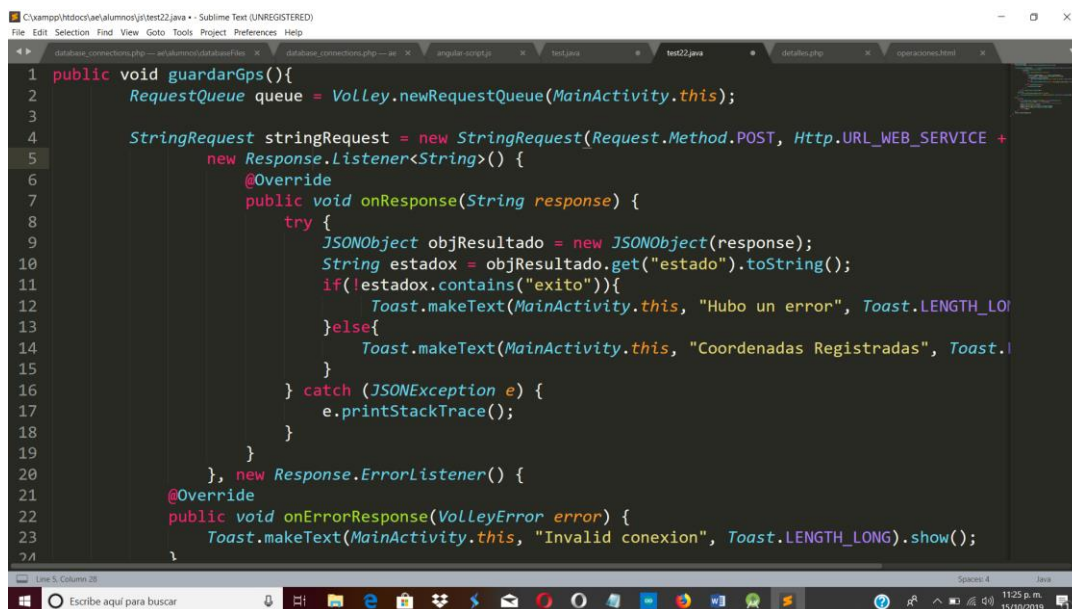
```

1 val queue = Volley.newRequestQueue(this) //Instancia de la nueva solicitud
2 val url = "http://www.google.com" //url solicitada
3
4 // Este metodo envia una solicitud y recibe una cadena de caracteres
5
6 Request.Method.GET indica que se trabajara con el metodo GET del protocolo HTTP
7 val stringRequest = StringRequest(Request.Method.GET, url,
8     Response.Listener<String> { response ->
9         // Toma los primeros 500 caracteres de la respuesta.
10            textView.text = "Response is: ${response.substring(0, 500)}"
11        },
12        Response.ErrorListener { textView.text = "Error" })
13
14 // Agrega la solicitud a la cola
15 queue.add(stringRequest)
  
```

Fig 3. Manipulación de los Objetos y métodos de la librería Volley

## RESULTADOS Y DISCUSIÓN

La implementación de la librería Volley se ha utilizado para incorporarla en el envío de coordenadas obtenidas de un teléfono inteligente y la recepción de la respuesta del servidor en formato JSON decodificado para su tratamiento.



```

1 public void guardarGps(){
2     RequestQueue queue = Volley.newRequestQueue(MainActivity.this);
3
4     StringRequest stringRequest = new StringRequest(Request.Method.POST, Http.URL_WEB_SERVICE +
5         new Response.Listener<String>() {
6             @Override
7             public void onResponse(String response) {
8                 try {
9                     JSONObject objResultado = new JSONObject(response);
10                    String estadox = objResultado.get("estado").toString();
11                    if(!estadox.contains("exito")){
12                        Toast.makeText(MainActivity.this, "Hubo un error", Toast.LENGTH_LO
13                    }else{
14                        Toast.makeText(MainActivity.this, "Coordenadas Registradas", Toast.
15                    }
16                } catch (JSONException e) {
17                    e.printStackTrace();
18                }
19            }
20        }, new Response.ErrorListener() {
21            @Override
22            public void onErrorResponse(VolleyError error) {
23                Toast.makeText(MainActivity.this, "Invalid conexion", Toast.LENGTH_LONG).show();
24            }
25        }
  
```

## TRABAJO A FUTURO

Se debe documentar más a fondo para que sea de utilidad para estudiantes y desarrolladores que deseen darles conectividad a sus aplicaciones móviles.

## CONCLUSIONES

Esta investigación demuestra que la librería trabaja como una interfaz de alto nivel, liberando al programador de la administración de hilos de control y del tedioso trabajo de descomponer las cadenas de caracteres recibidas de un servidor, trasladando todo a un solo hilo principal de trabajo.

## REFERENCIAS

- Amaro, J. (2014). El gran libro de programación Avanzada en Android. AlfaOmega
- Nolazco, J. (2010). Desarrollo de Aplicaciones Móviles en Android. Macro
- Amaro, J. (2012). Programación de Dispositivos Mviles. AlfaOmega
- Developer, G (2019), Training Developer Google. <https://developer.android.com/training/volley>