|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Materia:** |  | Programación Móvil II | | |
| **Práctica:** |  | BroadcastReceiver | | |
| **Alumno(s):** |  | Jesús García Páramo  Irving Nahum Paniagua Mora | **Fecha:** | 12/02/2020 |

**Objetivo**

Desarrollar una aplicación en Android que cargue una lista de tarjetas con imágenes desde Internet. Puedes utilizar el objeto ImageLoader de Volley, RecyclerView CardViews o cualquier control que permita cargar listas con imágenes.

**Temas del plan de estudios**

TEMA 3: Telefonía y redes

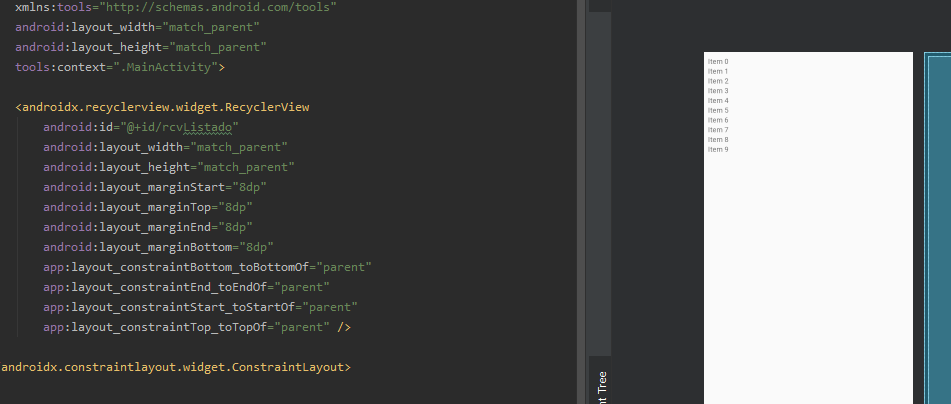
**Material**

Android Studio

**Marco Teórico**

**Desarrollo**

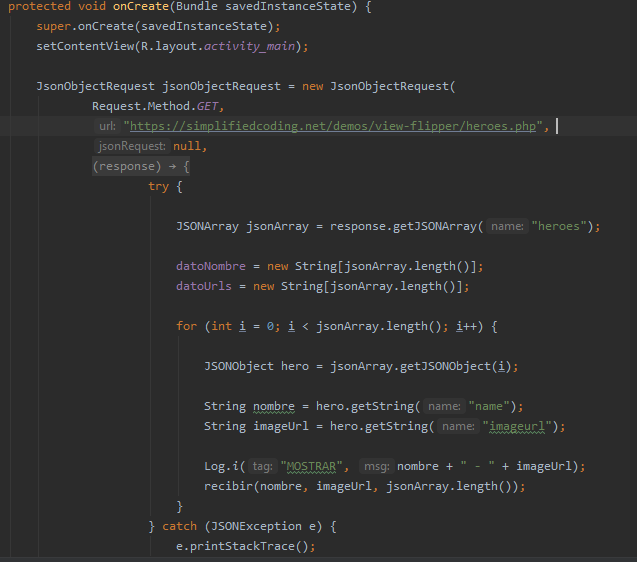
**Se creo la clase MainActivity con su respectivo layout activity\_main el cual contiene el recyclerview**

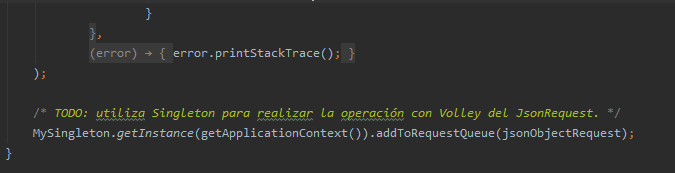


**En la clase MainActivity se creo una instancia de la clase RecyclerViewAdaptador la cual contiene el adaptador de recyclerview y se creo una instancia de recyclerview**

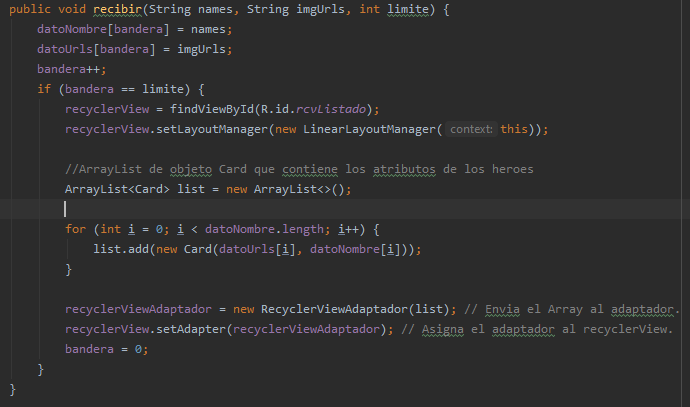


**En el método onCreate se realiza el JsonObjectRequest que permite extraer los datos contenidos en el JSON desde la dirección URL utilizada. Los datos obtenidos del JSON son parseados para obtener los valores de los atributos nombre y URL.**

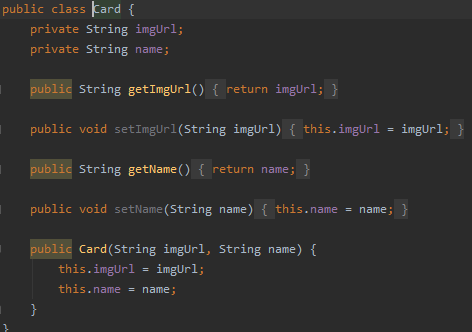




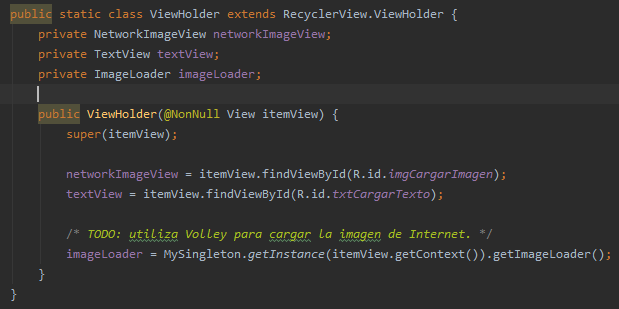
**Se creo el método Recibir el cual se encarga de agregar los datos de los heroes nombre y URL recibidos del método anterior a un ArrayList que pasa al adaptador del RecyclerView.**



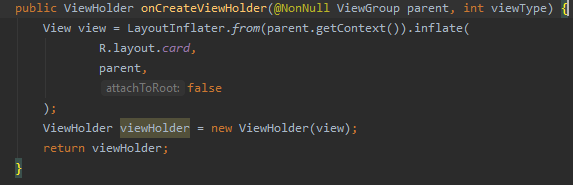
**Se creo la clase Card que contiene los atributos de nombre y url del héroe con sus respectivos getters y setters.**



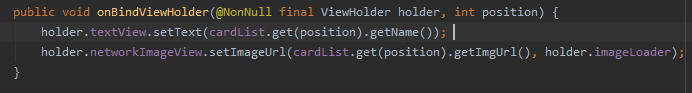
**Se creo la clase RecyclerViewAdaptador que extiende de RecyclerView.Adapter<RecyclerViewAdaptador.ViewHolder> el cual implementa los métodos de onCreateView holder(), onBindViewHolder() y getItemCount().**



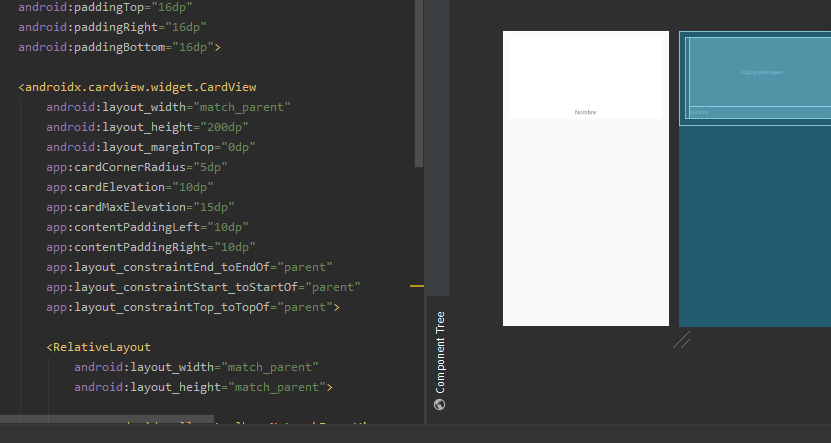
**El metodo onCreateViewHolder se crea la vista de cada Card, es el inflador del RecyclerView.**



**El metodo onBindViewHolder se encarga de asignar el nombre al textView y asignamos la imagen al networkManager el cual se encargará de cargar la imagen del héroe desde la url.**



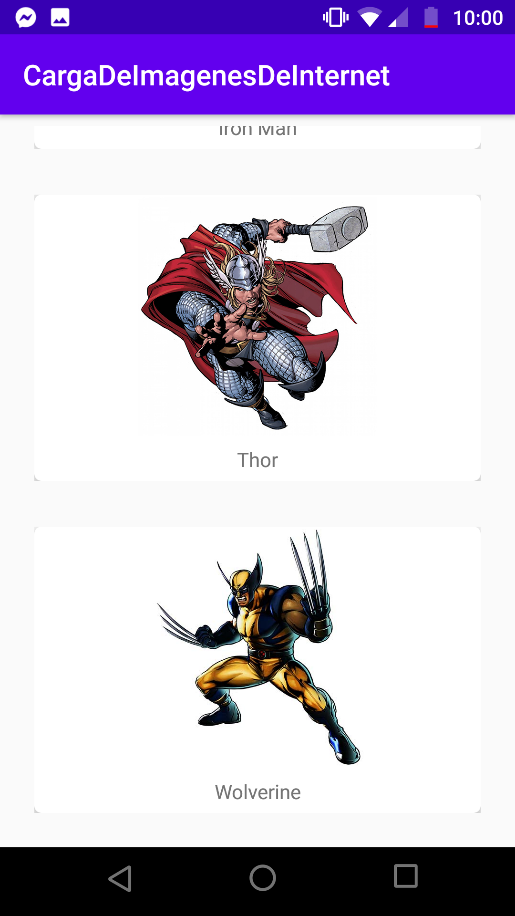
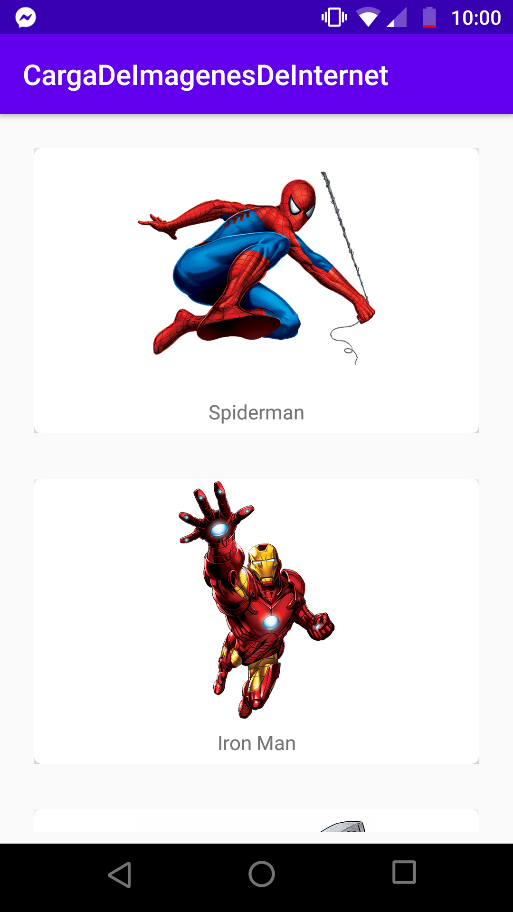
**Se creo también el layout card el cual contiene el networkImageView en el cual se colocara la imagen del héroe y textView el cual contendrá el nombre del heroe que se cargue el networkImageView.**



**Tambien asignamos el permiso de uso de internet en el manifiesto para que la app se pueda conectar a internet.**



**Resultado**



**Evaluación**

**Conclusiones**

**Bibliografía**