

FEUD

Fundación de egresados U. Distrital
Construyendo Profesionales



Somos el Centro de Entrenamiento Autorizado por marcas representativas en Gobierno TI y empresa, con el portafolio más amplio en Latinoamérica:



Todas las marcas, nombres comerciales, marcas de servicios y logotipos a los que se hace referencia en el presente documento pertenecen a sus respectivas empresas. ITIL® es una marca registrada de AXELOS Limited. El Swirl logo™ es una marca de AXELOS Limited. PMP, PMI, PMI-RMP, PMI-ACP, PMI-SP, PgMP, CAPM, REP Logo son marcas registradas del Project Management Institute.

FEUD
Fundación de egresados U. Distrital
Construyendo Profesionales



Desarrollo de aplicaciones utilizando SDK - Android

Diplomado Desarrollo de Software para móviles

Julio César Galeano García

2017

Android

- Fragments
- Colecciones
- Adapter
- Consumir un servicio
- Hilos

Android: Fragments

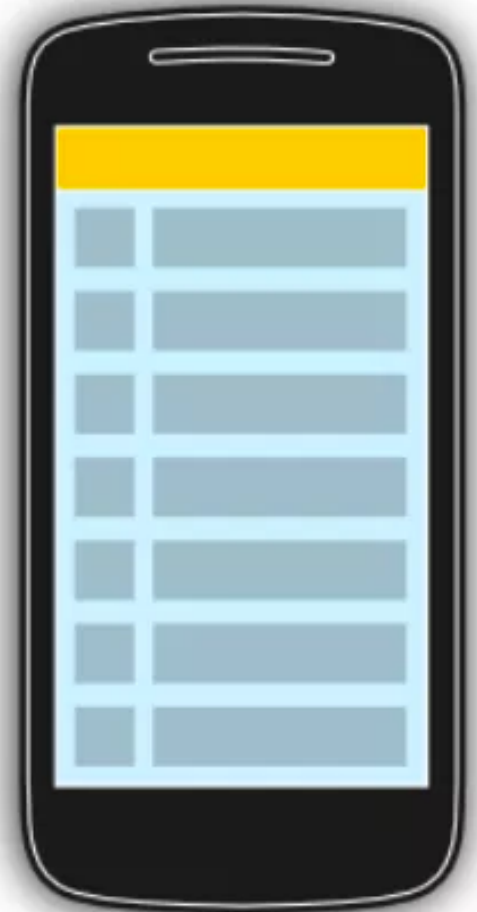
Un Fragment representa un comportamiento o una parte de la interfaz de usuario en una Activity. Puedes combinar múltiples fragmentos en una sola actividad para crear una IU multipanel y volver a usar un fragmento en múltiples actividades. Puedes pensar en un fragmento como una sección modular de una actividad que tiene su ciclo de vida propio, recibe sus propios eventos de entrada y que puedes agregar o quitar mientras la actividad se esté ejecutando (algo así como una "subactividad" que puedes volver a usar en diferentes actividades).

Android: ListView

ListView es un grupo de vistas que muestra una lista de elementos desplazables. Los elementos de la lista se insertan automáticamente en la lista con un Adapter que toma contenido de una fuente, como una matriz o consulta de base de datos, y convierte cada resultado en una vista que se dispone en la lista.

```
<ListView  
    android:id="@+id/listId"  
    android:layout_width="match_parent"  
    android:layout_height="match_parent" />
```

```
ListView listView = (ListView) findViewById(R.id.listId);
```

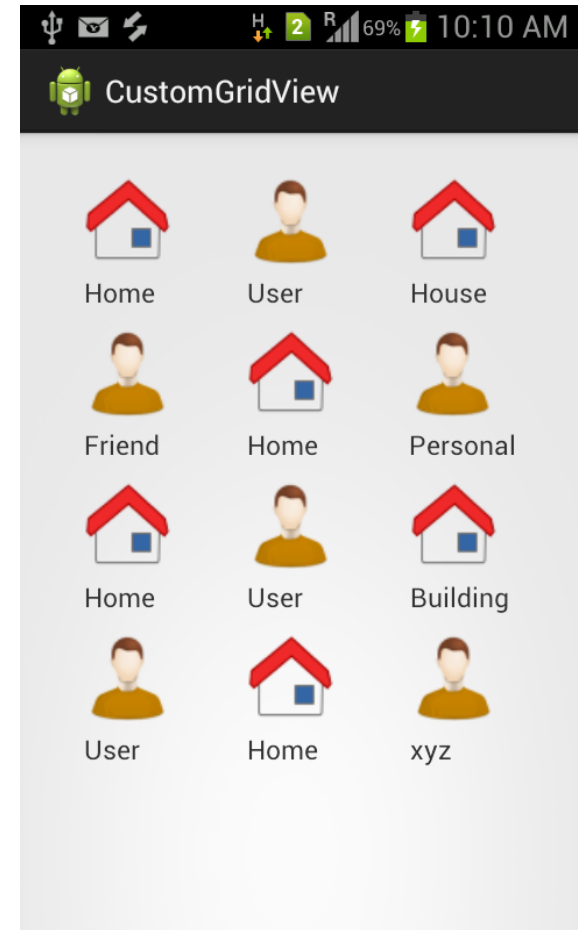


Android: GridView

GridView es una subclase de ViewGroup y, por lo tanto, también de View. Muestra datos a modo de rejilla bidimensional e incluye automáticamente un scroll para cuando los datos ocupan más que el tamaño de la pantalla. Los elementos de la lista se insertan automáticamente en la lista con un Adapter.

```
<GridView  
    android:id="@+id/gridId"  
    android:layout_width="match_parent"  
    android:layout_height="match_parent" />
```

```
GridView gridView = (GridView) findViewById(R.id.gridId);
```



Android: RecyclerView

El widget RecyclerView es una versión más flexible y avanzada de ListView. Este widget es un contenedor para mostrar grandes conjuntos de datos que se pueden desplazar de manera muy eficiente al mantener una cantidad limitada de vistas. Usa el widget RecyclerView cuando tengas conjuntos de datos cuyos elementos cambien en tiempo de ejecución sobre la base de la acción del usuario o los eventos de la red.



```
<android.support.v7.widget.RecyclerView  
    android:id="@+id/recyclerId"  
    android:layout_width="match_parent"  
    android:layout_height="match_parent" />
```

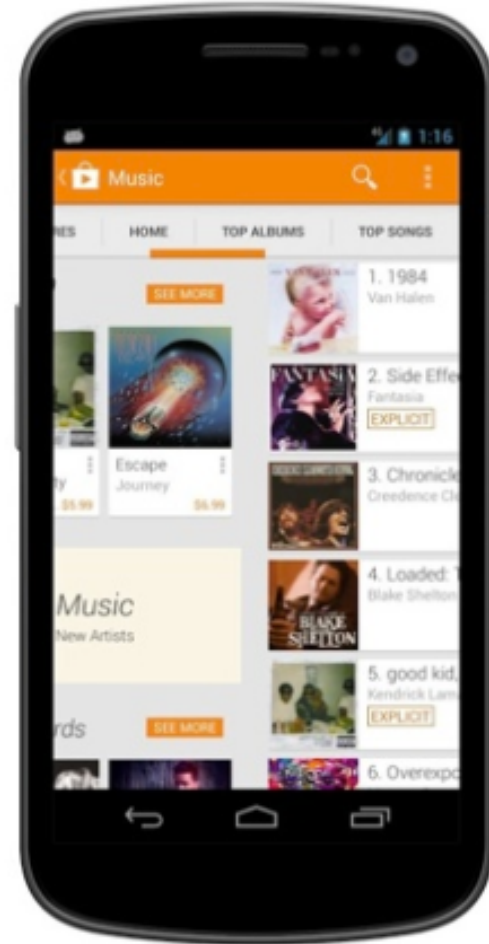
```
RecyclerView recyclerView =  
    ((RecyclerView) findViewById(R.id.recyclerId));
```

compile 'com.android.support:recyclerview-v7:25.2.0'

Android: ViewPager

ViewPager se utiliza con más frecuencia junto con Fragment, que es una manera conveniente de suministrar y administrar el ciclo de vida de cada página. Hay adaptadores estándar implementados para el uso de fragmentos con ViewPager, que cubren los casos de uso más comunes. Estos son `FragmentPagerAdapter` y `FragmentStatePagerAdapter`; Cada una de estas clases tiene un código simple que muestra cómo construir una interfaz de usuario completa con ellos.

```
<android.support.v4.view.ViewPager
    android:layout_width="match_parent"
    android:layout_height="match_parent">
```

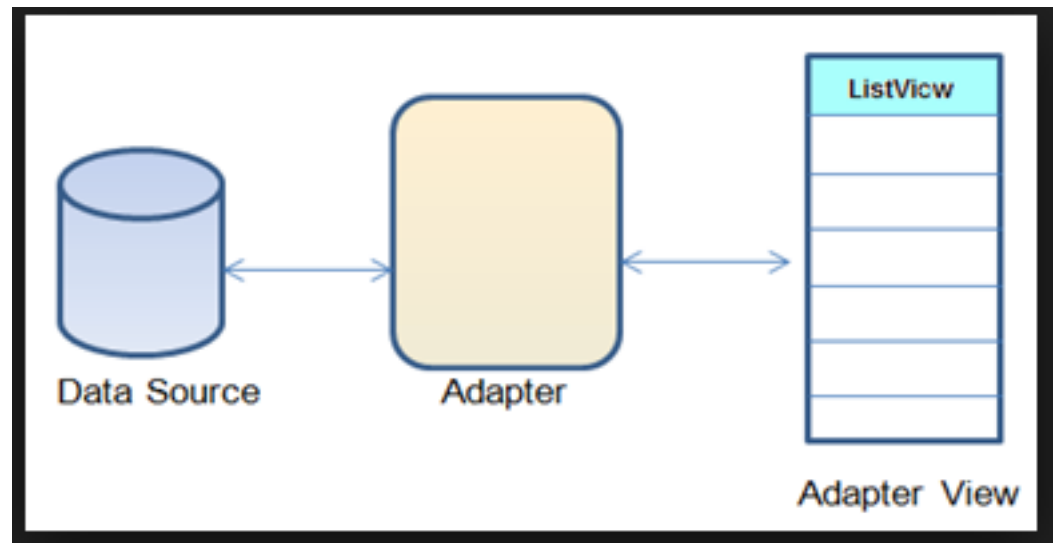


Android: Adapter

Un Adapter actúa como un puente entre un AdapterView y los datos subyacentes para esa vista. El Adapter proporciona acceso a los elementos de datos. El Adapter también es responsable de crear una vista para cada elemento del conjunto de datos.

Tipos de Adapter

- BaseAdapter
- ArrayAdapter
- CursorAdapter
- SimpleCursorAdapter
- RecyclerViewAdapter



Android: Consumir un servicio

```
public JSONObject getJSONObjectFromURL(String urlString) throws IOException, JSONException {  
    HttpURLConnection urlConnection = null;  
    URL url = new URL(urlString);  
    urlConnection = (HttpURLConnection) url.openConnection();  
    urlConnection.setRequestMethod("GET");  
    urlConnection.setReadTimeout(10000 /* milliseconds */);  
    urlConnection.setConnectTimeout(15000 /* milliseconds */);  
    urlConnection.setDoOutput(true);  
    urlConnection.connect();  
  
    BufferedReader br=new BufferedReader(new InputStreamReader(url.openStream()));  
    char[] buffer = new char[1024];  
    String jsonString = "";  
    StringBuilder sb = new StringBuilder();  
    String line;  
    while ((line = br.readLine()) != null) {  
        sb.append(line+"\n");  
    }  
    br.close();  
  
    jsonString = sb.toString();  
    System.out.println("JSON: " + jsonString);  
    return new JSONObject(jsonString);  
}
```

Android: Threads

```
new Thread(new Runnable() {  
    public void run() {  
        final Bitmap bitmap =  
            loadImageFromNetwork("http://example.com/image.png");  
        mImageView.post(new Runnable() {  
            public void run() {  
                mImageView.setImageBitmap(bitmap);  
            }  
        });  
    }  
}).start();
```

Android: AsyncTask

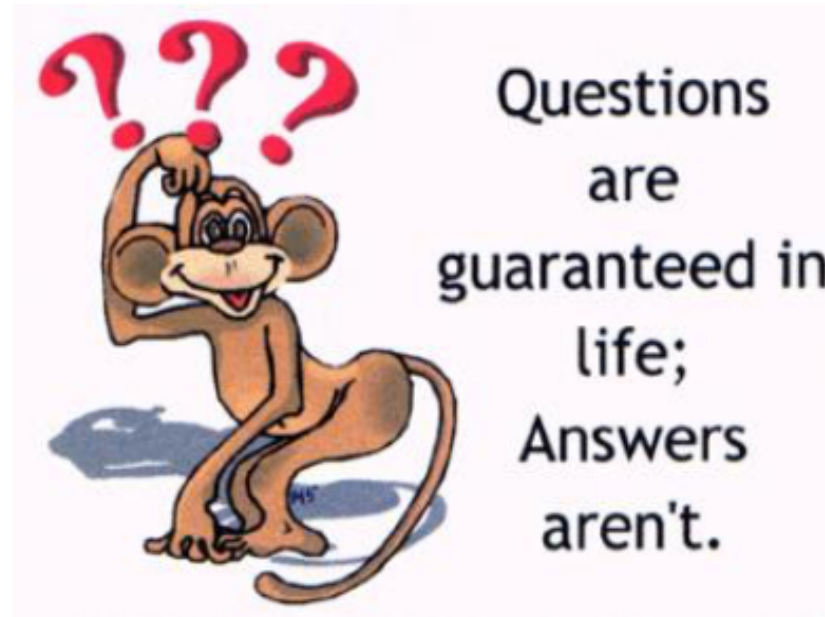
La AsyncTask te permite realizar trabajo asincrónico en la interfaz de usuario. Realiza las operaciones de bloqueo en un subproceso de trabajo y, luego, publica los resultados en el subproceso de IU, sin que tú tengas que manejar los subprocesos o los controladores.

Métodos importantes.

- `onPreExecute()`
- `doInBackground()`
- `onProgressUpdate()`
- `onPostExecute()`

Android: AsyncTask

```
public void onClick(View v) {  
    new DownloadImageTask().execute("http://example.com/image.png");  
}  
  
private class DownloadImageTask extends AsyncTask<String, Void, Bitmap> {  
    /** The system calls this to perform work in a worker thread and  
     * delivers it the parameters given to AsyncTask.execute() */  
    protected Bitmap doInBackground(String... urls) {  
        return loadImageFromNetwork(urls[0]);  
    }  
  
    /** The system calls this to perform work in the UI thread and delivers  
     * the result from doInBackground() */  
    protected void onPostExecute(Bitmap result) {  
        mImageView.setImageBitmap(result);  
    }  
}
```



¿Preguntas?

Julio César Galeano García