

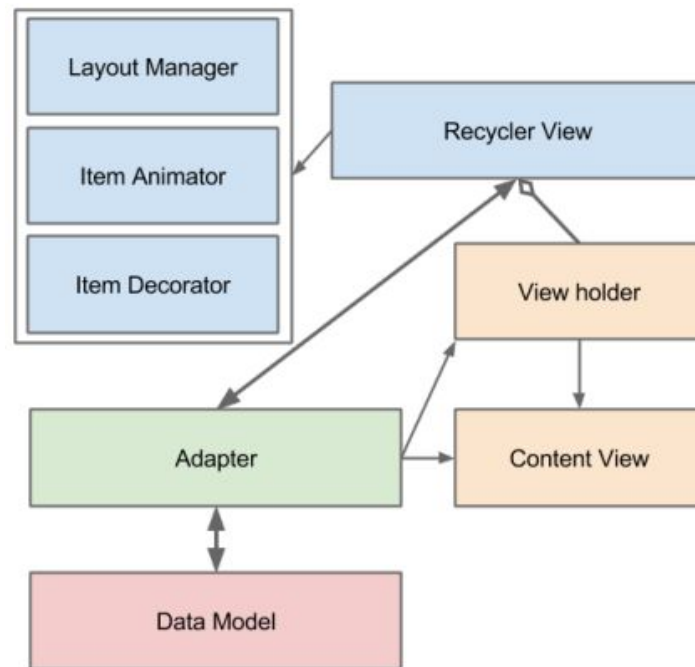
Recycler View

- Nos permite mostrar en pantalla grandes colecciones de datos
- Similar al ListView



Componentes complementarios

- RecyclerView.Adapter
- RecyclerView.ViewHolder
- LayoutManager
- ItemDecoration
- ItemAnimator



View Holder

- Obtener cada componente mediante el método `findViewById()` es costosa, dependiendo la complejidad del layout.

¿Por qué no guardamos una referencia a los componentes que forman un layout para no tener que volver a buscarlos?

View Holder

La idea será por tanto crear e inicializar el objeto ViewHolder la primera vez que inflemos un elemento de la lista y asociarlo a dicho elemento de forma que posteriormente podamos recuperarlo fácilmente.

```
static class ViewHolder
{
    TextView titulo;
    TextView subtítulo;
}
```

- Podemos asignarla y recuperarla mediante los métodos
 - `setTag()`
 - `getTag()`

Layout Manager

- Determina la forma en que se van a mostrar en pantalla los elementos de nuestra colección.
- El SDK incorpora de serie tres LayoutManager para las tres representaciones más habituales:
 - Lista vertical u horizontal (LinearLayoutManager)
 - Tabla tradicional (GridLayoutManager)
 - Tabla apilada (StaggeredGridLayoutManager)

Item Decoration y Animator

Estos componentes se encargan de definir cómo se representan algunos aspectos visuales de nuestra lista:

Ejemplo:

- Marcadores
- Separadores de elementos
- Animaciones al añadir o eliminar elementos.

¿Cómo utilizamos el RecyclerView?

- 1) Debemos modificar el .build del Gradle e importar las dependencias necesarias.

```
dependencies {  
    ...  
    compile 'com.android.support:recyclerview-v7:+'  
    ...  
}
```

2) Insertar el recycler view en el xml

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