

Paso a paso

1) Definir la dependencia en el archivo build de Gradle.

```
dependencies {  
    ...  
    com.android.support:recyclerview-v7:23.3.0  
    ...  
}
```

2) Agregar un recyclerView al XML. Además crear un nuevo layout para el detalle de cada item de nuestro RecyclerView.

Paso a paso

3) Implementar una clase que modele los objetos que mostraremos en nuestro RecyclerView.

- En nuestro ejemplo, creamos la clase Titulares.

4) Crear el Adapter y extenderlo.

- Recordar que los adapters para un RecyclerView deberán extender de la clase RecyclerView.Adapter

Paso a paso

5) Crear el ViewHolder. Es decir, una clase de tipo static que represente al ViewHolder y extenderla.

- Recordar que los View Holder extienden de la clase RecyclerView.ViewHolder
- En general, esta clase se define dentro del Adapter.
- En nuestro ejemplo, esa clase es TitularesViewHolder

Paso a paso

6) Sobrecribir el constructor de la clase creada anteriormente.

- Recordar que este constructor deberá realizar el findViewById de cada componente del detalle layout.

7) [Opcional] Crear un método que realice el bind de un View Holder. Es decir, que cargue el contenido de cada componente.

- En nuestro ejemplo, bindTitular(Titular unTitular)

Paso a paso

8) Sobrecribir e implementar los tres métodos que nos requiere el Adapter.

- **onCreateViewHolder(ViewGroup viewGroup, int viewType)**
 - Esta función es la encargada de inflar la vista del detalle de la lista RecyclerView y además de crear un objeto View Holder utilizando esa vista.
- **onBindViewHolder(RecyclerView.ViewHolder holder, int position)**
 - Esta función es la encargada de actualizar los datos reutilizando el ViewHolder que viene por parámetro (**holder**). Es decir, carga el view holder con los datos del objeto que se encuentra en la posición que me pasan por parámetro (**position**).
- **getItemCount().**
 - Indica el número de elementos de la colección de datos.

Paso a paso

En el MainActivity

1) Declarar y referenciar el recyclerView que definimos en el XML.

2) Setearle al recyclerView el Layout Manager.

- Recordar que podemos definir distintos tipos de Layout Manager según la forma que queremos tener en nuestra lista.

Paso a paso

En el MainActivity

3) [Opcional] Setearle al recyclerView el `setHasFixedSize` en true, si sabemos que no se modificará el tamaño de cada item del recycler view.

4) Crear un adapter, utilizando la clase definida anteriormente.

Paso a paso

En el MainActivity

5) Setearle al recyclerView el adapter creado.

6) [Opcional] Setearle un item decorator y/o animator para darle efectos y/o animación.