

UT7.3 LISTAS: RECYCLER VIEW

Programación multimedia y dispositivos móviles



OBJETIVO

Mostrar en el interfaz de usuario listas de elementos

RECYCLER VIEW



Biblioteca para mostrar listas de forma eficiente



Recibe

datos para mostrar
qué aspecto debe tener cada uno de los datos



RecyclerView

Crea dinámicamente los elementos visuales
Los reutiliza cuando hacemos desplazamientos
Eficiente

COMPONENTES CLAVE

1. **RecyclerView**: el contenedor de la lista
2. **RecyclerView.ViewHolder**: cada uno de los elementos visuales de los elementos de la lista hereda de esta clase
3. **RecyclerView.Adapter** responsable de enlazar los datos de la lista con el ViewHolder
4. **LayoutManager** se encarga de organizar los elementos dentro del RecyclerView.

PASOS

1

Elegir si vamos a mostrar los elementos de forma lineal o en una rejilla o cuadrícula

2

Decidir qué aspecto van a tener los elementos individuales de la lista

3

Definir el adaptador que asociará nuestros datos a los elementos visuales de la lista

PASO 0

- ¡Los datos!
- Creamos una clase con datos que queremos mostrar en la lista

PASO I

- Elegir el `LayoutManager`
- `LinearLayoutManager`: listas unidimensionales,
 - verticales
 - horizontales
- `GridLayoutManager`: ordenación en cuadrícula
 - Verticales: todos los elementos de las columnas con el mismo ancho
 - Horizontal: todos los elementos de la filas con el mismo alto

PASO 2

- Crear el diseño de los elementos de la lista

PASO 3

- Adapter
 - **Constructor:** recibir y guardar el conjunto de datos
 - **onCreateViewHolder:** inflar el diseño de los elementos individuales y crear un ViewHolder y pasarle el diseño
 - **onBindViewHolder:** rellenar los campos de la vista para un elemento concreto
 - **getItemCount:** devolver el número de elementos del conjunto de datos

PASO 4

- Juntar y agitar
- En la actividad
 - Localizar el RecyclerView
 - Configurar su layoutManager
 - Crear el Adapter y poblarlo con datos
 - Configurar el Adapter del RecyclerView



COMO REACCIONAR A LOS *TAP*

Conectar un ClickListener en el ViewHolder

Propagarlo al adaptador y de ahí a quien lo consuma