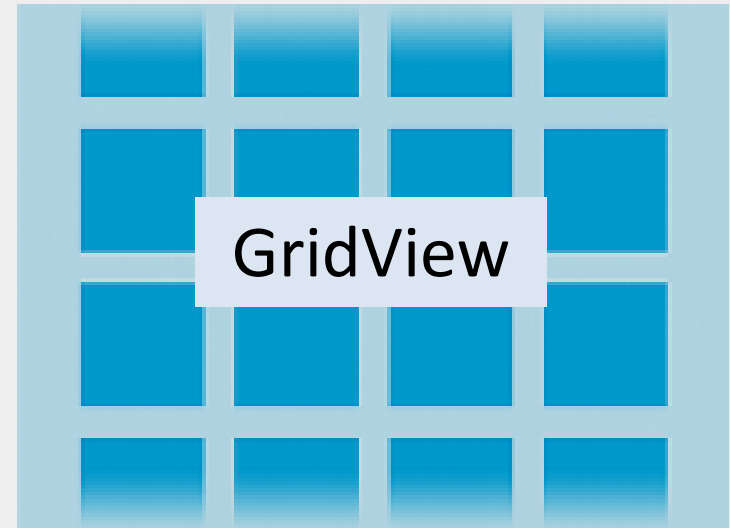
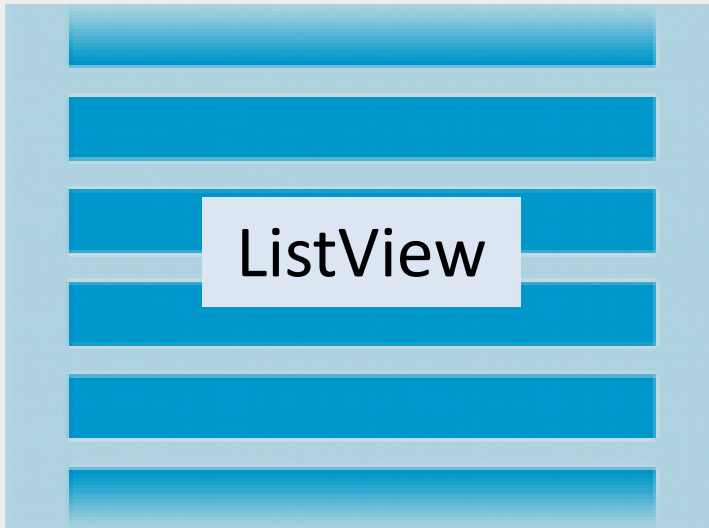




Listados

Ejemplos



Características

- **ListView**
 - Vista en orden, una detrás de otra
- **GridView**
 - Vistas colocadas ordenadas en filas y columnas
 - Atributos importantes: **numColumns**, **columnWidth**, **stretchMode**.
- Datos de **ListView** y **GridView**
 - Datos estáticos en fichero **xml**
 - Datos dinámicos básicos (array de **strings**)
 - Datos complejos (**Adapter personalizado**)

Adapter personalizado

1. Incluir la vista **ListView** en el layout
2. Crear un **layout** de elemento
3. Heredar de **ArrayAdapter**
 - a. Obtener contexto, item_layout y lista de datos en el **constructor**
 - b. Implementar el método **getView**
4. **Construir y vincular** el adapter

1. Incluir **ListView** en el layout

- En el xml del layout de una **Activity**:

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

2. Crear un layout de elemento

- En un nuevo fichero layout:

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent">

    <ImageView
        android:id="@+id/imagenIV"
        android:layout_width="100dp"
        android:layout_height="100dp"
        android:src="@color/colorAccent"/>

    <TextView
        android:id="@+id/textoTV"
        android:layout_width="wrap_content"
        android:layout_height="match_parent"
        android:layout_toRightOf="@+id/imagenIV"
        android:text="Prueba"/>
</RelativeLayout>
```

3. Heredar de **ArrayAdapter**: **Constructor**

- Crear una nueva clase, por ejemplo: **MiAdapter**:

```
public class MiAdapter extends ArrayAdapter {  
  
    Context context;  
    int itemLayout; // layout del elemento  
    Elemento[] datos; // También podría ser un ArrayList  
  
    public MiAdapter(@NonNull Context context, int resource, @NonNull Elemento[] objects) {  
        super(context, resource, objects);  
        this.context = context;  
        itemLayout = resource;  
        datos = objects;  
    }  
}
```

3. Heredar de **ArrayAdapter**: método **getView**

- En la clase que hereda de **ArrayAdapter** (**MiAdapter**):

```
public class MiAdapter extends ArrayAdapter {

    @NonNull
    @Override
    public View getView(int position, @Nullable View convertView, @NonNull ViewGroup parent) {
        if (convertView == null) {
            // Si la vista no está creada, se crea a partir del layout y el context
            LayoutInflater inflater = LayoutInflater.from(context);
            convertView = inflater.inflate(itemLayout, parent, false);
        }

        TextView textoTV = convertView.findViewById(R.id.textoTV);
        textoTV.setText(datos[position].texto);

        ImageView imagenIV = convertView.findViewById(R.id.imagenIV);
        imagenIV.setImageResource(datos[position].imagen);

        return convertView;
    }
}
```


4. Construir y vincular el adapter

- En una **Activity**:

```
Elemento[] datos;  
// Aquí se rellenarían los datos ...  
  
ListView lista = findViewById(R.id.lista);  
MiAdapter adapter = new MiAdapter(this, R.layout.item, datos);  
lista.setAdapter(adapter);
```

Algunos métodos útiles

- Actualizar listado después de modificar los datos:

```
ArrayAdapter adapter = (ArrayAdapter) lista.getAdapter();  
adapter.notifyDataSetChanged();
```

- Eventos:

setOnItemClickListener

setOnItemLongClickListener

setOnItemSelectedListener