

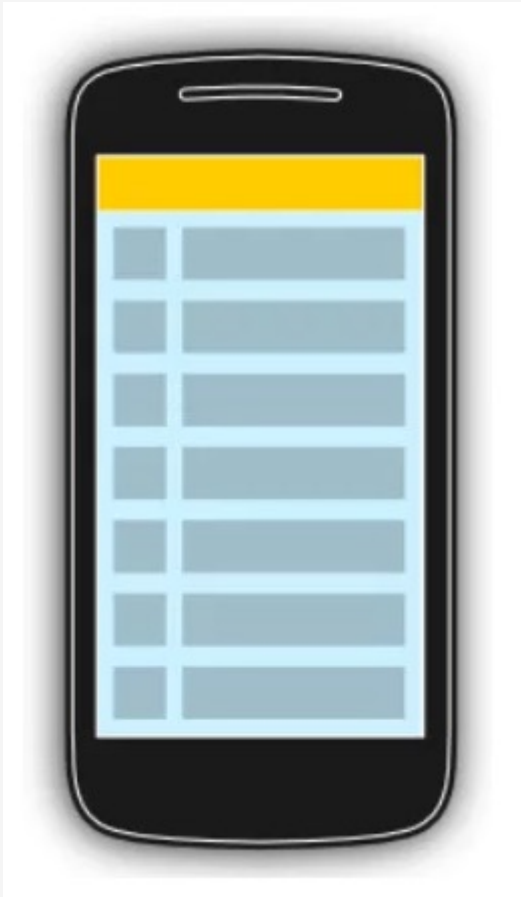
# Componentes de Listado (ListView, GridView, RecyclerView)

Clase Nro .09

Teófilo Copa F.

# ListView

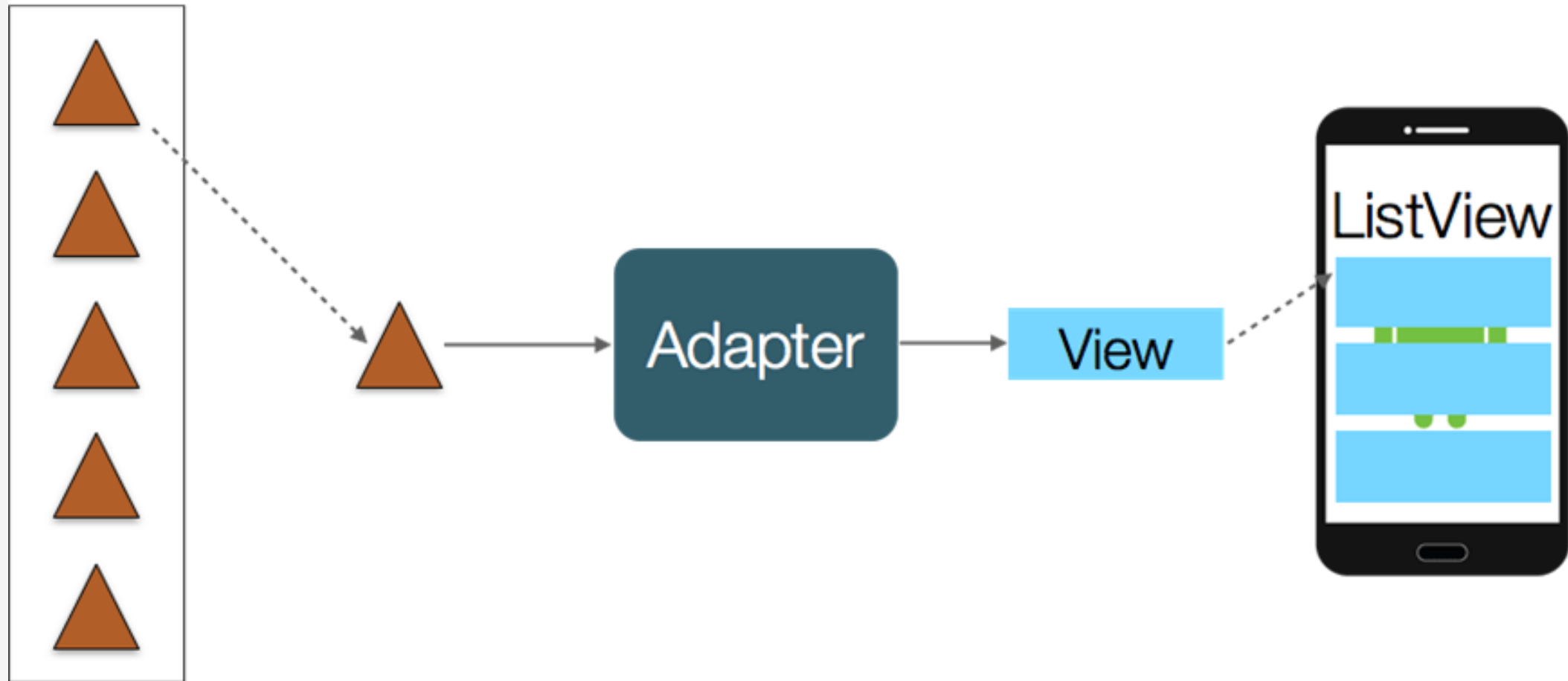
---



- **Visualiza una lista** de objetos java arbitrarios (data model)
- **Cada elemento** del *ListView* puede definirse **en un archivo de Layout**

## Pasos para implementar un ListView

---



## Pasos para implementar un ListView

---

1. **Diseñar un Layout** que lo contenga al `ListView`
2. Diseñar un **Layout individual** que se repetirá en la lista
3. Implementar una actividad que lo visualice el Layout con el `ListView`
4. Personalizar el ***Layout*** y ***Adapter*** según nuestros datos.

# 1. Diseñar un Layout que contiene un ListView

---

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:id="@+id/frameLayout3"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".FragmentListaPersonalizada">

    <!-- TODO: Update blank fragment layout -->

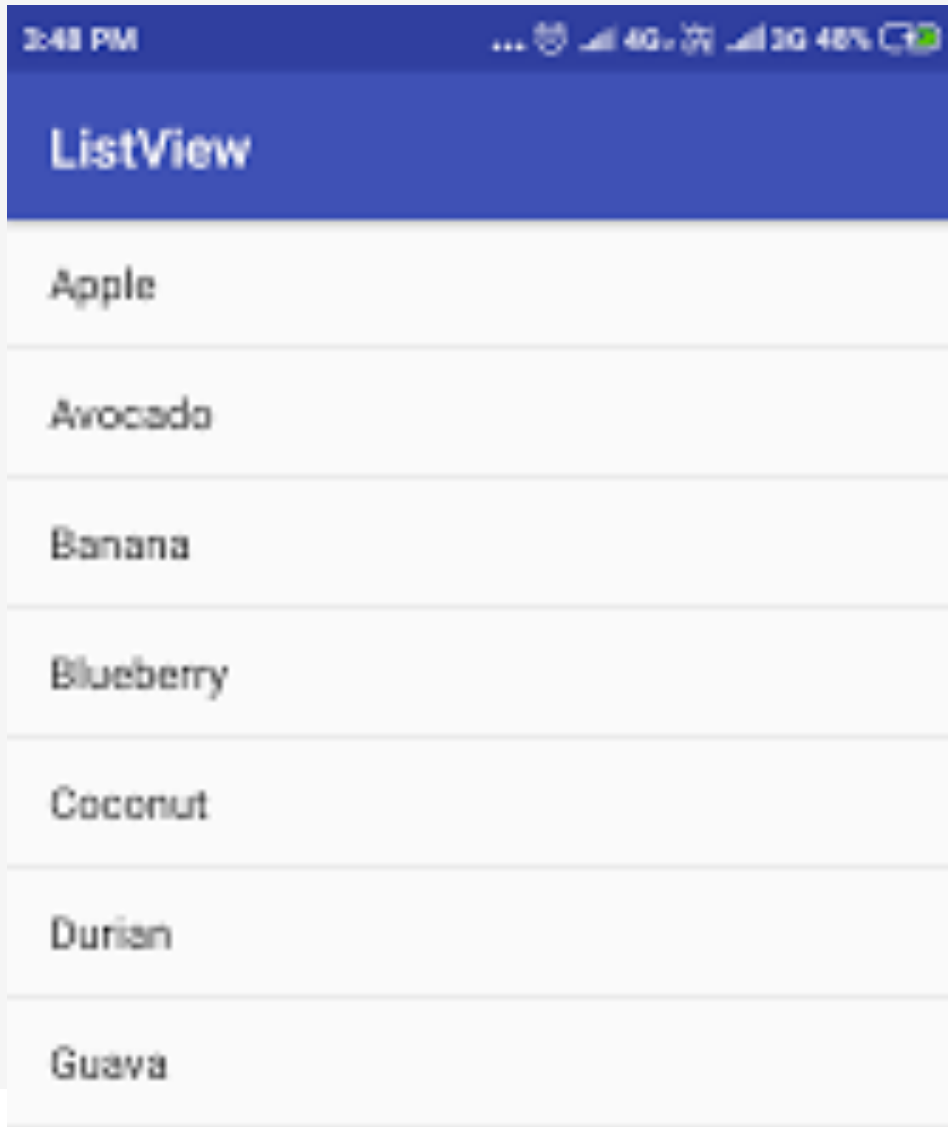
    <ListView
        android:id="@+id/listView3"
        android:layout_width="match_parent"
        android:layout_height="550dp"
        android:layout_marginTop="8dp"
        android:choiceMode="singleChoice"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent">

    </ListView>

</androidx.constraintlayout.widget.ConstraintLayout>
```

## 2. Diseñar un Layout individual para los elementos de la lista

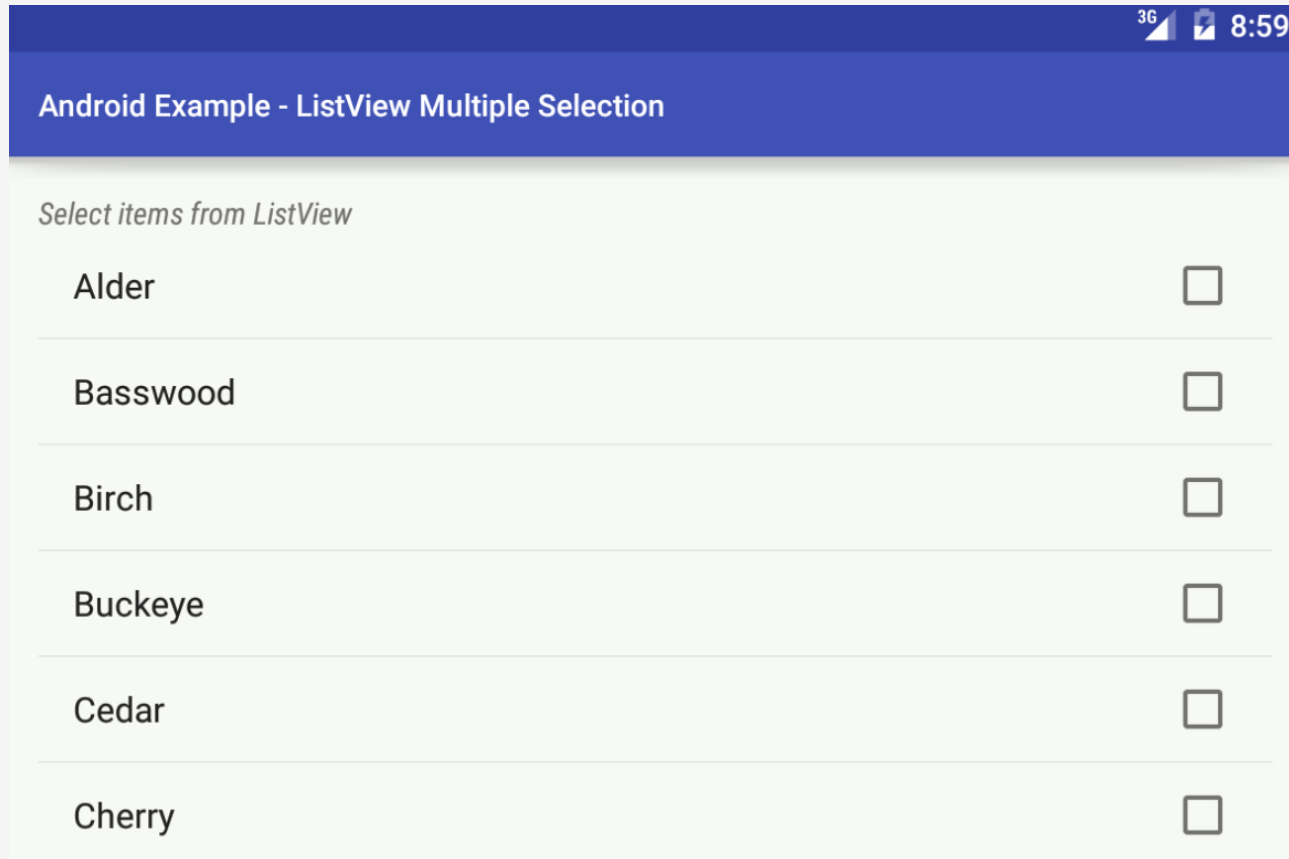
---



`android.R.layout.simple_list_item_1`

## 2. Diseñar un Layout individual para los elementos de la lista

---



```
android.R.layout.simple_list_item_multiple_choice
```

### 3. Implementar una Actividad que visualice el ListView

```
13  override fun onCreate(savedInstanceState: Bundle?) {
14      super.onCreate(savedInstanceState)
15      setContentView(R.layout.activity_main)
16
17
18      val personasList = mutableListOf<Persona>(
19          Persona(nombre: "JOSE", apellido: "MORALES", telefono: "2313526"),
20          Persona(nombre: "MARIA", apellido: "FERNANDEZ", telefono: "2343551"),
21          Persona(nombre: "ELENA", apellido: "PEÑA", telefono: "2343559"),
22          Persona(nombre: "ADOLFO", apellido: "CHOQUE", telefono: "2343557"),
23          Persona(nombre: "RENE", apellido: "PEREZ", telefono: "2343554"),
24          Persona(nombre: "WENDY", apellido: "JIMENEZ", telefono: "2343552"),
25          Persona(nombre: "JHONNY", apellido: "COLQUE", telefono: "2341556")
26      )
27      val listView = findViewById<ListView>(R.id.listView)
28      listView.choiceMode = AbsListView.CHOICE_MODE_MULTIPLE
29      val arrayAdapter = ArrayAdapter(applicationContext, android.R.layout.simple_list_item_multiple_choice, personasList)
30      listView.adapter = arrayAdapter
31
32
```

1 5 8



## 4. Personalizar un layout de acuerdo a nuestros datos



```
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="fill_parent"
    android:layout_height="?android:attr/listPreferredItemHeight"
    android:padding="6dip" >

    <ImageView
        android:id="@+id/icon"
        android:layout_width="wrap_content"
        android:layout_height="fill_parent"
        android:layout_alignParentBottom="true"
        android:layout_alignParentTop="true"
        android:layout_marginRight="6dip"
        android:contentDescription="TODO"
        android:src="@drawable/ic_launcher" />

    <TextView
        android:id="@+id/secondLine"
        android:layout_width="fill_parent"
        android:layout_height="26dip"
        android:layout_alignParentBottom="true"
        android:layout_alignParentRight="true"
        android:layout_toRightOf="@id/icon"
        android:ellipsize="marquee"
        android:maxLines="1"
        android:text="Description"
        android:textSize="12sp" />

    <TextView
        android:id="@+id/firstLine"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:layout_above="@id/secondLine"
        android:layout_alignParentRight="true"
        android:layout_alignParentTop="true"
        android:layout_alignWithParentIfMissing="true"
        android:layout_toRightOf="@id/icon"
        android:gravity="center_vertical"
        android:text="Example application"
        android:textSize="16sp" />

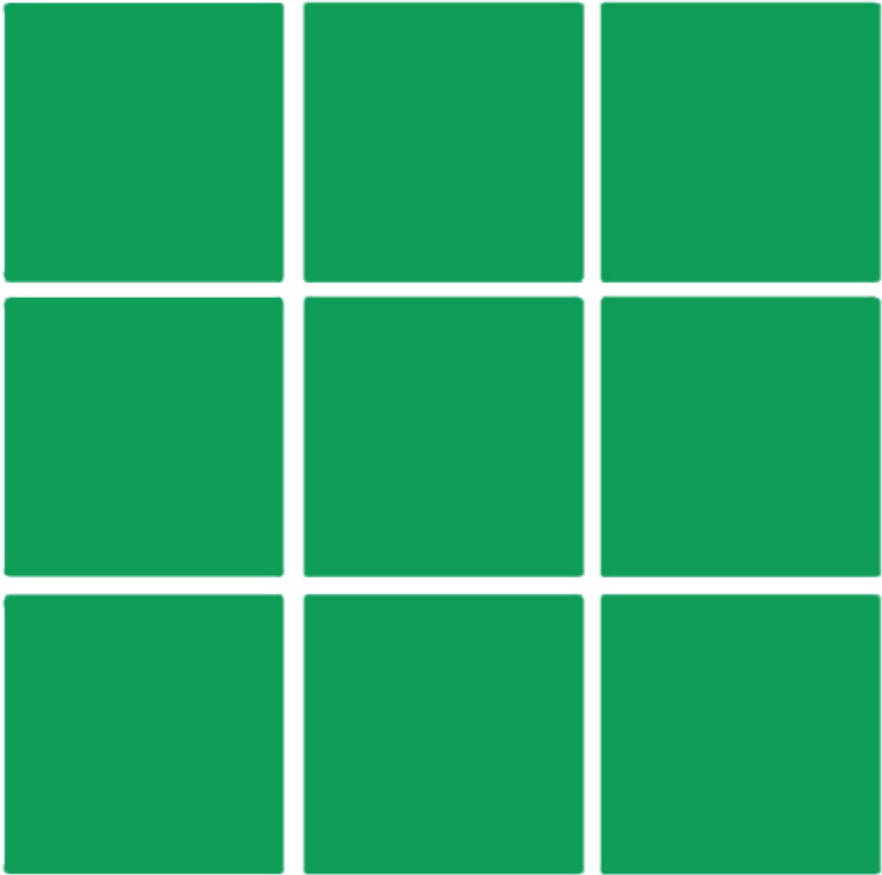
</RelativeLayout>
```

## 4. Personalizar un Adapter de acuerdo a nuestros datos

```
12
13 </> class PersonasAdapter(context: Context, personasList: List<Persona>) : ArrayAdapter<Persona>(context, resource: 0, personasList) {
14     private val ctx: Context = context
15
16     override fun getView(position: Int, convertView: View?, parent: ViewGroup): View {
17         var convertViewLocal = convertView
18         if (convertViewLocal == null) {
19             convertViewLocal = LayoutInflater.from(context).inflate(R.layout.item_views, parent, attachToRoot: false)
20         }
21
22         val img = convertViewLocal!!.findViewById<ImageView>(R.id.imagenPerfil)
23         val p = getItem(position)
24         Glide.with(ctx)
25             .load(p?.imagen)
26             .centerCrop()
27             .placeholder(R.drawable.usuarioicono)
28             .into(img)
29
30         val txtNombre = convertViewLocal.findViewById<TextView>(R.id.txtNombre)
31         txtNombre.text = "${p?.nombre} ${p?.apellido}"
32
33         val txtCelular = convertViewLocal.findViewById<TextView>(R.id.txtCelular)
34         txtCelular.text = p?.telefono
35
36         return convertViewLocal
37     }
38 }
39
```

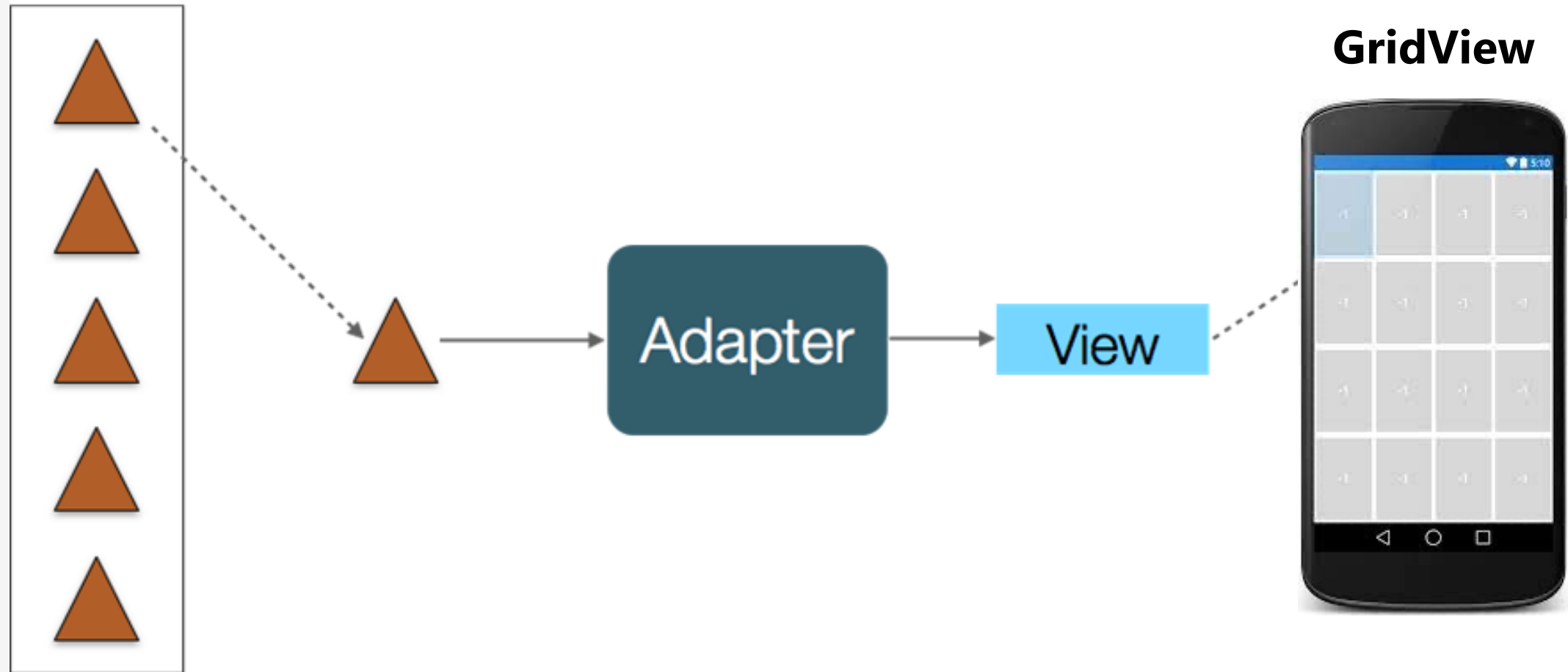
# GridView

---



- **Visualiza elementos en una cuadrícula**
- **Cada elemento del *GridView* puede definirse en un archivo de Layout**

# Pasos para implementar un ListView



## Pasos para implementar un GridView

---

1. **Agregar el widget** `GridView` al Layout
2. Diseñar un **Layout individual** que se repetirá en el grid
3. Personalizar el ***Adapter*** según el Layout Individual.
4. Implementar una actividad que lo visualice el Layout con el `GridView`

## Atributos de un GridView

---

1. **android:numColumns**: indica el número de columnas que se mostrarán en Grid.
2. **android:horizontalSpacing**: el espaciado entre dos columnas de GridView.
3. **android:verticalSpacing**: el espaciado entre dos filas de GridView.

# 1. Diseñar un Layout que contiene un GridView

---

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout xmlns:android="http://schemas.android.com/apk/res-auto"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:id="@+id/frameLayout"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".MainActivity">

    <GridView
        android:id="@+id/gridView1"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:numColumns="2"
        android:horizontalSpacing="6dp"
        android:verticalSpacing="6dp"
    />

</androidx.constraintlayout.widget.ConstraintLayout>
```

## 2. Diseñar el Layout individual para los elementos del GridView

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.cardview.widget.CardView xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    android:layout_width="match_parent"
    android:layout_height="200dp"
    android:layout_margin="5dp"
    app:cardCornerRadius="5dp"
    app:cardElevation="5dp">

    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:orientation="vertical">

        <ImageView
            android:id="@+id/imgViewFoto"
            android:layout_width="120dp"
            android:layout_height="120dp"
            android:layout_gravity="center" />

        <TextView
            android:id="@+id/txtTitulo"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:gravity="center"
            android:text="TextView"
            android:textAppearance="@style/TextAppearance.AppCompat.Large" />

        <TextView
            android:id="@+id/txtAutor"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:gravity="center"
            android:text="TextView"
            android:textAppearance="@style/TextAppearance.AppCompat.Body2" />

    </LinearLayout>
</androidx.cardview.widget.CardView>
```



### 3. Crear un Adapter de acuerdo a nuestros datos

```
public class PersonasAdapter extends ArrayAdapter<Personas> {
    private Context ctx;

    public PersonasAdapter(@NonNull Context context, List<Personas> personasList) {
        super(context, resource: 0, personasList);
        this.ctx = context;
    }

    @NonNull
    @Override
    public View getView(int position, @Nullable View convertView, @NonNull ViewGroup parent) {
        if (convertView == null)
            convertView = LayoutInflater.from(getContext()).inflate(R.layout.item_view, parent, attachToRoot: false);

        ImageView img = convertView.findViewById(R.id.imagenPerfil);
        Personas p = getItem(position);
        Glide.with(ctx)
            .load(p.getImagen())
            .centerCrop()
            .placeholder(R.drawable.usuarioicono)
            .into(img);

        TextView txtNombre = convertView.findViewById(R.id.txtNombre);
        txtNombre.setText(p.getNombre().concat(" ").concat(p.getApellido()));

        TextView txtCelular = convertView.findViewById(R.id.txtCelular);
        txtCelular.setText(p.getTelefono());

        return convertView;
    }
}
```

## 4. Asociar El Adapter al GridView

```
package com.teofilo.usogridview;

import ...

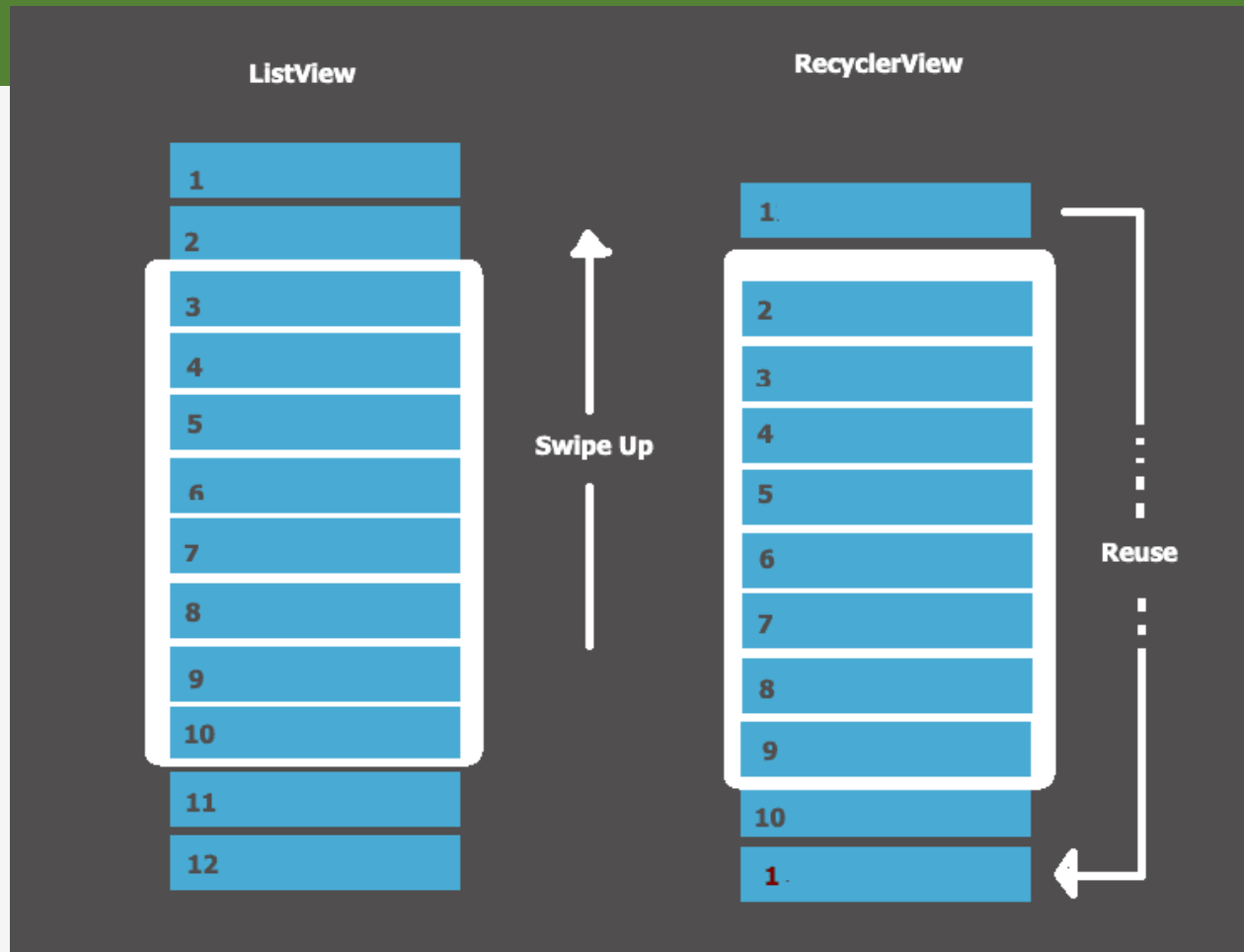
public class MainActivity extends AppCompatActivity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        GridView grid = findViewById(R.id.gridView1);
        List<Libros> librosModel = new ArrayList<Libros>();
        librosModel.add(new Libros( titulo: "Libro 1", imageUrl: "https://m.media-amazon.com/images/I/817R7YEMsvL._AC_UY2
        librosModel.add(new Libros( titulo: "Libro 2", imageUrl: "https://m.media-amazon.com/images/I/817R7YEMsvL._AC_UY2
        librosModel.add(new Libros( titulo: "Libro 3", imageUrl: "https://m.media-amazon.com/images/I/817R7YEMsvL._AC_UY2

        LibrosAdapter adapter = new LibrosAdapter(getApplicationContext(), librosModel);
        grid.setAdapter(adapter);
    }
}
```

# RecyclerView



- Es una versión avanzada de ListView y GridView.
- Utilizado para mostrar una lista de elementos.

## ListView (Ventajas, y Desventajas)

---

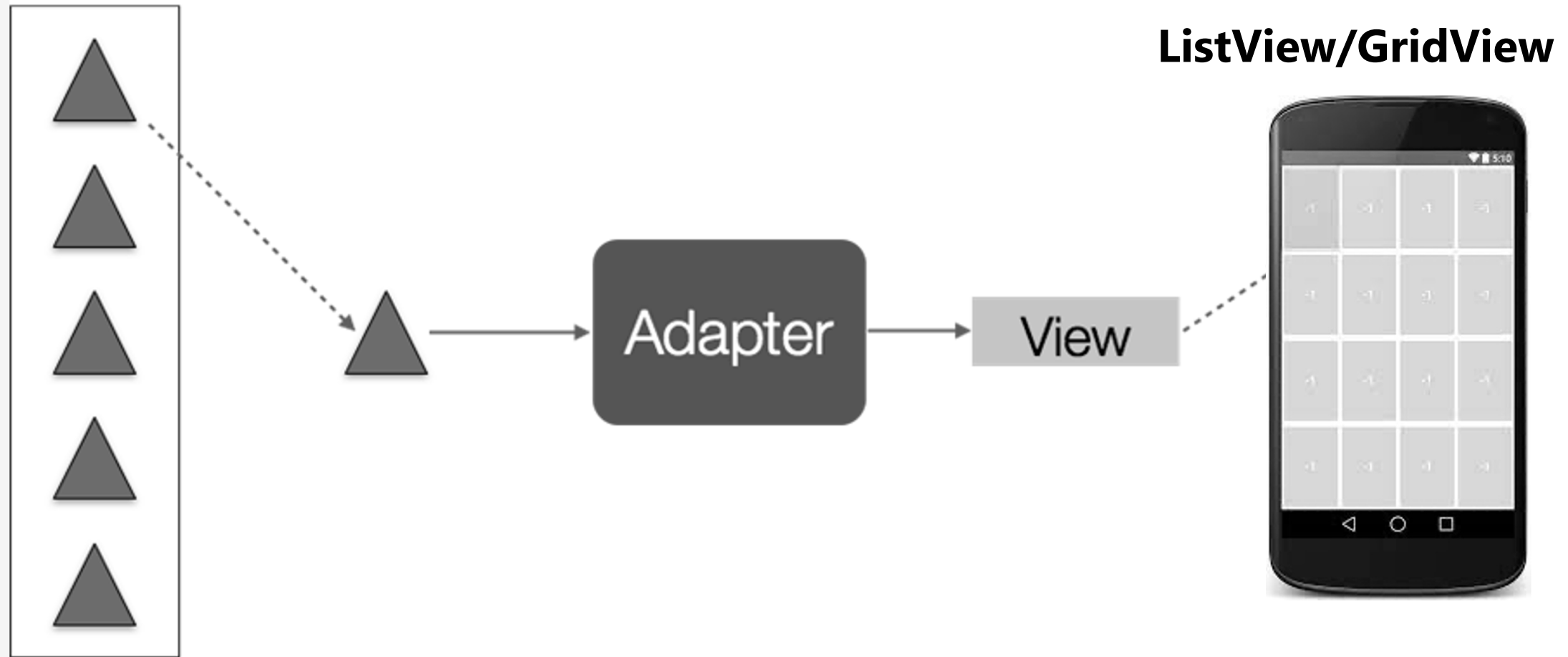
### **Ventajas**

- Fácil de implementar

### **Desventajas**

- Mal rendimiento con enormes datos.
- Sólo lista vertical

## Pasos para implementar un ListView/GridView



# RecyclerView (Ventajas, y Desventajas)

---

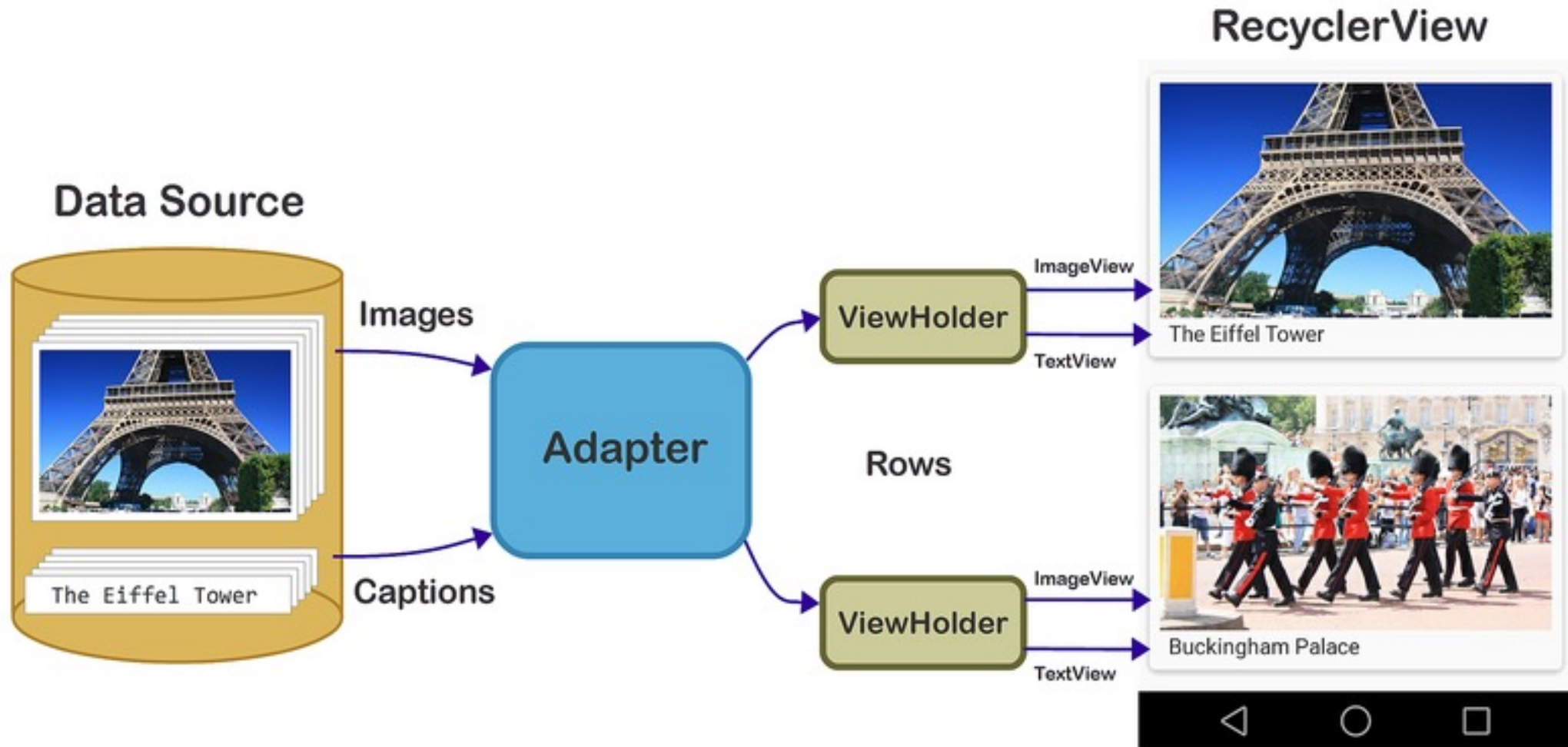
## **Ventajas**

- Animaciones al agregar, actualizar y eliminar elementos
- Decoración de artículos (bordes, separadores)
- Podemos usarlo como una lista o cuadrícula.
- Rendimiento más rápido.

## **Desventajas**

- Más código para su construcción

# Pasos para implementar un RecyclerView



## Pasos para implementar un RecyclerView

---

1. Crear un modelo de datos (Data class)
2. Crear el archivo de diseño en la actividad principal
3. Crear el archivo de diseño de los elementos individuales del RecyclerView.
4. Crear el ***ViewHolder*** y el ***Adapter***.
5. Definir un LayoutManager (*LinearLayoutManager*, *GridLayoutManager*, *StaggeredGridLayoutManager*)
6. Vincular el Adapter al RecyclerView



# 1. Creación del Modelo de Datos

---

```
data class Persona(  
    var nombre: String? = null,  
    var apellido: String? = null,  
    var telefono: String? = null,  
    var imagen: String? = null  
) {  
  
    override fun toString(): String {  
        return nombre ?: ""  
    }  
  
}
```

## 2. El archivo de diseño en la Actividad Principal

The image shows the Android Studio interface with the XML code for `activity_main.xml` on the left and a visual preview on the right.

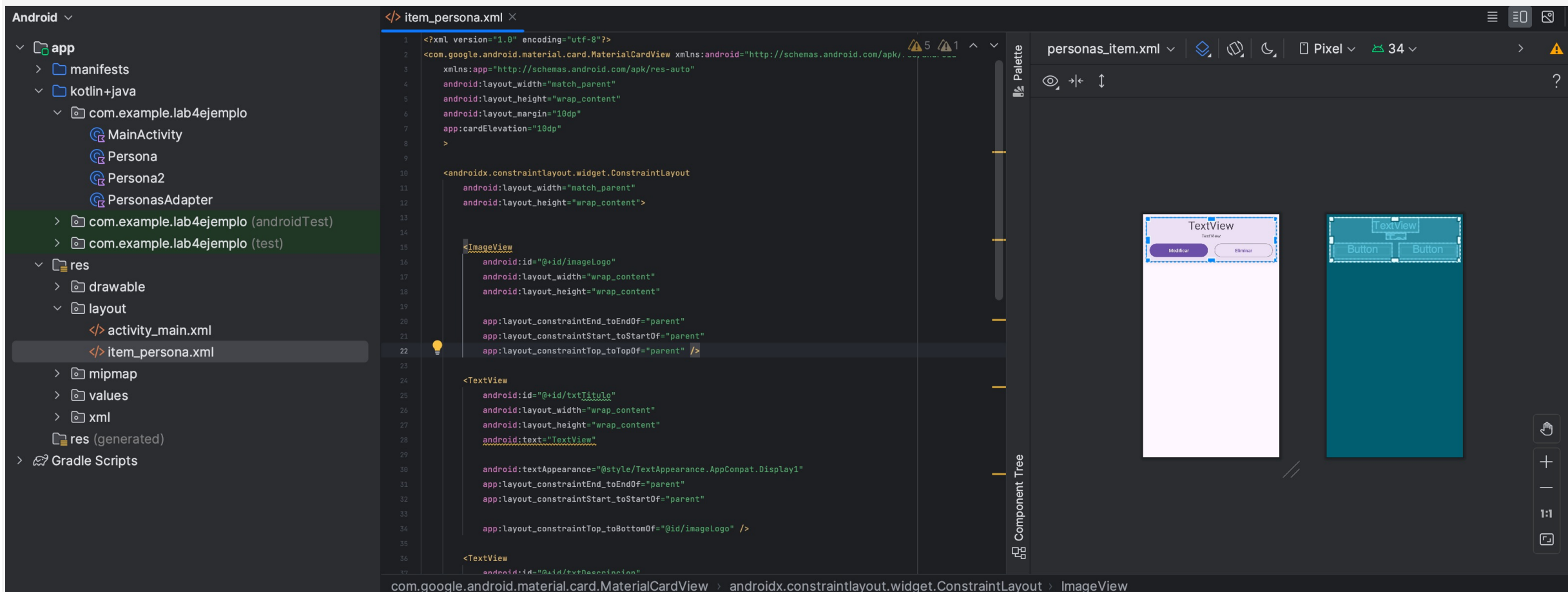
**XML Code:**

```
1 <?xml version="1.0" encoding="utf-8"?>
2 <androidx.constraintlayout.widget.ConstraintLayout xmlns:android="http://schemas.android.com/apk/res-auto"
3   xmlns:app="http://schemas.android.com/apk/res-auto"
4   xmlns:tools="http://schemas.android.com/tools"
5
6   android:layout_width="match_parent"
7   android:layout_height="match_parent"
8   tools:context=".MainActivity">
9
10  <androidx.recyclerview.widget.RecyclerView
11    android:id="@+id/recycler1"
12    android:layout_width="match_parent"
13    android:layout_height="match_parent"
14    app:layout_constraintBottom_toBottomOf="parent"
15    app:layout_constraintEnd_toEndOf="parent"
16    app:layout_constraintStart_toStartOf="parent"
17    app:layout_constraintTop_toTopOf="parent" />
18 </androidx.constraintlayout.widget.ConstraintLayout>
```

**Visual Preview:**

The visual preview on the right shows two versions of the layout: a light theme (left) and a dark theme (right). Both show a white background with a blue dashed border and four blue circular handles at the corners, indicating a resizable widget. The dark theme version is labeled `recycler1`.

### 3. Crear el archivo de diseño (layout) para los ítems del RecyclerView.



## 4.1. Crear el Adapter que contiene el ViewHolder

```
package com.example.lab4ejemplo

import android.content.Context
import android.view.LayoutInflater
import android.view.View
import android.view.ViewGroup
import android.widget.ImageView
import android.widget.TextView
import androidx.recyclerview.widget.RecyclerView

class PersonasAdapter (ctx: Context, private val personasModel: List<Persona> ):RecyclerView.Adapter<PersonasAdapter.PersonasViewHolder>(){

    inner class PersonasViewHolder(itemView: View):RecyclerView.ViewHolder(itemView){
        var imageUrl = itemView.findViewById<ImageView>(R.id.imageLogo)
        var titulo = itemView.findViewById<TextView>(R.id.txtTitulo)
        var descripcion = itemView.findViewById<TextView>(R.id.txtDescripcion)
    }

    override fun onCreateViewHolder(parent: ViewGroup, viewType: Int): PersonasViewHolder {
        val v = LayoutInflater.from (parent.context).inflate(R.layout.item_persona,parent, attachToRoot: false)
        return PersonasViewHolder(v)
    }

    override fun getItemCount(): Int {
        return personasModel.size
    }


    override fun onBindViewHolder(holder: PersonasViewHolder, position: Int) {
        val i = personasModel[position]
        holder.titulo.text = i.nombre
        holder.descripcion.text = i.apellido
    }
}
```

## 6. Vincular el Adapter con el RecyclerView

```
7
8 </> class MainActivity : AppCompatActivity() {
9     override fun onCreate(savedInstanceState: Bundle?) {
10         super.onCreate(savedInstanceState)
11         setContentView(R.layout.activity_main)
12         val r = findViewById<RecyclerView>(R.id.recycler1)
13         val personasList = mutableListOf<Persona>(
14             Persona( nombre: "TEOFILO", apellido: "COPA", telefono: "74859632", imagen: "https://fastly.picsum.photos/id/237/200/300.jpg?hmac=TmmQSbShHz9CdQm0NkEjx1Dyh_Y98",
15             Persona( nombre: "MARIBEL", apellido: "ACUÑA", telefono: "74859632", imagen: "https://fastly.picsum.photos/id/237/200/300.jpg?hmac=TmmQSbShHz9CdQm0NkEjx1Dyh_Y98",
16             Persona( nombre: "JUANA", apellido: "AGUIRRE", telefono: "74859632", imagen: "https://fastly.picsum.photos/id/237/200/300.jpg?hmac=TmmQSbShHz9CdQm0NkEjx1Dyh_Y98",
17             Persona( nombre: "GUADALUPE", apellido: "MEJIA", telefono: "74859632", imagen: "https://fastly.picsum.photos/id/237/200/300.jpg?hmac=TmmQSbShHz9CdQm0NkEjx1Dyh_Y98",
18             Persona( nombre: "SILVANA", apellido: "PAZ", telefono: "74859632", imagen: "https://fastly.picsum.photos/id/237/200/300.jpg?hmac=TmmQSbShHz9CdQm0NkEjx1Dyh_Y98",
19             Persona( nombre: "MARIO", apellido: "ORELLANA", telefono: "74859632", imagen: "https://fastly.picsum.photos/id/237/200/300.jpg?hmac=TmmQSbShHz9CdQm0NkEjx1Dyh_Y98"
20         )
21     }
22
23     val personasAdapter = PersonasAdapter(applicationContext, personasList)
24     r.adapter = personasAdapter
25
26     val layout = LinearLayoutManager(applicationContext)
27     layout.orientation = LinearLayoutManager.VERTICAL
28     r.layoutManager = layout
29
30
31
32 }
33 }
```

## 5. Definir el LayoutManager

---

```
25  
26  val layout = LinearLayoutManager(applicationContext)  
27 layout.orientation = LinearLayoutManager.VERTICAL  
28 r.layoutManager = layout  
29  
30  
31
```

# Demostración



# Demostración