

# 2025

Programación  
Multimedia y  
Dispositivos Móviles  
(PMDM)

# [ANDROID: COMPONENTE LISTVIEW]

## Tabla de contenido

1. Introducción.....	3
2. Adaptador de Android .....	3
3. Selección de elementos .....	5
4. Personalización de los ítems.....	6

## 1. Introducción

Un ListView en Android Studio es un control que nos permite mostrar una lista de elementos en varias filas, desplazables, y que contiene un adaptador que inserta automáticamente los elementos en la lista.

El objetivo principal del adaptador es obtener datos de un array, matriz o base de datos e insertar cada elemento que se coloca en la lista para obtener el resultado deseado. Por lo tanto, es la fuente principal para extraer datos del archivo strings.xml que contiene todas las cadenas de caracteres requeridas.

## 2. Adaptador de Android

El adaptador contiene los datos extraídos de una matriz y recorre cada elemento del conjunto de datos generando las vistas correspondientes para cada elemento de la lista.

Actúa como **intermediario** entre las fuentes de datos y las vistas del adaptador.

### Tipos de adaptadores:

- **Adaptador de matriz:** siempre acepta una matriz o lista como entrada. Los elementos de la lista se pueden almacenar en el archivo strings.xml.
- **Adaptador de cursor:** siempre acepta una instancia del cursor como medio de entrada.
- **Adaptador simple:** Permite crear configuraciones de menú alternativas para diferentes plataformas, tamaños de pantalla y otros parámetros de configuración.
- **Adaptador de base:** es una implementación genérica para los tres tipos de adaptadores y se puede utilizar en las vistas según necesitemos.

### Ejemplo:

#### Archivo list\_view.xml:

```
<?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/main"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".ListViewActivity">

    <TextView
        android:id="@+id/textView10"
        android:layout_width="0dp"
        android:layout_height="wrap_content"
        android:layout_marginStart="20dp"
```

```

        android:layout_marginTop="20dp"
        android:text="MONTAÑAS"
        android:textAppearance="@style/TextAppearance.AppCompat.Large"
        android:textStyle="bold"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent" />

<ListView
    android:id="@+id/list_view_montain"
    android:layout_width="0dp"
    android:layout_height="wrap_content"
    android:layout_marginStart="20dp"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/textView10" />
</androidx.constraintlayout.widget.ConstraintLayout>
```

### Archivo ListViewActivity.kt:

```

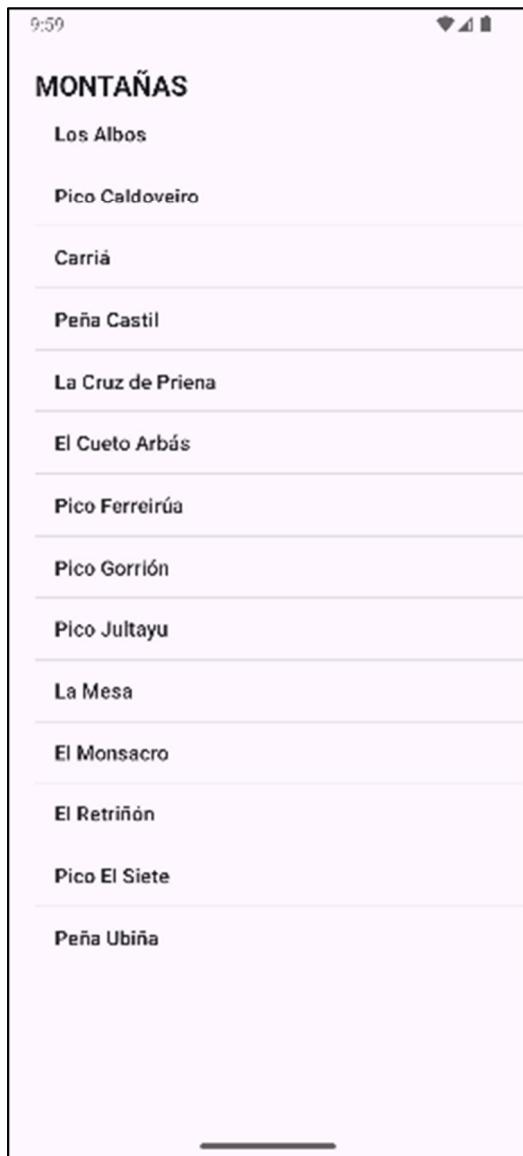
class ListViewActivity : AppCompatActivity() {

    val montanas = arrayOf("Los Albos", "Pico Caldoveiro", "Carriá",
        "Peña Castil", "La Cruz de Priena", "El Cueto Arbás", "Pico
        Ferreirúa", "Pico Gorrión", "Pico Jultayu", "La Mesa", "El
        Monsacro", "El Retriñón", "Pico El Siete", "Peña Ubiña")

    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        enableEdgeToEdge()
        setContentView(R.layout.list_view)

        ViewCompat.setOnApplyWindowInsetsListener(findViewById(R.id.mai)) {
            v, insets ->
            val systemBars =
                insets.getInsets(WindowInsetsCompat.Type.systemBars())
            v.setPadding(systemBars.left, systemBars.top,
                systemBars.right, systemBars.bottom)
            insets
        }

        val lista = findViewById<ListView>(R.id.list_view_montain)
        val adaptador: ArrayAdapter<String> = ArrayAdapter<String>(this,
            android.R.layout.simple_list_item_activated_1, montanas)
        lista.adapter = adaptador
    }
}
```



### 3. Selección de elementos

Al hacer click sobre cada elemento de la lista podemos saber la posición que éste ocupa dentro de la lista y recuperar la información del ítem a través del método **setOnItemClickListener**.

**Ejemplo** (continuando el código anterior):

```
lista.setOnItemClickListener { adapterView, view, i, l ->
    val nombre_montania = adaptador.getItem(i)
    Toast.makeText(this, nombre_montania, Toast.LENGTH_SHORT).show()
}
```

## 4. Personalización de los ítems

Por defecto, Android Studio nos facilita layouts prediseñados para los ítems de un ListView: simple\_list\_item\_1.xml, simple\_list\_item\_2.xml, simple\_list\_item\_activated\_1.xml, simple\_list\_item\_activated\_2.xml, etc.

En esta página <https://developer.android.com/reference/android/R.layout> se pueden consultar todos los layouts de Android, entre los que están los que afectan a los ítems de un ListView.

Puede que esos diseños no se ajusten a nuestras necesidades, así que podemos realizar **nuestros propios layouts** para los ítems.

**Ejemplo:** Mostrar una lista de montañas, donde cada ítem está formado por una imagen, un texto con el nombre de la montaña y otro texto con el concejo de la misma.



```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
    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="wrap_content"
    android:orientation="horizontal">

    <ImageView
        android:id="@+id/minatura"
        android:layout_width="0dp"
        android:layout_height="wrap_content"
        android:layout_gravity="start"
        android:layout_weight="1"
        android:contentDescription="@string/desc_icono"
        app:srcCompat="@android:drawable/ic_menu_mapmode" />

    <TextView
        android:id="@+id/nombre"
        android:layout_width="0dp"
        android:layout_height="wrap_content"
        android:layout_gravity="center"
        android:layout_weight="1"
        android:gravity="start"
        android:text="" />

    <TextView
        android:id="@+id/concejo"
        android:layout_width="0dp"
        android:layout_height="wrap_content"
        android:layout_gravity="center"
        android:layout_weight="1"
        android:gravity="start"
        android:text="" />
</LinearLayout>
```

Como ya se comentó, el adaptador es el intermediario entre los datos y la vista de los ítems, por lo tanto, generalmente, si tenemos un layout personalizado para los ítems se va a necesitar también:

- Una clase para los datos
- Un adaptador personalizado: suele heredar de la clase **BaseAdapter**.

**Ejemplo:** siguiendo con el ejemplo anterior, necesitamos almacenar la información para cada montaña, es decir, una imagen, su nombre y su concejo.

Además, el adaptador ha de cargar la información anterior en el diseño personalizado para cada ítem.

### ListData.kt

```
class ListData(val nombre: String, val concejo: String, val imagen: Int)
```

### ListAdapter.kt

```
class ListAdapter(): BaseAdapter(){  
    private var context: Context? = null  
    private var datos: ArrayList<ListData> = ArrayList<ListData>()  
    constructor(context: Context, datos: ArrayList<ListData>):this() {  
        this.context = context  
        this.datos = datos  
    }  
    override fun getCount(): Int {  
        return datos.size  
    }  
    override fun getItem(position: Int): Any {  
        return datos[position]  
    }  
    override fun getItemId(position: Int): Long {  
        return position.toLong()  
    }  
    override fun getView(position: Int, view: View?, parent:  
        ViewGroup? ) : View {  
        val vista = view?.let {  
            view  
        } ?:run{  
            LayoutInflater.from(context).inflate(R.layout.item_view,  
                parent, false)  
        }  
        val icono = vista.findViewById<ImageView>(R.id.miniatura)  
        val nombre_mountain = vista.findViewById<TextView>(R.id.nombre)  
        val concejo_mountain = vista.findViewById<TextView>(R.id.concejo)  
        val datosItem = getItem(position) as ListData  
        //icono.setImageResource(datosItem.imagen) -> si se comenta,  
        el mismo icono para todas las montañas. Si no, cada montaña  
        tiene su propio icono.  
        nombre_mountain.text = datosItem.nombre  
        concejo_mountain.setText(datosItem.concejo)  
        return vista;  
    }  
}
```

## MainActivity.kt

```
class MainActivity : AppCompatActivity() {
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        enableEdgeToEdge()
        setContentView(R.layout.list_view)
        ViewCompat.setOnApplyWindowInsetsListener(findViewById(R.id.main)) { v, insets -
            val systemBars =
                insets.getInsets(WindowInsetsCompat.Type.systemBars())
            v.setPadding(systemBars.left, systemBars.top, systemBars.right,
                systemBars.bottom)
            insets
        }
        val datos = this.obtenerDatos()
        val lista = findViewById<ListView>(R.id.list_view_montain)
        val adaptador =ListAdapter(this, datos)
        lista.adapter = adaptador
        lista.setOnItemClickListener { parent, view, position, id ->
            val montania = adaptador.getItem(position) as ListData
            val intent = Intent(this, DetalleMontania::class.java)
            intent.putExtra("nombre", montania.nombre)
            intent.putExtra("concejo", montania.concejo)
            intent.putExtra("imagen", montania.imagen)
            startActivity(intent)
        }
    }
    private fun obtenerDatos(): ArrayList<ListData>{
        val datos = arrayListOf<ListData>()
        datos.add(ListData("Los Albos", "Somiedo", R.drawable.albos))
        datos.add(ListData("Pico Caldoveiro", "Teverga",
            R.drawable.caldoveiro))
        datos.add(ListData("Pico Carriá", "Ponga", R.drawable.carria))
        datos.add(ListData("Peña Castil", "Cabrales", R.drawable.castil))
        datos.add(ListData("La Cruz de Priena", "Cangas de Onís",
            R.drawable.cruzpriena))
        datos.add(ListData("El Cueto Arbás", "Cangas del Narcea",
            R.drawable.cuetobarbas))
        datos.add(ListData("Pico Ferreirúa", "Teverga",
            R.drawable.ferreirua))
        datos.add(ListData("Pico Gorrión", "Quirós", R.drawable.gorrion))
        datos.add(ListData("Pico Jultayu", "Cangas de Onís",
            R.drawable.jultayu))
        datos.add(ListData("La Mesa", "Lena", R.drawable.lamesa))
        datos.add(ListData("El Monsacro", "Morcín", R.drawable.monsacro))
        datos.add(ListData("El Retriñón", "Aller", R.drawable.retrinon))
        datos.add(ListData("Pico El Siete", "Lena", R.drawable.siete))
        datos.add(ListData("Peña Ubiña", "Lena", R.drawable.ubina))
        return datos;
    }
}
```