

Apuntes

[Descargar estos apuntes](#)

Tema 8. Interfaz de Usuario III

Índice

1. [Spinner](#)

Spinner

Las listas desplegables en Android se llaman **Spinners**. El usuario selecciona de la lista emergente un elemento que es el seleccionado para el control. El código de la vista de un Spinner será el siguiente:

```
<Spinner
    android:id="@+id/spinner"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content" />
```

Los elementos que mostrará nuestro Spinner no se pueden indicar directamente, sino que es necesario crear una estructura de elementos que podamos asociar a éste en nuestro código. Una de las maneras sería usando un Array de Strings para luego asignarlo al spinner:

```
var colores = arrayOf("Rojo", "Verde", "Azul")
```

A continuación debemos definir el adaptador, en nuestro caso un objeto ArrayAdapter, que será el objeto que usaremos con nuestro Spinner.

```
val adaptador = ArrayAdapter(this, android.R.layout.simple_list_item_1, colores)
```

Creamos el adaptador en sí, al que pasamos 3 parámetros:

1. El contexto, que normalmente será simplemente una referencia a la actividad donde se crea el adaptador.
2. El ID del layout sobre el que se mostrarán los datos del control. En este caso le pasamos el ID de un layout predefinido en Android (android.R.layout.simple_spinner_item), formado únicamente por un control textView, pero podríamos pasar el ID de cualquier layout de nuestro proyecto con cualquier estructura y conjunto de controles.
3. El elemento que contiene los datos a mostrar.

Con estas simples líneas tendremos un control similar al de la siguiente imagen:



📁 Vamos a crear un proyecto de ejemplo llamado Spinner al que añadiremos el siguiente código en la **MainActivity.kt**, y no olvides añadir un spinner con *id=spinner* en el layout **activity_main.xml**:

```
class MainActivity : AppCompatActivity(),
    AdapterView.OnItemSelectedListener {
    var colores = arrayOf("Rojo", "Verde", "Azul")
    lateinit var listaColores: Spinner
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_main)
        8      val adaptador = ArrayAdapter(this, android.R.layout.
                                   simple_list_item_1, colores)
        listaColores = findViewById(R.id.spinner)
        11     listaColores.adapter=adaptador
        12     listaColores.setOnItemSelectedListener(this)
    }
```

📌 **Línea 8** creamos el adaptador, como hemos explicado anteriormente, para asignarlo al Spinner en la **Línea 11**.

En cuanto a los eventos lanzados por el control Spinner, el más utilizado será el generado al seleccionarse una opción de la lista desplegable, `onItemSelected`. Para capturar este evento se procederá de forma similar a lo ya visto para otros controles anteriormente, asignándole su controlador mediante el método **`setOnItemSelectedListener()`**, en este caso lo hacemos mediante el método de derivar de la interfaz, **Línea 12**.

En el código inferior vemos implementado el método, que muestra un Toast con el color elegido.

```
override fun onItemSelected(adapterView: AdapterView<*>,
    view: View?, i: Int, l: Long){
    val x = Toast.makeText(this, "El color elegido es: " +
        colores[i], Toast.LENGTH_LONG)
    x.show()
}
```

Nota ampliativa

Una alternativa a tener en cuenta, si los datos a mostrar en el control son estáticos, sería definir la lista de posibles valores como un recurso de tipo string-array. Para ello, primero crearíamos un nuevo fichero XML en la carpeta **res/values** llamado por ejemplo **spinner.xml** e incluiríamos en él los valores seleccionables de la siguiente forma:

```
<?xml version="1.0" encoding="utf-8"?>
<resources>
  <string-array name="colores">
    <item>ROJO</item>
    <item>VERDE</item>
    <item>AZUL</item>
  </string-array>
</resources>
```

En este caso, a la hora de crear el adaptador utilizaríamos el método

`createFromResource()` para hacer referencia al array XML que acabamos de crear:

```
val adaptador=ArrayAdapter.createFromResource(this,R.array.colores,
                                              android.R.layout.simple_list_item_
```

Ejercicio Propuesto Spinner