

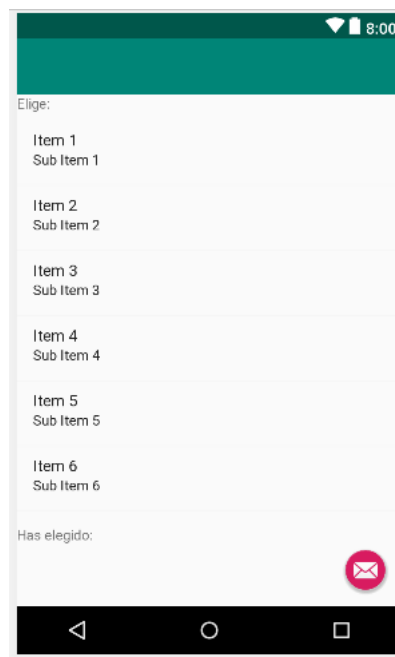
Widgets selección

Elementos que permiten al usuario seleccionar un valor de una lista evitándonos cientos de comprobaciones para validar entradas del usuario.

El más típico en Android es **ListView**, que también es una subclase de la clase ViewGroup, por tanto también es un contenedor pero con la peculiaridad de los adaptadores. Los adaptadores son un objeto que actúa de puente entre un widget y sus datos.

1.1.- Selección simple

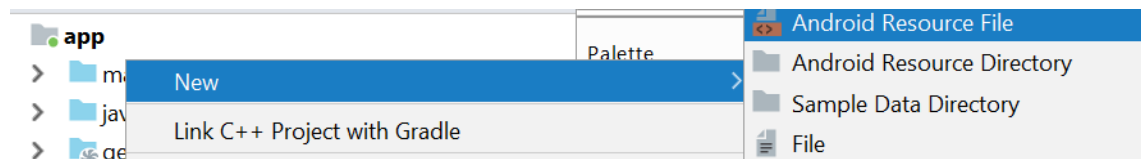
Para este apartado, crea un proyecto con un ListView y dos etiquetas:



Para poder llenar la lista de contenido necesitarás dos cosas:

- Un adaptador
- Un nuevo recurso a nodo de layout file con la definición de un TextView que se usará como elemento simple de tu ListView

Para ello, en el código fuente declaras un objeto de esta clase y lo rellenas con los elementos del array. Estos elementos, a través de un adaptador, serán canalizados hasta los elementos individuales que contenga la lista. Lo primero es crear un fichero de recursos, por ejemplo “fila.xml”



Y en dicho fichero inserta el siguiente código:

```
<TextView xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="vertical"
    android:layout_width="match_parent"
```

```

        android:layout_height="wrap_content"
        android:textStyle="italic"
        android:textSize="20dp"
        android:textColor="#FF0000"
    />

```

A continuación vamos a programar el método onCreate:

```

@Override
protected void onCreate(Bundle savedInstanceState) {
    //Crear un array con los elementos seleccionables
    String [] elementos ={ "Jaén", "Córdoba", "Sevilla", "Huelva", "Cádiz", "Málaga", "C
    //Declara un adaptador de texto (String)
    ArrayAdapter<String> adaptador;

    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);

    //Obtienes la referencia de la lista
    ListView l =(ListView)findViewById(R.id.lista);

    //Creas el adaptador
    adaptador = new ArrayAdapter<String>( context: this, R.layout.fila,elementos);

    //Le das el adaptador a la lista
    l.setAdapter(adaptador);

```

Fíjate que para crear el adaptador, el constructor de la clase ArrayAdapter recibe tres parámetros:

- La referencia a la propia actividad (this)
- Referencia al fichero fila.xml
- Los elementos que tendrá en array

Si lo pruebas verás algo como la imagen lateral.

Solo falta que la seleccionada aparezca a continuación de

“Has elegido:”

Para ello tendremos que registrar el evento con setOnItemClickListener(this) y recibirlo con el método OnItemClickListener de la interface OnItemClickListener:

```

public class MainActivity extends AppCompatActivity implements
    ListView.OnItemClickListener{
    . . .
    @Override
    public void onItemClick(AdapterView<?> adapterView, View view, int i, long l) {
        TextView t = (TextView)findViewById(R.id.TextoSelect);
        t.setText("Has elegido:" +adapterView.getItemAtPosition(i).toString());
    }
    . . .
}

```

Fíjate que como parámetros tiene el padre con que hizo la selección (AdapterView<?>), ña View (el widget) que se pulsó, la posición en la lista y el identificador de la fila que se seleccionó.

```

public void onItemClick(AdapterView<?> adapterView, View view, int i, long l)

```



Este método es un callback que se invoca cuando se hace clic en un elemento de la lista. Puedes con la posición (i) elegir el elemento seleccionado con el método `getItemAtPosition` de la siguiente forma:

```
t.setText("Has elegido:" +adapterView.getItemAtPosition(i).toString());
```

O de la siguiente alternativa:

```
t.setText("Has elegido:" + ((TextSelect)view).getText());
```

Recuerda registrar el listener de la lista para que se realice el callback

```
l.setOnItemClickListener(this);
```

1.2.- Selección multiple

Un ListView se puede configurar para poder escoger selección múltiple. Dos formas:

- En el XML mediante el atributo `android:choiceMode="multipleChoice"`
- En el código con el método `setChoiceMode(CHOICE_MODE_MULTIPLE)`

Además deberás utilizar como Layout para la selección el predefinido por Android para la selección múltiple, que será el segundo elemento del adaptador:

```
adaptador = new ArrayAdapter<String>(this,  
    android.R.layout.simple_list_item_multiple_choice, elementos);
```

Los métodos para capturar los eventos de selección funcionan exactamente igual que en un ListView de selección simple. Para mostrar todos los elementos seleccionados tendremos que usar un bucle como el for siguiente:

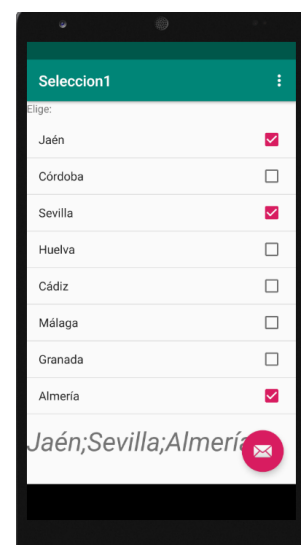
```
public void onItemClick(AdapterView<?> adapterView, View view, int i, long l) {  
    TextView t = (TextView)findViewById(R.id.TextoSelect);  
    ListView lis=(ListView)findViewById(R.id.lista);  
    String seleccionado = new String();  
    SparseBooleanArray checked = lis.getCheckedItemPositions();  
  
    for (int k=0; k<checked.size();k++){  
        if (checked.valueAt(k)){  
            seleccionado = seleccionado+  
                adapterView.getItemAtPosition(checked.keyAt(k)).toString()+" ";  
        }  
        t.setText(seleccionado);  
    }  
}
```

Esta fracción de código responde al evento de hacer una selección en uno de los elementos de la ListView. Utiliza un array booleano (`SparseBooleanArray`) para obtener de la lista los elementos que están seleccionados (`checked`):

```
SparseBooleanArray checked = lis.getCheckedItemPositions();
```

El método `getCheckedItemPositions` de la ListView devuelve este tipo de array para que puedas consultar los valores.

Para saber más sobre `SparseBooleanArray` puedes consultar la documentación de Android:



<https://developer.android.com/reference/android/util/SparseBooleanArray>

pero en resumen, el método `valueAt(k)` devuelve un boolean indicando si el elemento `k` del array fue seleccionado o no. Gracias al método `keyAt(k)` de este array, devuelve el número indicando la posición de un elemento seleccionando dentro de la `ListView`. De esta manera se puede hacer un bucle desde 0 hasta el tamaño del array, así como extraer los valores múltiples seleccionados.