

Programación Multimedia y dispositivo móviles

CONTROLES DE ENTRADA

Teresa Carrero

Desarrollo Aplicaciones Multiplataforma

Table of contents

1. Objetivos

2. TextView

Propiedades del TextView

3. EditText

Propiedades del EditText

4. Conclusión

Objetivos

- Conocer los controles básicos de Android.
- Manejar las etiquetas de texto y sus atributos.
- Utilización de las etiquetas de texto en la clase MainActivity
- Utilización de las etiquetas de texto en el fichero activity_main

TextView



Este control es uno de los más usados, tiene la única función de mostrar un texto en pantalla.

Atributos mas importantes

- **android:text**. Define cuál será el texto que se mostrará en pantalla.
- **android:background**. Establece el color de fondo.
- **android:textColor**. Asigna el color del texto.
- **android:textSize**. Determina el tamaño del texto.
- **android:textStyle**. Elige el estilo del texto: normal, negrita (bold) o cursiva (italic).
- **android:typeface**. Fija el tipo de letra por el que podemos optar: sans, monospace o serif.

TextView en main_activity

Para incluirlo en el fichero de la vista (XML) de la aplicación tan solo deberá ser añadido dentro de un objeto ViewGroup (Layout) de la siguiente manera:

```
1 <TextView
2     android:id="@+id/textview_titulo"
3     android:layout_width="wrap_content"
4     android:layout_height="wrap_content"
5     android:layout_marginTop="32dp"
6     android:text="Hello World!"
7     app:layout_constraintLeft_toLeftOf="parent"
8     app:layout_constraintRight_toRightOf="parent"
9     app:layout_constraintTop_toTopOf="parent" />
```

TextView en ActivityMain

Para manipular este objeto de texto desde Java, primero deberá asociarse a la vista, para lo que es necesario importar la librería del widget (esto suele hacerlo de manera automática Android Studio cuando se declara la variable) que da acceso a los objetos incluidos en el interface. En este caso, se necesitará import `android.widget.TextView`

```
import androidx.appcompat.app.AppCompatActivity;

import android.os.Bundle;
import android.widget.TextView; //importar la libreria

public class MainActivity extends AppCompatActivity {

    //iniciar variable
    private TextView tvTitulo; //declarar la variable

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


TextView en ActivityMain

A continuación se deberá instanciar el objeto y acceder al recurso, se hará de la siguiente manera:

```
1 //instanciar el textview
2 tvTitulo = findViewById(R.id.textview_titulo);
```

Donde **tvTitulo** es el nombre que adquirirá esta instancia en Java y **R.id.textview_titulo** es el nombre con el que lo identificamos en el fichero XML de la vista asociada a este objeto (**android:id="@+id/textview_titulo"**).

TextView en ActivityMain

Una vez hecho esto, se pueden obtener (get) o modificar (set) los atributos de este objeto, y así, si se quisiera cambiar el contenido de la etiqueta de texto, tan solo se tendría que escribir:

```
1 // cambiar el contenido de el textview  
2 tvTitulo.setText("Listado de elementos");
```

TextView en ActivityMain

Puede ser manipulado un gran número de propiedades de este objeto desde Java y desde XML.

Vamos a modificar el color de una etiqueta de texto desde Java, se utilizará el comando **setTextColor()**, incluyendo entre paréntesis el color que se va a aplicar, lo que puede hacerse de variadas formas (en todas ellas se aplica el color rojo a la etiqueta de texto):

```
1 //modificar el color del TextView
2 tvTitulo.setTextColor(0xffff0000); //en hexadecimal
3 //utilizando la libreria Color con la constante
4 tvTitulo.setTextColor(Color.RED);
5 //utilizando la libreria Color con rgb
6 tvTitulo.setTextColor(Color.rgb(255,0,0));
7 //utilizando la libreria Color con argb con transparencia
8 tvTitulo.setTextColor(Color.argb(255,255,0,0));
9 //utilizando la libreria Color utilizando una cadena
10 tvTitulo.setTextColor(Color.parseColor("#ff0000"));
```

TextView en main_activity

Desde el fichero main_activity también se puede añadir y modificar propiedades. Solo hay que indicar la propiedad y poner el valor.

```
1 <TextView
2     android:id="@+id/textview_titulo"
3     android:layout_width="wrap_content"
4     android:layout_height="wrap_content"
5     android:layout_marginTop="32dp"
6     android:text="Hello World!"
7     android:textColor="@color/purple_500"
8     app:layout_constraintLeft_toLeftOf="parent"
9     app:layout_constraintRight_toRightOf="parent"
10    app:layout_constraintTop_toTopOf="parent" />
```

Edit Text



Es el objeto más simple que nos ofrece Android para leer entradas de texto. Este objeto hereda todas las propiedades de TextView.

Algunos atributos específicos

- **android:inputType**. Define cuál será el tipo de texto que se puede introducir, nos mostrara el teclado que mas se ajuste al tipo de texto a introducir.
- **android:hint**. Establece un texto que desaparece cuando se pincha en el elemento.
- **android:ems**. Establece el tamaño de la fuente

TextView en main_activity

Para incluirlo en el fichero de la vista (XML) de la aplicación tan solo deberá ser añadido dentro de un objeto ViewGroup (Layout) de la siguiente manera:

```
1 <EditText
2     android:id="@+id/editTextTextPersonName"
3     android:layout_width="wrap_content"
4     android:layout_height="wrap_content"
5     android:layout_marginTop="32dp"
6     android:ems="10"
7     android:inputType="textPersonName"
8     android:hint="Name"
9     app:layout_constraintEnd_toEndOf="parent"
10    app:layout_constraintStart_toStartOf="parent"
11    app:layout_constraintTop_toBottomOf="@+id/
    textview_titulo" />
```

EditText en ActivityMain

Para manipular este objeto de texto desde Java, primero deberá asociarse a la vista, para lo que es necesario importar la librería del widget (esto suele hacerlo de manera automática Android Studio cuando se declara la variable) que da acceso a los objetos incluidos en el interface. En este caso, se necesitará import `android.widget.EditText`

```
import android.os.Bundle;
import android.widget.EditText; //importa la libreria
import android.widget.TextView; //importar la libreria

public class MainActivity extends AppCompatActivity {

    //iniciar variable
    private EditText etNombre; //declaramos la variable

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

        //instanciamos el editText
        etNombre = findViewById(R.id.editTextTextPersonName);
    }
}
```


TextView en ActivityMain

A continuación se deberá instanciar el objeto y acceder al recurso, se hará de la siguiente manera:

```
1 //instanciaremos el editText
2 etNombre = findViewById(R.id.editTextTextPersonName);
```

Donde **etNombre** es el nombre que adquirirá esta instancia en Java y **R.id.editTextTextPersonName** es el nombre con el que lo identificamos en el fichero XML de la vista asociada a este objeto (**android:id="@+id/editTextTextPersonName"**).

Propiedad `inputType`

Por defecto, `EditText` lee texto plano y asocia el teclado textual del dispositivo móvil, pero dispone de numerosas opciones, gracias al modificador **`inputType`**.

- **`textUri`**. Texto que se usará como URI.
- **`textEmailAddress`**. Texto que se usará como dirección de correo.
- **`textPersonName`**. Nombre de una persona.
- **`textPassword`**. Contraseña.
- **`number`**. Entrada numérica.
- **`phone`**. Entrada de un número de teléfono.
- **`textMultiLine`**. Permite multilínea.
- **`textNoSuggestions`**. Desactiva la escritura predictiva.

Conclusión



Dudas?

