PROGRAMACIÓN MULTIMEDIA Y DISPOSITIVOS MÓVILES

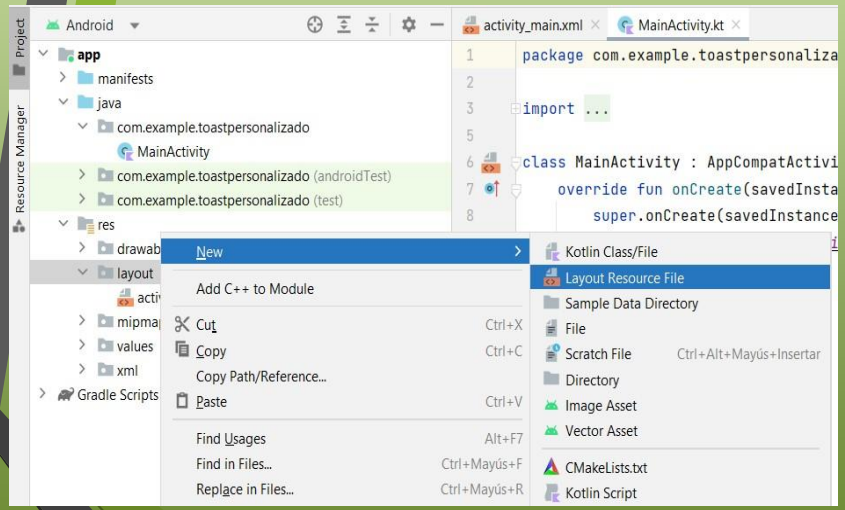
ANEXO TEMA 6:

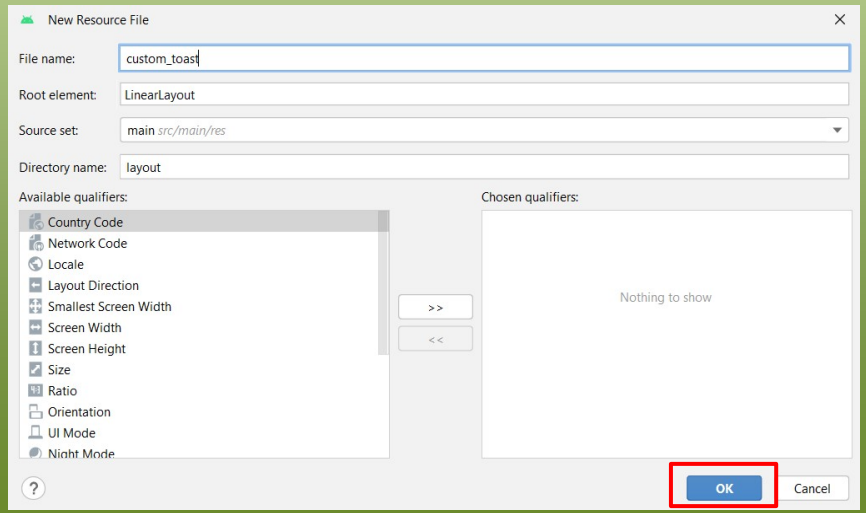
Crear Toast personalizado

UNION EUROPEA
Fondo Social Europeo
"Una manera de hacer Europa"
"Una manera de hacer Europa". Cofinanciación a cargo del Programa Operativo del FSE 2014-2020 para Extremadura gastos de Ciclos Formativos de Grados Medio y Superior.

Crear Toast personalizado

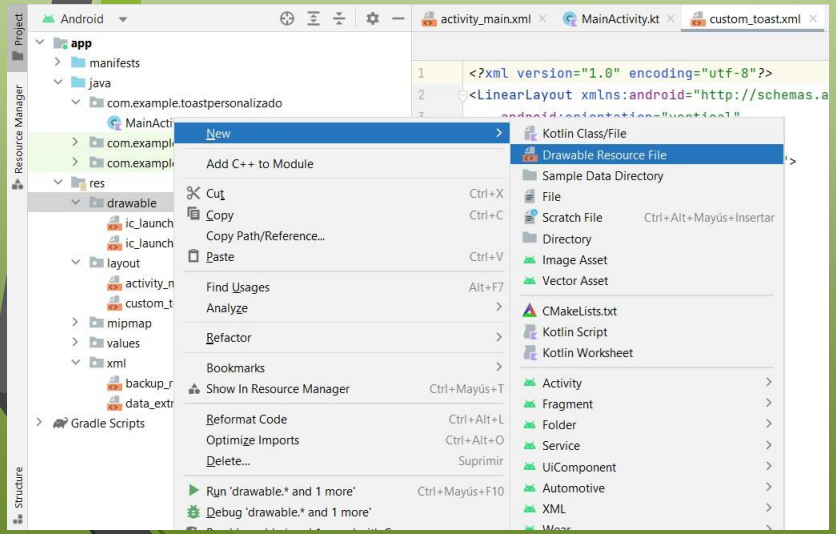
1. Crear un nuevo xml (*Layout Resource File*) dentro del directorio *res/layout*. Este xml lo diseñaremos con un *Linear Layout que* contendrá el estilo y los componentes de nuestro Toast.

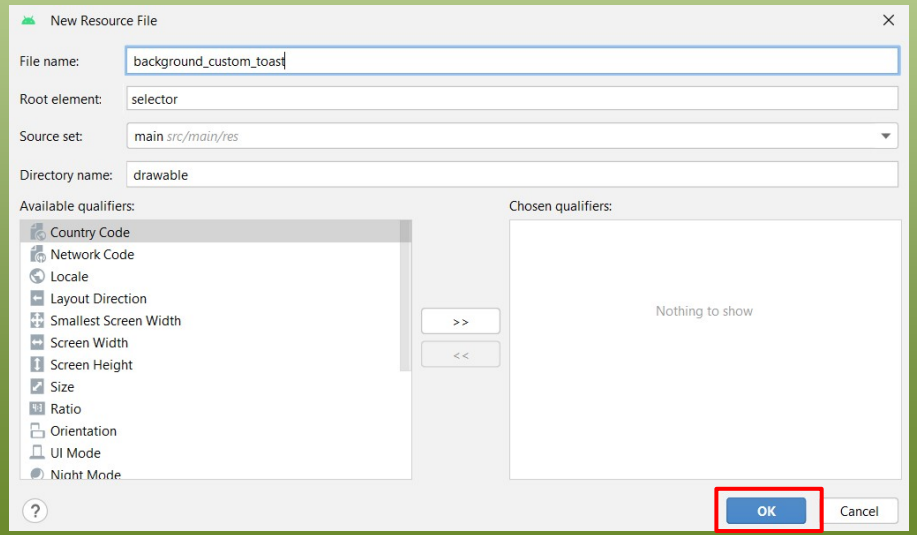




Crear Toast personalizado

2. *OPCIONAL*: Si queremos diseñar un fondo personalizado para nuestro Toast, debemos crear un nuevo recurso gráfico (*Drawable Resource File*) que contenga un elemento *selector*, dentro el directorio *res/drawable.*





Crear Toast personalizado

OPCIONAL: Dentro del elemento *selector* podemos definir un elemento *item* con los atributos que den forma a nuestro fondo personalizado. Por ejemplo, con un elemento *shape* podemos establecer el color de fondo o las esquinas redondeadas.

background_custom_toast.xml | custom_toast.xml | MainActivity.kt

```xml
<?xml version="1.0" encoding="utf-8"?>
<selector xmlns:android="http://schemas.android.com/apk/res/android">
    <item>
        <shape>
            <solid android:color="#0AA0C9"/>
            <corners android:radius="10dp"/>
        </shape>
    </item>

</selector>
```

Crear Toast personalizado

3. Diseñamos el xml para nuestro Toast. Este debe contener un *TextView* para insertar el texto que queremos mostrar. Adicionalmente puede contener otros elementos, como por ejemplo un *ImageView* para insertar un icono.

Nota: En el atributo *background* del LinearLayout, debemos cargar el xml del fondo personalizado que hemos creado previamente.
Si no utilizamos un fondo personalizado para nuestro Toast, otra opción es establecer un color de fondo que elijamos.

custom_toast.xml | strings.xml | MainActivity.kt

```xml
<?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:id="@+id/layoutCustomToast"
    android:orientation="horizontal"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:background="@drawable/background_custom_toast"
    android:padding="15dp">

    <ImageView
        android:id="@+id/imageViewIconCheck"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_weight="1"
        android:src="@drawable/ic_check" />

    <TextView
        android:id="@+id/textViewCustom"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_weight="1"
        android:layout_marginLeft="10dp"
        android:text="@string/custom_toast_message"
        android:textStyle="bold"
        android:textColor="@color/white" />

</LinearLayout>
```

Crear Toast personalizado

4. Por último implementamos el código necesario para crear, configurar y mostrar nuestro Toast, en la clase KOTLIN de la Activity.

```
private fun mostrarToast() {
    //"Inflamos" un objeto View para añadir en él el layout que hemos diseñado para el Toast
    val view: View = layoutInflater.inflate(R.layout.custom_toast, findViewById(R.id.layoutCustomToast))

    //Si queremos modificar el mensaje del toast antes de mostrarlo, debemos modificar el textView
    // que contiene el layout del Toast
    /* val textViewCustomToast: TextView = view.findViewById<TextView>(R.id.textViewCustomToast)
    textViewCustomToast.setText("Nuevo mensaje del toast") */

    //Creamos, configuramos y mostramos nuestro Toast
    val customToast: Toast = Toast(applicationContext)
    customToast.setGravity(Gravity.TOP,  xOffset: 0,  yOffset: 200)
    customToast.duration = Toast.LENGTH_SHORT
    customToast.view=view
    customToast.show()
}
```