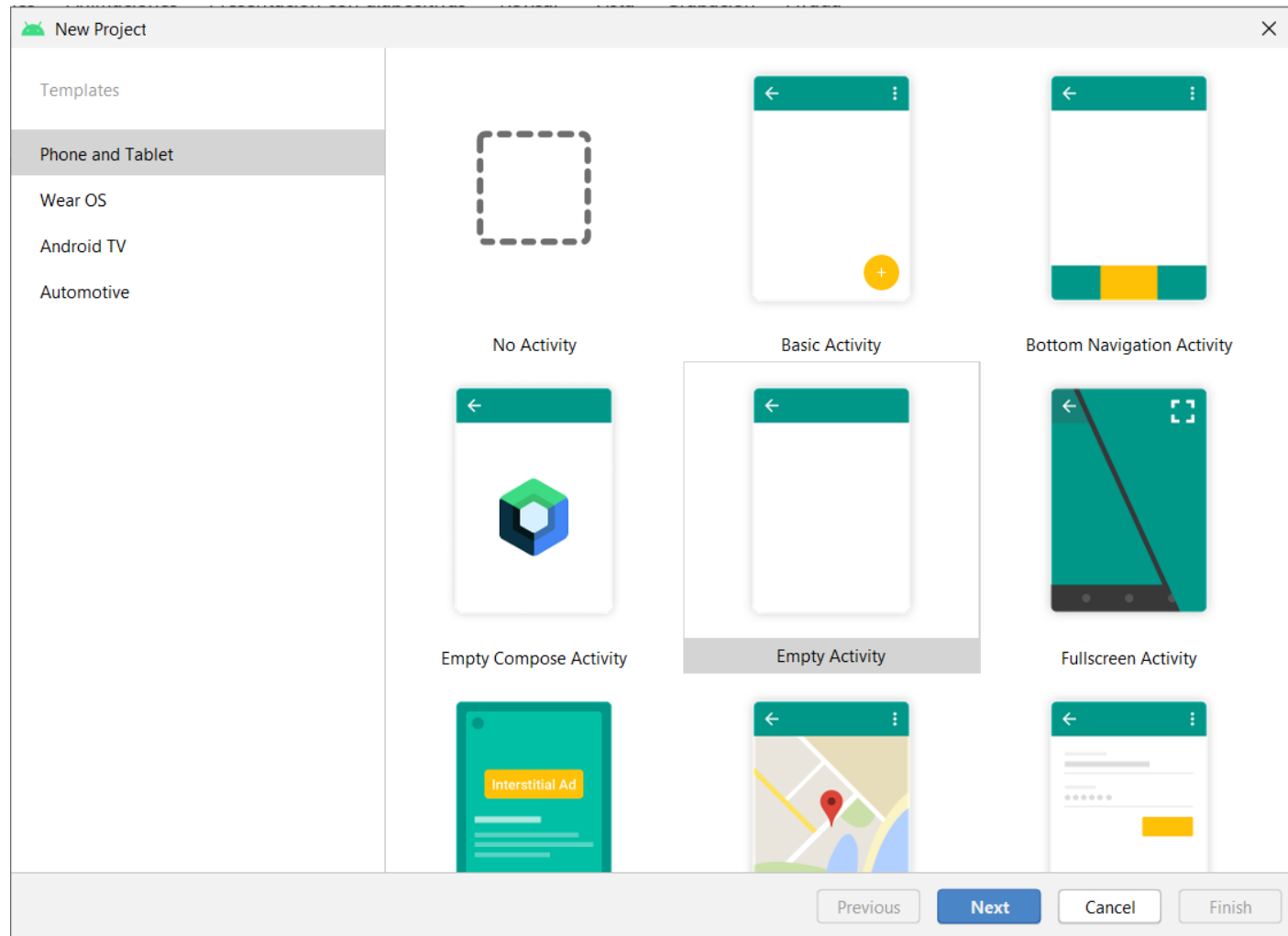


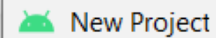
Mi primer ejercicio en ▶ Android

Laura Sacristán Matesanz

Vamos a crear un nuevo proyecto. Seleccionaremos una ventana vacía



Ponemos el nombre del nuevo proyecto. Elegiremos lenguaje java y la versión SDK que aparece por defecto



Empty Activity

Creates a new empty activity


Name


Package name

Save location


Language

Minimum SDK

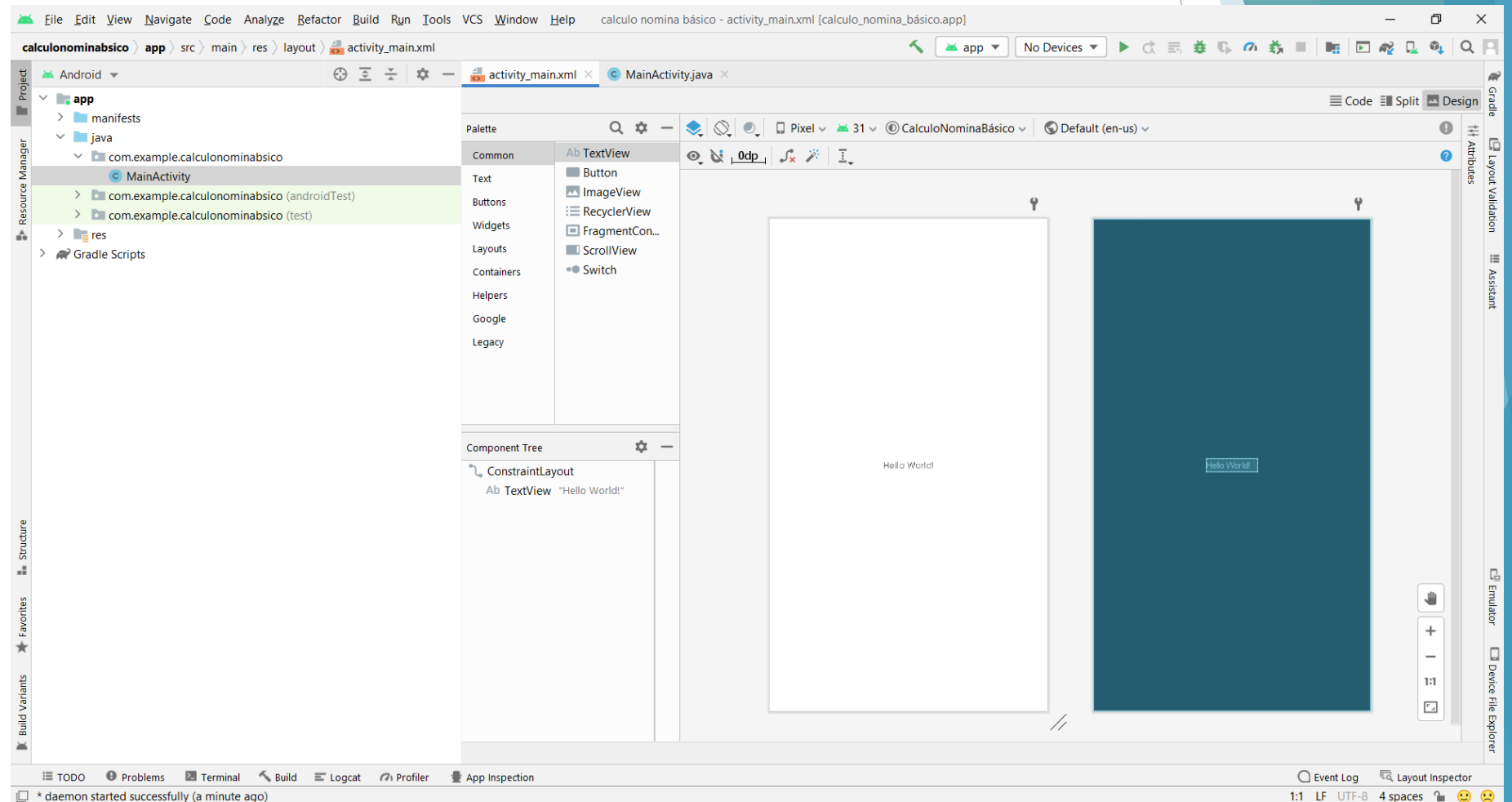
 Your app will run on approximately **94,1%** of devices.
[Help me choose](#)

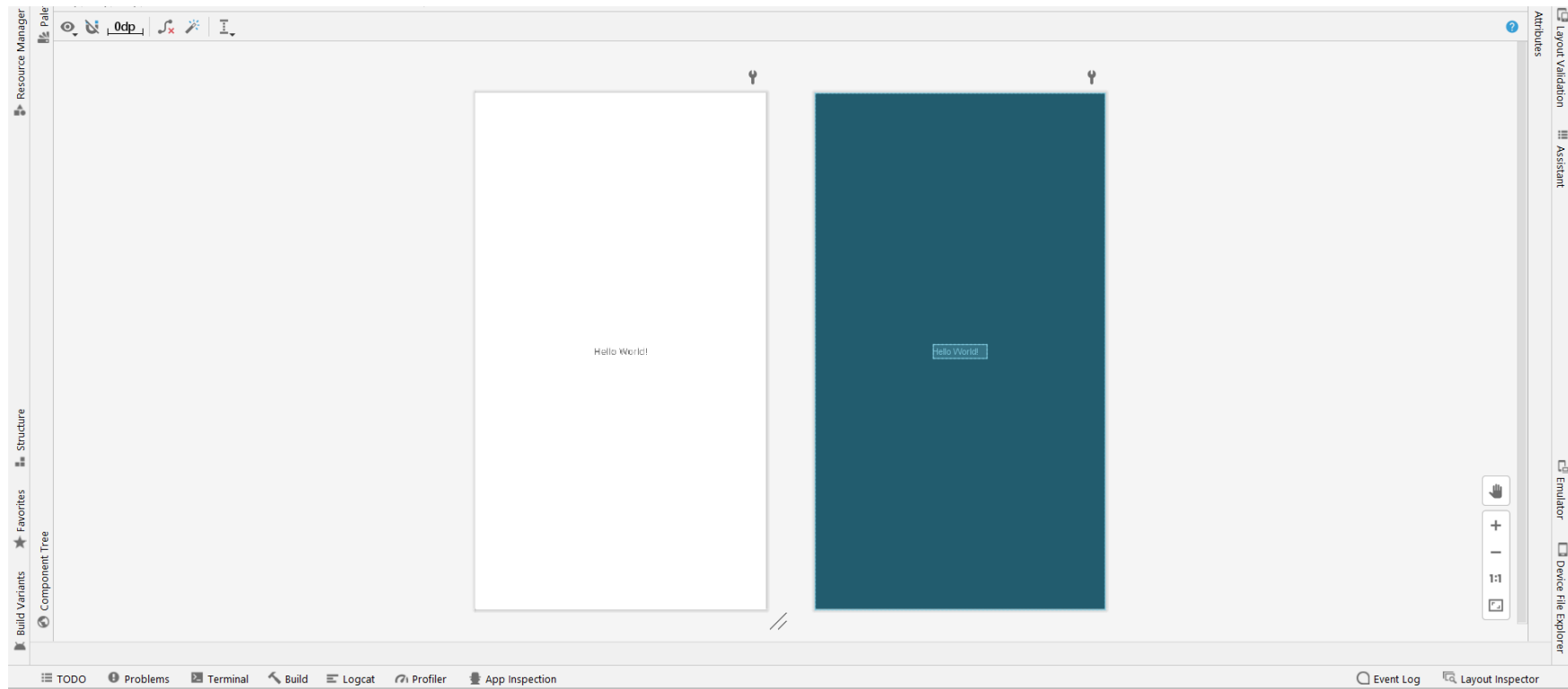
☐ Use legacy android.support libraries 

Using legacy android.support libraries will prevent you from using the latest Play Services and Jetpack libraries

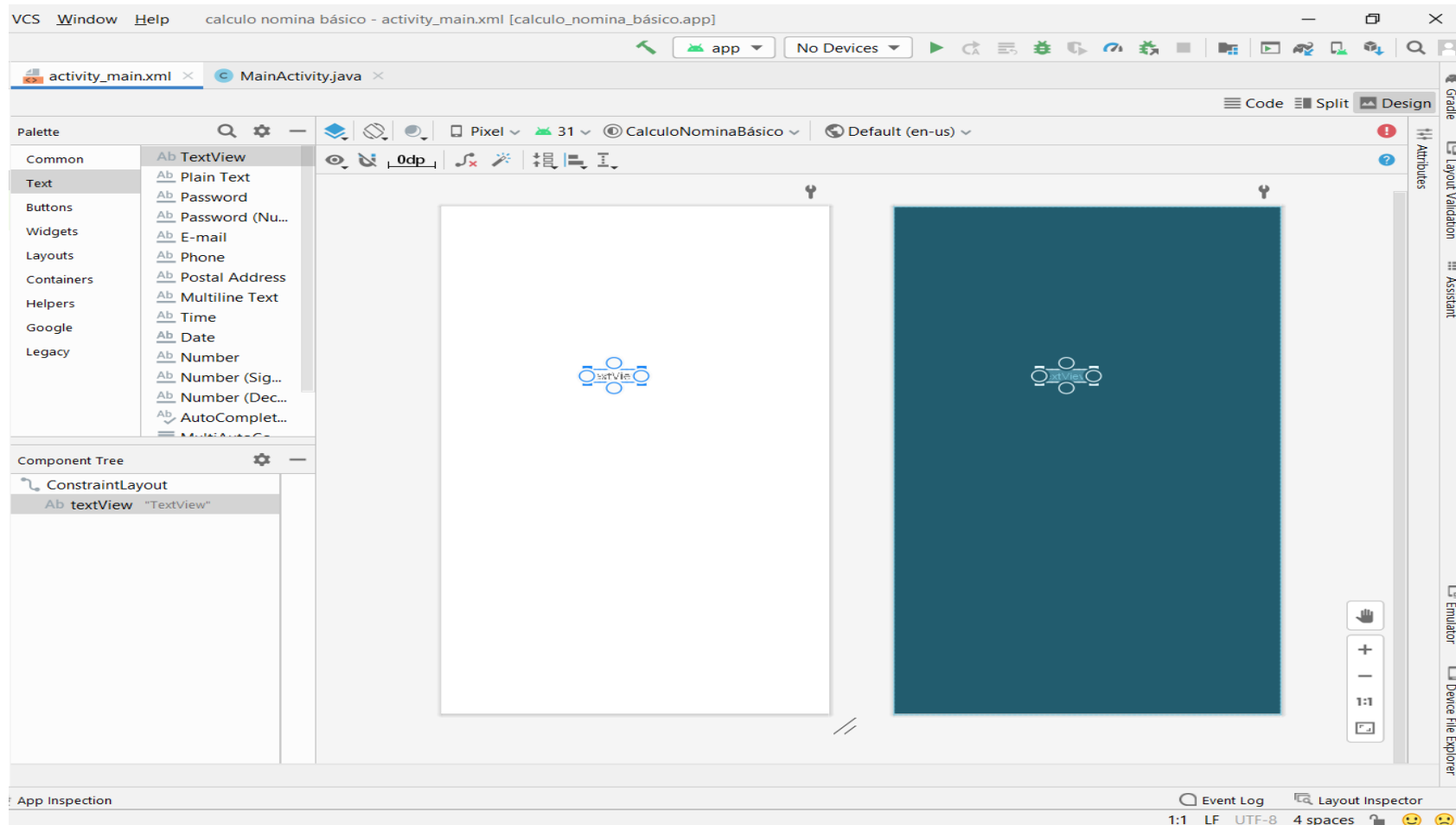
 The application name for most apps begins with an uppercase letter

Una vez nos ha generado el proyecto (puede tardar unos minutos) vamos a configurar el activity_main.xml (parte gráfica de nuestro proyecto)

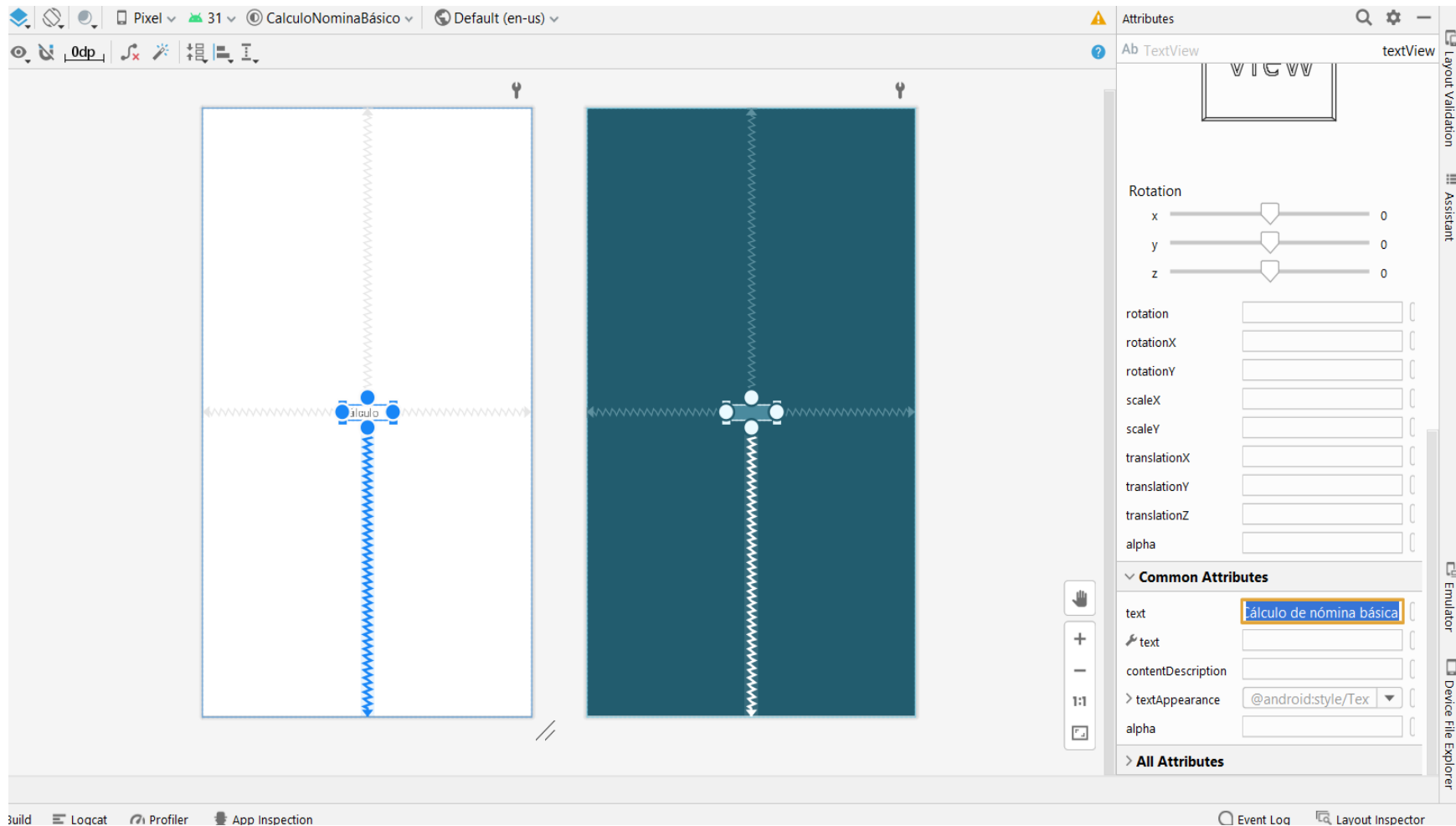




Ya sabemos que podemos utilizar el entorno gráfico o el entorno de texto y programar directamente la aplicación. Nosotros utilizaremos el entorno gráfico

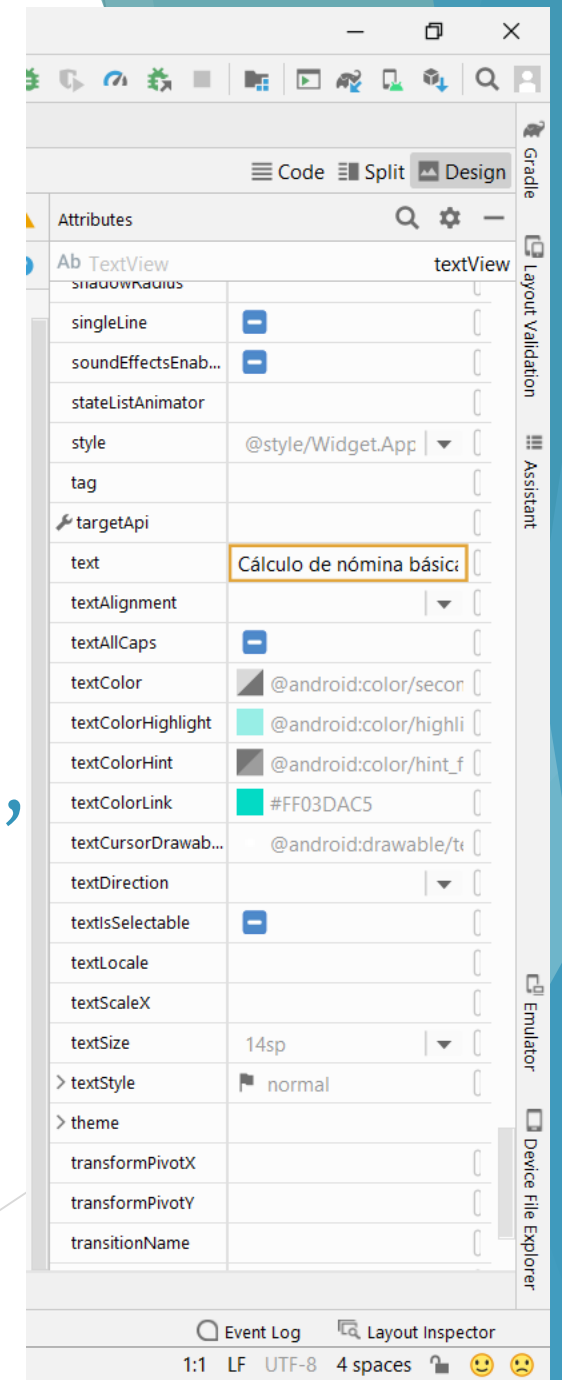


Lo primero que vamos a hacer es eliminar el textview que aparece por defecto y comenzar a diseñar nuestros propios controles. El primer elemento que crearemos es un textview.

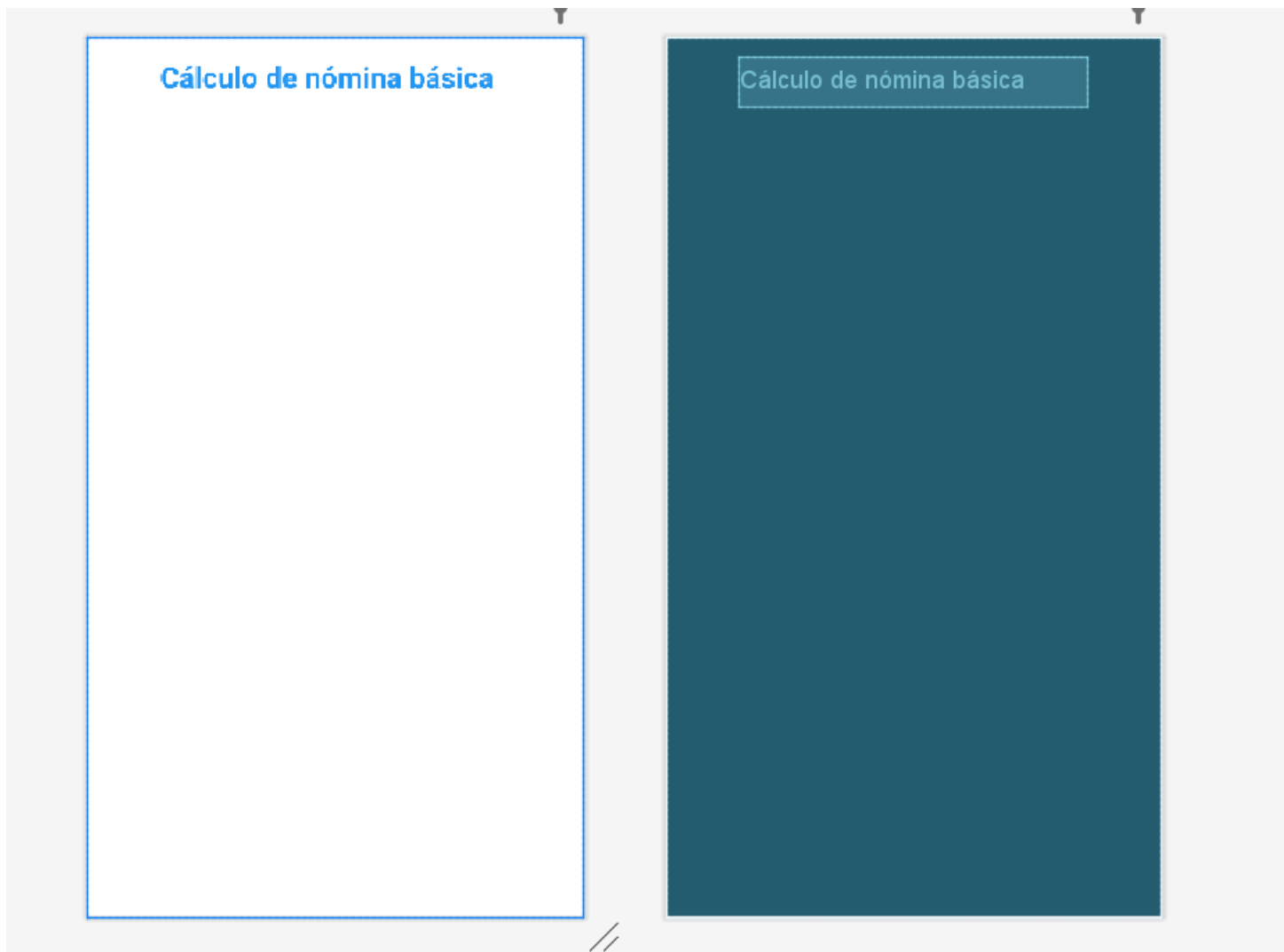


A ese textview le cambiaremos el nombre y le pondremos Cálculo de nómina básica. Es muy importante marcar los márgenes (los 4) porque si no la caja de texto se pondrá en un lugar que no es el que nosotros le vamos a indicar.

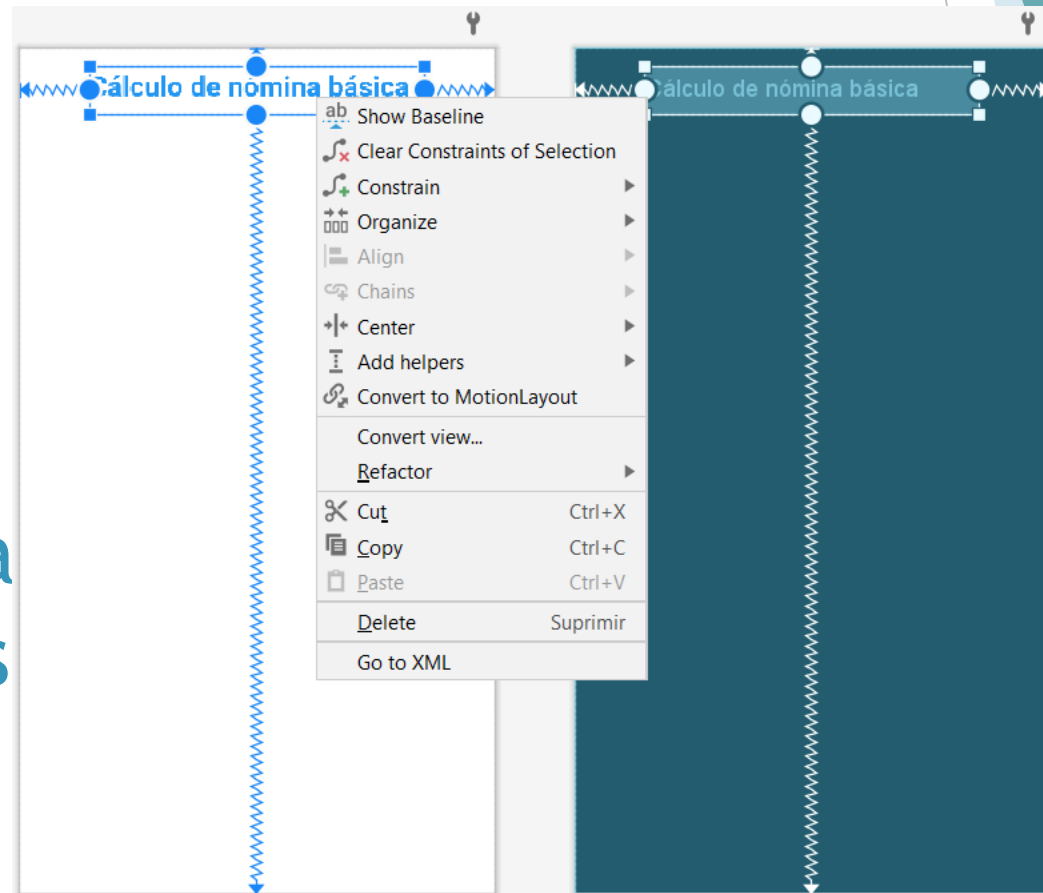
Vamos a ver todos los atributos del control para poder modificar lo que consideremos necesario. En este caso modificaremos el tamaño (24p), lo pondremos en negrita y lo cambiaremos de color. Los atributos están ordenados alfabéticamente. Como los tres atributos que vamos a modificar son de texto, buscaremos por la t de text. Cambiaremos textSize, textStile y textColor. Si queremos que nuestro texto esté centrado tendremos que modificar el atributo gravity.

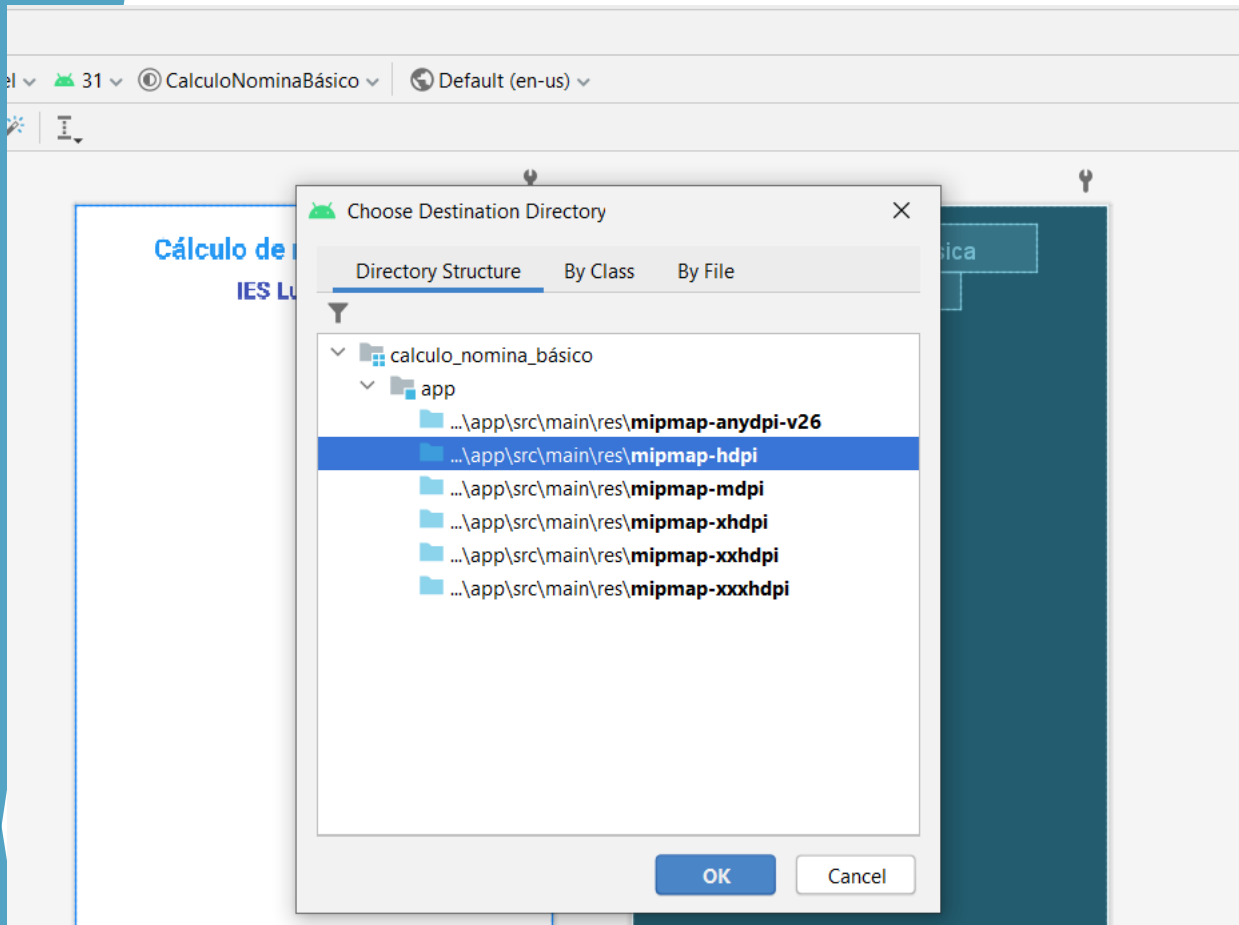


Nos quedará algo así



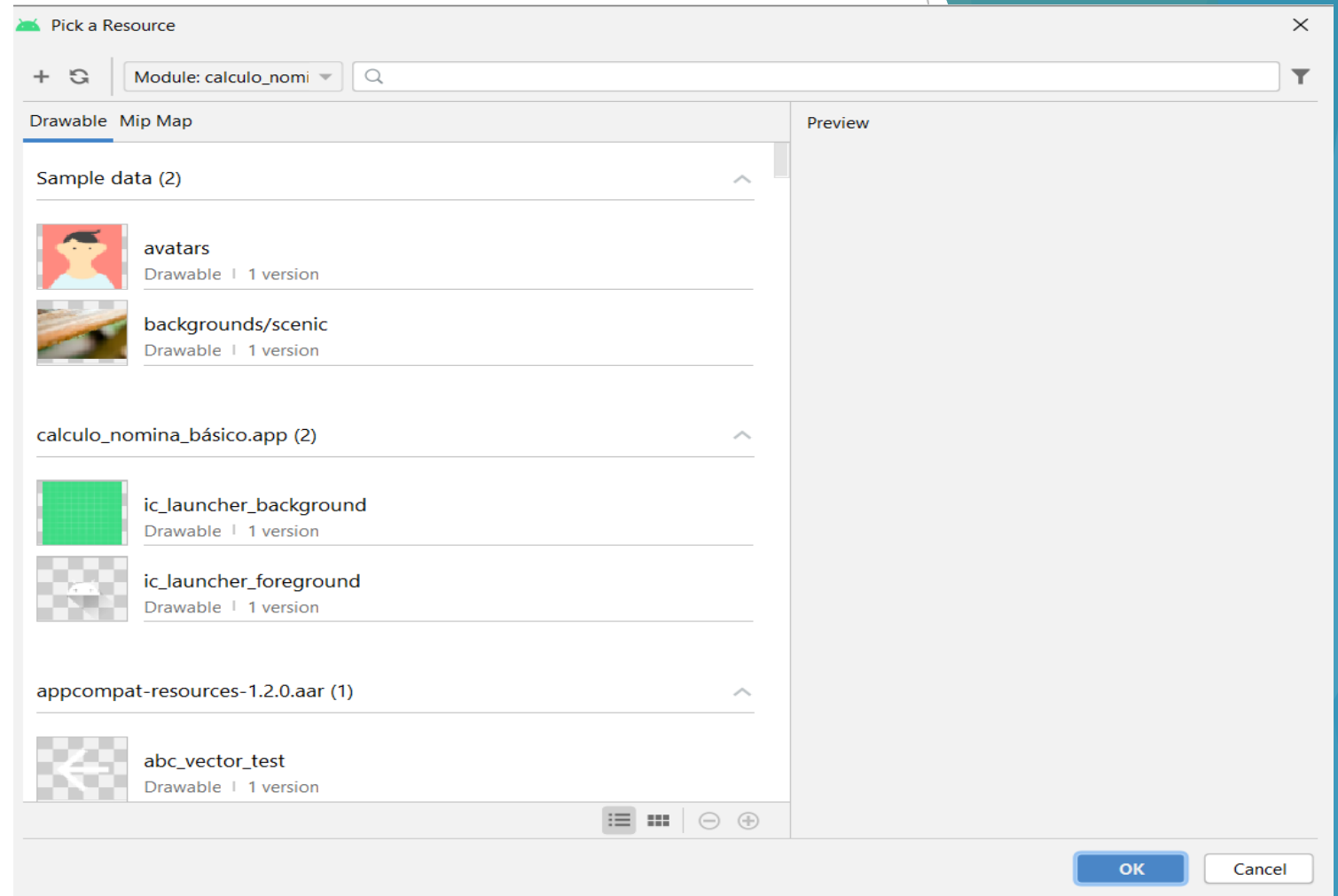
Lo siguiente que vamos a hacer es duplicar el textview que habíamos creado y le vamos a modificar el nombre. Pondremos IES Luis Braille. Pulsamos el botón derecho del ratón copiamos y pegamos. La copia se queda encima así que la moveremos hacia abajo para poderla modificar. Lo cambiamos de tamaño y de color.

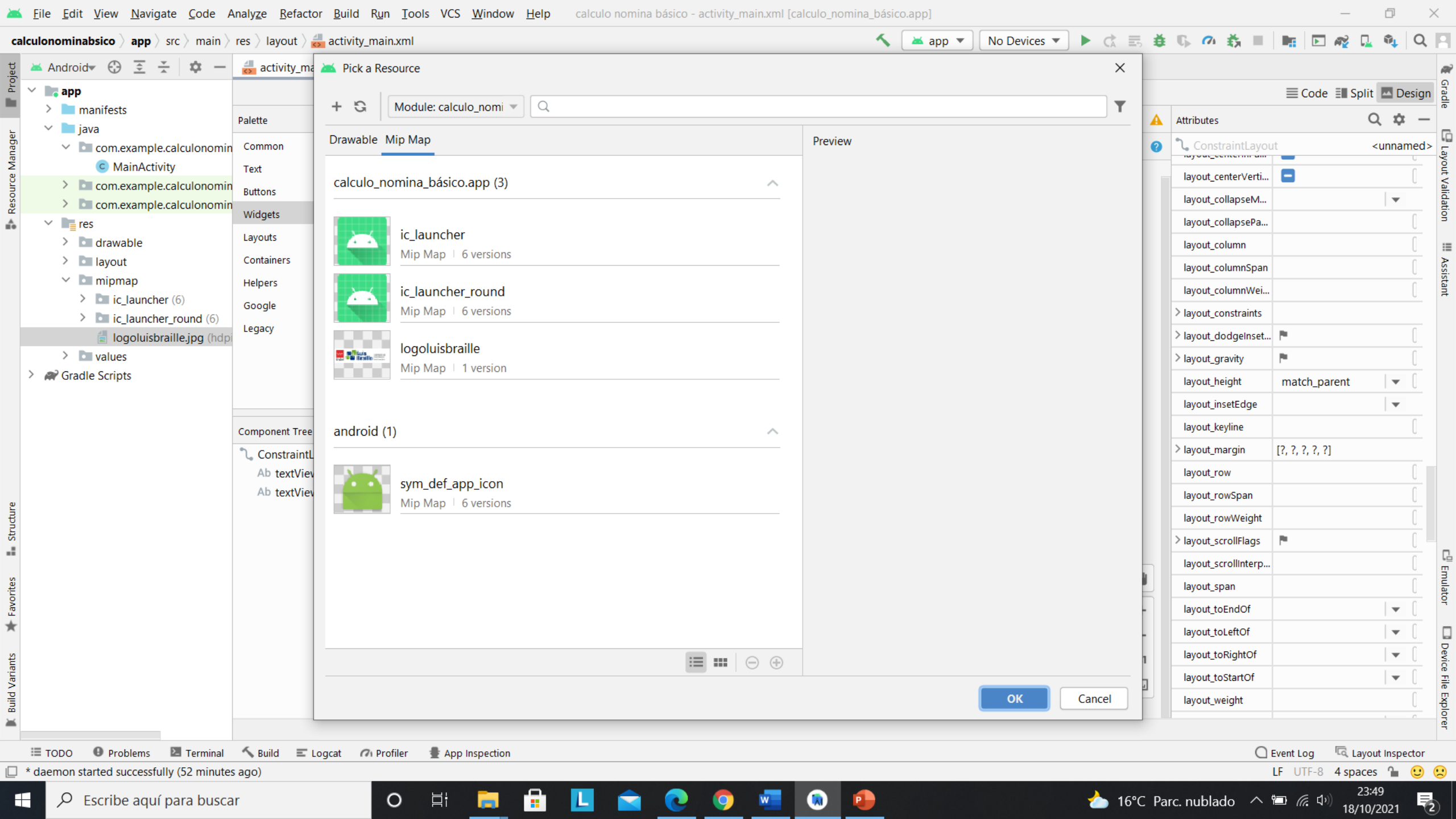




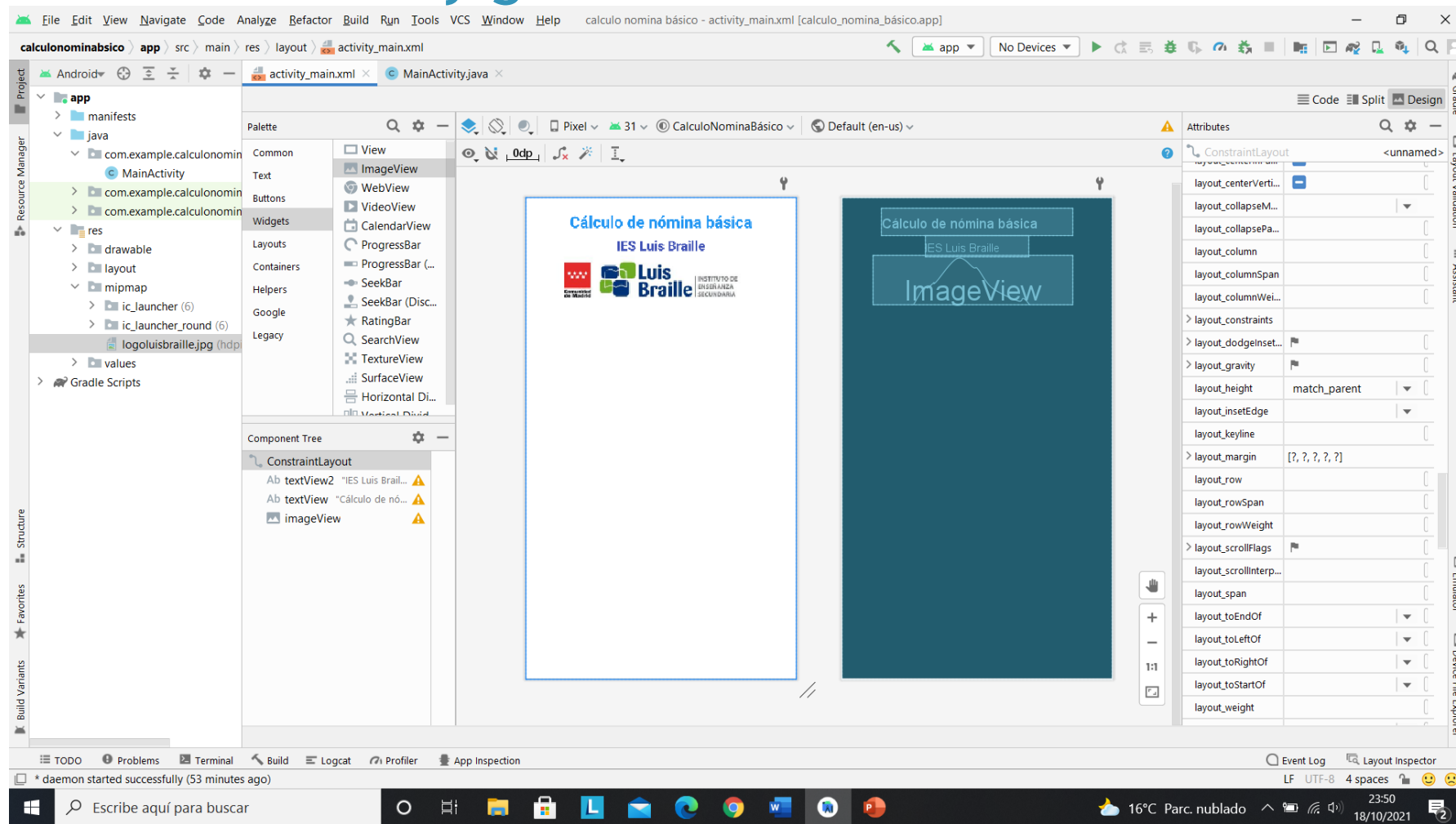
Ahora vamos a añadir una imagen. Yo he puesto el logotipo de instituto. Vosotros buscad la imagen que queráis y guardarla como jpg o png. Vamos a copiar la foto y la vamos a pegar dentro de nuestro Android Studio en la carpeta mipmap que está en la carpeta res. Como mi imagen no es me pide que calidad quiero que tenga. Le ponemos la que ponga por defecto.

Una vez que la hemos copiado en mipmap la podemos utilizar como imagen para incluirla en nuestro diseño. Las imágenes están en la opción widgets dentro de ImageView. Arrastramos a nuestro diseño y nos muestra lo siguiente. Temos que elegir la opción mipmap (es donde tenemos nuestra imagen)





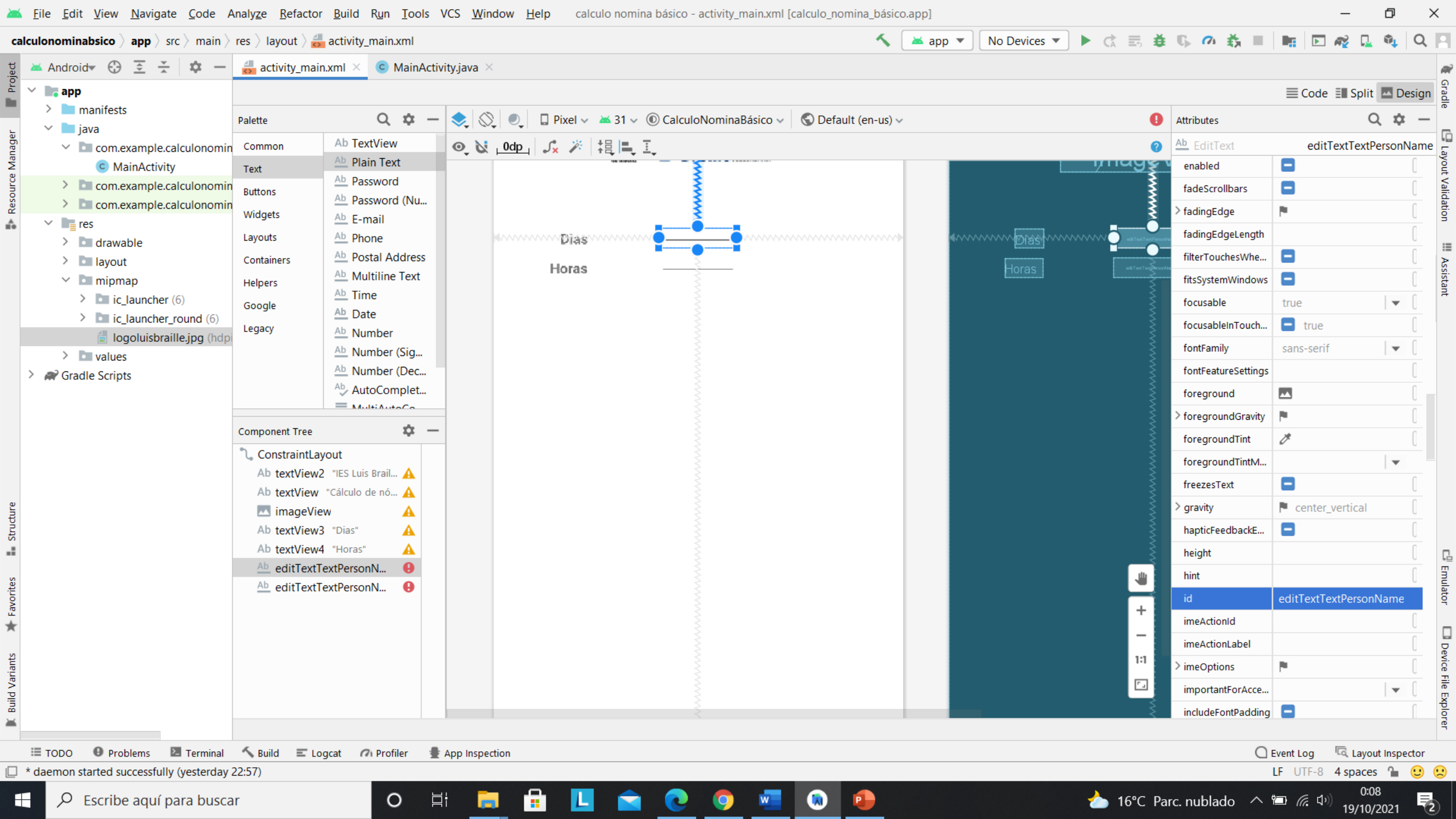
Elegimos nuestra imagen, marcamos los márgenes y nos quedará algo así. Podemos cambiar el icono de tamaño si es muy grande



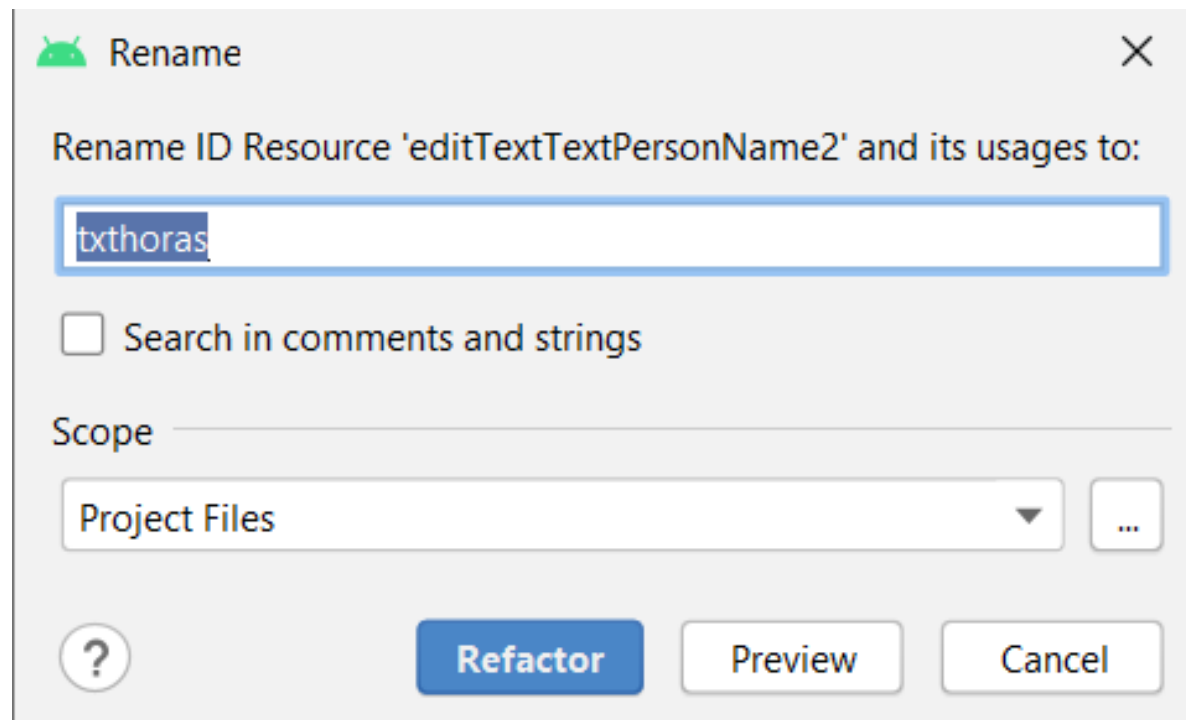
Ahora vamos a incluir controles en los que vamos a indicar los días y las horas que se trabajan cada día. Vamos a poner un par de textview, uno para días y otro para horas. Estos textview son informativos para saber que datos tenemos que introducir. Les pondremos el tamaño, formato y color que consideremos oportuno.



Además de los textviews tendremos que tener donde vamos a introducir esos datos. Para ello insertaremos en nuestro diseño un control plaintext por cada dato que necesitemos introducir. En nuestro caso 2, uno de días y otro de horas. Acordaros de marcar los márgenes. Lo que acabamos de hacer es crear dos variables, por lo tanto debemos asignarle un nombre que sea significativo para nosotros, ya que tendremos que usarlas después. Les quitamos el nombre y les ponemos el id.



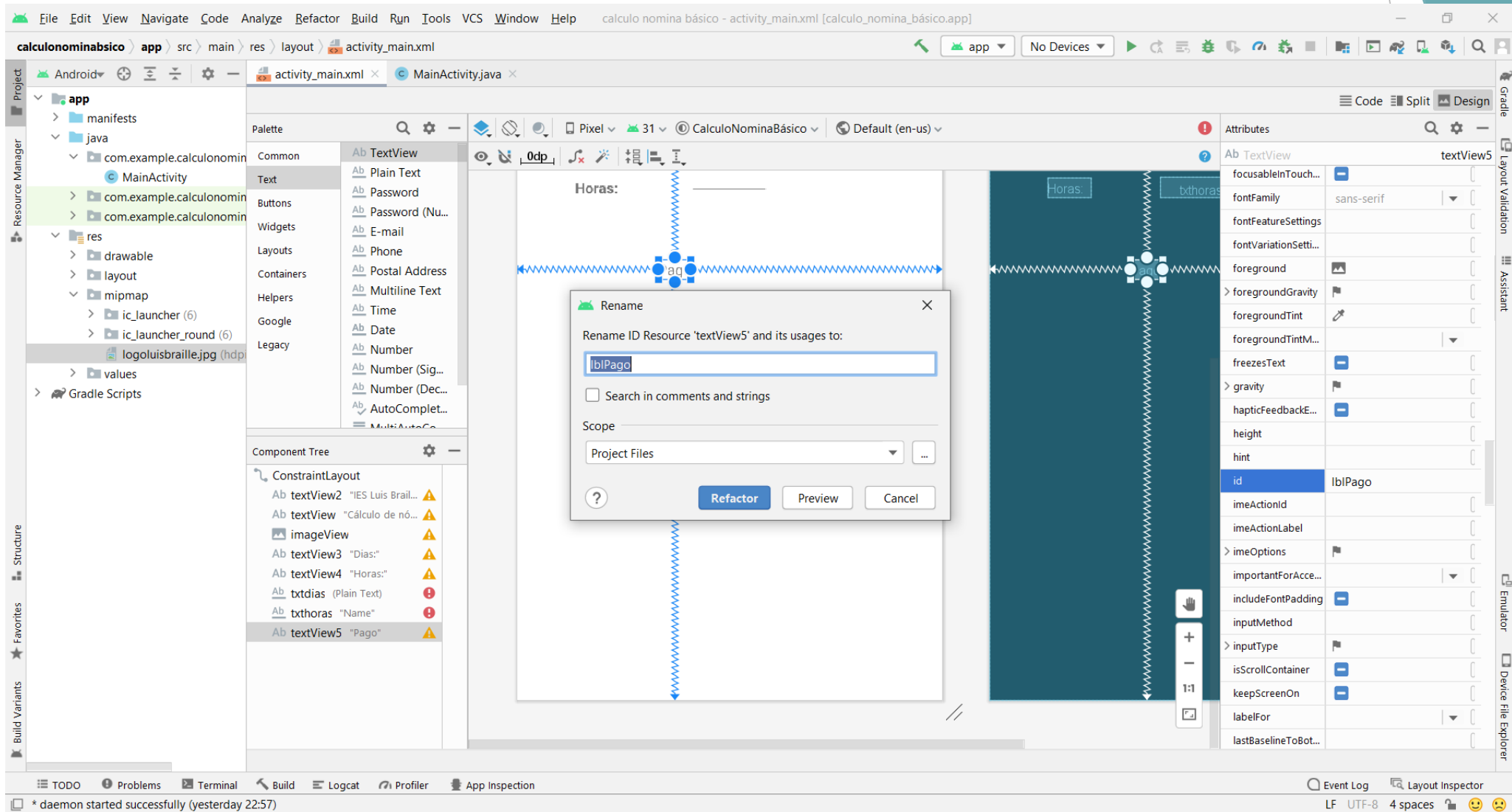
Vamos a poner como nombre id a la variable de día txtdias y a la de horas txthoras. Nos va a pedir que confirmemos que queremos modificar el nombre de la variable. Le diremos que si.



Nuestra aplicación lo que va a hacer es calcular el pago en base a las horas y días trabajados. Por cada hora vamos a pagar 13,75 €. Para poder ver lo que nos van a pagar necesitaremos añadir otro control textview que después utilizaremos para poder mostrar lo que vamos a cobrar. Le cambiaremos el nombre y le pondremos como nombre pago. También le pondremos nombre a la variable en este caso lblPago Acordaros de los márgenes.

Cuando calculemos lo que nos van a pagar, el texto se sustituirá por un número.

Nos queda algo así:

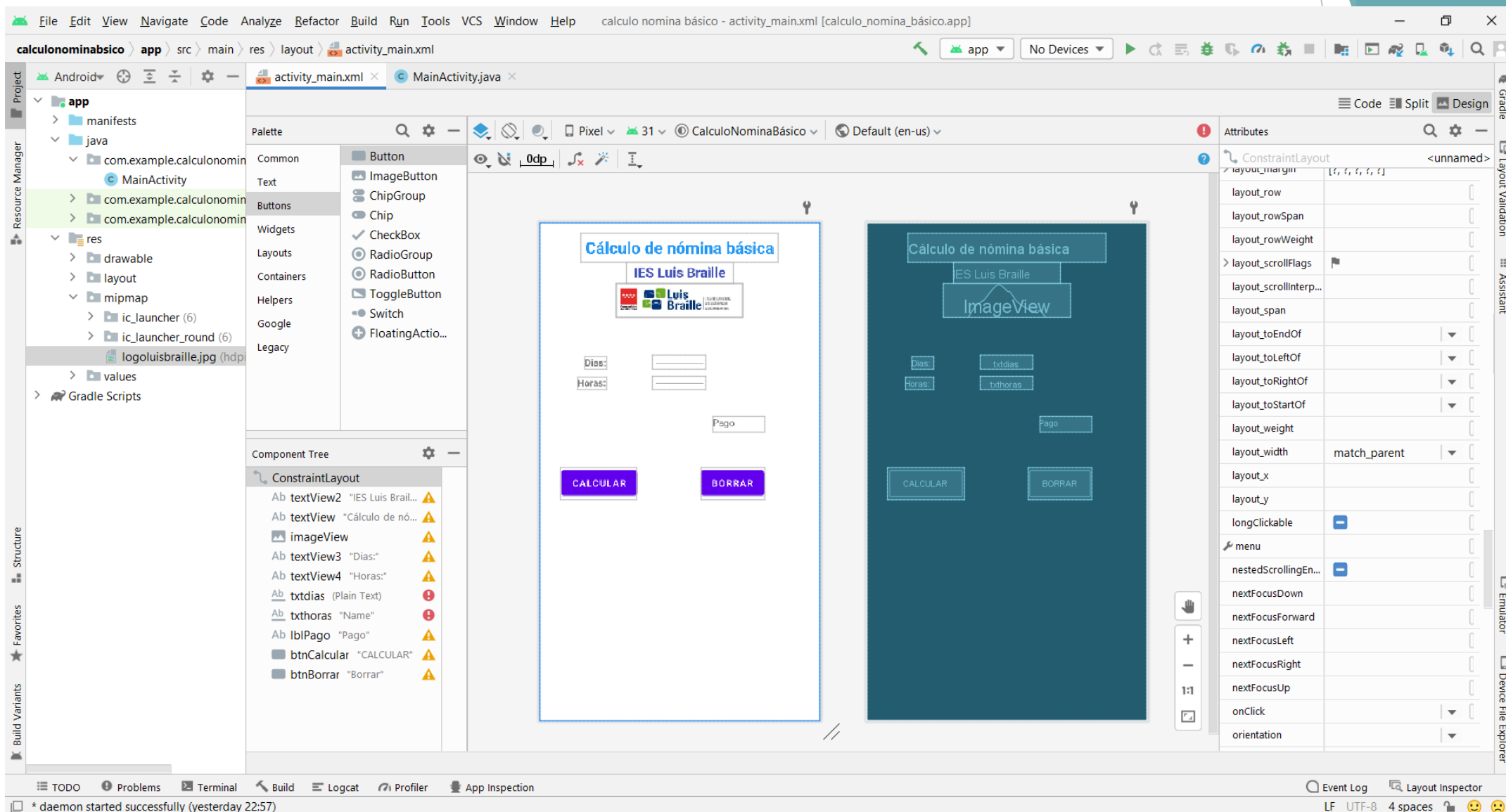


Nos falta incluir los botones que ejecutarán el cálculo y borrarán los datos cuando queramos hacer otro cálculo nuevo.

Los botones que vamos a incluir son calcular y borrar. Los botones están en la opción buttons. Cambiamos los nombres en text (como siempre) Cambiamos también los nombres de las variables porque después lo vamos a utilizar en la programación.

Para el botón calcular la variable será btnCalcular y para el botón de borrar btnBorrar.

El resultado es:



Lo siguiente que tenemos que hacer es desarrollar nuestro programa en el `mainActivity`.

Antes vamos a poner nombre al método de cálculo. Para ello iremos al atributo `onClick` y le daremos el nombre `calcular`.

Lo mismo haremos con el método de borrado y lo daremos el nombre `borrar`, en el atributo `onClick` del botón `borrar`.

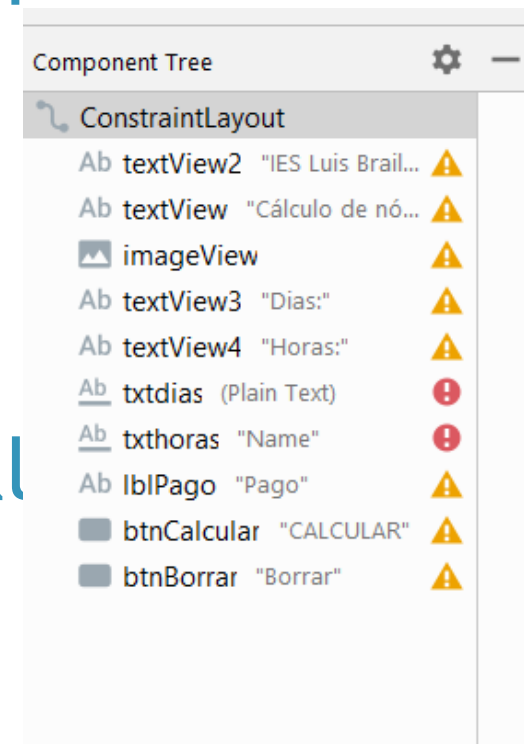
Estos métodos se desarrollarán en el `mainActivity`.

Antes de ir al `mainActivity` os invito a que veáis como se ha modificado el archivo `activity_main` en formato texto. Cada uno de los controles que hemos incluido en el entorno gráfico se ha implementado en el formato texto.
Es curioso ¿verdad?

Comenzamos a programar. Lo primero que haremos es definir cada una de las variables que vamos a necesitar y que hemos creado en el activity_main.

Las vamos a generar privadas y aquí tenemos las variables que hemos creado:

Las variables deben llamarse igual en los dos archivos.



Las variables declaradas quedarían así:

```
File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help calculo nomina básico - MainActivity.java [c]

calculonominabsico > app > src > main > java > com > example > calculonominabsico > MainActivity > lblPago

Project
Android
app
  manifests
  java
    com.example.calculonominabsico
      MainActivity
    com.example.calculonominabsico
    com.example.calculonominabsico
  res
    drawable
    layout
    mipmap
      ic_launcher (6)
      ic_launcher_round (6)
    logoluisbraille.jpg (hdp)
    values
  Gradle Scripts

activity_main.xml x MainActivity.java x

1 package com.example.calculonominabsico;
2
3 import ...
4
5
6
7
8
9
10 public class MainActivity extends AppCompatActivity {
11     private EditText txtdias, txthoras;
12     private Button btnCalcular, btnBorrar;
13     private TextView lblPago;
14
15     @Override
16     protected void onCreate(Bundle savedInstanceState) {
17         super.onCreate(savedInstanceState);
18         setContentView(R.layout.activity_main);
19     }
20 }
```

Ahora vamos a asociar las variables. Para ello haremos lo siguiente:

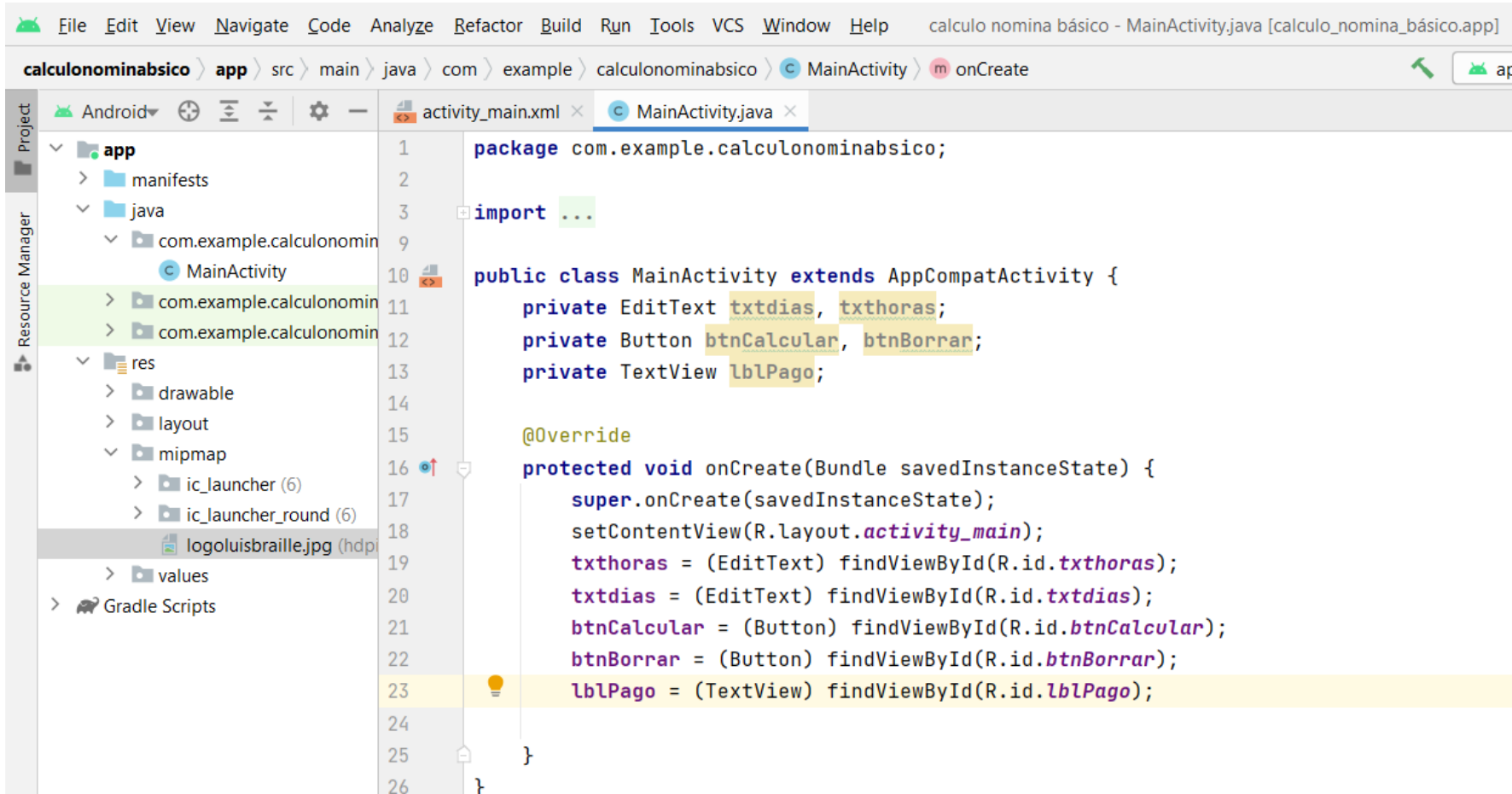
Ponemos el nombre de la variable y lo igualamos a una variable de tipo editText (que es del mismo tipo) después llamamos a `findViewById(R.id.nombrevariable)`. Esta función enlaza un recurso de la interfaz gráfica con una variable en nuestro código.

Un ejemplo:

```
txthoras = (EditText) findViewById(R.id.txthoras);
```

Hacemos lo mismo para todas las variables declaradas.

Y nos queda así:



```
File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help calculo nomina básico - MainActivity.java [calculo_nomina_básico.app]
calculonominabsico > app > src > main > java > com > example > calculonominabsico > MainActivity > onCreate
Project
  app
    manifests
    java
      com.example.calculonominabsico
        MainActivity
    res
      drawable
      layout
      mipmap
        ic_launcher (6)
        ic_launcher_round (6)
      logoluisbraille.jpg (hdp)
      values
    Gradle Scripts
Resource Manager
activity_main.xml x MainActivity.java x
1 package com.example.calculonominabsico;
2
3 import ...
4
5
6
7
8
9
10 public class MainActivity extends AppCompatActivity {
11     private EditText txtdias, txthoras;
12     private Button btnCalcular, btnBorrar;
13     private TextView lblPago;
14
15     @Override
16     protected void onCreate(Bundle savedInstanceState) {
17         super.onCreate(savedInstanceState);
18         setContentView(R.layout.activity_main);
19         txthoras = (EditText) findViewById(R.id.txthoras);
20         txtdias = (EditText) findViewById(R.id.txtdias);
21         btnCalcular = (Button) findViewById(R.id.btnCalcular);
22         btnBorrar = (Button) findViewById(R.id.btnBorrar);
23         lblPago = (TextView) findViewById(R.id.lblPago);
24     }
25 }
26 }
```

Ahora vamos a desarrollar el método calcular. El método lo vamos a crear fuera del onCreate, que es donde hemos definido y enlazado todas las variables.

Lo primero es definir el método y después definir las variables que vamos a utilizar para el cálculo. Vamos a crear una variable integer que va a coger el datos de días. Días es un valor introducido en texto. Para poder calcular tendremos que hacer un casting (transformación) de texto a integer para poder operar. Lo mismo haremos con horas.

El método es el siguiente

```
public void Calcular (View view) {  
    int dias = Integer.parseInt(txtdias.getText().toString());  
    int horas = Integer.parseInt(txthoras.getText().toString());  
    int horas_mensuales = horas*dias;  
    Double pago= horas_mensuales*13.75;  
    lblpago.setText(String.valueOf(pago));  
}
```

Nos queda realizar el método borrar, para dejar limpios todos los campos antes de volver a realizar un nuevo cálculo. El método es el siguiente:

```
public void Borrar (View view){  
    txtdias.setText("");  
    txthoras.setText("");  
    lblPago.setText("Pago");  
}
```

Ahora solo queda ejecutar y
comprobar que funciona
correctamente.
¿Qué tal te ha ido?

Ampliación de la práctica:

Sueldo base: Si el pago al trabajador supera el sueldo base se le reduce el sueldo un 10%. El sueldo base es de 2000 (checkbox uno con pago y otro con descuento)

RadioGroup para aplicar redondeo o no.

Dentro introducimos los radiobuttons (uno con redondeo y otro sin redondeo).

Visualizar el pago tanto con descuento como sin descuento y con redondeo o no.