UNIVERSIDAD DON BOSCO
FACULTAD DE INGENIERÍA
ESCUELA DE INGENIERÍA EN COMPUTACIÓN
DESARROLLO DE SOFTWARE PARA MÓVILES DSM104 G01T

INTRODUCCIÓN ANDROID

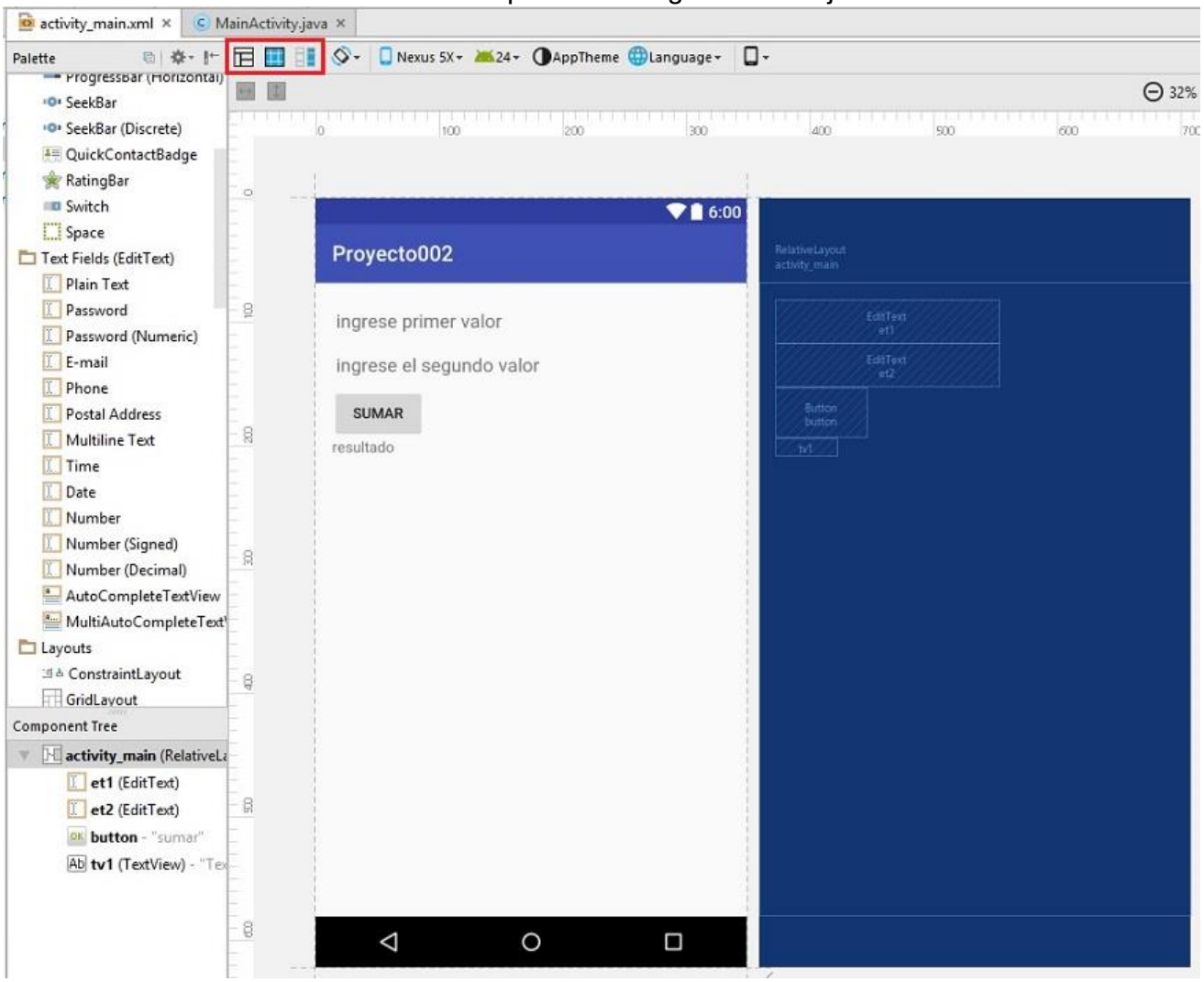
Tabla de contenido

1. Capturar el clic de un botón

Problema:

Confeccionar un programa que permita la carga de dos números enteros en controles de tipo EditText (Number). Mostrar dentro de los mismos controles EditText mensajes que soliciten la carga de los valores. Disponer un Button para sumar los dos valores ingresados. Mostrar el resultado en un control de tipo TextView.

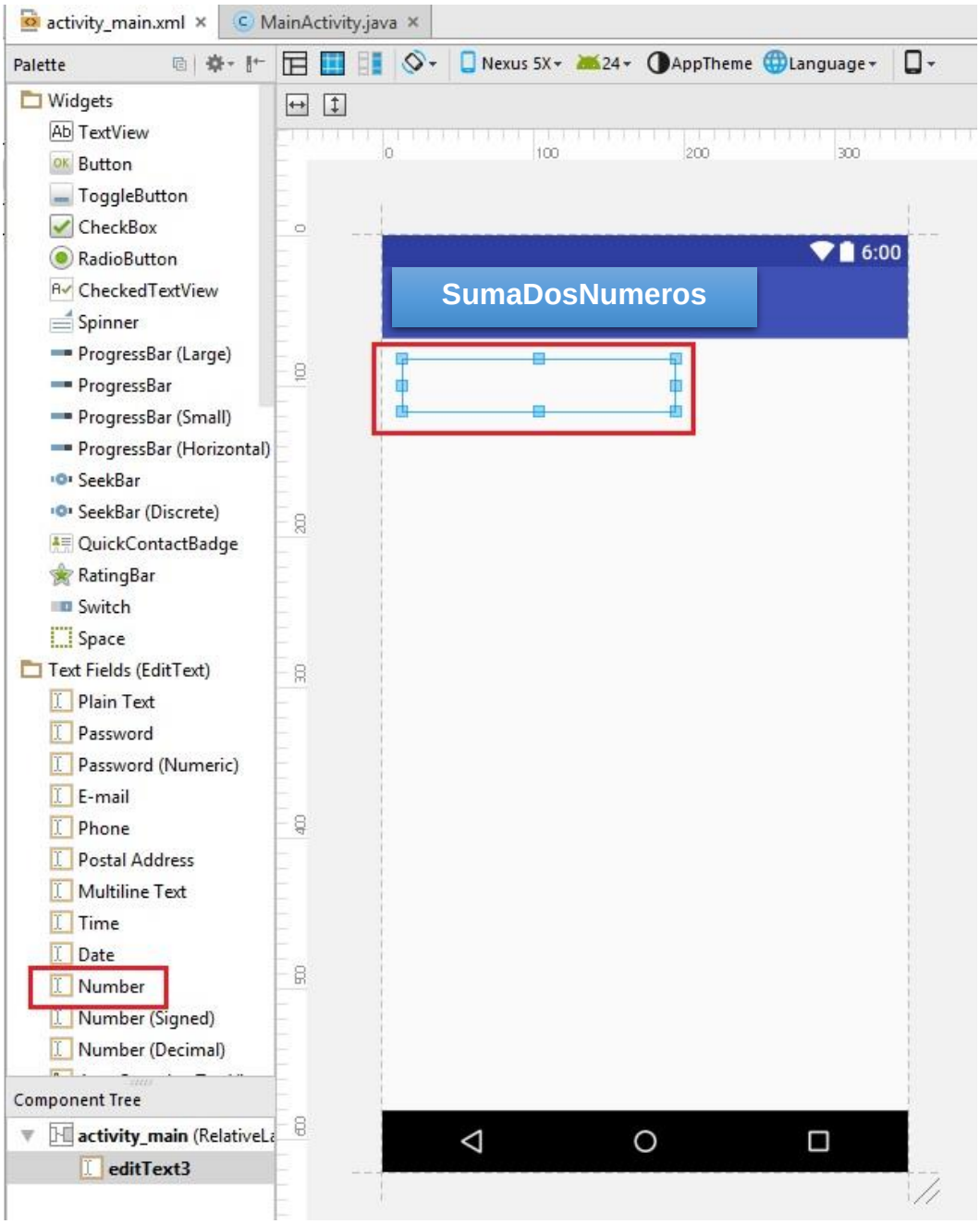
La interfaz visual debe quedar algo semejante a esto:



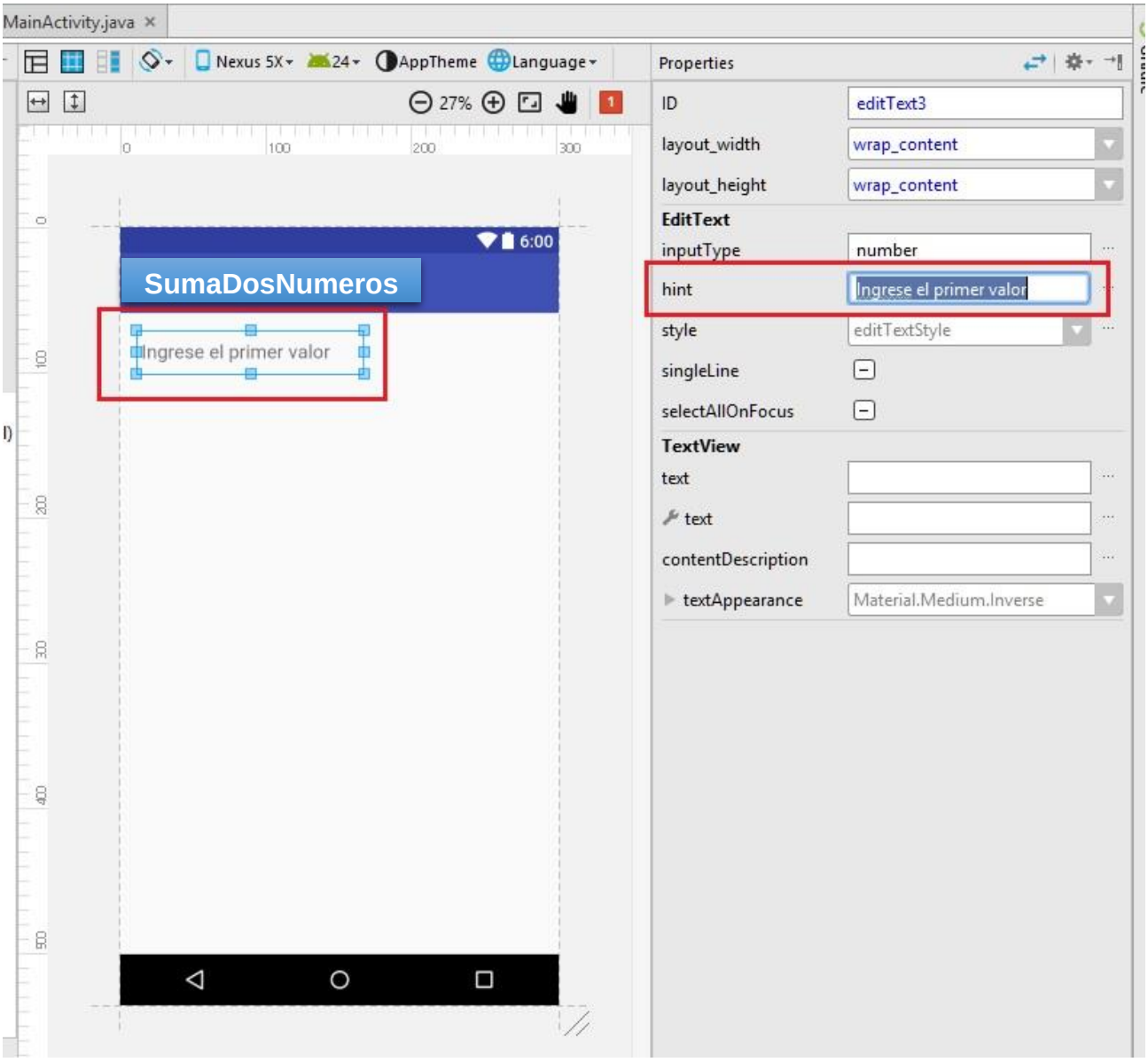
Recordar que si queremos ocultar o volver a mostrar el diseño "blueprint" tenemos tres íconos en la parte superior del diseñador.

Crear un proyecto llamado: **SumaDosNumeros**.

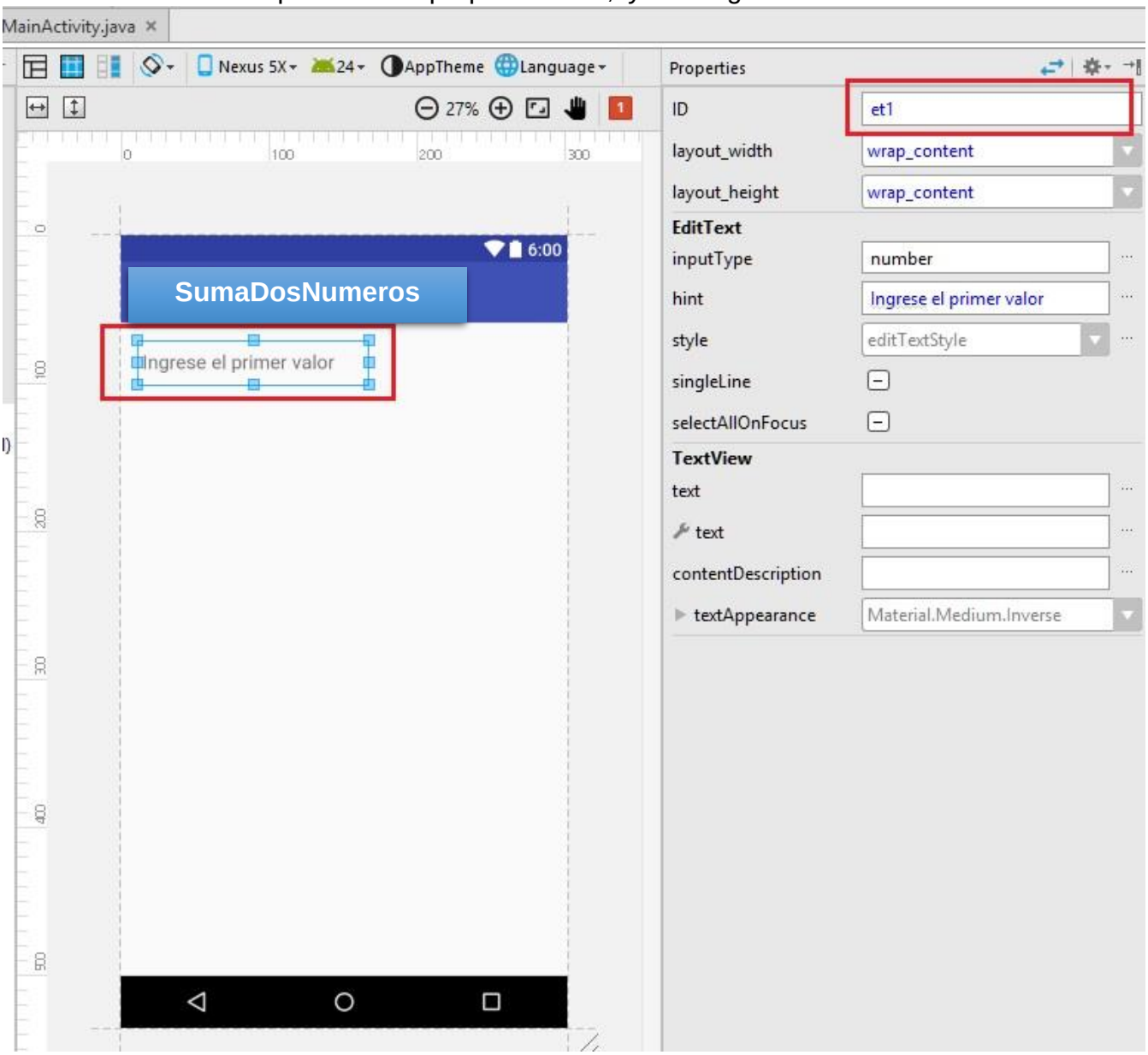
Veamos paso a paso como creamos la interfaz visual de nuestro programa. Primero borramos el TextView que aparece por defecto cuando se crea un proyecto con el Android Studio. Ahora desde la ventana "Palette" seleccionamos de la pestaña "Text Fields (EditText)" el control "Number" (es de la clase EditText) y lo arrastramos a la ventana de diseño de nuestra interfaz a la parte superior izquierda:



Ahora lo seleccionamos y en la ventana de propiedades (Properties) especificamos la propiedad hint, disponemos el texto "Ingrese el primer valor":

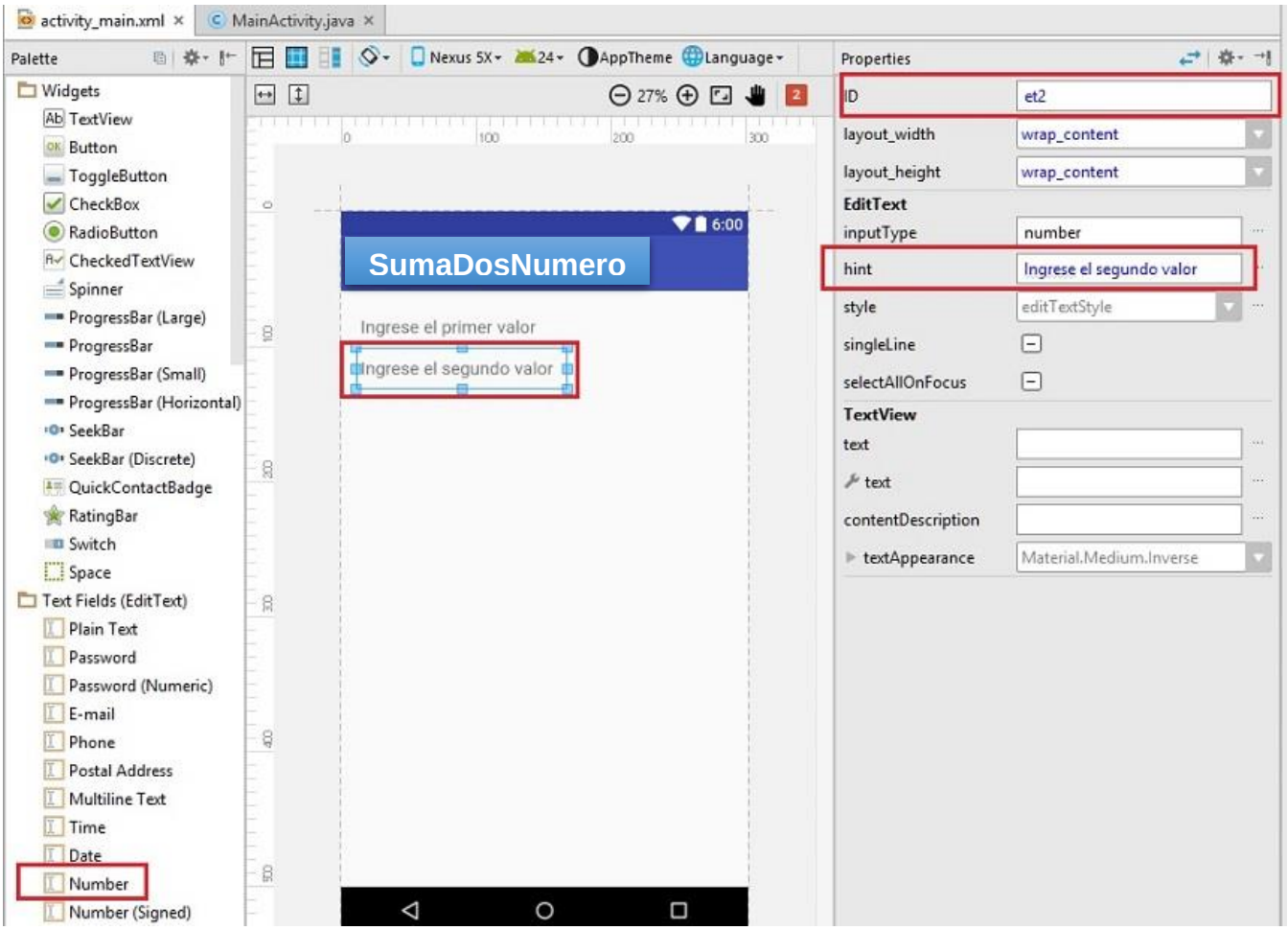


También vamos a especificar la propiedad "id", y le asignaremos el valor et1

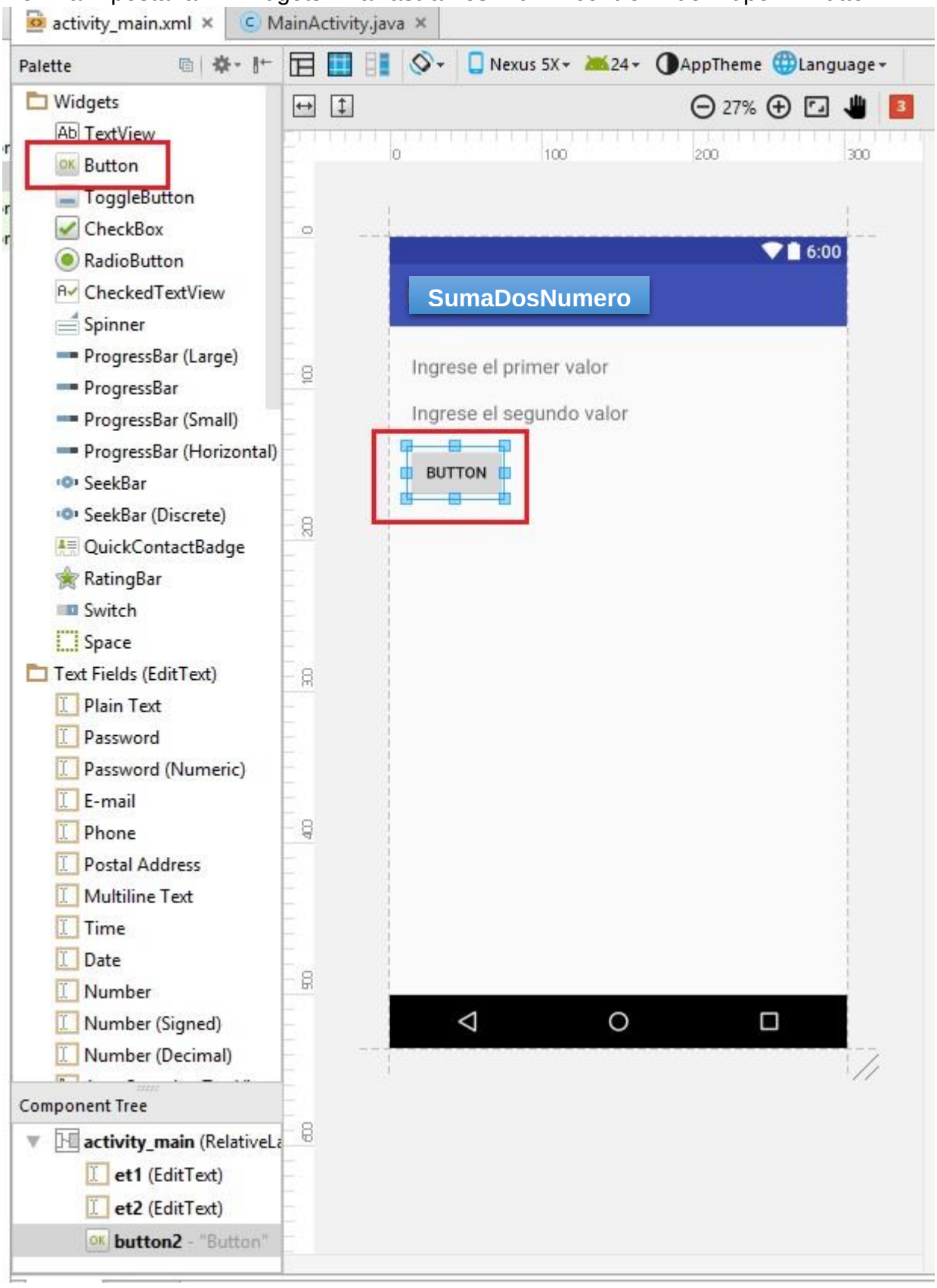


Entonces asignado como nombre a este objeto: et1 (recordemos que se trata de un objeto de la clase EditText), este nombre haremos referencia posteriormente desde el programa en Java.

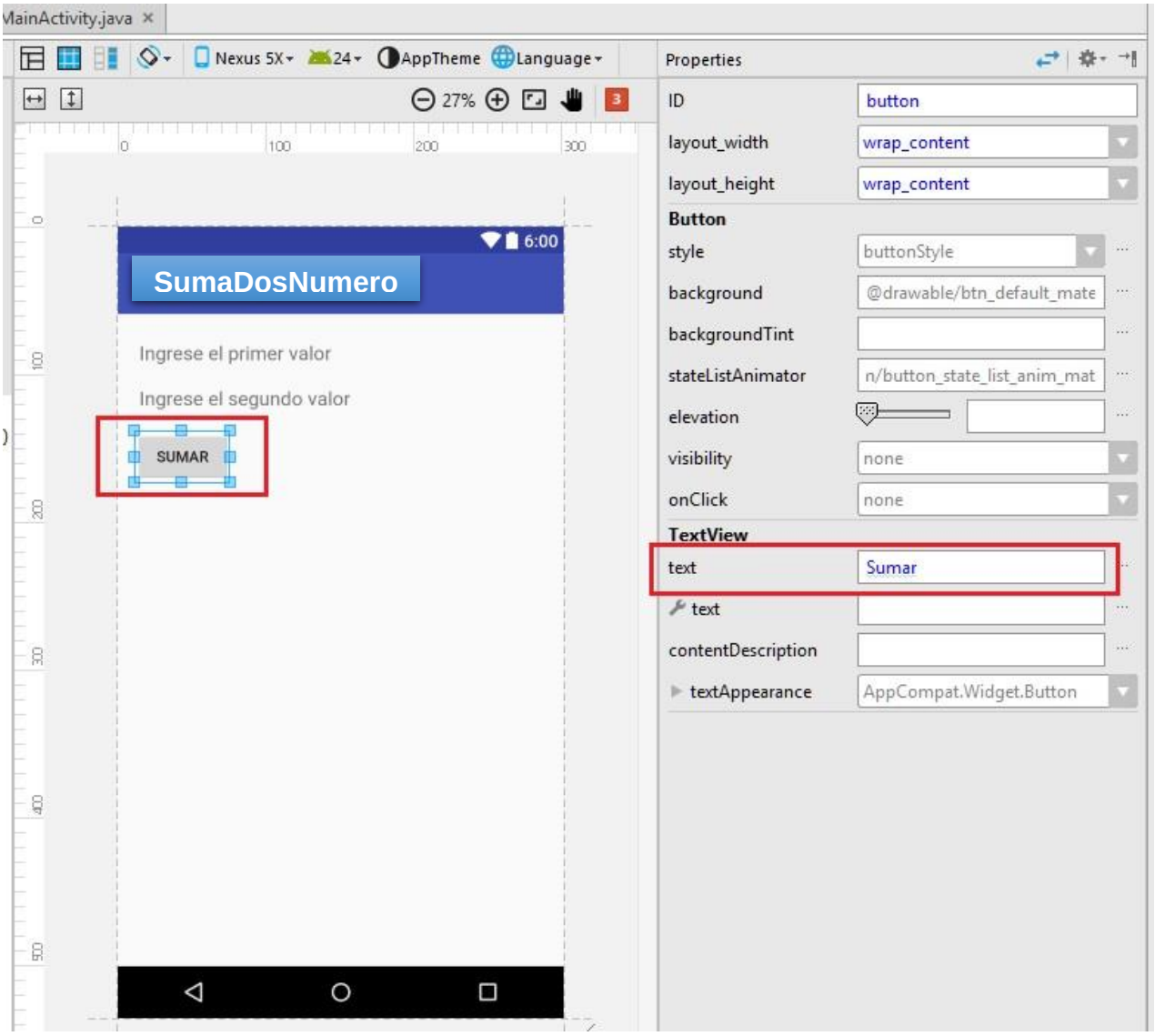
Efectuamos los mismos pasos para crear el segundo EditText de tipo "Number" (iniciamos las propiedades respectivas) Definimos el id con el nombre et2 y la propiedad hint con el mensaje "Ingrese el segundo valor", el resultado visual debe ser algo semejante a esto:



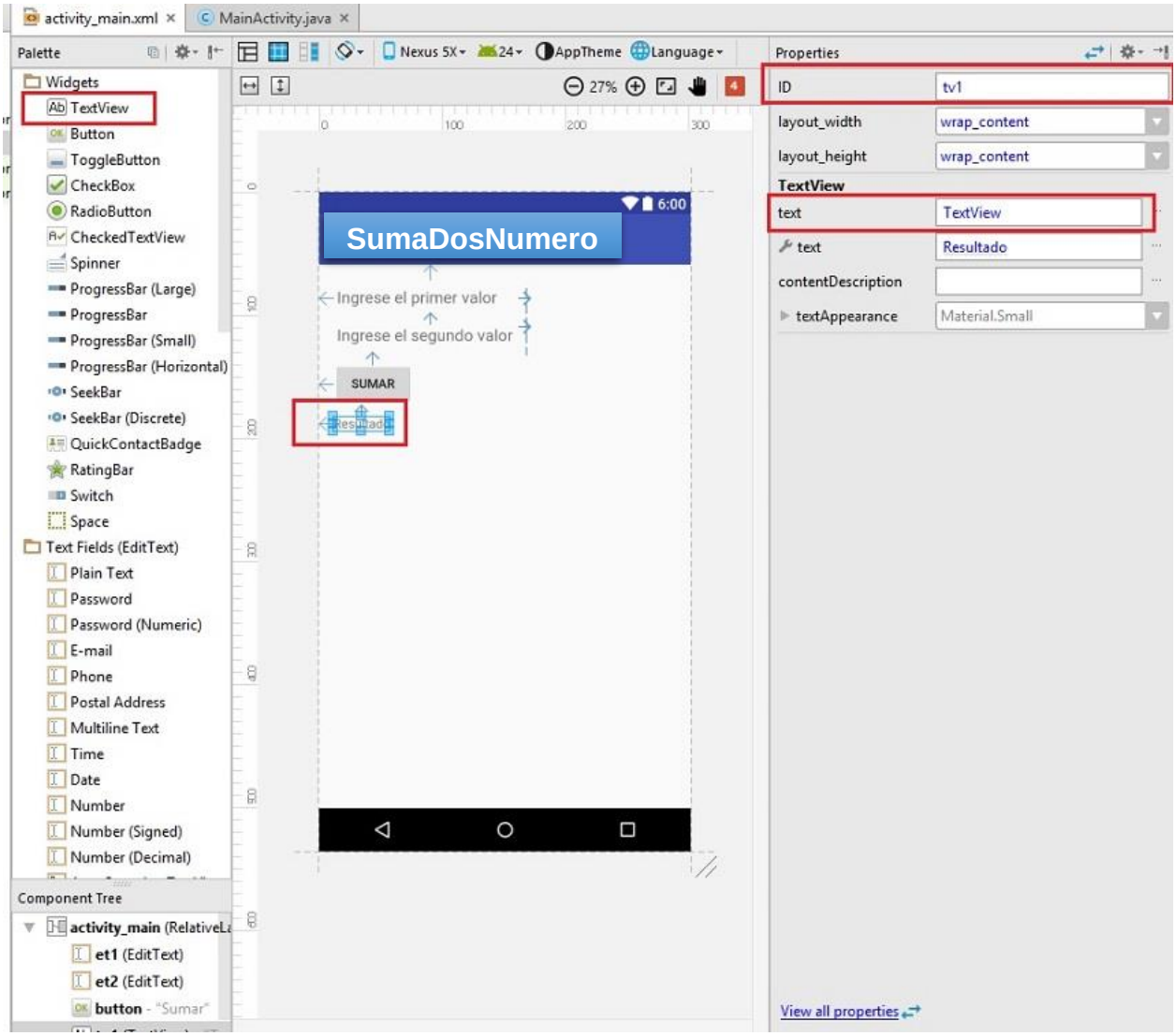
De la pestaña "Widgets" arrastramos un control de tipo "Button":



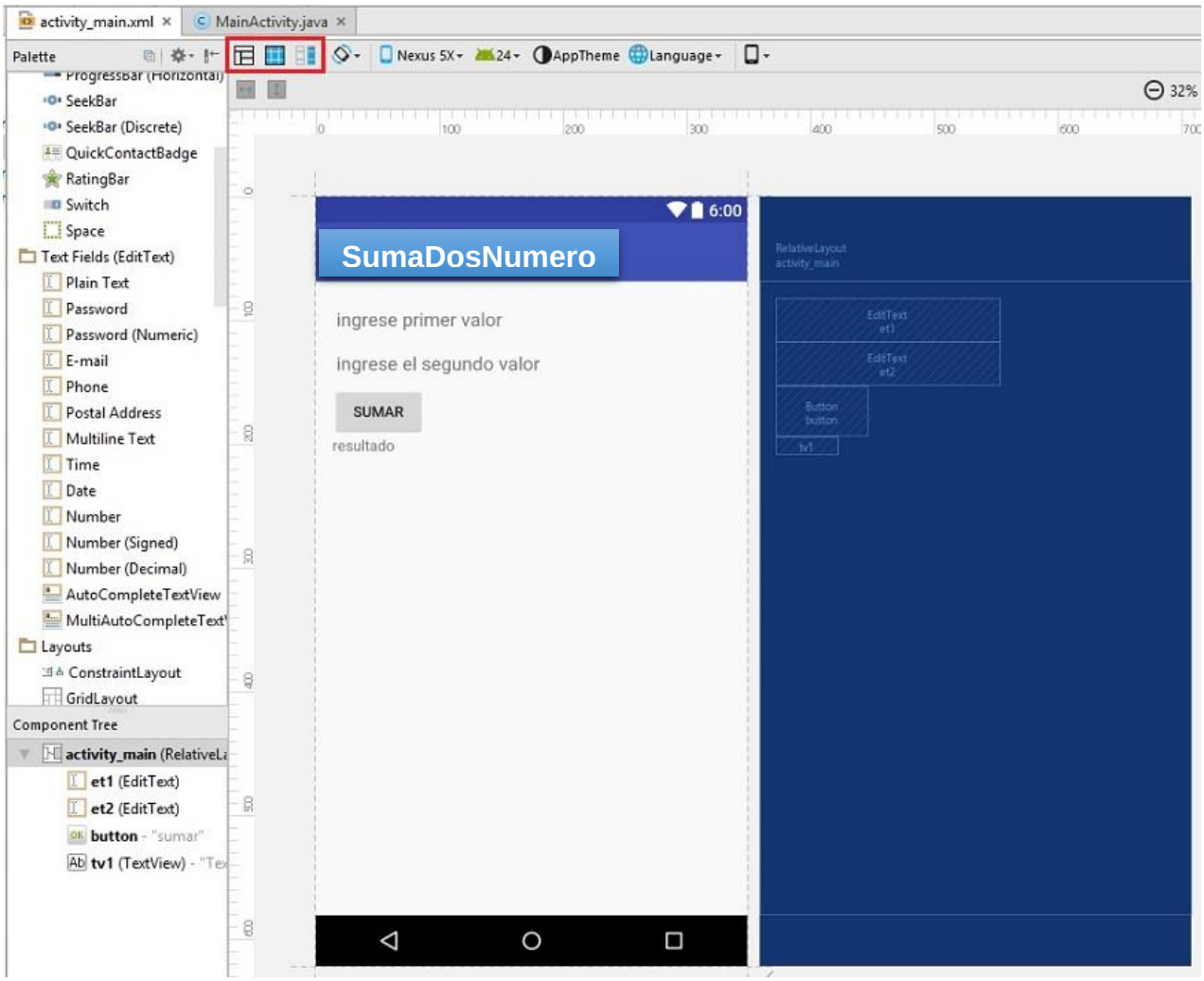
Iniciamos la propiedad text con el texto "Sumar" y la propiedad id la dejamos con el valor ya creado llamado "button":



Para terminar con nuestra interfaz visual arrastramos una componente de tipo "TextView" de la pestaña "Widgets". Definimos la propiedad id con el valor "tv1" y la propiedad text con el texto "Resultado":



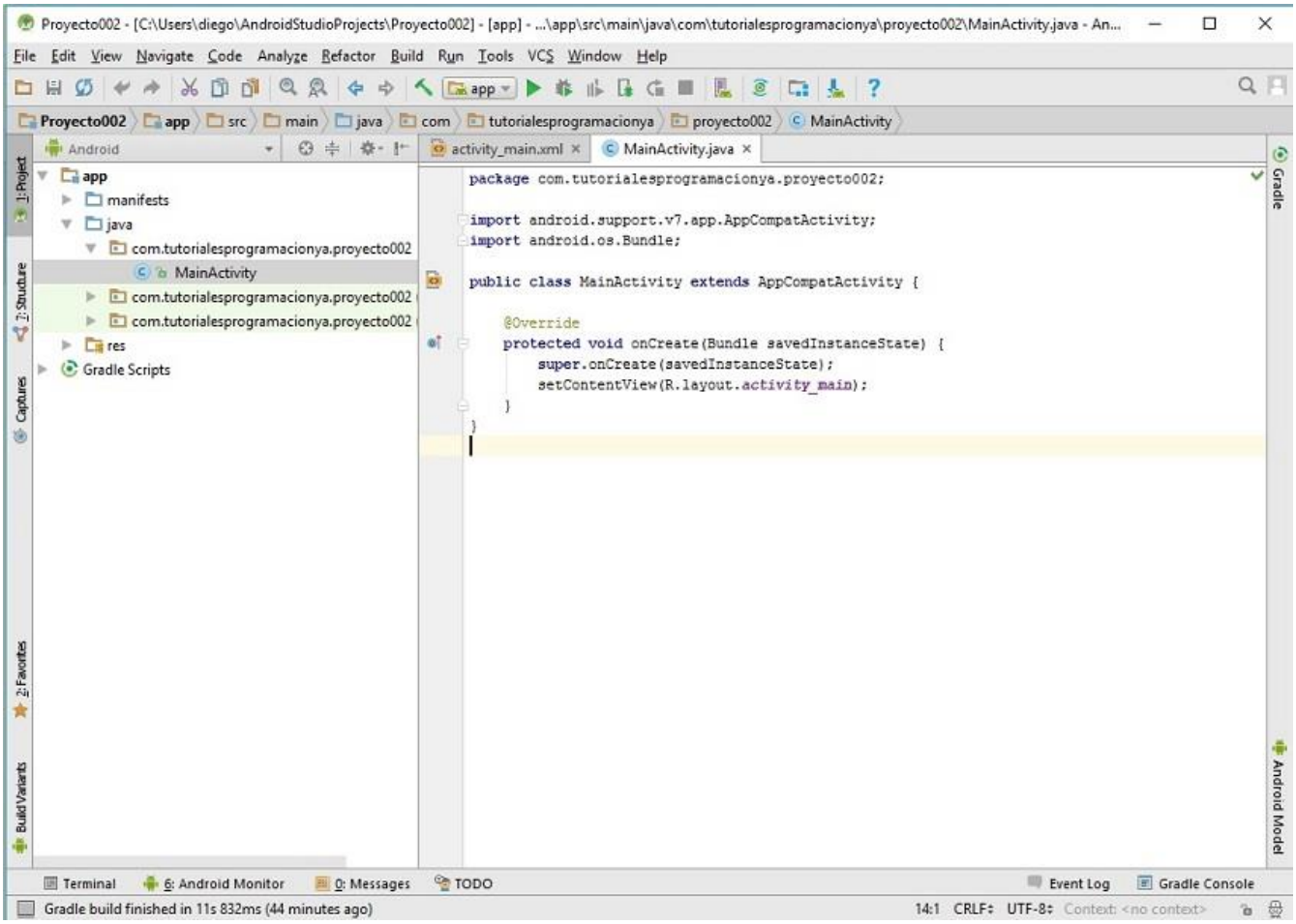
La interfaz final debe ser semejante a esta:



Si en este momento ejecutamos la aplicación aparece la interfaz visual correctamente pero cuando presionemos el botón no mostrará la suma.

Hasta ahora hemos trabajado solo con el archivo xml (activity_main.xml) donde se definen los controles visuales de la ventana que estamos creando.

Abrimos seguidamente el archivo MainActivity.java



La clase MainActivity hereda de la clase AppCompatActivity. La clase AppCompatActivity representa una ventana de Android y tiene todos los métodos necesarios para crear y mostrar los objetos que hemos dispuesto en el archivo xml.

El código fuente de la clase MainActivity.java es:

```
package com.tutorialesprogramacionya.proyecto002;

import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;

public class MainActivity extends AppCompatActivity {

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

Como mínimo se debe sobrescribir el método onCreate heredado de la clase AppCompatActivity donde procedemos a llamar al método setContentView pasando como referencia un valor almacenado en una constante llamada activity_main contenida en una clase llamada layout que a su vez la contiene una clase llamada R (veremos más adelante que el Android Studio se encarga de crear la clase R en forma automática y sirve como puente entre el archivo xml y nuestra clase MainActivity)

Captura de eventos.

Ahora viene la parte donde definimos variables en java donde almacenamos las referencias a los objetos definidos en el archivo XML.

Definimos tres variables, dos de tipo EditText y finalmente una de tipo TextView (estas dos clases se declaran en el paquete android.widget, es necesario importar dichas clases para poder definir las variables de dichas clases, la forma más fácil de importar las clases es una vez que definimos el objeto por ejemplo private EditText et1; veremos que aparece en rojo el nombre de la clase y nos invita el Android Studio a presionar las teclas "Alt" e "Intro" en forma simultánea. Luego el Android Studio codifica automáticamente la línea que importa la clase: import android.widget.EditText;):

```
package com.tutorialesprogramacionya.proyecto002;

import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.widget.EditText;
import android.widget.TextView;

public class MainActivity extends AppCompatActivity {

    private EditText et1;
    private EditText et2;
```

```
    private TextView tv1;

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

Recordar que la forma más fácil de importar las clases EditText y TextView es escribir las tres líneas:

```
    private EditText et1;
    private EditText et2;
    private TextView tv1;
```

y luego presionar las teclas "Alt" y "Enter" en cada nombre de clase que se debe importar
Esto hace que se escriban automáticamente los import:

```
import android.widget.EditText;
import android.widget.TextView;
```

Los nombres que le dí a los objetos en este caso coinciden con la propiedad id (no es obligatorio):

```
    private EditText et1;
    private EditText et2;
    private TextView tv1;
```

Para la clase Button no es necesario definir un atributo.
En el método onCreate debemos enlazar estas variables con los objetos definidos en el archivo XML, esto se hace llamando al método findViewById:

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

        et1=(EditText)findViewById(R.id.et1);
        et2=(EditText)findViewById(R.id.et2);
        tv1=(TextView)findViewById(R.id.tv1);
    }
```

Al método findViewById debemos pasar la constante creada en la clase R (recordemos que se crea automáticamente esta clase) el nombre de la constante si debe ser igual con el nombre de la propiedad del objeto creado en el archivo XML.

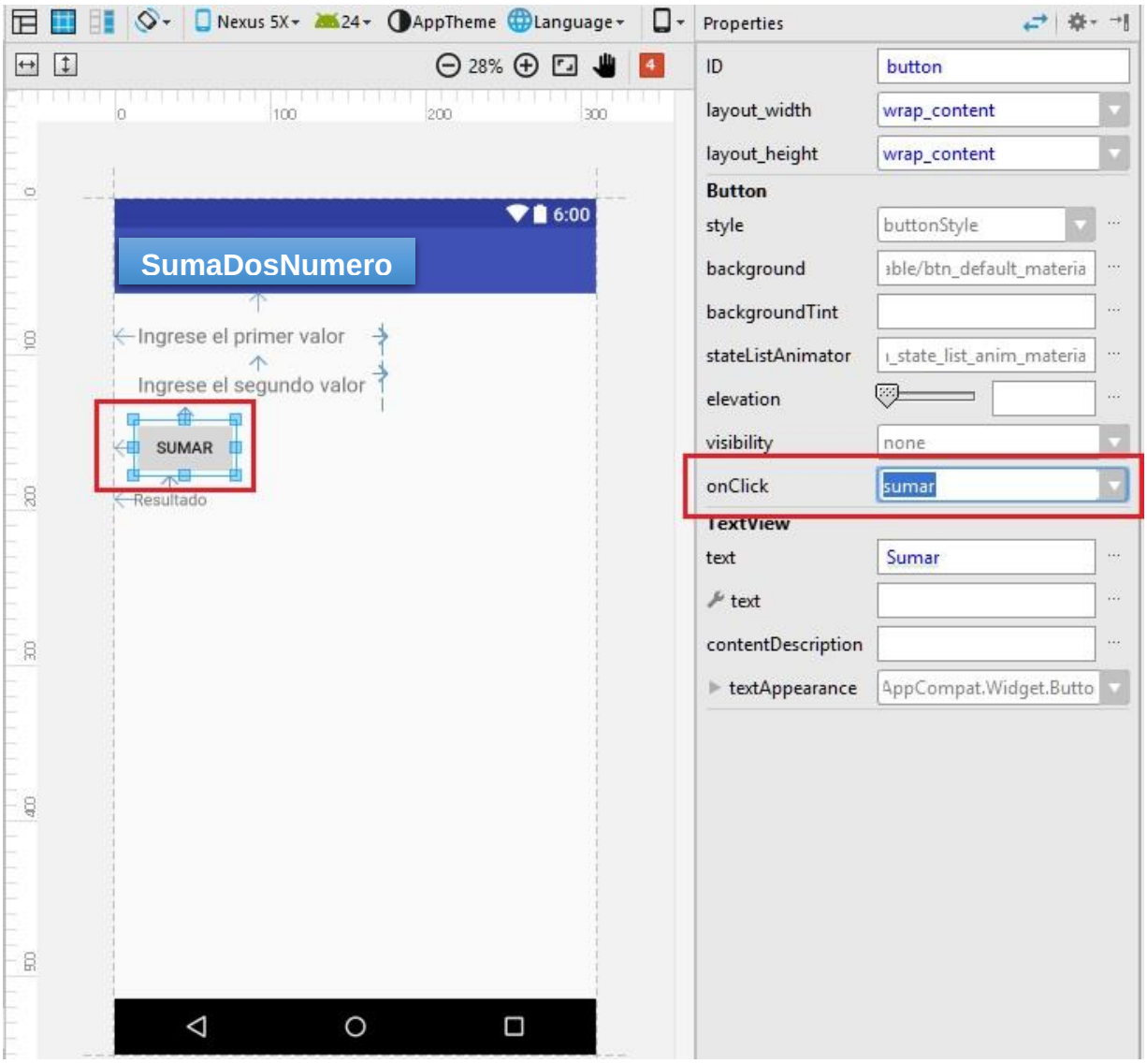
Como el método findViewById retorna un objeto de tipo View luego debemos utilizar el operador cast (es decir le antecedemos entre paréntesis el nombre de la clase)

Ya tenemos almacenados en las variables las referencias a los tres objetos que se crean al llamar al método:setContentView(R.layout.main); .
Ahora planteamos el método que se ejecutará cuando se presione el botón (el método debe recibir como parámetro un objeto de la clase View) En nuestro ejemplo lo llamé sumar:

```
package com.tutorialesprogramacionya.proyecto002;

import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.view.View;
import android.widget.EditText;
import android.widget.TextView;

public class MainActivity extends AppCompatActivity {

    private EditText et1;
    private EditText et2;
    private TextView tv1;

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

        et1=(EditText)findViewById(R.id.et1);
        et2=(EditText)findViewById(R.id.et2);
        tv1=(TextView)findViewById(R.id.tv1);
    }

    //Este método se ejecutará cuando se presione el botón
    public void sumar(View view) {

    }
}
```

Debemos importar lar clase View (presionamos las teclas "Alt" y luego "Enter" en forma simultanea)

Ahora debemos ir al archivo XML (vista de diseño) e inicializar la propiedad onClick del objeto button con el nombre del método que acabamos de crear (este paso es fundamental para que el objeto de la clase Button pueda llamar al método sumar que acabamos de crear):



Finalmente implementaremos la lógica para sumar los dos valores ingresados en los controles EditText:

```
public void sumar(View view) {
    String valor1=et1.getText().toString();
    String valor2=et2.getText().toString();
    int nro1=Integer.parseInt(valor1);
    int nro2=Integer.parseInt(valor2);
    int suma=nro1+nro2;
    String resu=String.valueOf(suma);
```

```
        tv1.setText(resu);
    }
```

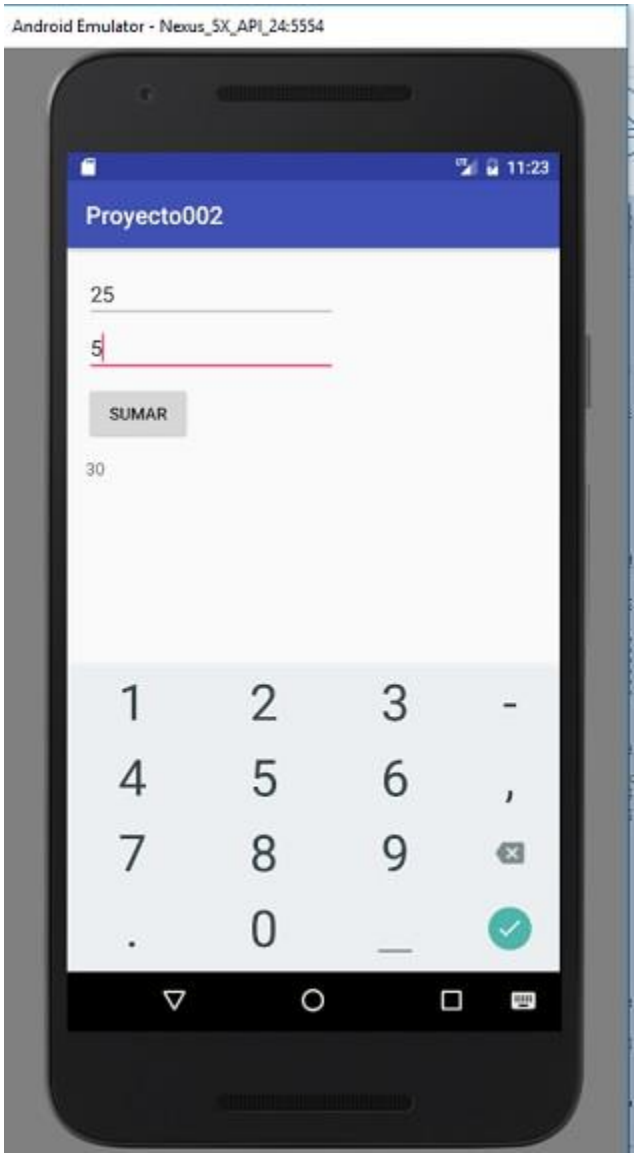
Extraemos el texto de los dos controles de tipo EditText y los almacenamos en dos variables locales de tipo String. Convertimos los String a tipo entero, los sumamos y el resultado lo enviamos al TextView donde se muestra la suma (previo a convertir la suma a String)

La clase completa queda entonces como:

```
package com.tutorialesprogramacionya.proyecto002;

import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.view.View;
import android.widget.EditText;
import android.widget.TextView;

public class MainActivity extends AppCompatActivity {

    private EditText et1;
    private EditText et2;
    private TextView tv1;

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

        et1=(EditText)findViewById(R.id.et1);
        et2=(EditText)findViewById(R.id.et2);
        tv1=(TextView)findViewById(R.id.tv1);
    }

    //Este método se ejecutará cuando se presione el botón
    public void sumar(View view) {
        String valor1=et1.getText().toString();
        String valor2=et2.getText().toString();
        int nro1=Integer.parseInt(valor1);
        int nro2=Integer.parseInt(valor2);
        int suma=nro1+nro2;
        String resu=String.valueOf(suma);
        tv1.setText(resu);
    }
}
```

Si ejecutamos nuestro programa podemos ver ahora que los controles EditText muestran los mensajes "Ingrese primer valor" e "Ingrese segundo valor" (la propiedad hint de los EditText muestran un mensaje que se borra automáticamente cuando el operador carga los enteros):

Luego de cargar dos valores al presionar el botón aparece en el TextView el resultado de la suma de los dos EditText :



2. Controles RadioGroup y RadioButton

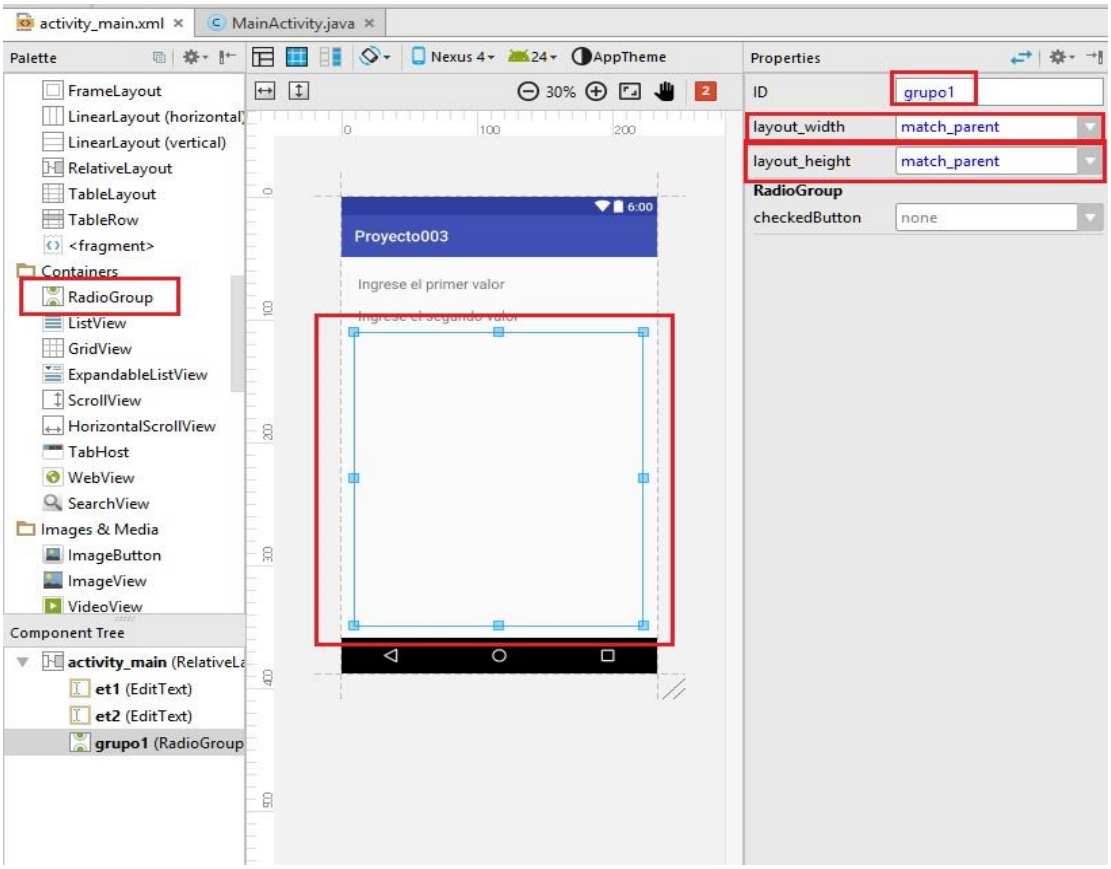
El objetivo de este concepto es practicar la implementación de un programa que requiera controles de tipo RadioButton para seleccionar una actividad. Aprenderemos como agrupar un conjunto de RadioButton y verificar cual está seleccionado.

Crear un proyecto llamado Proyecto003.

Problema:

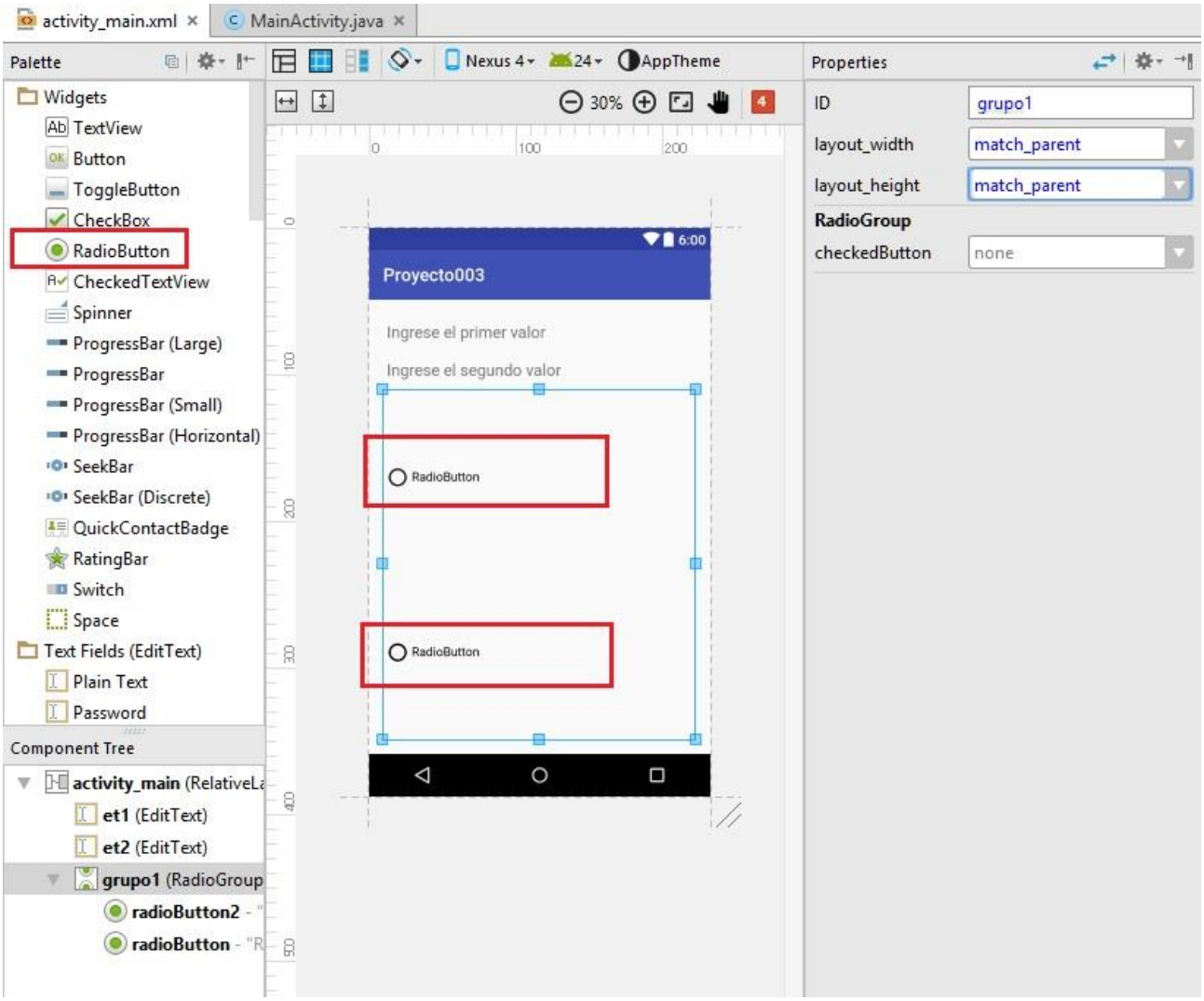
Realizar la carga de dos números en controles de tipo EditText. Mostrar mensajes que soliciten la carga de los valores dentro de los mismos EditText (propiedad hint). Disponer dos controles de tipo RadioButton para seleccionar si queremos sumar o restar dichos valores. Finalmente, mediante un control de tipo Button efectuamos la operación respectiva. Mostramos el resultado en un TextView.

El problema es similar al anterior. Disponemos dos controles EditText (Number) y configuramos sus propiedades id y hint. Para disponer los controles de tipo RadioButton debemos en realidad primero insertar un control de tipo RadioGroup (este control se encuentra en la paleta de componentes en la pestaña Containers de la "Palette"):



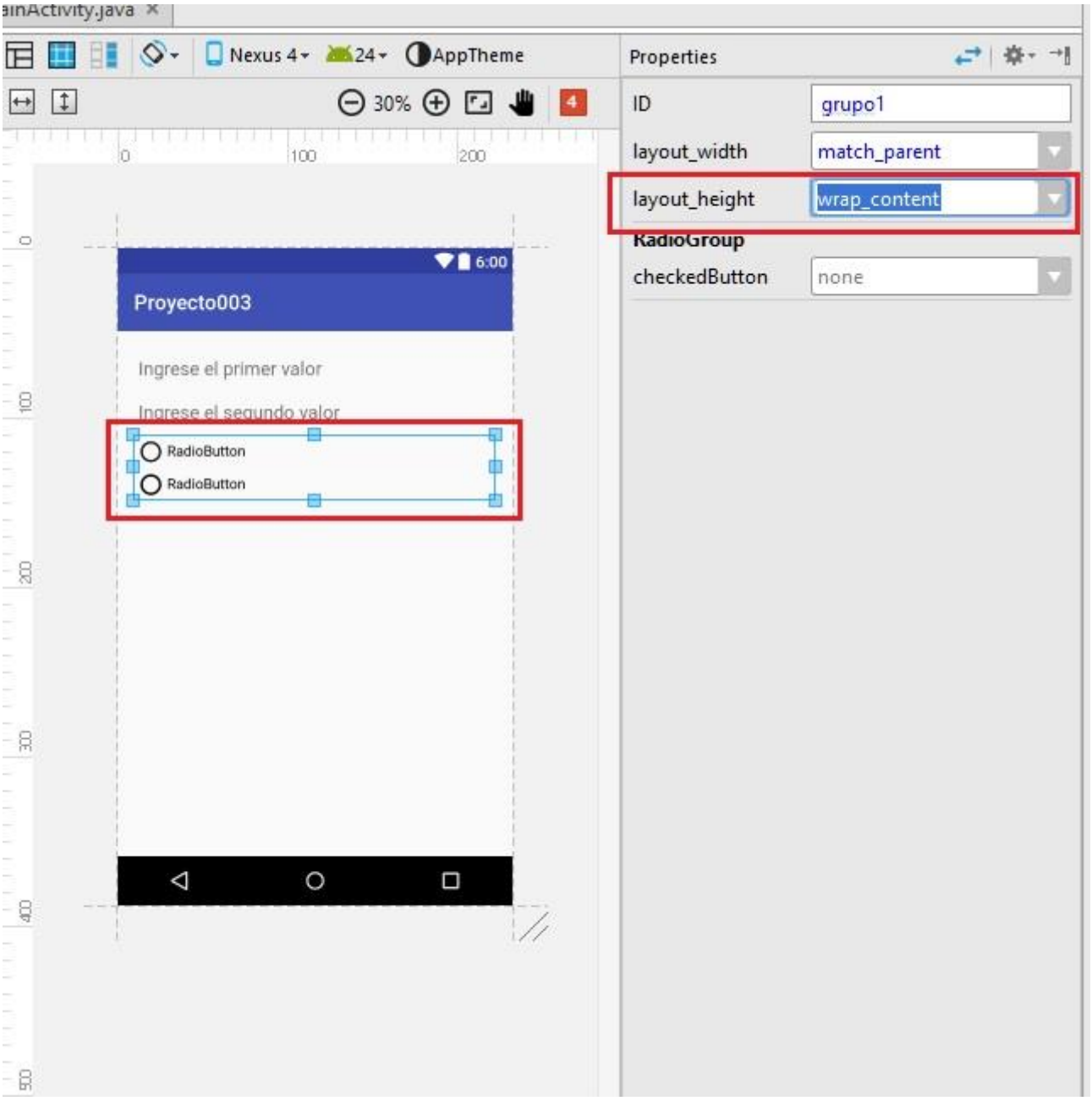
Luego de arrastrar el control RadioGroup iniciamos las propiedades layout_width y layout_height con el valor "match_parent". También definimos su id con el valor grupo1.

Ahora debemos arrastrar dos controles de la clase RadioButton de la pestaña "Widgets" dentro del RadioGroup



Nuestro problema solo requiere dos controles de tipo RadioButton.

Otra cosa muy importante es seleccionar nuevamente el control RadioGroup y cambiar la propiedad layout_height con el valor wrap_parent (para que el control RadioGroup solo ocupe el espacio de los dos controles RadioButton):



Ahora a los dos controles de tipo RadioButton definimos sus id (los llamaremos r1 y r2 respectivamente)
Cambiamos sus propiedades text por los textos "sumar" y "restar".

No olvidemos también cambiar los id de los controles EditText por et1 y et2 (igual que en el problema anterior)

Por último agreguemos un botón y un TextView para mostrar el resultado Inicializamos las propiedades del botón con los valores:

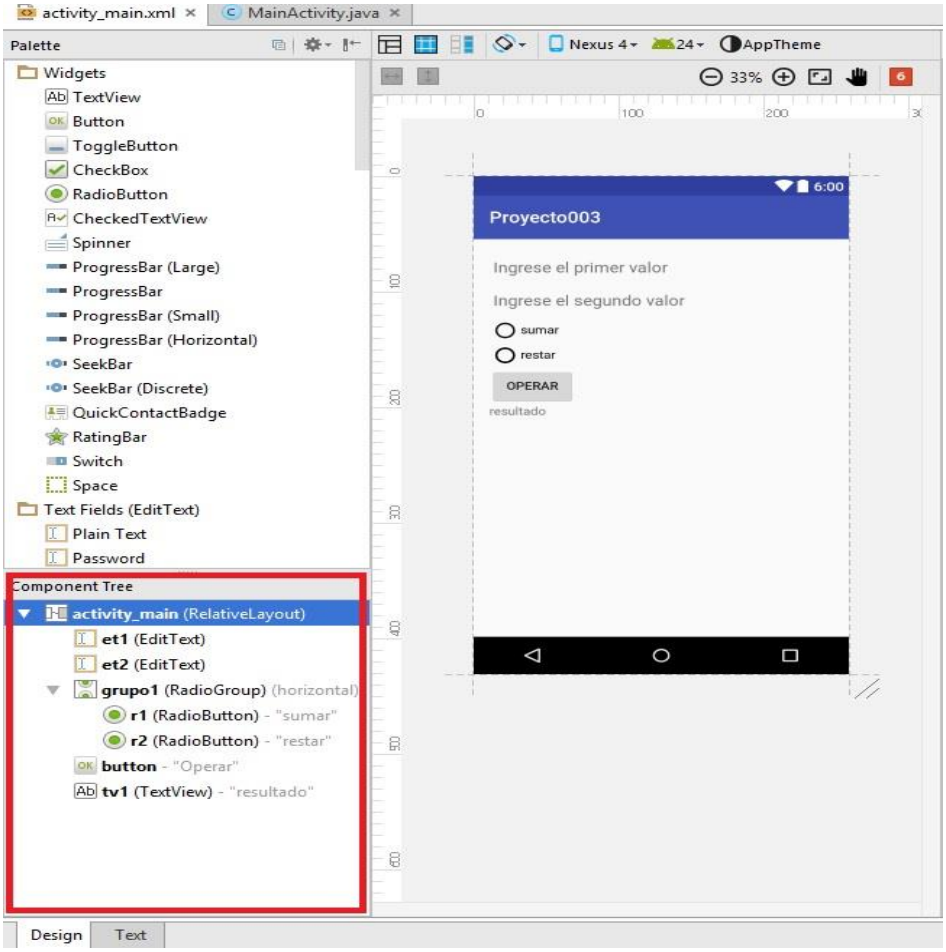
```
id : button
text : operar
```

Y el TextView con los valores:

```
id : tv1
text : resultado
```

Podemos controlar en la ventana "Component Tree" el id definido para cada control (et1, et2, grupo1, r1, r2, tv1)

También podemos observar de que clase es cada control visual y el texto de la propiedad text para aquellos controles que tienen sentido su inicialización.

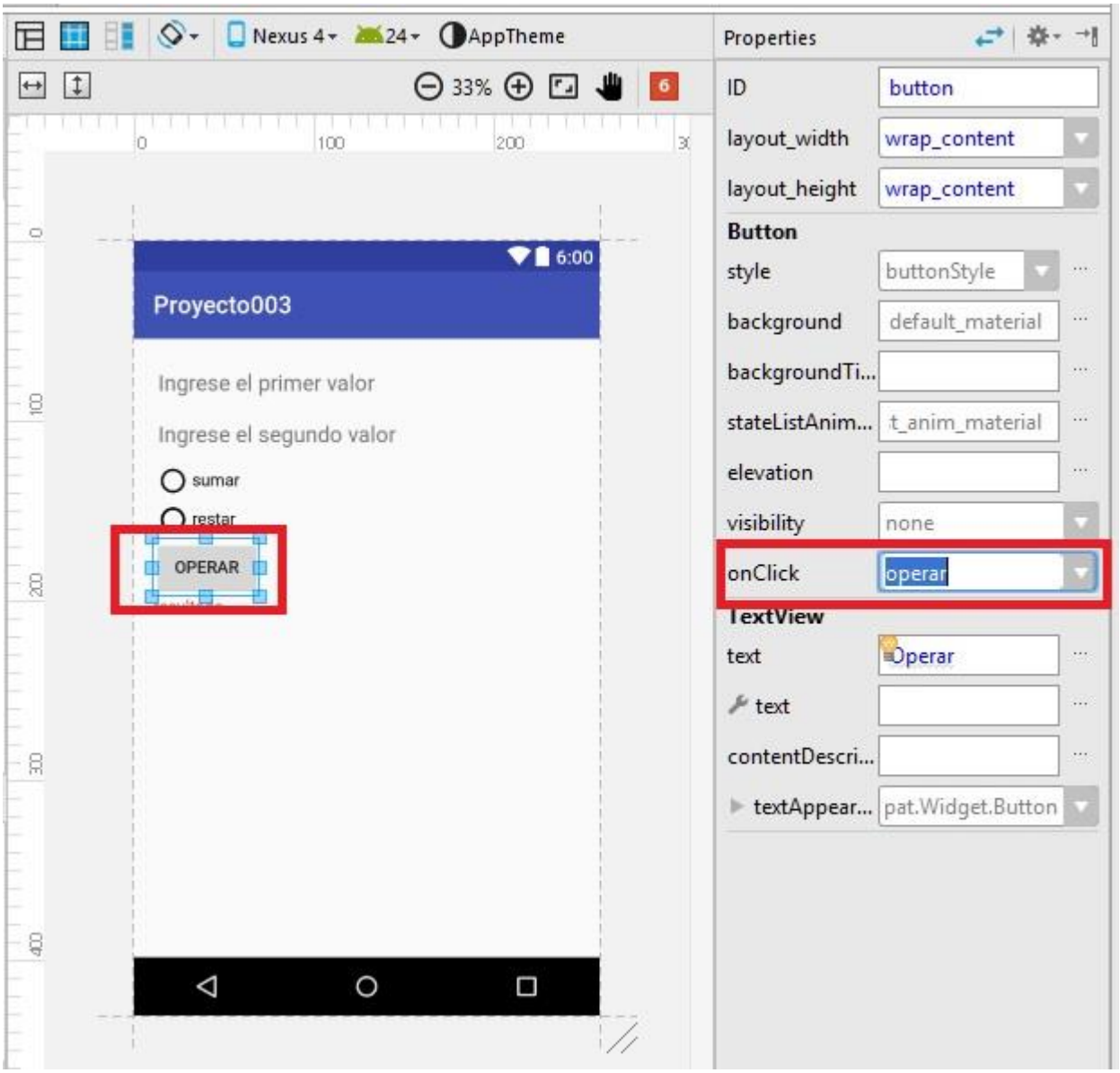


Captura del evento clic del button e identificación del RadioButton seleccionado.

El código fuente de la clase MaintActivity es:

```
package com.tutorialesprogramacionya.proyecto003;

import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.view.View;
import android.widget.EditText;
import android.widget.RadioButton;
import android.widget.TextView;

public class MainActivity extends AppCompatActivity {
    private EditText et1,et2;
    private TextView tv1;
    private RadioButton r1,r2;

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

        et1=(EditText)findViewById(R.id.et1);
        et2=(EditText)findViewById(R.id.et2);
        tv1=(TextView)findViewById(R.id.tv1);
        r1=(RadioButton)findViewById(R.id.r1);
        r2=(RadioButton)findViewById(R.id.r2);
    }

    //Este método se ejecutará cuando se presione el botón
    public void operar(View view) {
        String valor1=et1.getText().toString();
        String valor2=et2.getText().toString();
        int nro1=Integer.parseInt(valor1);
        int nro2=Integer.parseInt(valor2);
        if (r1.isChecked()==true) {
            int suma=nro1+nro2;
            String resu=String.valueOf(suma);
            tv1.setText(resu);
        } else
        if (r2.isChecked()==true) {
            int resta=nro1-nro2;
            String resu=String.valueOf(resta);
            tv1.setText(resu);
        }
    }

}
```

Primero debemos enlazar el objeto button con el método operar. Para esto similar al problema anterior seleccionamos el control button y cambiamos la propiedad onClick por el valor operar (si no hacemos esto nunca se ejecutará el método operar de la clase MainActivity):



Como podemos ver el código fuente es igual al problema anterior. Tenemos dos objetos más que debemos inicializar en el método onCreate:

```
r1=(RadioButton)findViewById(R.id.r1);
r2=(RadioButton)findViewById(R.id.r2);
```

Las variables r1 y r2 son de la clase RadioButton y son necesarios en el método operar para verificar cual de los dos RadioButton están seleccionados. La clase

RadioButton tiene un método llamado isChecked que retorna true si dicho elemento está seleccionado:

```
public void operar(View view) {
    String valor1=et1.getText().toString();
    String valor2=et2.getText().toString();
    int nro1=Integer.parseInt(valor1);
    int nro2=Integer.parseInt(valor2);
    if (r1.isChecked()==true) {
        int suma=nro1+nro2;
        String resu=String.valueOf(suma);
        tv1.setText(resu);
    } else
    if (r2.isChecked()==true) {
        int resta=nro1-nro2;
        String resu=String.valueOf(resta);
        tv1.setText(resu);
    }
}
```