

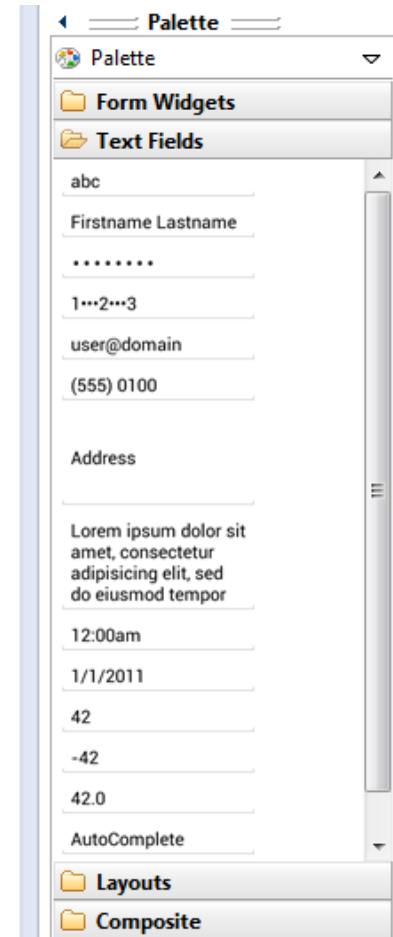
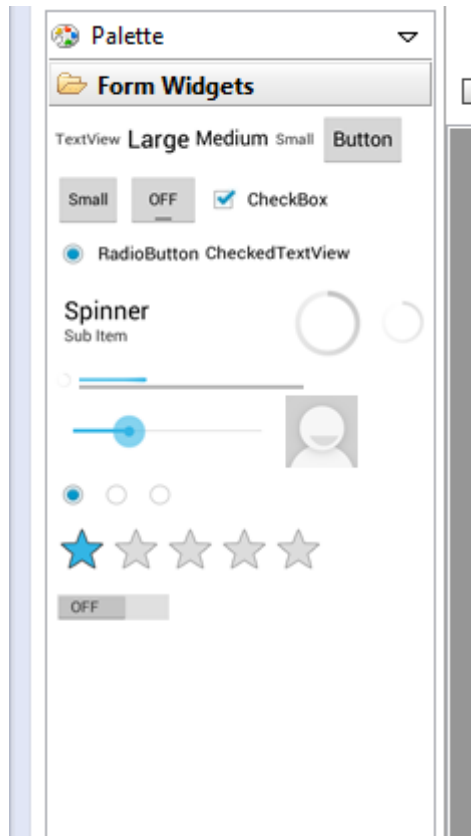


Interfaz de Usuario I

Curso: Desarrollando aplicaciones con Android

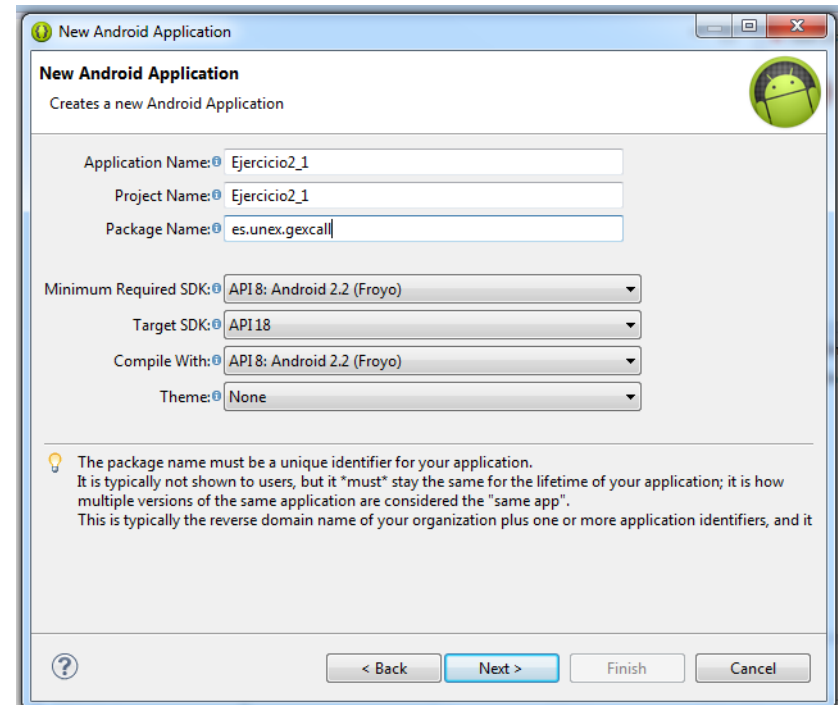
Widgets I

- ▶ En este tema vamos a probar los Widgets básicos en el diseño gráfico y su funcionamiento



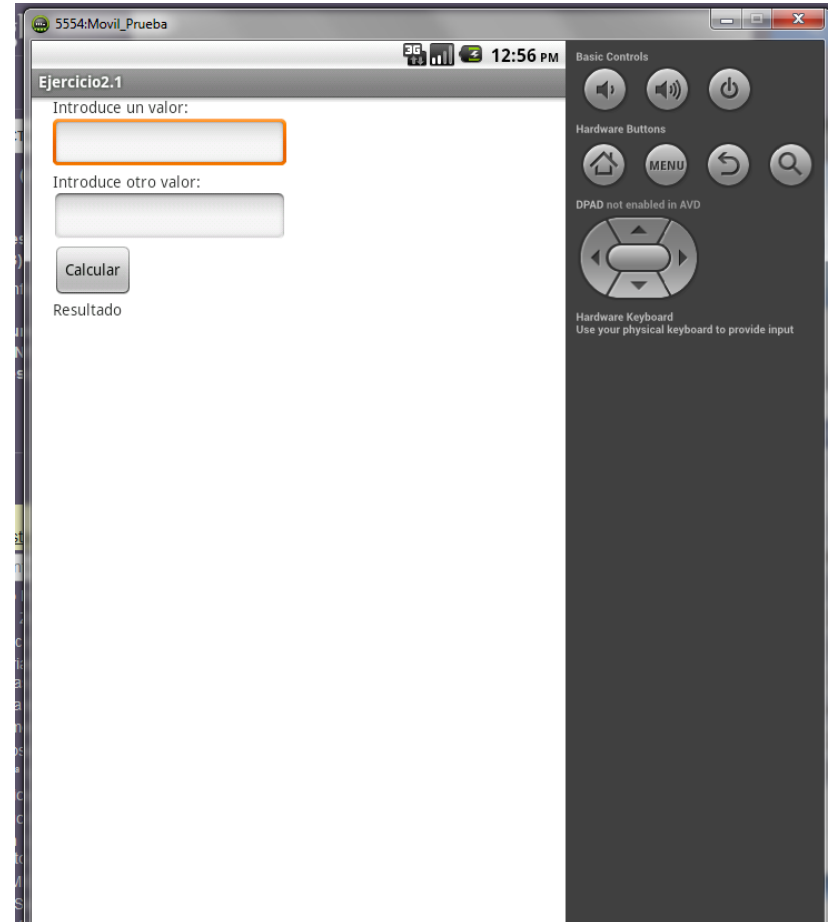
Widgets II

- ▶ Crea un proyecto por cada ejercicio
 - ▶ Ejercicio 2.1: Button
 - ▶ Ejercicio 2.2: Radio Button
 - ▶ Ejercicio 2.3: Spinner
 - ▶ Ejercicio 2.4: Listview
 - ▶ Ejercicio 2.5: ImageButton
 - ▶ Ejercicio 2.6: CheckBox



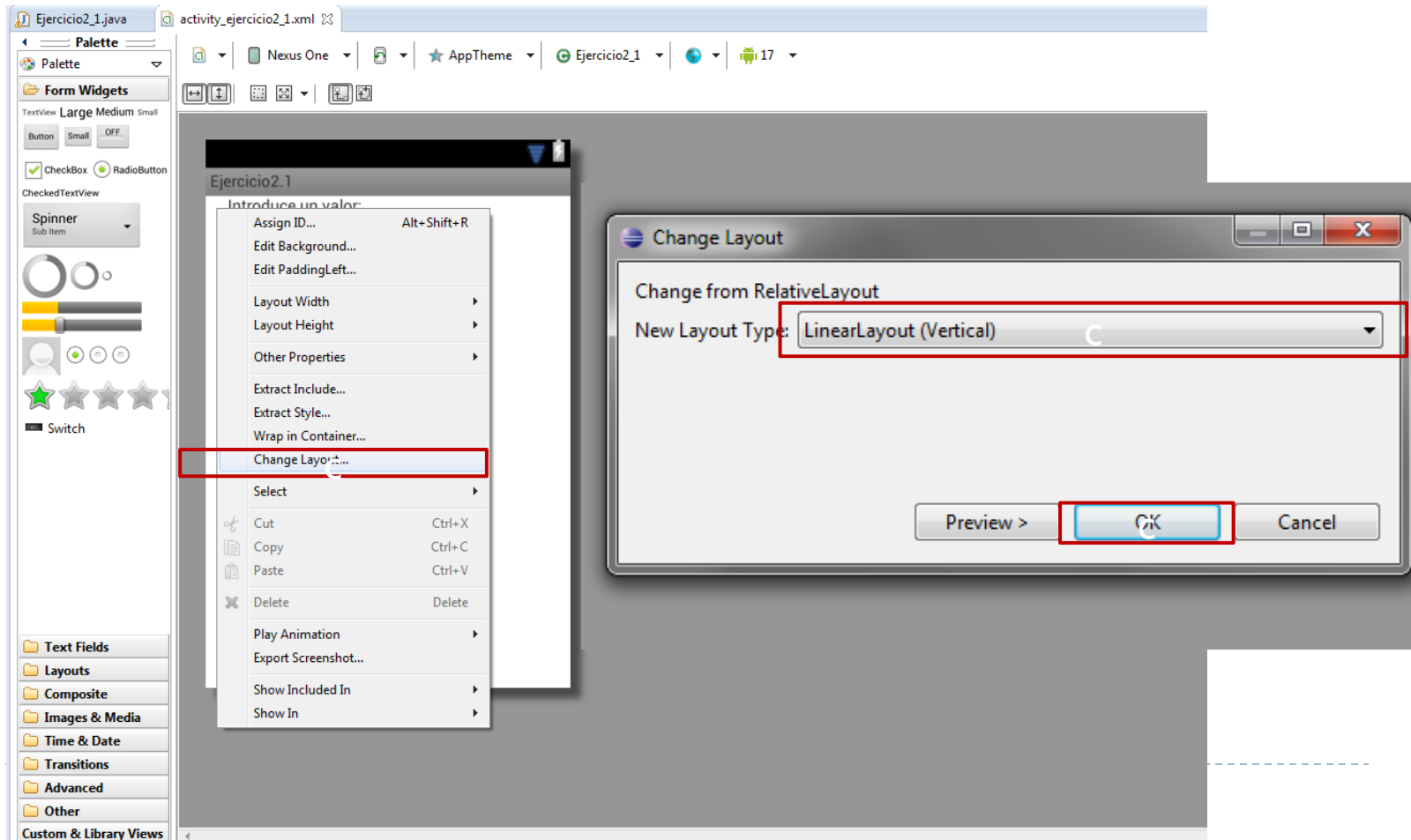
Ejercicio 2.1 – Calculadora (Botón)

Crear una calculadora sencilla que sume dos números



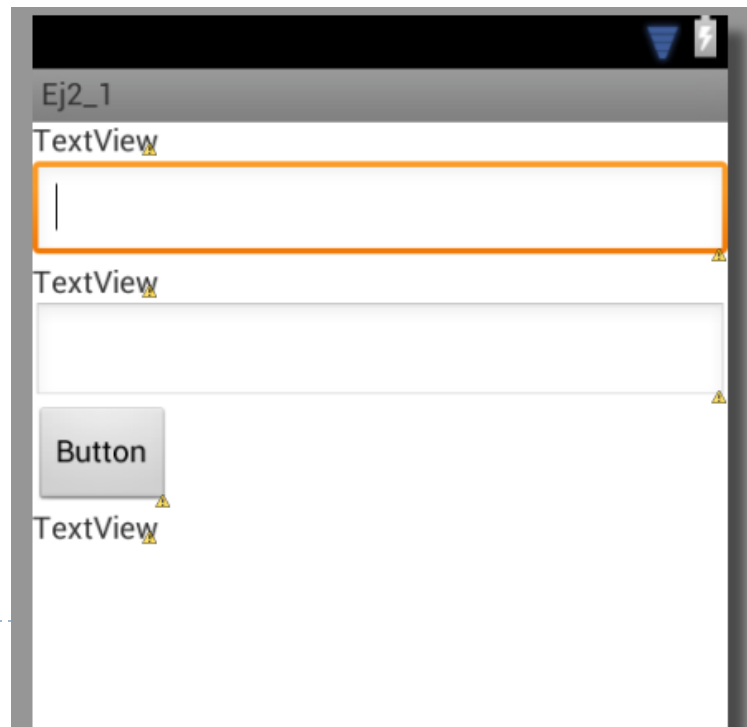
Ejercicio 2.1 – Calculadora (Botón)

- Un vez creado el proyecto, cambiar Layout a LinearLayout (Vertical), botón derecho sobre el layout.



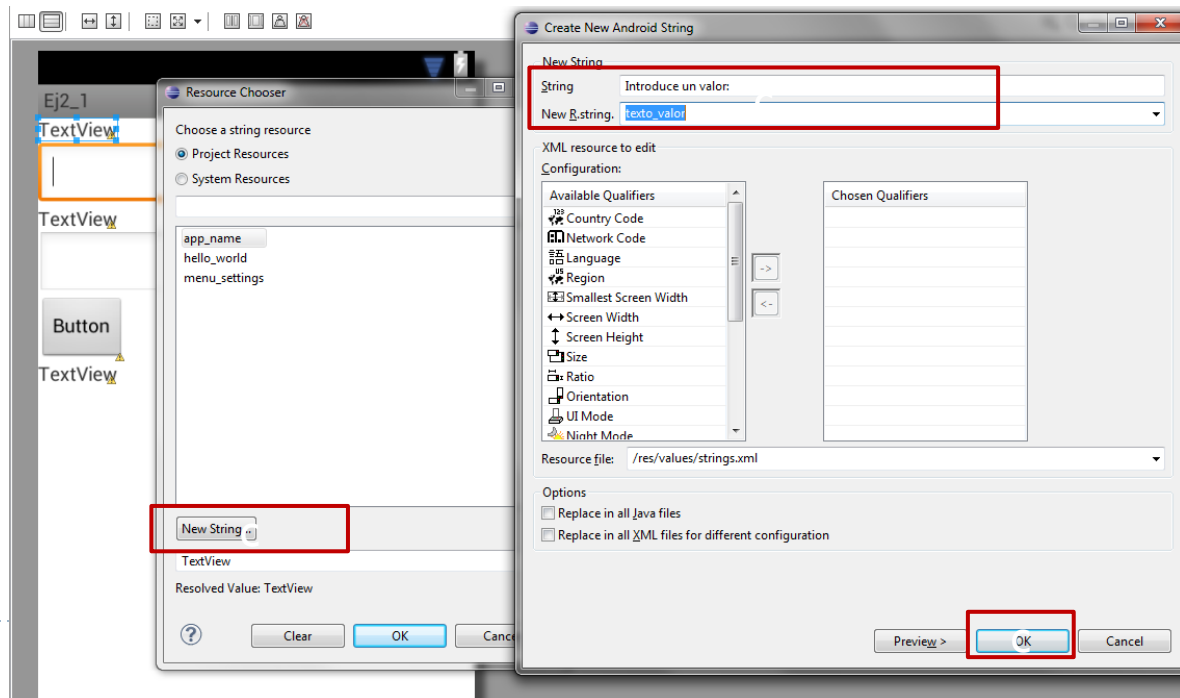
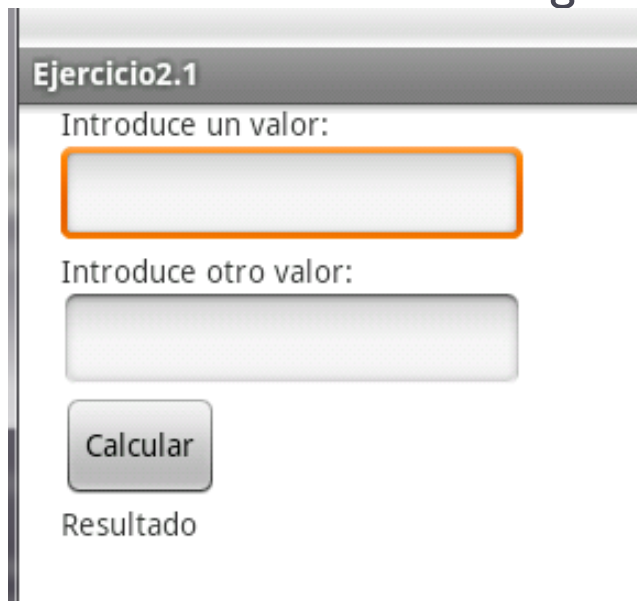
Ejercicio 2.1 – Calculadora (Botón)

- ▶ Desde la paleta Añade:
- ▶ Un TextView (Form widgets) y un EditText (Text fields-> Plain Text) para el valor1
- ▶ Un TextView y un EditText para el valor2
- ▶ Un Botón para calcular el resultado
- ▶ Un TextView para mostrar el resultado



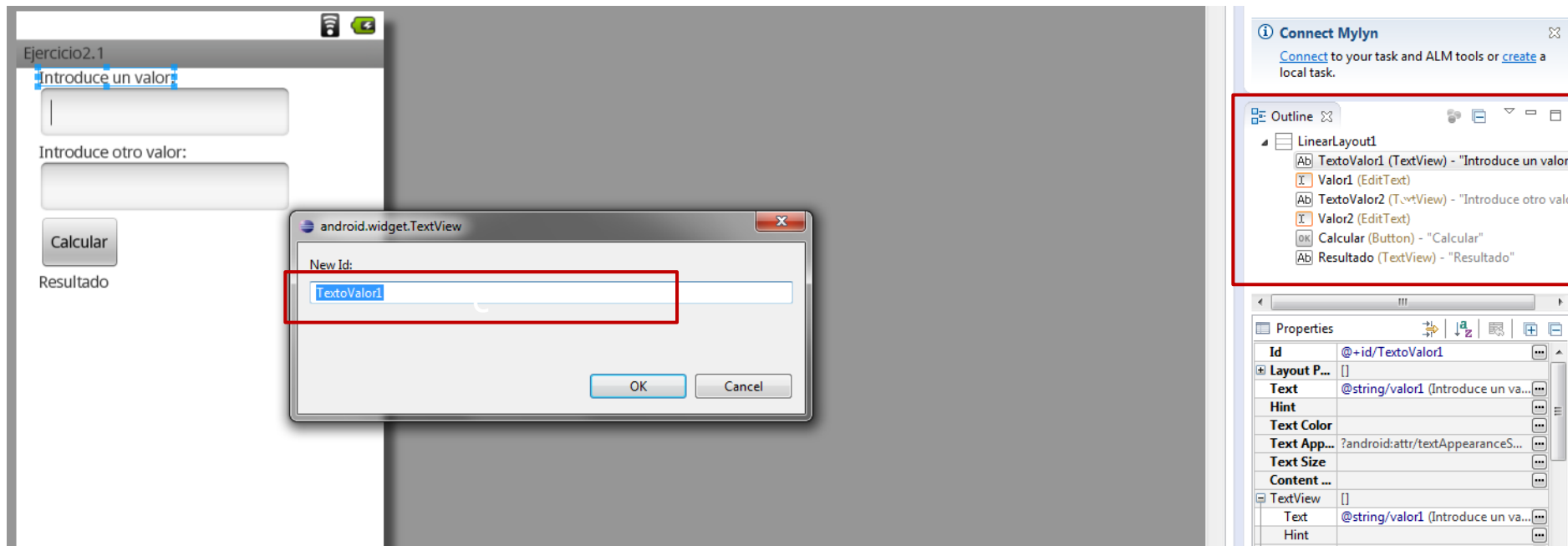
Ejercicio 2.1 – Calculadora (Botón)

- ▶ Modifica los textos de los TextViews y el botón.
 - ▶ No se recomienda asignarle valor directamente sino que habría que añadir el texto al strings.xml para futuras traducciones de la interfaz.
 - ▶ Esto lo podemos hacer directamente con el botón derecho sobre el widget -> Edit Text y en el menú que aparece crear un nuevo string con “New String”



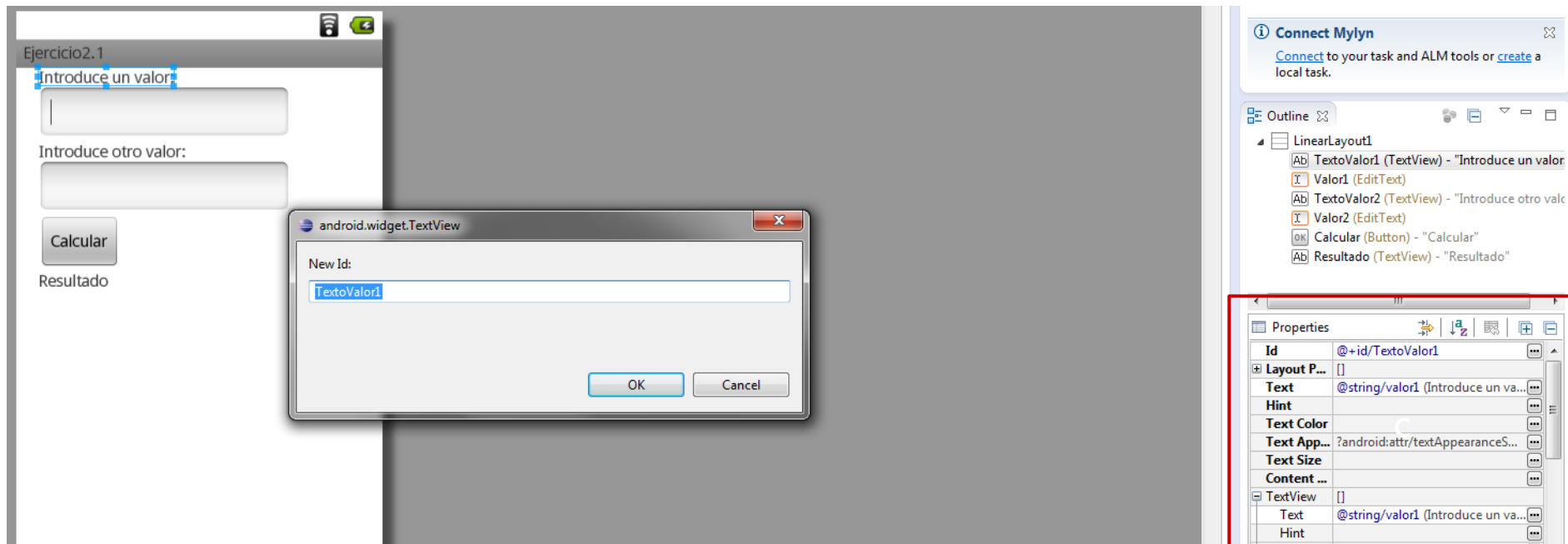
Ejercicio 2.1 – Calculadora (Botón)

- ▶ También es muy importante que modifiquemos los identificadores de todos los widgets. Botón derecho sobre el Widget y “Edit Id”
- ▶ A la derecha en outline, os aparecerán los nombres y será más fácil identificarlos.



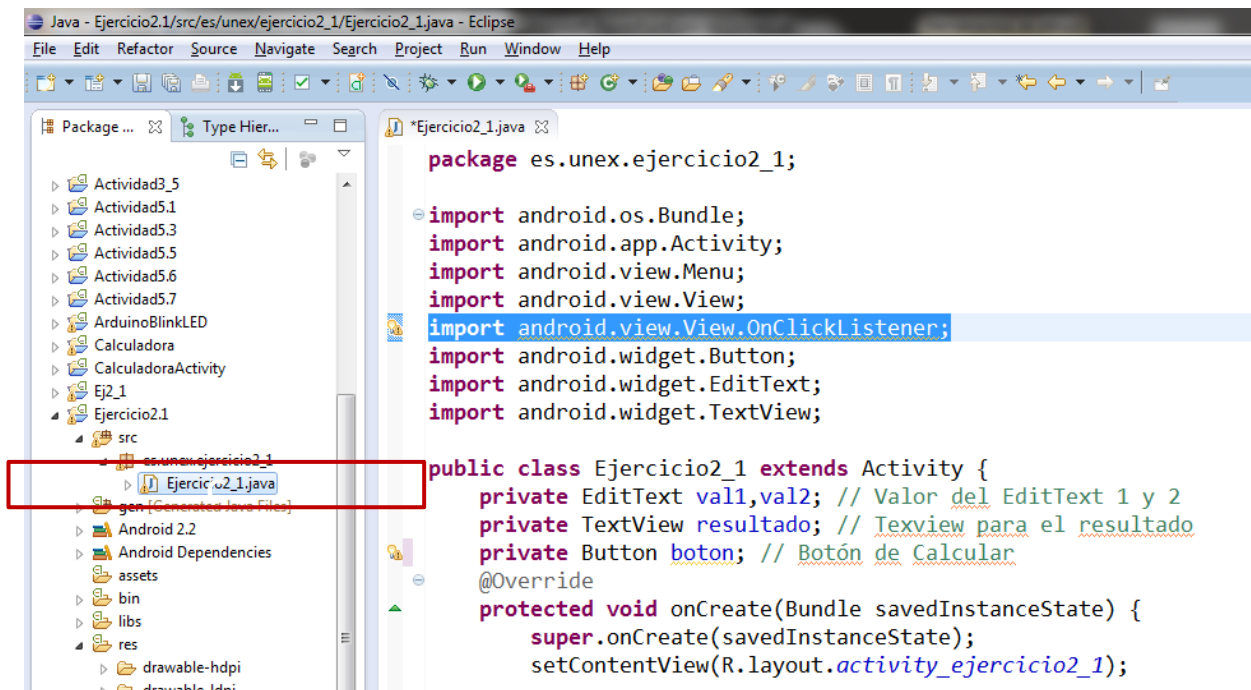
Ejercicio 2.1 – Calculadora (Botón)

- ▶ Con los dos pasos anteriores hemos cambiado las propiedades “Id” y “Text” de nuestro Widget. Como podemos ver abajo a la derecha.



Ejercicio 2.1 – Calculadora (Botón)

- Dentro de la carpeta “src” encontraremos el .java correspondiente a nuestro Activity.



Ejercicio 2.1 – Calculadora (Botón)

- ▶ Ahora vamos a añadir variables al código de nuestro Activity para poder acceder y modificar sus propiedades.
- ▶ Lo añadiremos justo después de la declaración de la clase:

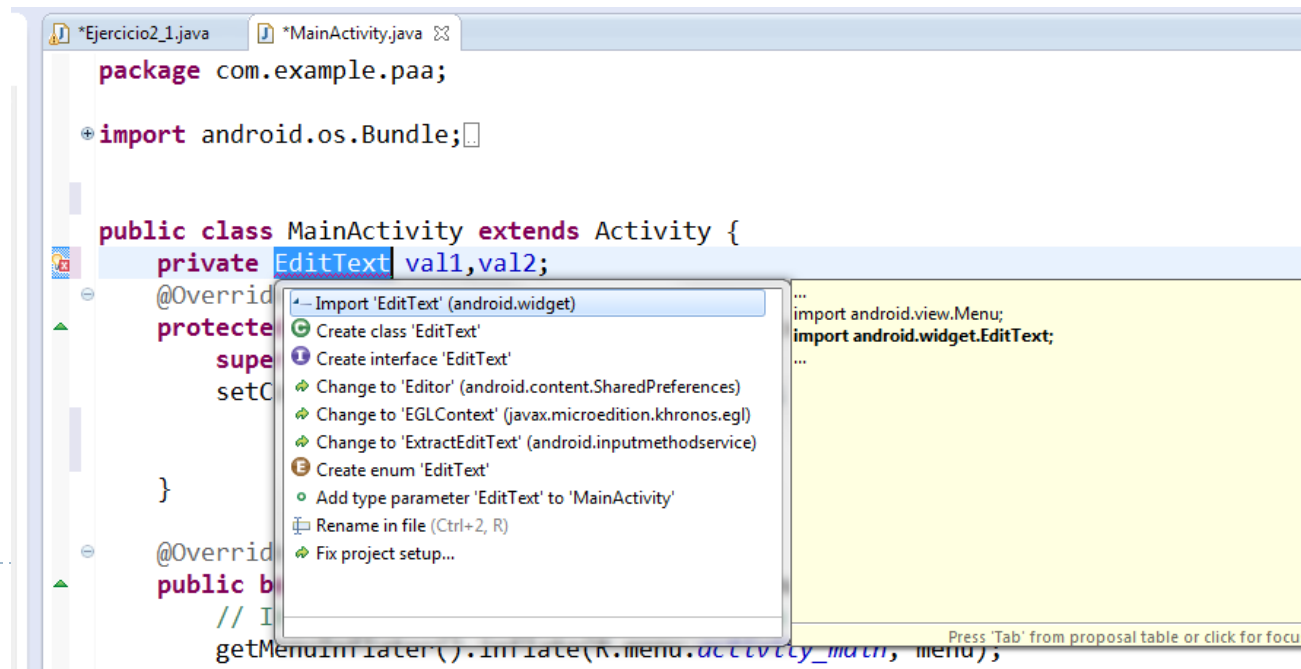
```
public class Ejercicio2_1 extends Activity {  
    private EditText val1, val2; // Valor del EditText 1 y 2  
    private TextView resultado; // Texview para el resultado  
    private Button boton; // Botón de Calcular  
    @Override  
    protected void onCreate(Bundle savedInstanceState) {
```

... .

Ejercicio 2.1 – Calculadora (Botón)

- ▶ Para cada nuevo tipo de Widget (View) tendremos que incorporar la librería correspondiente.
- ▶ Cuando os aparezca el error por ejemplo al no estar declarado EditText, pinchando en la bombilla podemos añadir la importación de la biblioteca y se añadirá la siguiente línea al area de importación:

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



Ejercicio 2.1 – Calculadora (Botón)

- ▶ El siguiente paso es asociar las variables a los recursos dentro del “onCreate”. Este método se ejecuta cuando se crea la ventana.

```
val1=(EditText)findViewById(R.id.Valor1);
```

- ▶ Asignamos al “val1” el View, lo encontramos con el método “findViewById” indicando R.id.Valor1 que es el nombre que especificamos en el Layout. Si ponemos R.id y añadimos un punto se nos listaran todos los recursos disponibles en el layout.

```
protected void onCreate(Bundle savedInstanceState) {  
    super.onCreate(savedInstanceState);  
    setContentView(R.layout.activity_ejercicio2_1);  
  
    //Asociamos las variables a los recursos  
    val1=(EditText)findViewById(R.id.Valor1);  
    val2=(EditText)findViewById(R.id.Valor2);  
    resultado=(TextView)findViewById(R.id.Resultado);  
    boton =(Button)findViewById(R.id.Calcular);  
}
```

Ejercicio 2.1 – Calculadora (Botón)

- ▶ Ahora debemos de crear el método que se llamara desde el botón (debe llevar un View como parámetro):

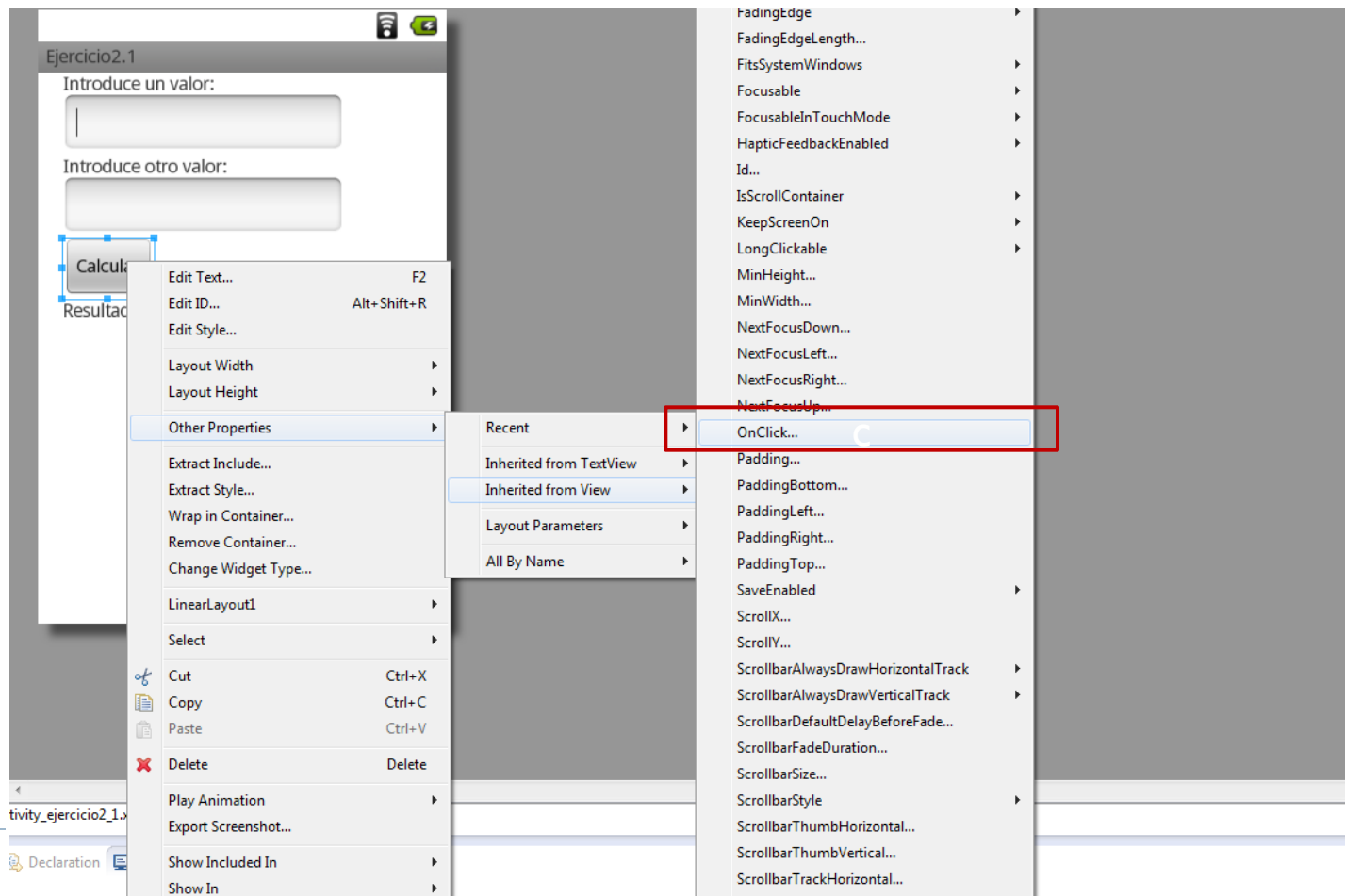
```
public void sumar(View view) { ...}
```

- ▶ Añadimos el método después del “OnCreateOptionsMenu”

```
public boolean onCreateOptionsMenu(Menu menu) {  
    // Inflate the menu; this adds items to the action bar if it is present.  
    getMenuInflater().inflate(R.menu.activity_ejercicio2_1, menu);  
    return true;  
}  
  
//Suma los valores al hacer clic  
public void sumar(View view) {  
    //Obtenemos el valor del primer operando  
    String valor1 = val1.getText().toString();  
    //Obtenemos el valor del segundo operando  
    String valor2 = val2.getText().toString();  
    //Los convertimos a entero  
    int numero1 = Integer.parseInt(valor1);  
    int numero2 = Integer.parseInt(valor2);  
    //Realizamos la suma  
    int suma = numero1+numero2;  
    //Convertimos el resultado a String y lo asignamos al recurso setText  
    resultado.setText(String.valueOf(suma));  
}  
}
```

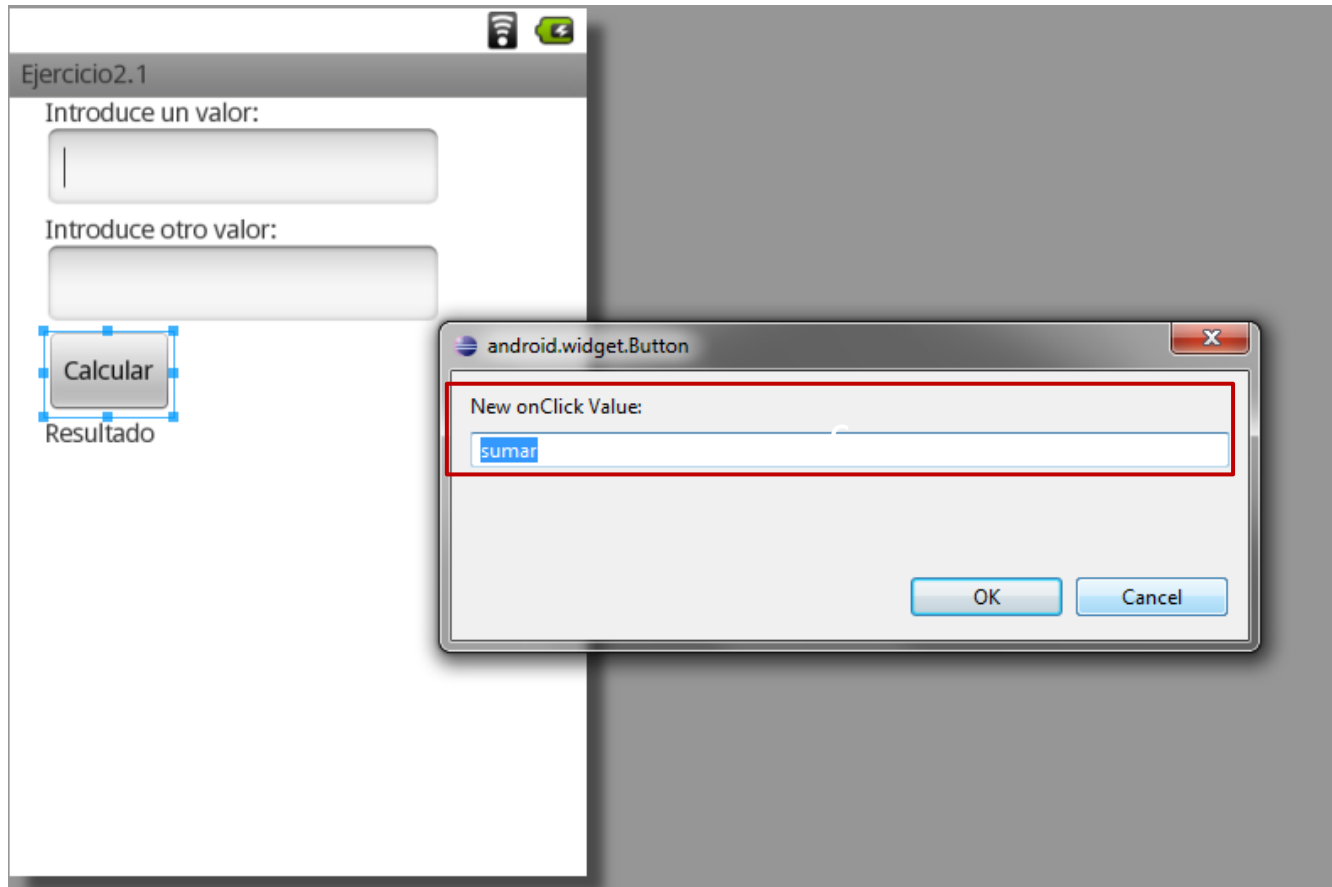
Ejercicio 2.1 – Calculadora (Botón)

- ▶ Finalmente tendremos que añadir el evento al botón.
- ▶ Volvemos a la vista del layout y accedemos a la propiedad onclick:



Ejercicio 2.1 – Calculadora (Botón)

- ▶ En la ventana que nos aparece, escribimos el nombre del método: (¡¡OjO con mayúsculas y minúsculas!!)



Ejercicio 2.1 – Calculadora (Botón)

- ▶ Ya podéis probar vuestra calculadora sencilla:



Ejercicio 2.1 – Calculadora (Botón)- Opcional

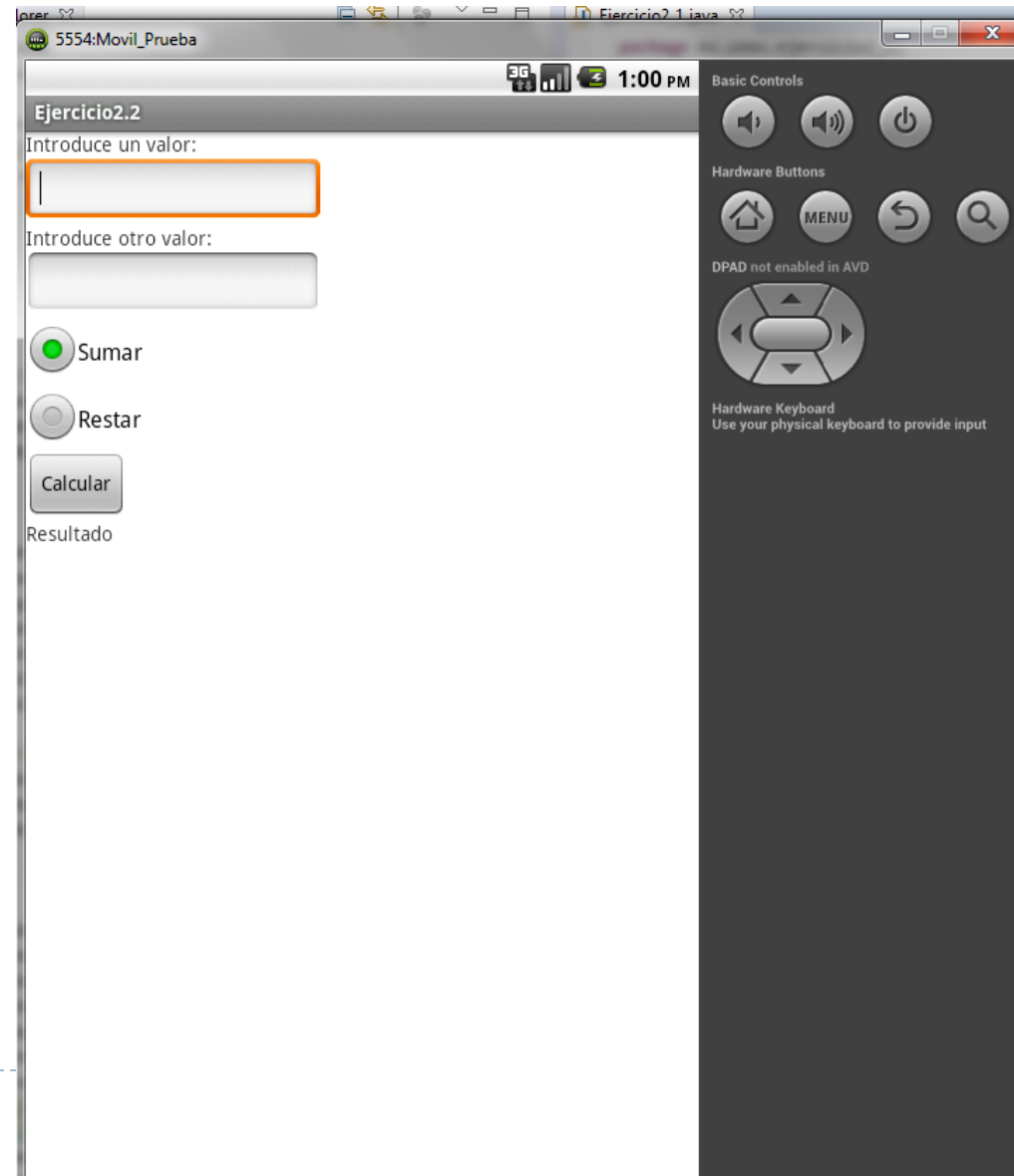
- **Ejercicio Opcional:** Otrá opción es crear el evento desde código a través de un listener, con el código siguiente:

```
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_ejercicio2_1);
    //Asociamos las variables a los recursos
    val1=(EditText)findViewById(R.id.Valor1);
    val2=(EditText)findViewById(R.id.Valor2);
    resultado=(TextView)findViewById(R.id.Resultado);
    boton =(Button)findViewById(R.id.Calcular);
    //Con un listener
    OnClickListener listener = new OnClickListener(){
        @Override
        public void onClick(View arg0) { //Equivalente al método sumar
            // TODO Auto-generated method stub
            String valor1 = val1.getText().toString();
            String valor2 = val2.getText().toString();

            int numero1 = Integer.parseInt(valor1);
            int numero2 = Integer.parseInt(valor2);
            resultado.setText(String.valueOf(numero1+numero2)+" con listener");
        }
    };
    boton.setOnClickListener(listener);
}
```

Ejercicio 2.2 – Calculadora (Radio Button)

Crear una calculadora sencilla que sume o reste dos números al pulsar un botón. Utilizaremos dos `RadioButton` para las operaciones

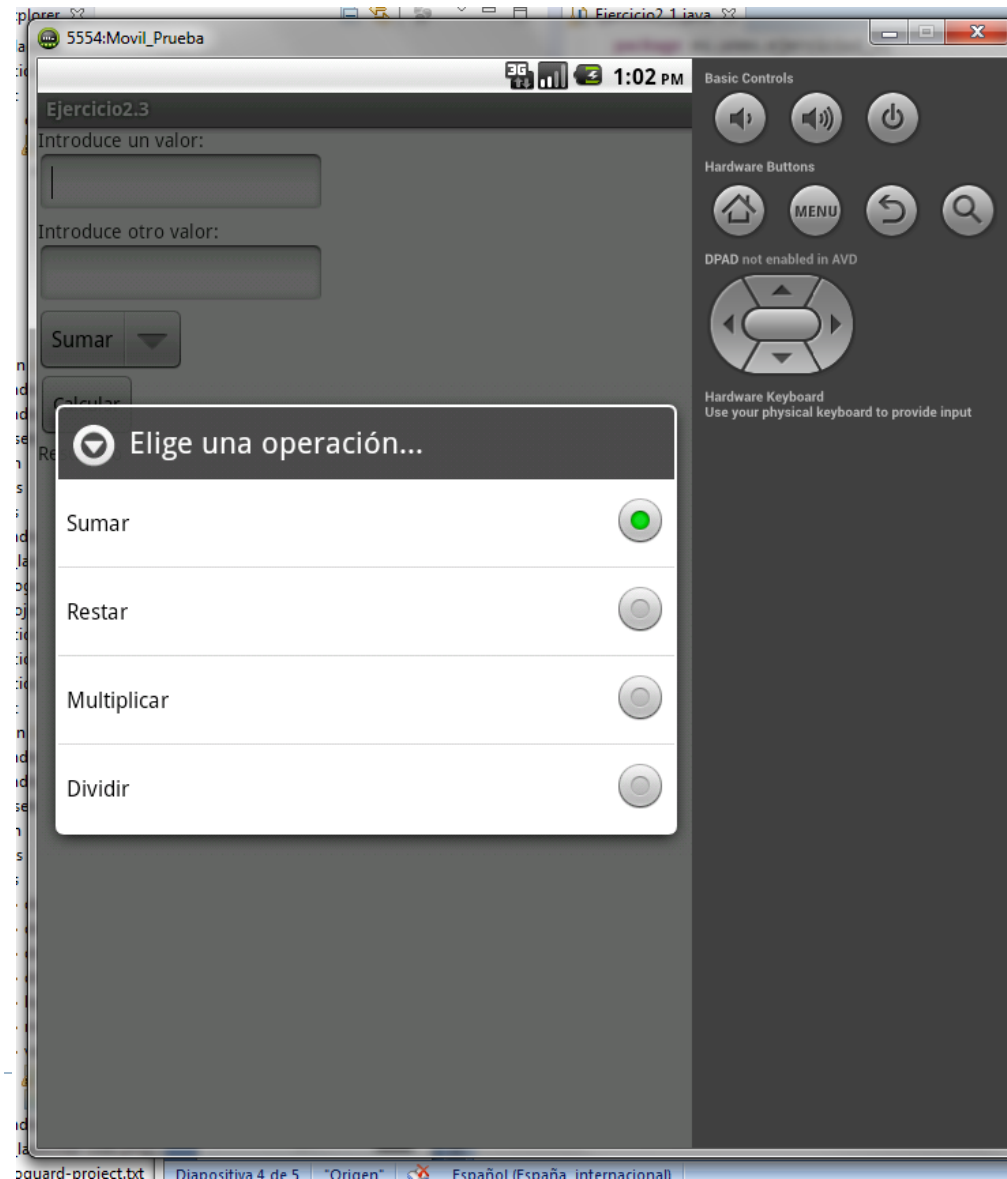


Ejercicio 2.2 – Calculadora (Radio Button)

1. Crear un proyecto nuevo.
2. Crear los widgets en el Layout. (Añadir un RadioGroup no un RadioButton)
3. Nombrarlos adecuadamente (Id)
4. Crear los textos como Strings (Text)
5. Ir al código del activity:
 1. Añadimos las variables necesarias
 2. Asociamos las variables a los recursos
 - ▶ Para RadioButton: `sumar=(RadioButton)findViewById(R.id.Sumar);`
 3. Creamos un método calcular: `public void calcular(View view)`
 - ▶ Para saber si un checkBox esta marcado :
 - ▶ `if (sumar.isChecked()) { //sumar ... }`
 4. Asociamos el método al evento.

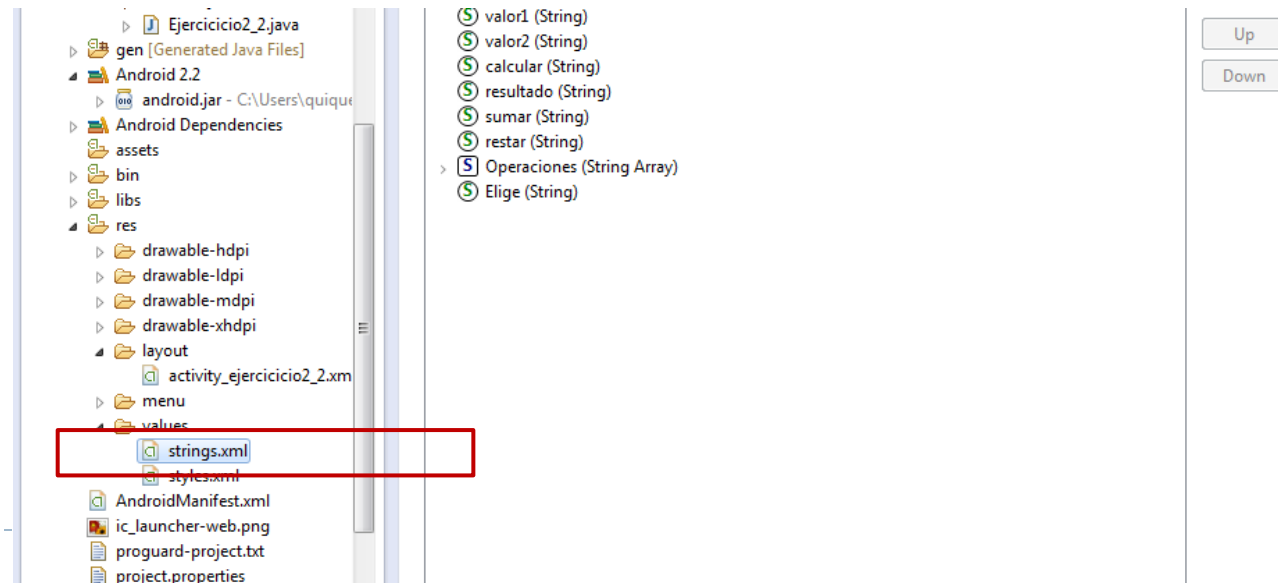
Ejercicio 2.3 – Calculadora (Spinner)

Crear una calculadora sencilla que sume, reste, multiplique o divida dos números



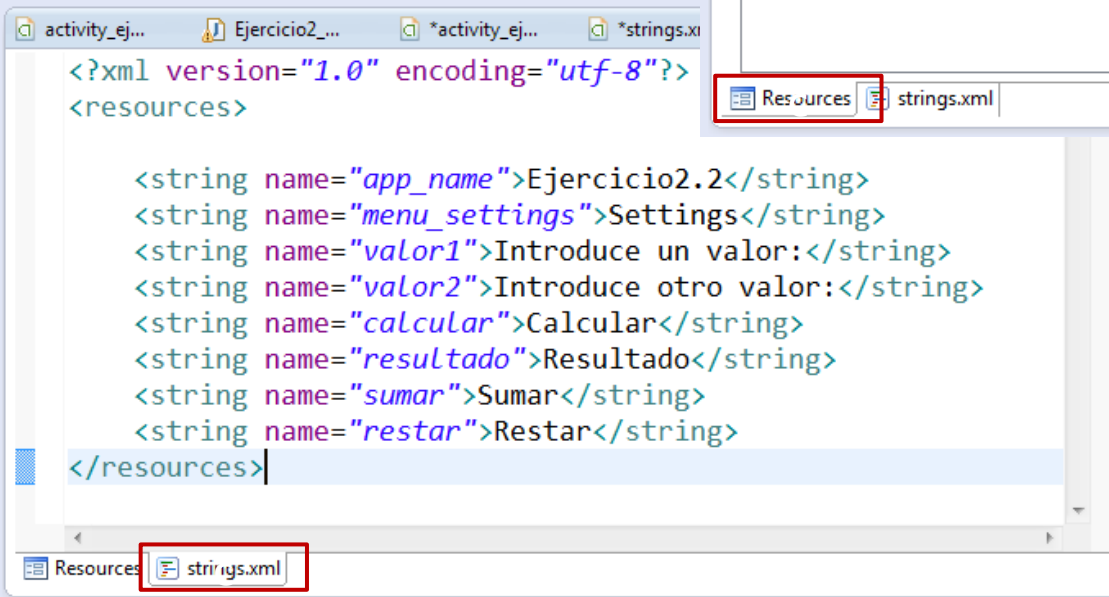
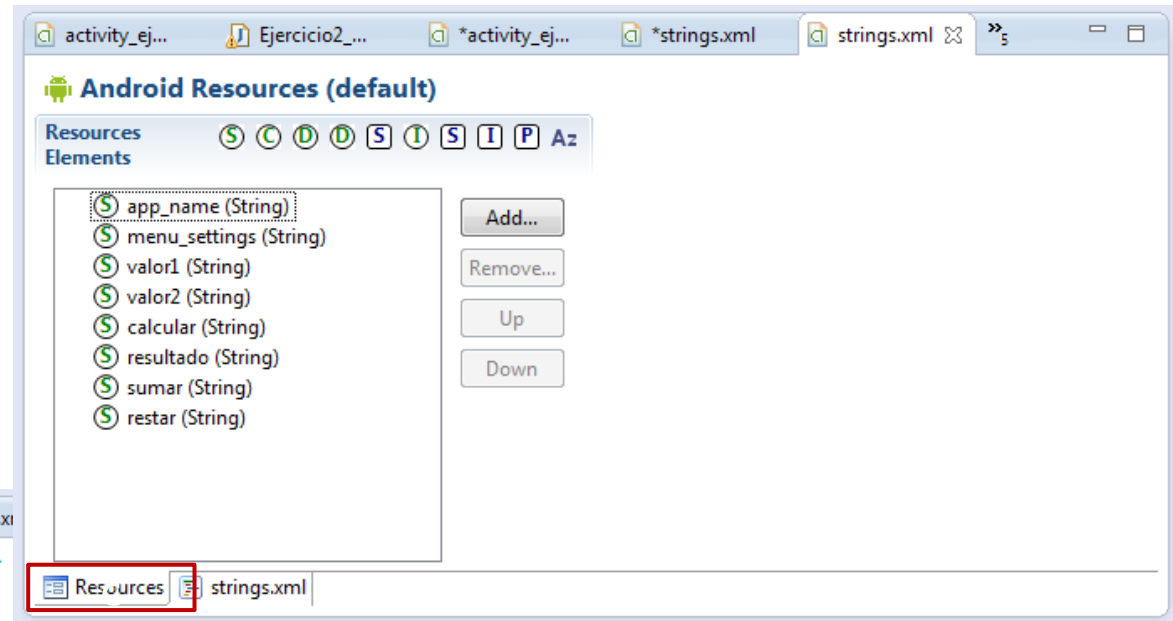
Ejercicio 2.3 – Calculadora (Spinner)

- ▶ En este caso utilizaremos un Spinner para seleccionar las operaciones.
- ▶ En este caso tenemos que asignarle las opciones a través de un String Array.
- ▶ Todos los String que creamos están en la carpeta values, en el archivo string.xml.



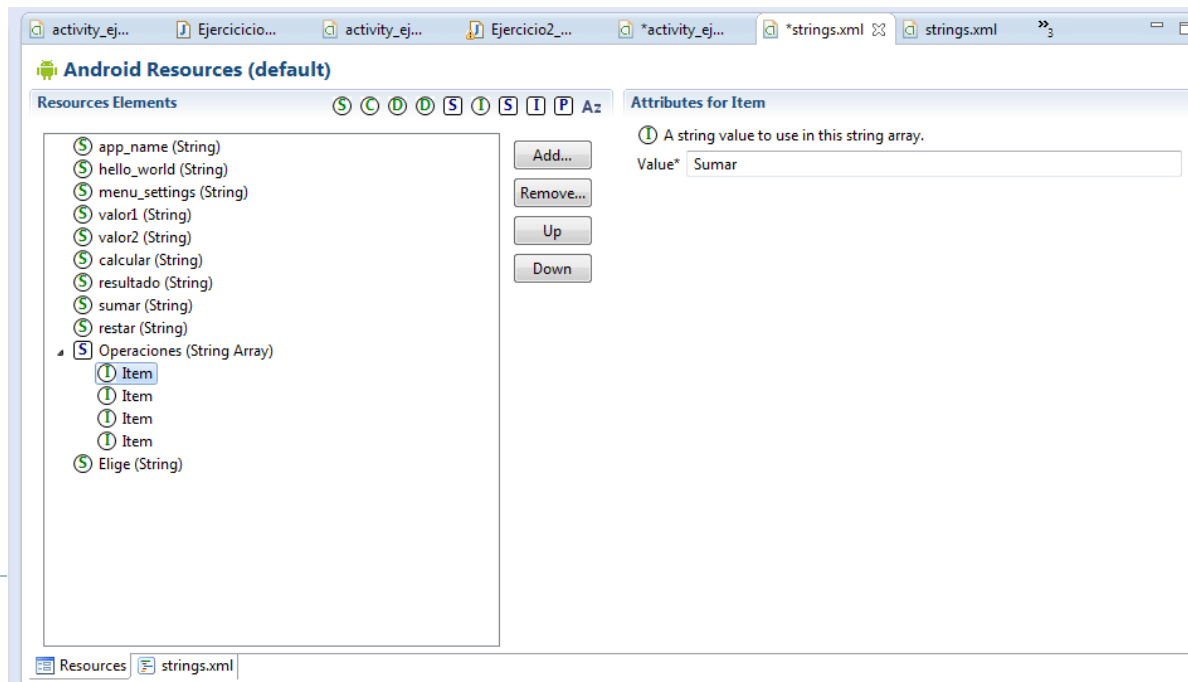
Ejercicio 2.3 – Calculadora (Spinner)

- ▶ string.xml
- ▶ Dos vistas



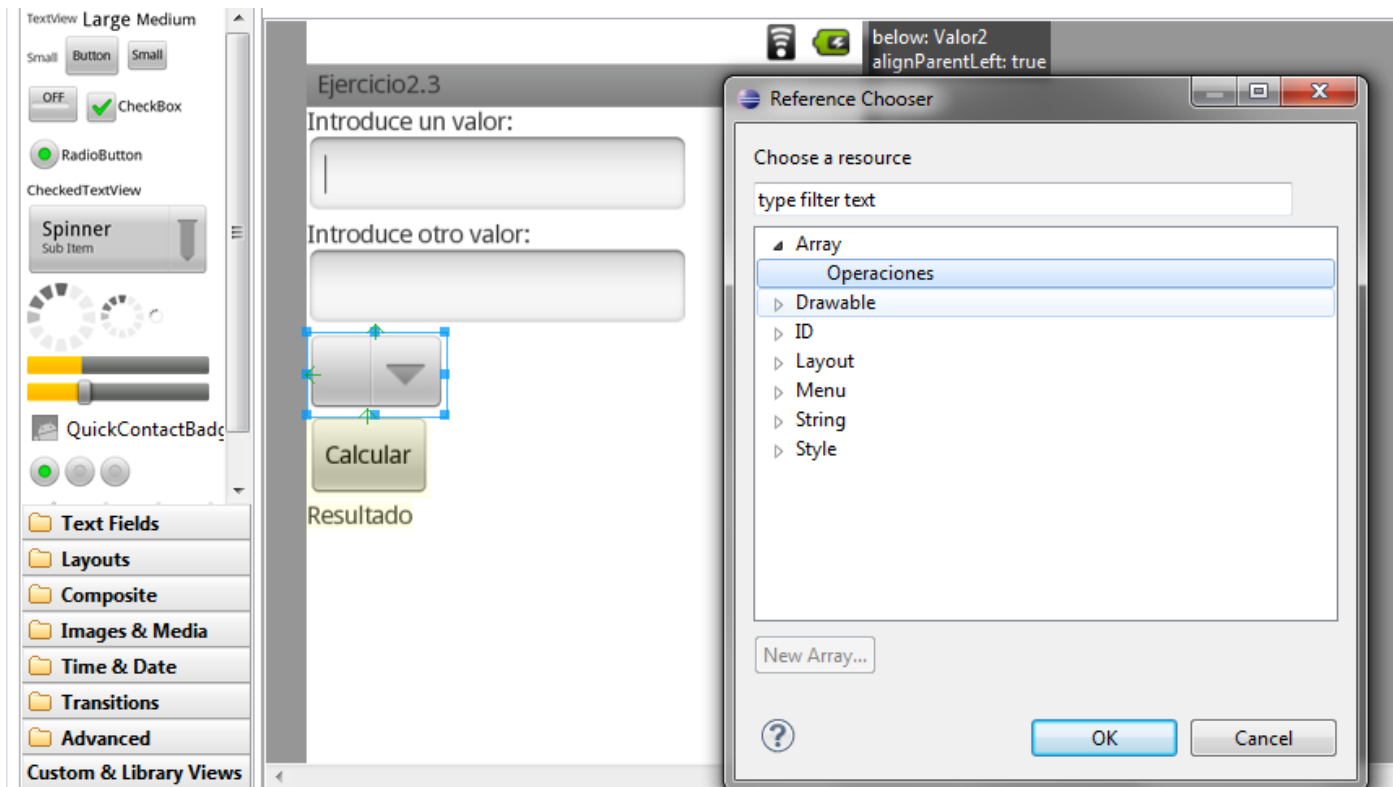
Ejercicio 2.3 – Calculadora (Spinner)

- ▶ Añadimos un `ArrayString` con las opciones que queremos poner en el spinner:
 - ▶ A través del botón `StringArray` añadimos el Array y lo llamamos “Operaciones”
 - ▶ Luego añadimos los “Item” y en value indicamos el nombre: sumar, restar, multiplicar y dividir.



Ejercicio 2.3 – Calculadora (Spinner)

- ▶ Luego volvemos al Layout y sobre el Spinner con el botón derecho pulsamos en “Edit entries”
- ▶ Y seleccionamos el array “Operaciones”.



Ejercicio 2.3 – Calculadora (Spinner)

- ▶ Como en ejercicios anteriores tenemos que hacer los mismos pasos que en el ejercicio 2.1 para su programación:
- ▶ Solo hay que hacer una operación diferente según la opción seleccionada:

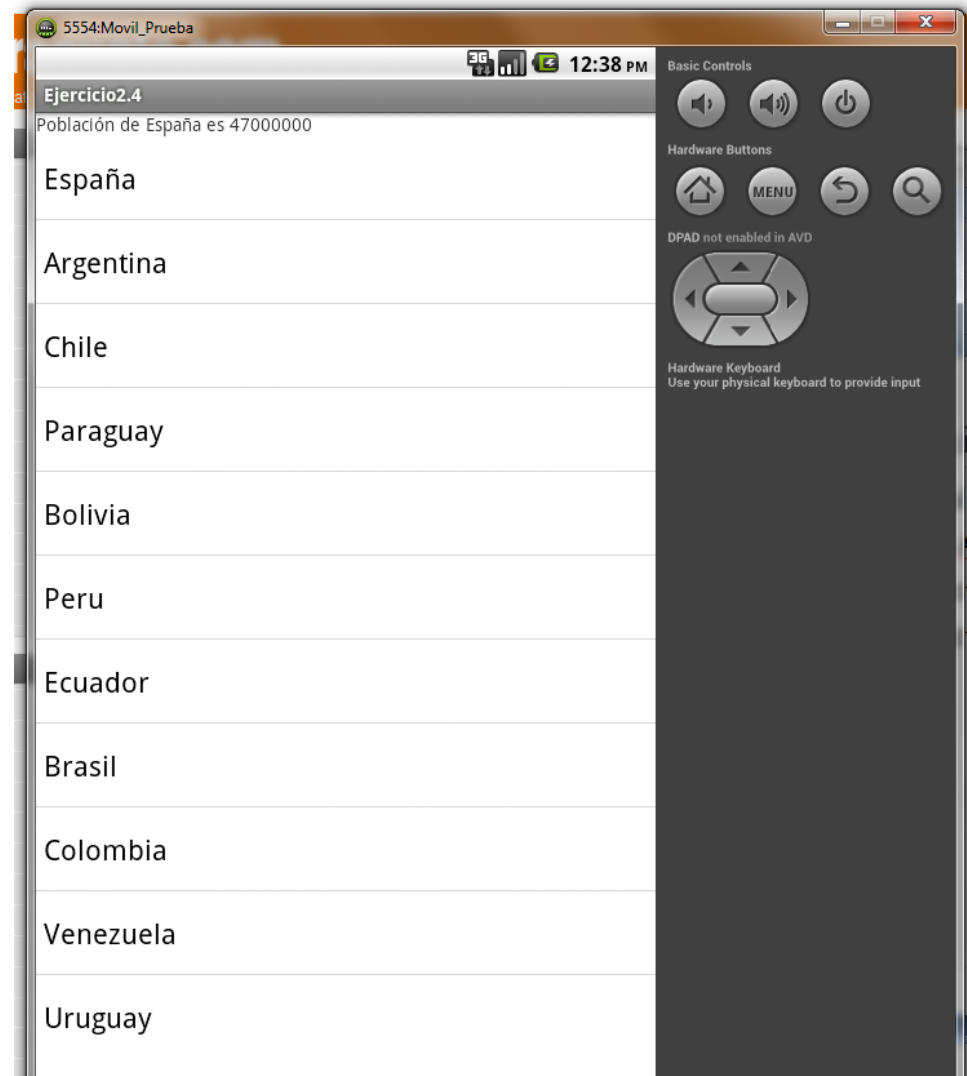
...

```
//Obtenemos la operación seleccionada
String selec=operacion.getSelectedItem().toString();
//Realizamos la operación
if (selec.equals("Sumar")) {
    resultado.setText(String.valueOf(nro1+nro2));
}
```

...

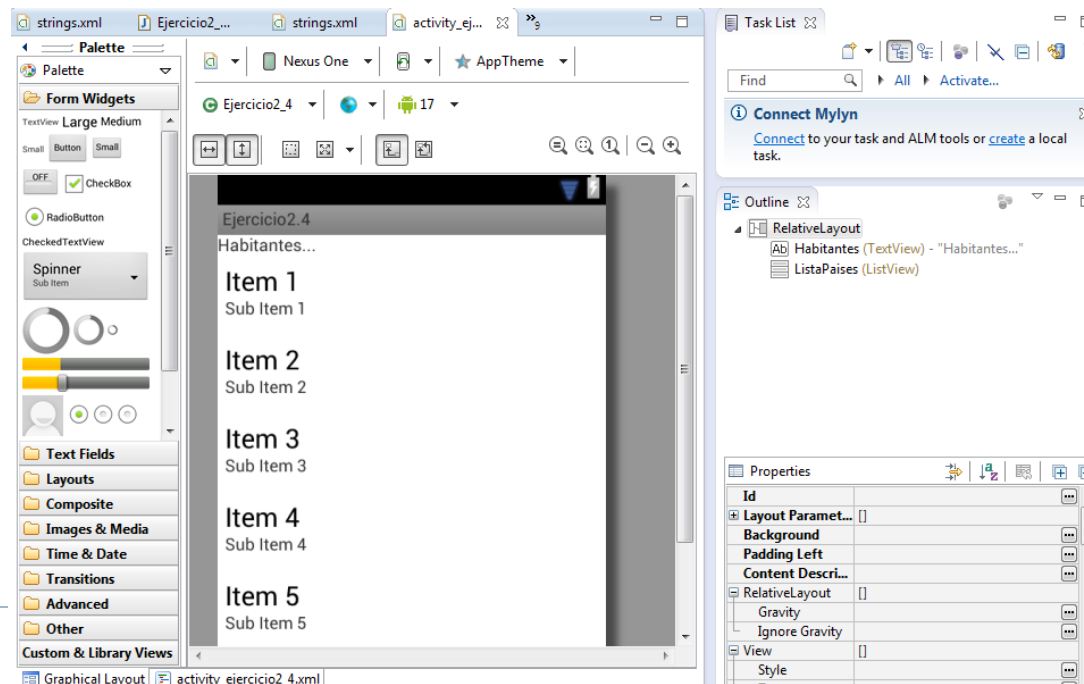
Ejercicio 2.4 – ListView con países ¡Actividad a Entregar!

Pais	Habitantes
España	47000000
Argentina	40000000
Chile	17000000
Paraguay	6500000
Bolivia	10000000
Peru	30000000
Ecuador	14000000
Brasil	183000000
Colombia	44000000
Venezuela	29000000
Uruguay	3500000



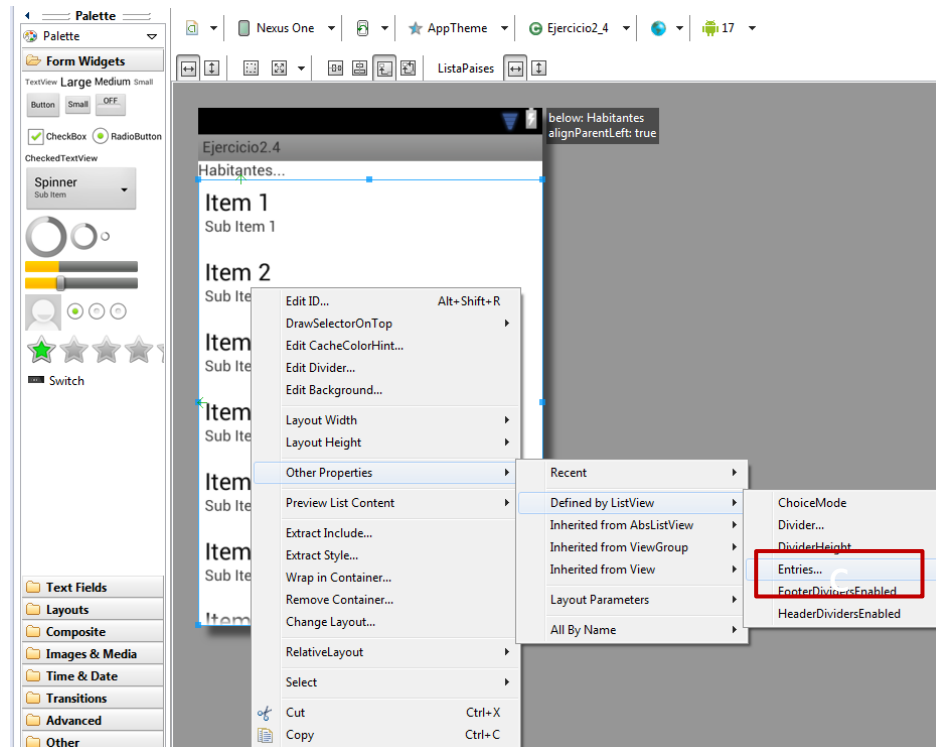
Ejercicio 2.4 – ListView con países

- ▶ La interfaz estará compuesta de un TextView donde mostraremos información y ListView (Composite) donde se seleccionará el país.
- ▶ Al hacer clic en un país deberemos modificar el TextView con la información de cada país.



Ejercicio 2.4 – ListView con países

- ▶ En este caso tendremos que crear 2 array lists, 1 con países y otro con las poblaciones, con los datos que se ven en la diapositiva anterior.
- ▶ Le añadiremos el Array de países a un ListView insertado en el layout.



Ejercicio 2.4 – ListView con países

```
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_ejercicio2_4);
    //Buscamos los elementos del layout
    habitantes = (TextView)findViewById(R.id.Habitantes);
    listaPaises = (ListView)findViewById(R.id.ListaPaises);

    //Creamos un listener de eventos
    OnItemClickListener listener = new OnItemClickListener() {
        @Override
        public void onItemClick(AdapterView<?> arg0, View arg1, int posicion,
            long arg3) {
            // TODO Auto-generated method stub
            //Obtenemos los array de String de paises y de habitantes
            Resources res = getResources();
            String[] arrayHabitantes = res.getStringArray(R.array.NumeroHabitantes);
            String[] arrayPaises = res.getStringArray(R.array.Paises);
            //Mostramos el texto correspondiente a la posición seleccionada
            habitantes.setText("Población de " + arrayPaises[posicion] + " es " +
                arrayHabitantes[posicion]);
        }
    };
    listaPaises.setOnItemClickListener(listener);
}
```

Ejercicio 2.4 – ListView con países

- ▶ En este caso el evento que se genera no es un clic sino un `onItemClickListener`, por lo que deberemos de crearnos el listener para poder capturar la posición donde se ha hecho clic
- ▶ Una vez tenemos la posición, podremos obtener el nombre del país y de los recursos a través de los recursos (`StringArray`) correspondientes que tendremos que almacenar en un `Array` de `Strings`.
- ▶ Es muy importante añadir el listener al objeto, sino no capturamos el evento:

```
listaPaises.setOnItemClickListener(listener);
```

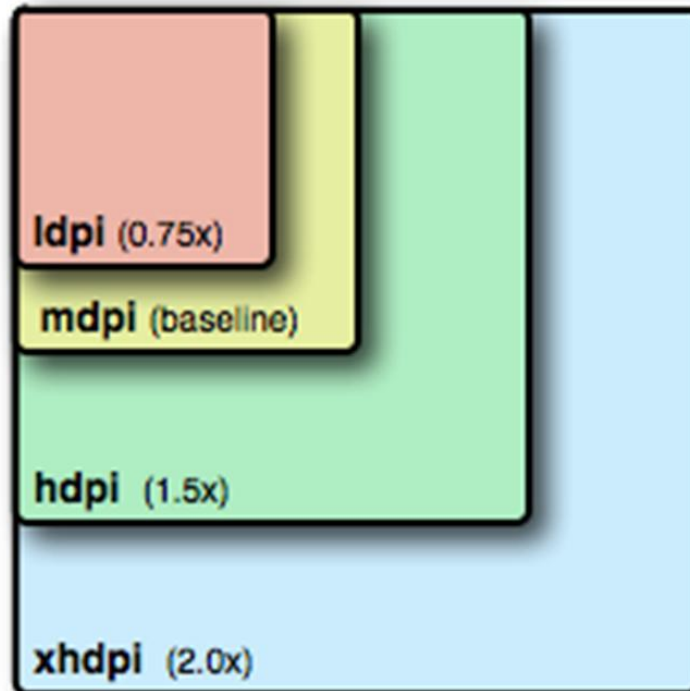
Ejercicio 2.5 – Botón Imagen(ImageButton)

Crear una botón a partir de una imagen que modifique un TextView.



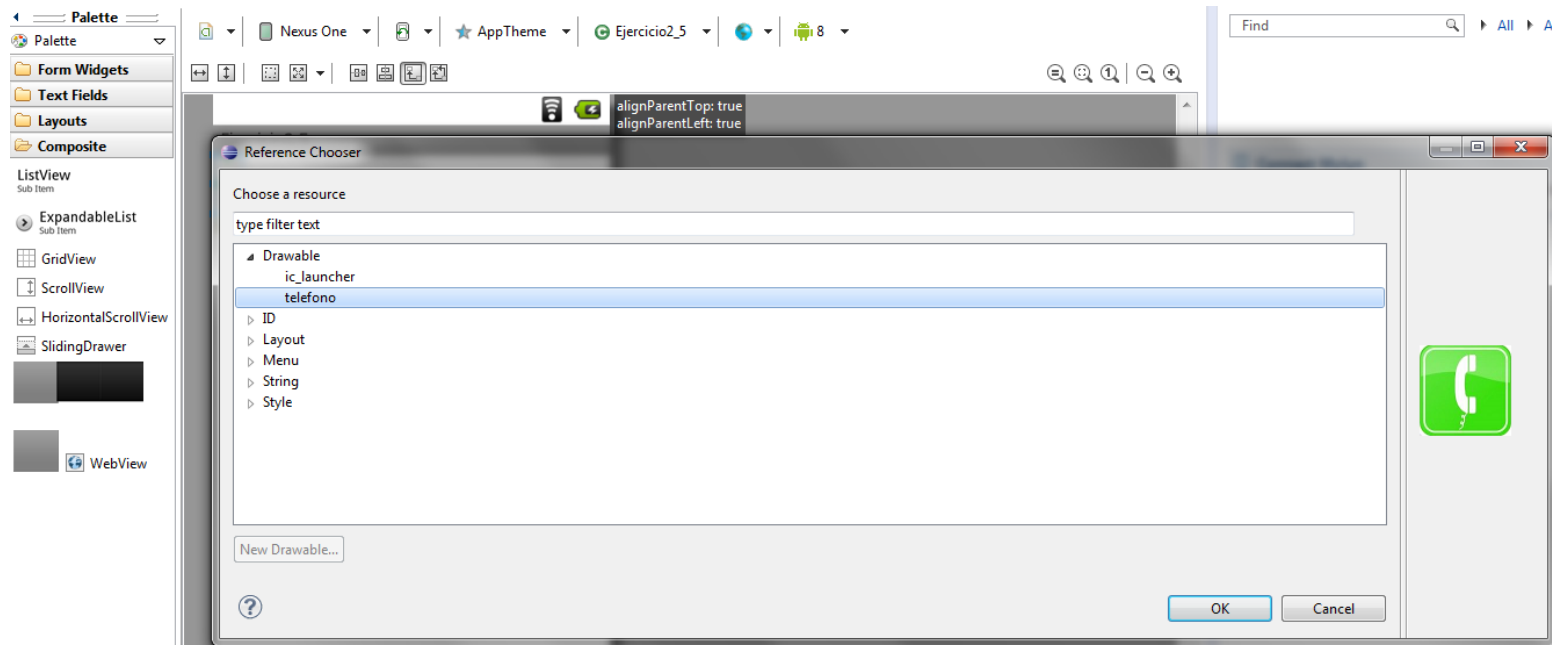
Ejercicio 2.5 – Botón Imagen(ImageButton)

- ▶ En este caso veremos como hacer iconos para un botón.
- ▶ Podemos añadir recursos, en este caso imágenes dentro de la carpeta recursos, normalmente dentro de la carpeta “res” en la subcarpeta “drawable-mdpi”, el resto de carpeta es para diferentes resolución conforme a:



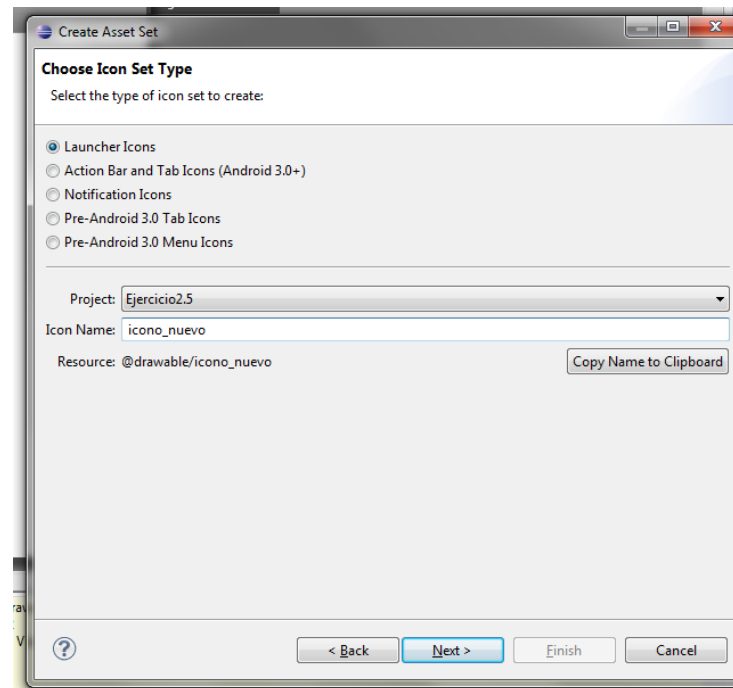
Ejercicio 2.5 – Botón Imagen(ImageButton)

- ▶ Añadiremos un archivo telefono.png a la carpeta drawable-mdpi.
- ▶ En el layout tendremos un ImageButton y un TextView.
- ▶ Con el botón derecho sobre el ImageButton, pulsaremos sobre Edit src y elegiremos el teléfono desde Drawable.
- ▶ Lo siguiente sería añadir el evento al botón para que al hacer clic cambiara el TextView a llamando



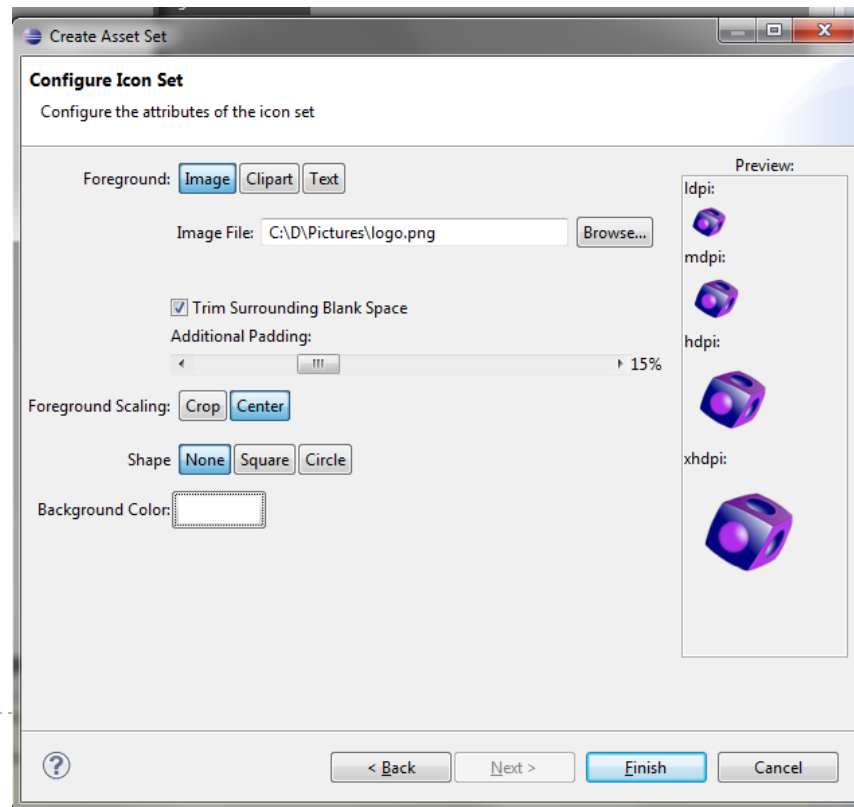
Ejercicio 2.5 – Botón Imagen(ImageButton

- ▶ Otra opción para crear un icono es crear un Android Icon Set.
- ▶ File -> New -> Other
- ▶ Y en la carpeta de Android seleccionamos Android Icon Set.



Ejercicio 2.5 – Botón Imagen(ImageButton)

- ▶ Tenemos tres opciones: Image, Clipart o Texto.
- ▶ Probarlas, por ejemplo creando tres diferentes y podéis cambiar la imagen del botón o crear otros ImageButton.
- ▶ En este caso nos creará una copia en cada carpeta: ldpi, mdpi, hdpi y xhdpi automáticamente



Ejercicio 2.6 – Calculadora (CheckBox)

¡Actividad a Entregar!

Crear una calculadora sencilla que sume y/o reste dos números al pulsar un botón

