HLC

2º DAW

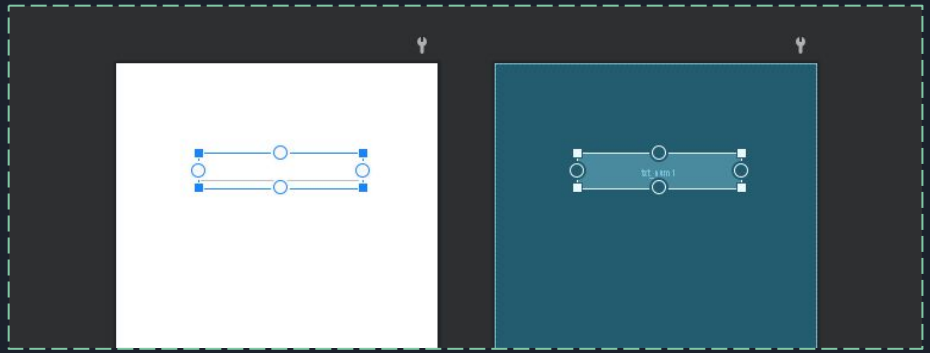
NUESTRA PRIMERA APP. CALCULADORA

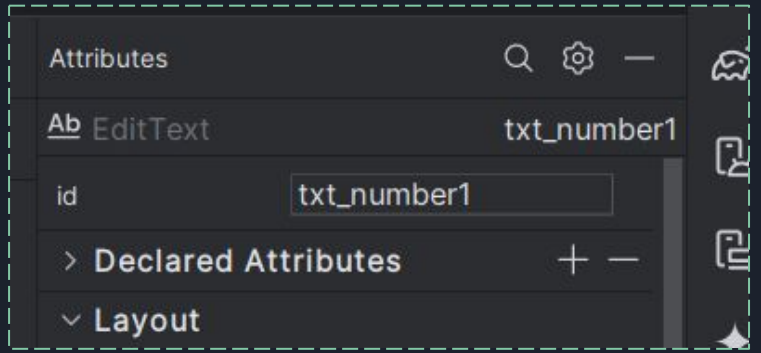
Borraremos o comentaremos todo el código de la clase anterior que añadimos para depurar ciertas variables que nos indicaban si estábamos “aprobados o suspensos”.

Con lo que nuestro MainActivity.java quedará tal y como nos lo generó Android Studio.

Desde la vista de diseño, eliminaremos el holaMundo para dejar nuestra pantalla “limpia”.

Añadiremos un campo de texto numérico desde Palette > Text > Number a nuestra pantalla, y a continuación le indicaremos un id. Práctica que como programadores, debe ser casi obligatoria.

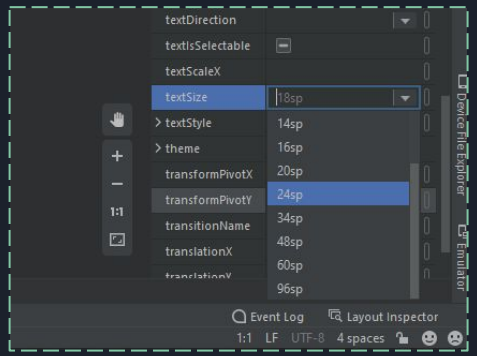


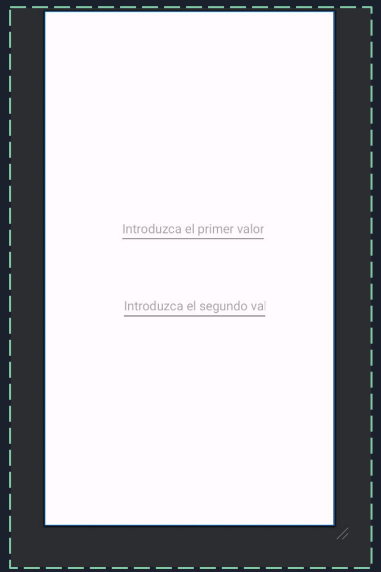


NUESTRA PRIMERA APP. CALCULADORA

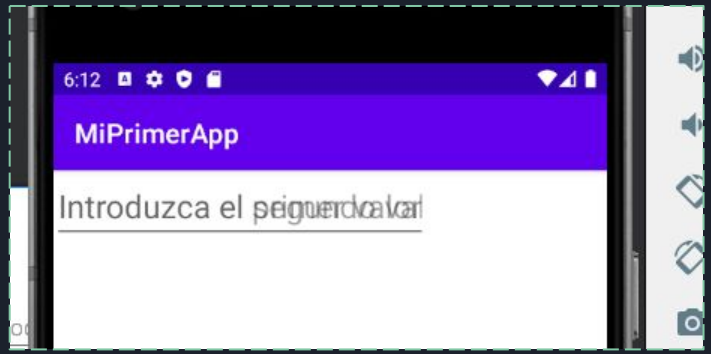
Indicaremos un texto para que aparezca por defecto, antes de añadir ningún valor, en el campo que hemos creado. Se hará desde la propiedad hint, en Common Attributes.

Para que el campo se vea más grande, desplegaremos todos los atributos y modificaremos textSize.



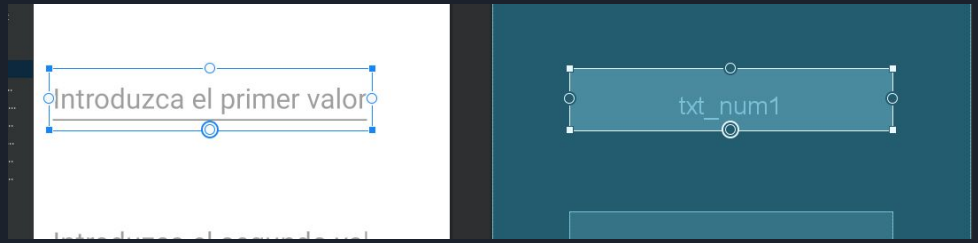


De la misma forma, añadimos el segundo campo de número. Si ejecutamos, se vería algo así:

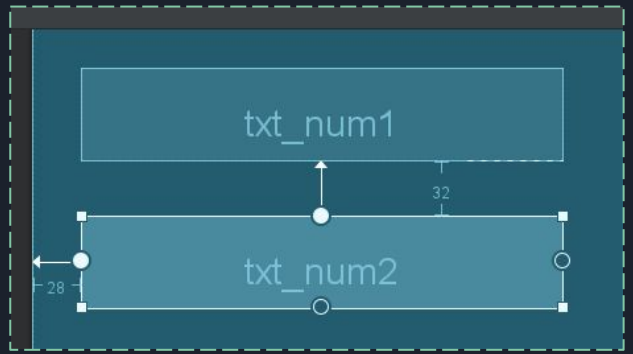


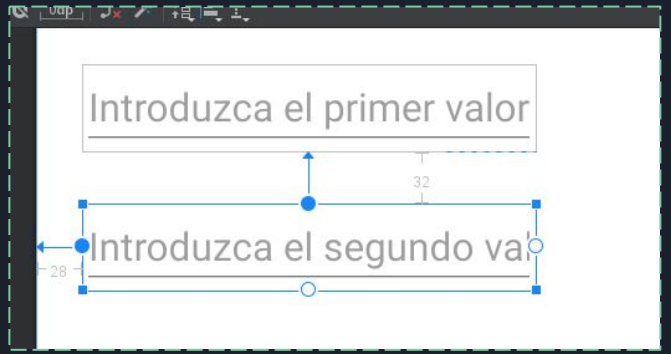
NUESTRA PRIMERA APP. CALCULADORA

Para solucionar dicho inconveniente, está la vista Blueprint.

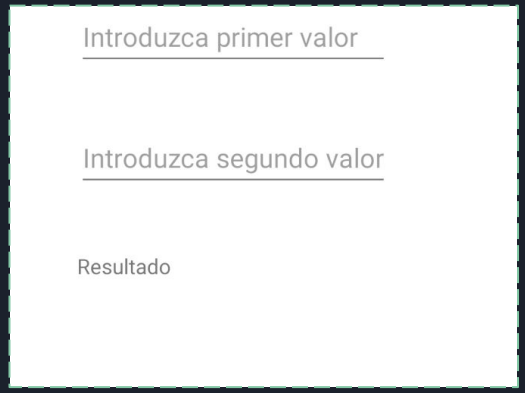


Desde Blueprint, podemos seleccionar el comportamiento de nuestros elementos a la hora de mostrarse en pantalla. Para ello nos ayudamos de las guías circulares.



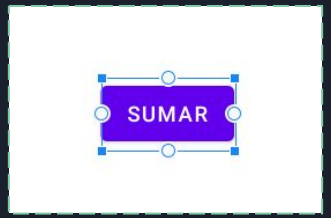


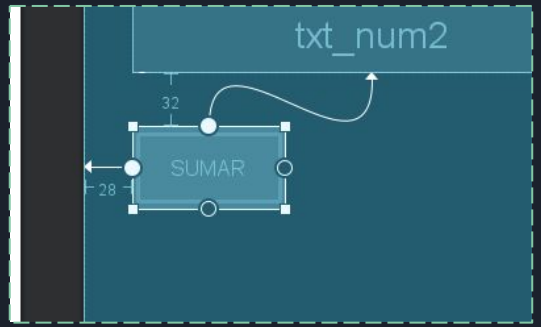
NUESTRA PRIMERA APP. CALCULADORA

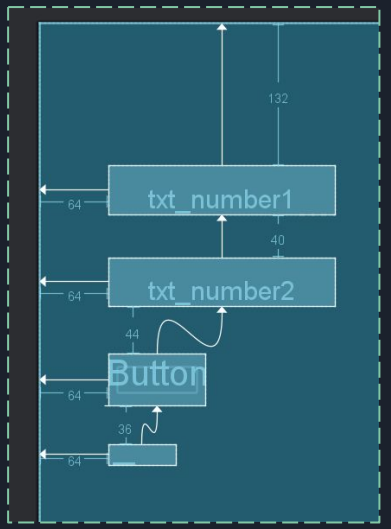


Tenemos que añadir un lugar donde se muestre el resultado de nuestra suma. Para ello arrastraremos un TextView. Modificaremos su id a *txt_resultado* y su text a *Resultado*. También podemos poner su textSize a *24sp*. Conectamos los puntos.

NUESTRA PRIMERA APP. CALCULADORA







Tenemos que añadir un botón para proceder a realizar la suma. Para ello arrastraremos un Button. Modificaremos su atributo text a *Sumar*. También podemos poner su textSize a 24sp. Conectamos los puntos.

NUESTRA PRIMERA APP. CALCULADORA

Pasamos a la parte lógica. Crearemos los elementos que nos ayudan a mostrar o recoger texto, para ello usaremos la clase EditText y TextView. Nosotros tenemos 3 elementos de dicho tipo. Serán atributos.

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

Y establecemos una comunicación entre la parte lógica y la gráfica a través de la clase R en el método onCreate().

```
et1= findViewById(R.id.txt_num1);
et2= findViewById(R.id.txt_num2);
tv1= findViewById(R.id.txt_resultado);
```

NUESTRA PRIMERA APP. CALCULADORA

Nos salimos del método onCreate() y creamos uno nuevo, llamado sumar(). El argumento será un objeto de tipo View.

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

Al igual que en Java, o similar a JavaScript, recogemos los valores numéricos introducidos en forma de cadenas de texto.

```
String valor1 = et1.getText().toString();
String valor2 = et2.getText().toString();
```

Ahora *parseamos* dichos elementos, ya que tendremos que aplicar funciones aritméticas sobre ellos.

```
int num1 = Integer.parseInt(valor1);
int num2 = Integer.parseInt(valor2);
```

Y realizamos la suma:

```
int suma = num1 + num2;
```

NUESTRA PRIMERA APP. CALCULADORA

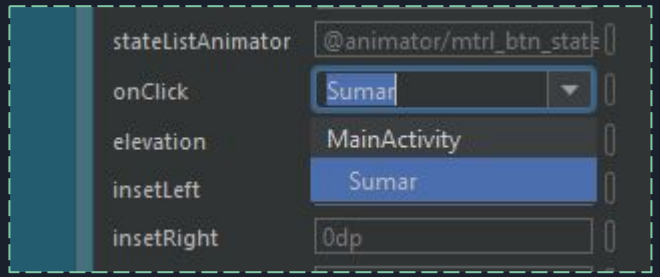
El resultado, al mostrarse por pantalla en nuestro elemento creado anteriormente “Resultado”, debe convertirse a cadena de texto:

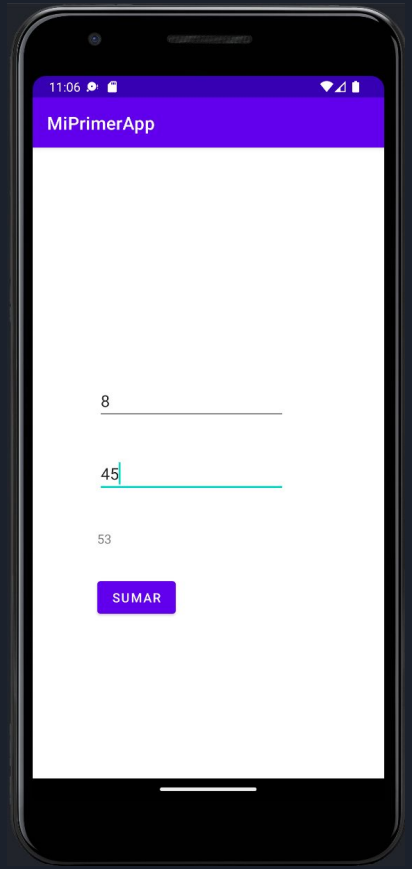
```
String result = String.valueOf(suma);
```

Y, por supuesto, indicarle que debe establecerse en la variable tv1:

```
tv1.setText(result);
```

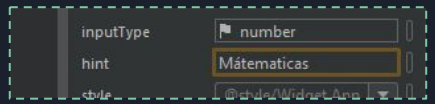
Tras esto, tenemos que indicar a nuestro botón que sea el encargado de realizar la llamada a nuestro método sumar().

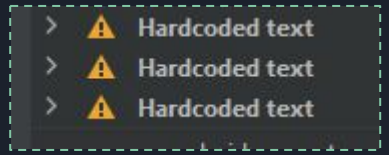




MEJORAS EN NUESTRAS APP | Hardcoded string

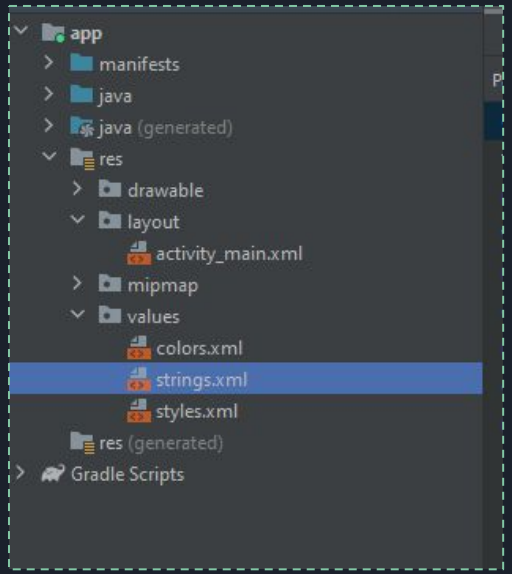
¿Por qué nos salen estos Warning y cómo solucionarlo?



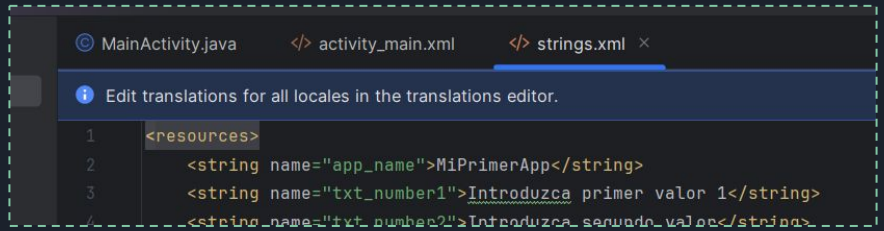


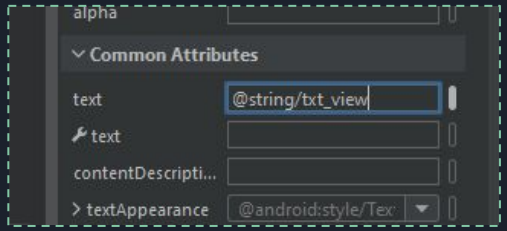
Son advertencias de recurso de cadena. Son una sugerencia que nos hace Android Studio para que no usemos texto dentro de nuestros elementos de forma directa.

Dejaremos de escribir texto dentro del atributo hint y text.



MEJORAS EN NUESTRAS APP | Hardcoded string

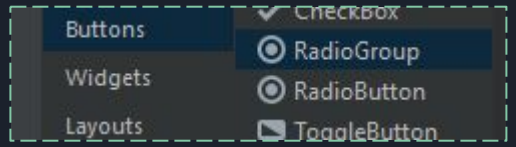


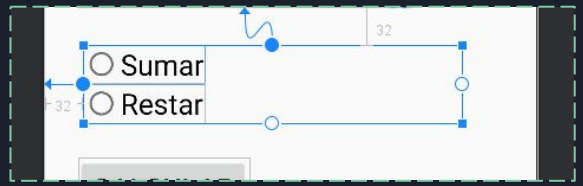


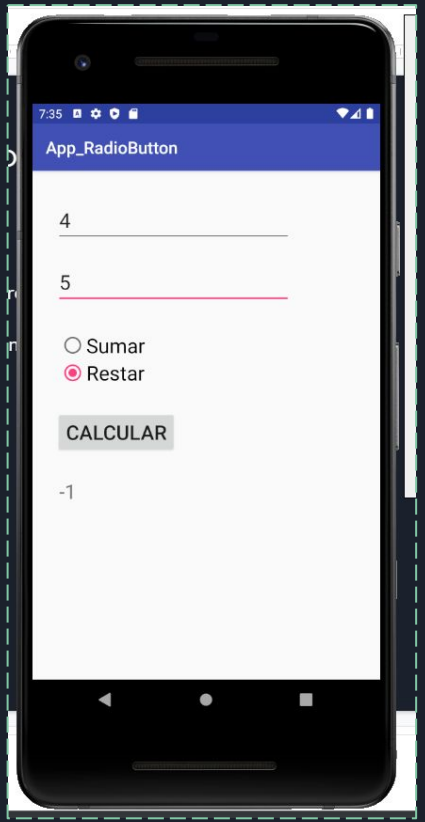
Creamos en dicho XML tantas etiquetas como referencias necesitemos.
Borramos el texto que teníamos en nuestros elementos desde la vista diseño.
Llamamos ahí a nuestro elemento de strings.xml.
Hacemos lo mismo con los hint.
Apreciamos cómo desaparecen los warnings.

Controles RadioGroup y RadioButton

Haciendo uso de todo lo aprendido hasta ahora, crearemos la siguiente App. La idea se aprecia de una forma bastante clara en la imagen.
Aplicaremos las buenas prácticas que conocemos, como referenciar elementos a otros en la blueprint y usar el strings.xml para albergar nuestro texto.
Para los botones de opción usaremos:

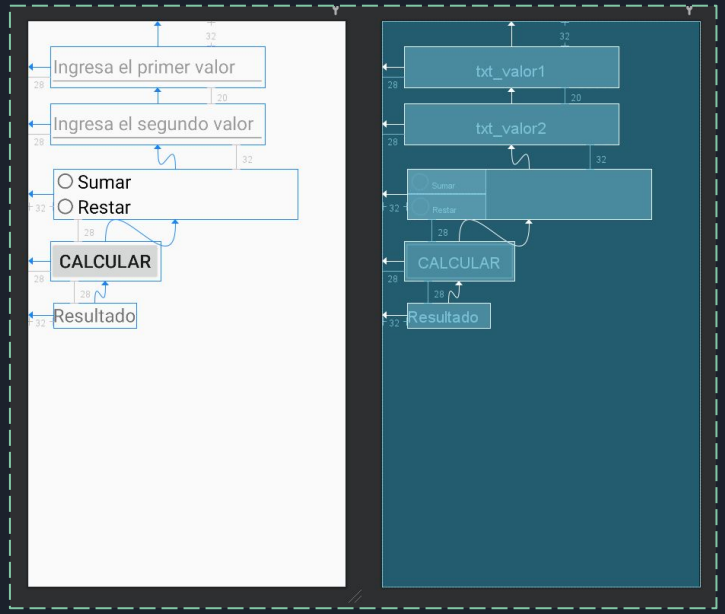






Controles RadioGroup y RadioButton

Así quedarían alguno de nuestros componentes en Android Studio:




```
<resources>
  <string name="app_name">App_RadioButton</string>
  <string name="txt_PrimerValor">Ingresa el primer
valor</string>
  <string name="txt_SegundoValor">Ingresa el segundo
valor</string>
  <string name="rb1">Sumar</string>
  <string name="rb2">Restar</string>
  <string name="Button">Calcular</string>
  <string name="TextView_resultado">Resultado</string>
</resources>
```

Controles RadioGroup y RadioButton

```
public class MainActivity extends AppCompatActivity {

    private EditText et1, et2;
    private TextView tv1;
    private RadioButton rb1, rb2;

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

        et1 = findViewById(R.id.txt_valor1);
        et2 = findViewById(R.id.txt_valor2);
        tv1 = findViewById(R.id.textView);
        rb1 = findViewById(R.id.rb_sumar);
        rb2 = findViewById(R.id.rb_restar);
    }

    //Método para el botón calcular
    public void calcular(View view){
        String valor1_String = et1.getText().toString();
        String valor2_String = et2.getText().toString();

        int valor1_int = Integer.parseInt(valor1_String);
        int valor2_int = Integer.parseInt(valor2_String);

        if(rb1.isChecked()){
            int suma = valor1_int + valor2_int;
            String resultado = String.valueOf(suma);
            tv1.setText(resultado);
        } else if(rb2.isChecked()){
            int resta = valor1_int - valor2_int;
            String resultado = String.valueOf(resta);
            tv1.setText(resultado);
        }
    }
}
```