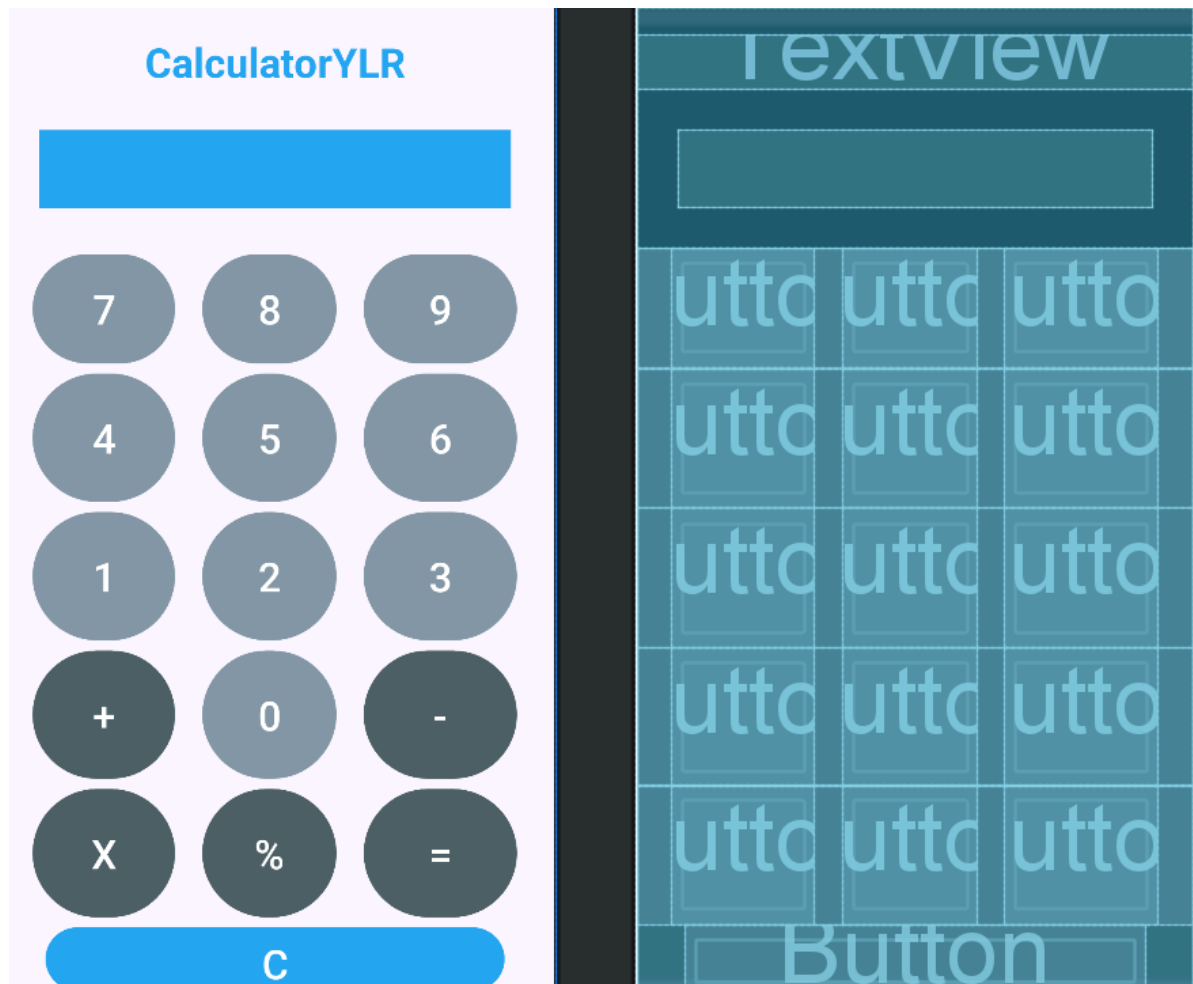


## CalculatorYLR

### 1. Estructura de la interfaz

Para hacer una calculadora básica en mi caso la interfaz de la calculadora, el orden de los botones lo he colocado con un `table_layout` para que los botones se alineen correctamente y estén bien ordenados.



## 2. Código de la calculadora

En primer lugar del código he definido las variables que voy a utilizar y también he declarado los botones y el cuadro de texto de la calculadora

```
//buttons and textbox definitions
TextView etnumbers; 8 usages
Button btn0, btn1, btn2, btn3, btn4, btn5, btn6, btn7, btn8, btn9; 2 usages
Button btnplus, btnminus, btnmultiply, btndivide, btnequals, delete; 2 usages

//vars i am going to use
private double firstNumber = 0; 6 usages
private double secondNumber = 0; 7 usages
private String operator = ""; 2 usages
private boolean isOperatorSelected = false; 3 usages
```

```
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);

    setContentView(R.layout.activity_main);

    etnumbers = findViewById(R.id.numbersbox);
    btn0 = findViewById(R.id.button0);
    btn1 = findViewById(R.id.button1);
    btn2 = findViewById(R.id.button2);
    btn3 = findViewById(R.id.button3);
    btn4 = findViewById(R.id.button4);
    btn5 = findViewById(R.id.button5);
    btn6 = findViewById(R.id.button6);
    btn7 = findViewById(R.id.button7);
    btn8 = findViewById(R.id.button8);
    btn9 = findViewById(R.id.button9);
    btnplus = findViewById(R.id.plus);
    btnminus = findViewById(R.id.minus);
    btnmultiply = findViewById(R.id.multiply);
    btndivide = findViewById(R.id.btndivide);
    btnequals = findViewById(R.id.equals);
    delete = findViewById(R.id.delete);

    ViewCompat.setOnApplyWindowInsetsListener(findViewById(R.id.numbersbox)) {v, insets} -> {
        Insets systemBars = insets.getInsets(WindowInsetsCompat.Type.systemBars());
        v.setPadding(systemBars.left, systemBars.top, systemBars.right, systemBars.bottom);
        return insets;
    });

    loadComponent();
}
```

A continuación, tengo dos métodos que sirven para cargar los botones de las operaciones y los botones de los números y también les añado la funcionalidad que estos van a tener.

```
//method to load the operations buttons and numbers buttons functionality
protected void loadComponent() { 1 usage
    btnplus.setOnClickListener(v -> setOperator("+"));
    btnminus.setOnClickListener(v -> setOperator("-"));
    btnmultiply.setOnClickListener(v -> setOperator("*"));
    btndivide.setOnClickListener(v -> setOperator("/"));

    delete.setOnClickListener(v -> {
        setOperator("");
        etnumbers.setText("");
        firstNumber = 0;
        secondNumber = 0;
    });

    btnequals.setOnClickListener(view -> calculateResult());

    loadNumbers();
}
```

```
protected void loadNumbers() { 1 usage
    btn0.setOnClickListener(view -> addNumber("0"));
    btn1.setOnClickListener(view -> addNumber("1"));
    btn2.setOnClickListener(view -> addNumber("2"));
    btn3.setOnClickListener(view -> addNumber("3"));
    btn4.setOnClickListener(view -> addNumber("4"));
    btn5.setOnClickListener(view -> addNumber("5"));
    btn6.setOnClickListener(view -> addNumber("6"));
    btn7.setOnClickListener(view -> addNumber("7"));
    btn8.setOnClickListener(view -> addNumber("8"));
    btn9.setOnClickListener(view -> addNumber("9"));
}
```

Después de cargar la funcionalidad de los botones e creado unos métodos, cuya funcionalidad es hacer que los botones pulsados se escriban en el cuadro de texto para poder visualizarlos.

```
//method to add operators to the TextView
protected void setOperator(String operator) { 5 usages
    firstNumber = Double.parseDouble(etnumbers.getText().toString());
    this.operator = operator;
    isOperatorSelected = true;
}

//method to add numbers to the TextView
private void addNumber(String number) { 10 usages
    if (isOperatorSelected) {
        etnumbers.setText("");
        isOperatorSelected = false;
    }
    etnumbers.append(number);
}
```

Por último un método que realiza los cálculos introducidos en la calculadora, mostrando en el cuadro de texto el resultado.

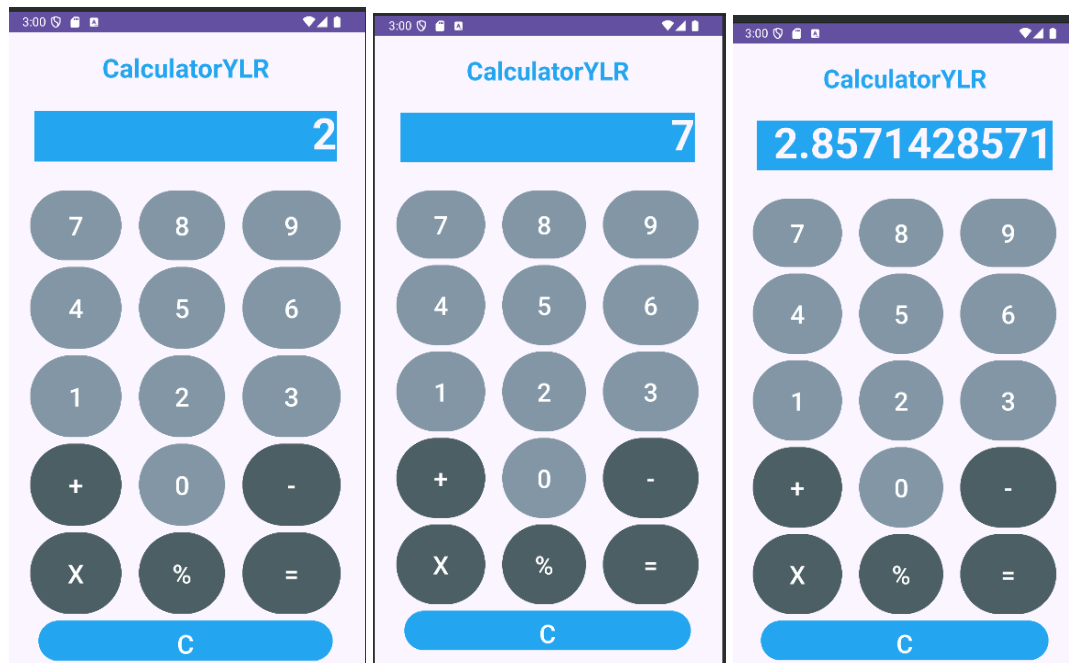
```
//method to do the calculations and set the result on the screen
private void calculateResult() { 1 usage
    secondNumber = Double.parseDouble(etnumbers.getText().toString());

    double result = 0;
    switch (operator) {
        case "+":
            result = firstNumber + secondNumber;
            break;
        case "-":
            result = firstNumber - secondNumber;
            break;
        case "*":
            result = firstNumber * secondNumber;
            break;
        case "/":
            if (secondNumber != 0) {
                result = firstNumber / secondNumber;
            } else {
                etnumbers.setText("Error");
                return;
            }
            break;
    }

    etnumbers.setText(String.valueOf(result));
}
```

### 3. Cómo usar la aplicación

Para usar esta aplicación debes introducir primero un número, una operación y por último el segundo número. Si queremos comprobar el resultado de esa operación deberemos pulsar el botón = el cual nos mostrará el resultado, si queremos borrar los datos introducidos debemos pulsar el botón C. En este ejemplo hemos probado  $20 / 7 = 2.857$ .



[Enlace de github](#)