

# Calculadora

## Interfaz - Código



## Índice

|                                   |   |
|-----------------------------------|---|
| 1) Interfaz gráfica.....          | 3 |
| 1 Entorno Visual.....             | 3 |
| 2 Layouts y Vistas.....           | 3 |
| 3 Interfaz Gráfica - Código.....  | 4 |
| Botones:.....                     | 4 |
| TextView:.....                    | 4 |
| 2) Código.....                    | 5 |
| 1 Operaciones.....                | 6 |
| 2 Métodos con “Operaciones” ..... | 7 |

# 1) Interfaz gráfica

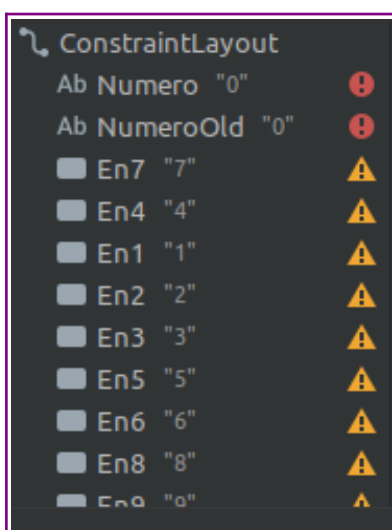
## 1 Entorno Visual

Se presenta la interfaz gráfica / visual, consta:

- Botones con las distintas funciones y números propios de una calculadora.
- Los 2 layouts superiores muestran 2 tipos de información, número introducido, número anterior y resultados.
- Se han elegido distintas variaciones de colores.



## 2 Layouts y Vistas



Cuenta con distintos objetos tipo Button y TextView

Cada tecla se considera un objeto tipo Button

Resultados TextView

### 3 Interfaz Gráfica - Código

#### Botones:

Todos los botones poseen la misma configuración.

Altura y anchura de 75 dp

Sus constraint van encadenadas al objetivo superior o inferior

```
<Button
    android:id="@+id/En2"
    android:layout_width="75dp"
    android:layout_height="75dp"
    android:layout_marginStart="20dp"
    android:layout_marginTop="20dp"
    android:background="#9200BCD4"
    android:text="2"
    app:layout_constraintStart_toEndOf="@+id/En1"
    app:layout_constraintTop_toBottomOf="@+id/En5" />
```

#### TextView:

Características:

Anchura de 408dp.

Altura de 118dp.

Family: serif.

Color fuente: blanca.

Se ha centrado el texto en el centro de la vista, utilizando **android:gravity**

```
<TextView
    android:id="@+id/Numero"
    android:layout_width="408dp"
    android:layout_height="118dp"
    android:layout_marginTop="104dp"
    android:background="#8BC34A"
    android:fontFamily="serif"
    android:gravity="center"
    android:text="0"
    android:textAlignment="center"
    android:textColor="@color/white"
    android:textSize="20sp"
    app:layout_constraintTop_toTopOf="parent"
    tools:ignore="MissingConstraints"
    tools:layout_editor_absoluteX="1dp" />
```

## 2) Código

Tras crear las variables que almacenan los id de los botones y los textos visuales

Ejemplo:

n0 almacena el id del botón 0

oIgual almacena el botón igual

oCE almacena el botón CE

Se implementa la interfaz **OnClickListener**.

La cual obliga a implementar el método **onClick**

Cada vez que pulses un botón accede al método **onClick**

```
@Override
public void onClick(View view) {
    switch(view.getId()) {
        case R.id.En0:
            aux+=0;
            limpiar=false;
            mostrar();
            break;
        case R.id.En1:
            aux+=1;
            limpiar=false;
            mostrar();
            break;
        case R.id.En2:
            aux+=2;
            limpiar=false;
            mostrar();
            break;
        case R.id.En3:
            aux+=3;
            limpiar=false;
            mostrar();
            break;
        case R.id.En4:
```

```
//Llamada al método
n0.setOnClickListener(this);
n1.setOnClickListener(this);
n2.setOnClickListener(this);
n3.setOnClickListener(this);
n4.setOnClickListener(this);
n5.setOnClickListener(this);
n6.setOnClickListener(this);
n7.setOnClickListener(this);
n8.setOnClickListener(this);
n9.setOnClickListener(this);
nPunto.setOnClickListener(this);
oMulti.setOnClickListener(this);
oDivi.setOnClickListener(this);
oSuma.setOnClickListener(this);
oResta.setOnClickListener(this);
oIgual.setOnClickListener(this);
oCE.setOnClickListener(this);
```

### (Método onClick)

Tras pulsar un botón accede a su posición en el switch

Los botones (números) añaden números a las variables utilizadas para hacer operaciones

### (Método mostrar())

Muestra en pantalla los números introducidos

### (suma(), resta(), multi(), etc)

Métodos encargados de realizar operaciones

# 1 Operaciones

Se utiliza una variable final **OPERACION\_VALOR** encargada de asignar una operación:

1. Al utilizar **suma()** **OPERACION\_VALOR** pasa a 1 y accede a su bloque en el **switch**
2. En el caso (**OPERACION\_VALOR=2**) realiza una resta, así sucesivamente.

```
public void igual () {
    double resultado =0;
    switch (OPERACION_VALOR) {
        case 1:
            resultado = Double.parseDouble(n) + Double.parseDouble(aux);
            Numero.setText(String.valueOf(resultado));
            NumeroOld.setText("0");
            aux=String.valueOf(resultado);
            break;
        case 2:
            resultado = Double.parseDouble(n) - Double.parseDouble(aux);
            Numero.setText(String.valueOf(resultado));
            NumeroOld.setText("0");
            aux=String.valueOf(resultado);
            break;
        case 3:
            resultado = Double.parseDouble(n) * Double.parseDouble(aux);
            Numero.setText(String.valueOf(resultado));
            NumeroOld.setText("0");
            aux=String.valueOf(resultado);
            break;
        case 4:
            resultado = Double.parseDouble(n) / Double.parseDouble(aux);
            if (Double.isInfinite(resultado)) {
                Numero.setText("No se puede dividir por 0");
            }
    }
}
```

- **División \***

En el caso de la división se añade una funcionalidad extra, al dividir un valor **Double** entre 0 Este devuelve "Infinity", utilizamos un if con la condición "**Double.isInfinite**" si el resultado es infinity, accederá a ese bloque y mostrará en pantalla que no es posible dividir.

- **aux**

La variable **aux** se encarga de almacenar el número introducido y el resultado de una operación.

- **Resultado**

Variable Double encargada de realizar operaciones (Es imposible realizar operaciones con variables String)

## 2 Métodos con “Operaciones”

Los métodos con operaciones en realidad no realizan operaciones, indican **cual** es la operación

### 1. Variable n

Variable utilizada para almacenar el valor introducido y realizar operaciones con ella

### 2. OPERACION\_VALOR

Se asigna cual es la operación a realizar

1. Suma
2. Resta
3. divide
4. Multiplicación

### 3. Aux (Usos)

Se muestra en el ViewText superior el contenido anteriormente añadido

Se realizan operaciones entre

**aux Y n**

```
public void suma() {  
    n = aux;  
    Numero01d.setText(aux);  
    Numero.setText("+");  
    aux="0";  
    OPERACION_VALOR=1;  
}  
  
public void divide() {  
    n = aux;  
    Numero01d.setText(aux);  
    Numero.setText("/");  
    aux="0";  
    OPERACION_VALOR=4;  
}  
  
public void resta() {  
    n = aux;  
    Numero01d.setText(aux);  
    Numero.setText("-");  
    aux="0";  
    OPERACION_VALOR=2;  
}
```