

Calculadora

Funcionamiento

Lo primero de todo. Importamos todas las clases que vayamos a utilizar:

```
//Estas son todas las clases que utilizaré para el funcionamiento
import androidx.appcompat.app.AppCompatActivity;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.TextView;
import android.widget.Toast;

import com.example.moviles_22_23.R;
```

En mi clase principal yo elijo importar la vista OnClickListener.

Con eso tengo 1 sólo listener para todos los botones.

La alternativa es que para cada botón hago un clicklistener y crear 30 líneas de código para cada botón.

```
public class Calculadora extends AppCompatActivity implements View.OnClickListener {
```

Antes del onCreate, creo todas las variables que vaya a usar en mi código. Así no necesito una variable diferente para cada método, las puedo usar directamente de arriba.

```
TextView resultado;
Button borrar, sumar, restar, dividir, num0, num1, num2, num3, num4, num5, num6, num7, num8, num9, igual, multi;
float numero1= 0.0f;
float numero2= 0.0f;
float valor= 0.0f;
String op="";
```

A continuación, digo a mi programa que las variables que voy a usar en el código son las mismas que las del layout.

```
//aqui asigno los botones del layout a los del código y con el this hago que un solo listener ejecute todas las ordenes
//dependiendo de que botón he pulsado
resultado=findViewById(R.id.resultado);
multi=findViewById(R.id.multi);
multi.setOnClickListener(this);
```

Eso se hace con el findViewById(R.id."nombre de la variable del layout")

El `setOnClickListener` sirve para que interprete que cada botón tiene una orden diferente. Eso lo explicaré más adelante.

Cada botón tiene su id del layout y su respectivo `setOnClickListener`

```
borrar=findViewById(R.id.borrar);
borrar.setOnClickListener(this);
sumar=findViewById(R.id.sumar);
sumar.setOnClickListener(this);
restar=findViewById(R.id.restar);
restar.setOnClickListener(this);
dividir=findViewById(R.id.dividir);
dividir.setOnClickListener(this);
num0=findViewById(R.id.num0);
num0.setOnClickListener(this);
num1=findViewById(R.id.num1);
num1.setOnClickListener(this);
num2=findViewById(R.id.num2);
num2.setOnClickListener(this);
num3=findViewById(R.id.num3);
num3.setOnClickListener(this);
num4=findViewById(R.id.num4);
num4.setOnClickListener(this);
num5=findViewById(R.id.num5);
num5.setOnClickListener(this);
num6=findViewById(R.id.num6);
num6.setOnClickListener(this);
num7=findViewById(R.id.num7);
num7.setOnClickListener(this);
num8=findViewById(R.id.num8);
num8.setOnClickListener(this);
num9=findViewById(R.id.num9);
num9.setOnClickListener(this);
igual=findViewById(R.id.igual);
igual.setOnClickListener(this);
```

Nos adentramos en el corazón del código y la magia del `this`.

```
public void principal(int a){
    switch(a){
```

Con un método que recibe un número y no devuelve nada, creo un switch.

El switch recibe el número de arriba.

Para explicarlo de la forma más simple posible:

Cada botón tiene un código interno “12345”, ese código lo llamamos `id(“botón1”)`.

Por así decirlo botón1==12345

A partir de aquí.

Como implementé el método setOnClick a mi onCreate. El programa me obliga a tener un método OnClick

Para no hacerlo muy largo, creo un método void interno.

Ese método es el principal que recibe un número("botón1==12345") y con el switch digo:

Si he pulsado el "12345"(que es a su vez el botón 1) ejecuta esto:

Con esto explicado vamos con los casos de cada botón:

```
case R.id.num0:  
    valor = Float.parseFloat(resultado.getText().toString());  
    if(valor== 0.0f){  
        resultado.setText("0");  
    }else{  
        resultado.setText(resultado.getText()+"0");  
    }  
    break;
```

Para cada número es igual.

Valor es un float que acumula números.

A valor le asigno lo que contiene el botón que acabo de pulsar.

Para ello paso lo que contiene la caja de texto(un text view) a String(cojo el resultado con getText y lo paso a String con .toString) y luego lo paso a un número.

Si el valor numérico es 0 se deja en 0, y muestro 0.

Si no es 0, muestro lo que contenía antes(resultado.getText()) y concateno en este caso el 0 porque es el caso R.id.numero0

Vemos que el código es igual para todos los números.

```

case R.id.num1:

    valor = Float.parseFloat(resultado.getText().toString());
    if(valor== 0.0f){
        resultado.setText("1");
    }else{
        resultado.setText(resultado.getText()+"1");
    }
    break;

```

Si es 0 muestro el botón que acabo de pulsar y si es distinto a 0 concateno el contenido + el valor del botón.

```

case R.id.sumar:
    numero1=Float.parseFloat(resultado.getText().toString());
    resultado.setText("0");
    op="+";
    break;

```

Para la suma, cojo todo el contenido acumulado en la caja resultado y lo paso a número.

Pongo la caja a 0 y a la variable op(de operación) le asigno el valor “+” para utilizarlo más adelante.

```

case R.id.restar:
    numero1=Float.parseFloat(resultado.getText().toString());
    op="-";
    resultado.setText("0");
    break;
case R.id.multi:
    numero1=Float.parseFloat(resultado.getText().toString());
    op="*";
    resultado.setText("0");
    break;

```

Para la resta, multiplicación y división, el código es exactamente igual.

A la variable número1 le damos todo lo que contiene en la caja resultado y le asignamos a 0.

```
case R.id.borrar:
    resultado.setText("0");
    numero1= 0.0f;
    numero2= 0.0f;
    break;
```

Para borrar, simplemente pongo el resultado a 0 y las variables numero1 y numero2 también se vuelven 0.

```
case R.id.igual:
    numero2=Float.parseFloat(resultado.getText().toString());
```

En el igual es donde se hacen los cálculos.

Lo primero, al presionar una operación, cojo todo lo que contiene resultado y lo meto dentro de la variable numero1.

Al pulsar igual, lo primero que hago es coger todo lo que contiene resultado (que se reseteó tras la operación) y lo meto dentro de la variable número2.

A partir de aquí es mucho más sencillo.

Cuando pulsamos cualquier operación le asignamos un valor a la variable op.

He utilizado ifs para comparar.

```
if(op.equals("/")){
    if(numero2==0.0f){
        resultado.setText("0");
        Toast.makeText( context: this, text: "Division entre 0",Toast.LENGTH_LONG).show();
    }else{
        float result= numero1/numero2;
        resultado.setText(result+"");
    }
}
```

En este caso si op contiene / . Hago otro if para validar que el segundo número no sea 0.

Si es 0, muestro el resultado en 0 y saco un mensaje por pantalla con el Toast.

En este caso mi mensaje es división entre 0.

Si el segundo número no es 0:

Creo una variable result y divido los dos números.

A continuación, muestro ese resultado por pantalla con el setText.

```
if(op.equals("+")){  
    resultado.setText((int)(numero1+numero2)+"");  
}  
if(op.equals("-")){  
    resultado.setText((int)(numero1-numero2)+"");  
}  
if(op.equals("*")){  
    resultado.setText((int)(numero1*numero2)+"");  
}
```

Dependiendo de lo que contenga op, sumará restará o multiplicará.

Por último, pongo las dos variables en 0.

```
}  
  
numero1=0.0f;  
numero2=0.0f;  
  
break;
```

Lo último sería llamarlo en el OnClick

```
@Override  
public void onClick(View view) { principal(view.getId()); }
```

Aquí es donde el programa coge un id con el getId , principal recibe ese Id y como 1234==boton1 ejecuta el código.