



Instituto Politécnico Nacional

Omar Montoya Romero

7CM1

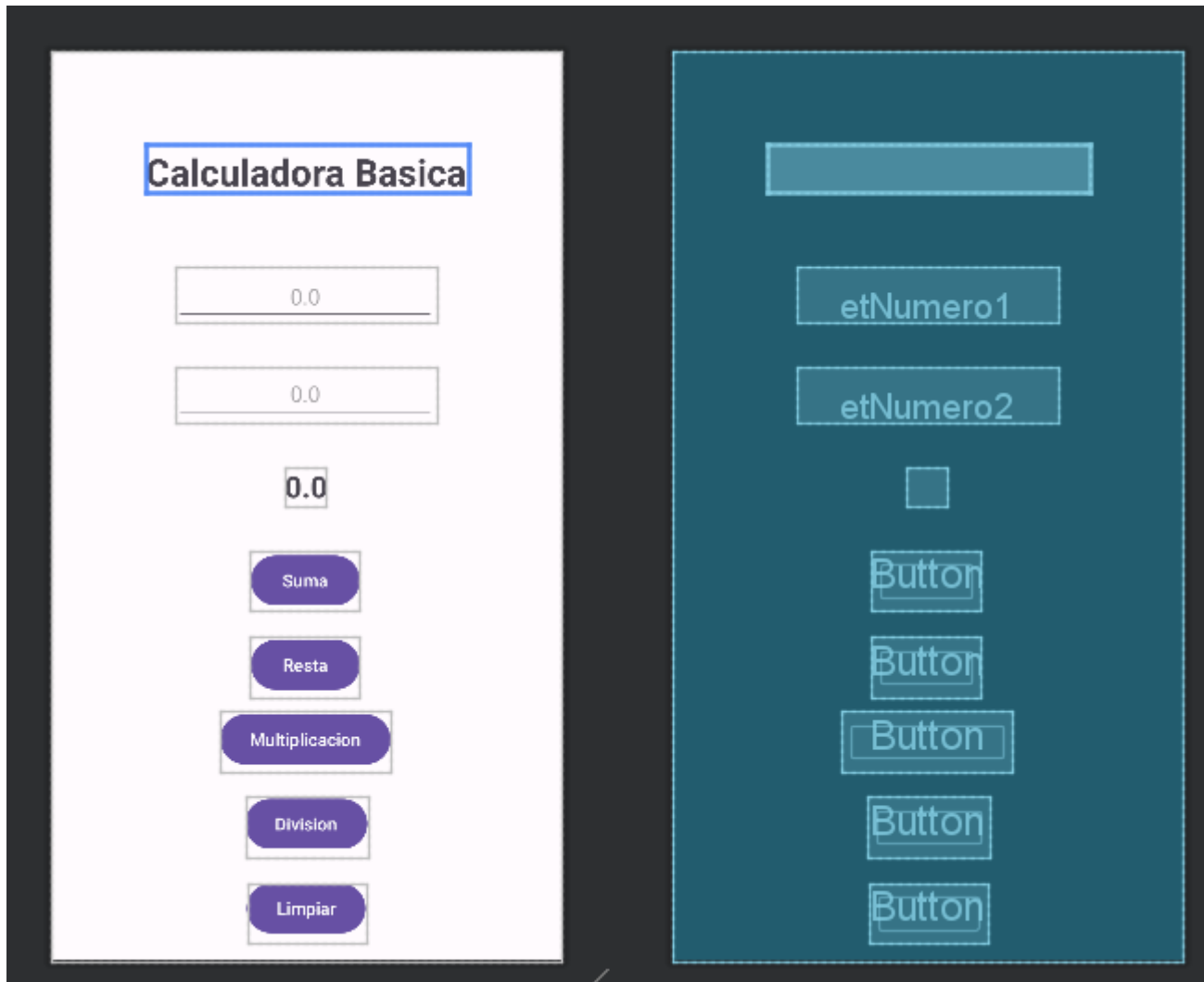
Aplicaciones Móviles Nativas

Sistemas Computacionales

Ejercicio2.- Calculadora Básica Nativa

Mtro. Efraín Arredondo Morales

08/09/2023



Primero diseñamos la aplicación arrastrando los diferentes componentes a usar en la vista, los componentes que se usaron fueron 2 TextView, 2 EditText y 5 Button, al inicio dio problemas el acomodo ya que daba error por no haberlos acomodado correctamente, se tenían que sincronizar con los extremos para que se acomoden bien y al momento de usarlo en diferentes dispositivos estos no se desacomoden.

```
lateinit var num1:EditText
lateinit var num2:EditText
lateinit var resultado:TextView
lateinit var btnSuma:Button
lateinit var btnResta:Button
lateinit var btnMultiplicacion:Button
lateinit var btnDivision:Button
lateinit var btnLimpiar:Button
```

Una vez acomodados todos correctamente y con su id especial pasamos a declararlos en el main para poder llamarlos después se utiliza lateinit var para declarar una nueva variable donde se pone el nombre y el tipo de variable que va a ser.

```
//Hacer referencia a la vista
num1 = findViewById(R.id.etNumero1)
num2 = findViewById(R.id.etNumero2)
resultado = findViewById(R.id.tvResultado)
btnSuma = findViewById(R.id.btnSuma)
btnDivision = findViewById(R.id.btnDivision)
btnMultiplicacion = findViewById(R.id.btnMultiplicacion)
btnResta = findViewById(R.id.btnResta)
btnLimpiar = findViewById(R.id.btnLimpiar)
```

Después de declararlos pasamos a hacer referencia a los componentes de la vista anteriormente inicializados en la vista se hace referencia mediante el id que se le definió al crearlos, esto para poder usarlos en los métodos y funciones.

```
btnSuma.setOnClickListener { it: View?
    //Mandar llamar el metodo
    realizarSuma()
}
```

Para el uso de los botones se le activa la función de OnClick el cual cada vez que se le de click manda llamar la función que este asignada en este caso para resolver las operaciones matemáticas básicas.

```

fun realizarSuma(){
    // 1.- Obtener valores de cajas de texto
    val num1Text:String = num1.text.toString()
    val num2Text:String = num2.text.toString()
    // 2.- Determinar si no estan vacias las cajas de texto
    if (num1Text.isNotEmpty() && num2Text.isNotEmpty())
    {
        // 3.- Asignar valores con numeros
        var num1count:Double = num1Text.toDouble()
        var num2count:Double = num2Text.toDouble()
        // 4.- Realizar la operacion
        var res1 = num1count + num2count
        // 5.- Nostrar el resultado
        resultado.setText(res1.toString())
    }
    // 2.1.-En caso de estar vacios mostrar un mensaje apropiado
    else
    {
        resultado.text= "Error de datos"
    }
}
}

```

Despues de definir el metodo para el funcionamiento del botón tenemos que desarrollar la función que este va a llamar, primero se obtienes los valores de los de las cajas de texto de la aplicación donde se mandan llamar por el identificador asignado anteriormente, se le asigna a una nueva variable la cual nos va a permitir hacer las operaciones, antes de que se haga cualquier operación primero comparamos que los campos no estén vacíos, si las cajas contienen números procede a realizar las operaciones donde se crean otras dos nuevas variables las cuales se les asigna el contenido de las variables anteriormente pero ahora como double, una vez echa la operación se guarda en otra variable llamada res1 la cual es imprimida en el TextView llamado Resultado, en caso de ingresar datos pero estos no sean números lanza el Error de “Error de datos”.

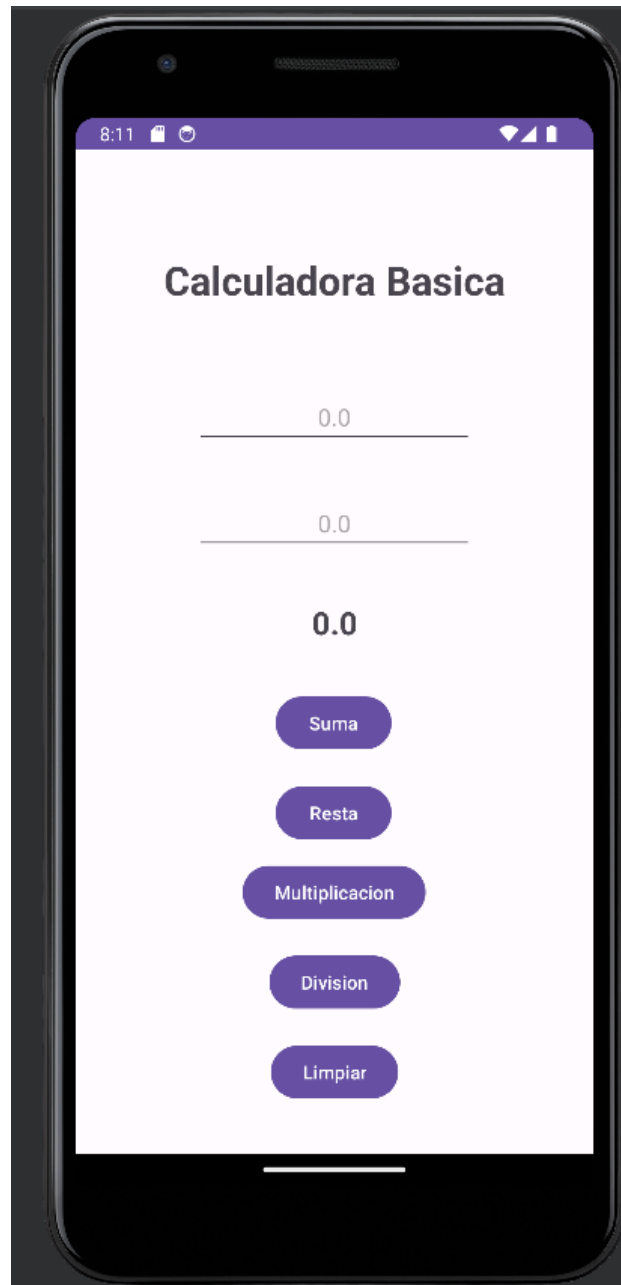
```

fun realizarDivision(){
    // 1.- Obtener valores de cajas de texto
    val num1Text:String = num1.text.toString()
    val num2Text:String = num2.text.toString()
    // 2.- Determinar si no estan vacias las cajas de texto
    if (num1Text.isNotEmpty() && num2Text.isNotEmpty())
    {
        if (num2Text.contentEquals(charSequence: "0")){
            resultado.text= "No se divide entre 0"
        }else{
            // 3.- Asignar valores con numeros
            var num1count:Double = num1Text.toDouble()
            var num2count:Double = num2Text.toDouble()
            // 4.- Realizar la operacion
            var res1:Double = num1count / num2count
            // 5.- Nostrar el resultado
            resultado.setText(res1.toString())
        }
    }

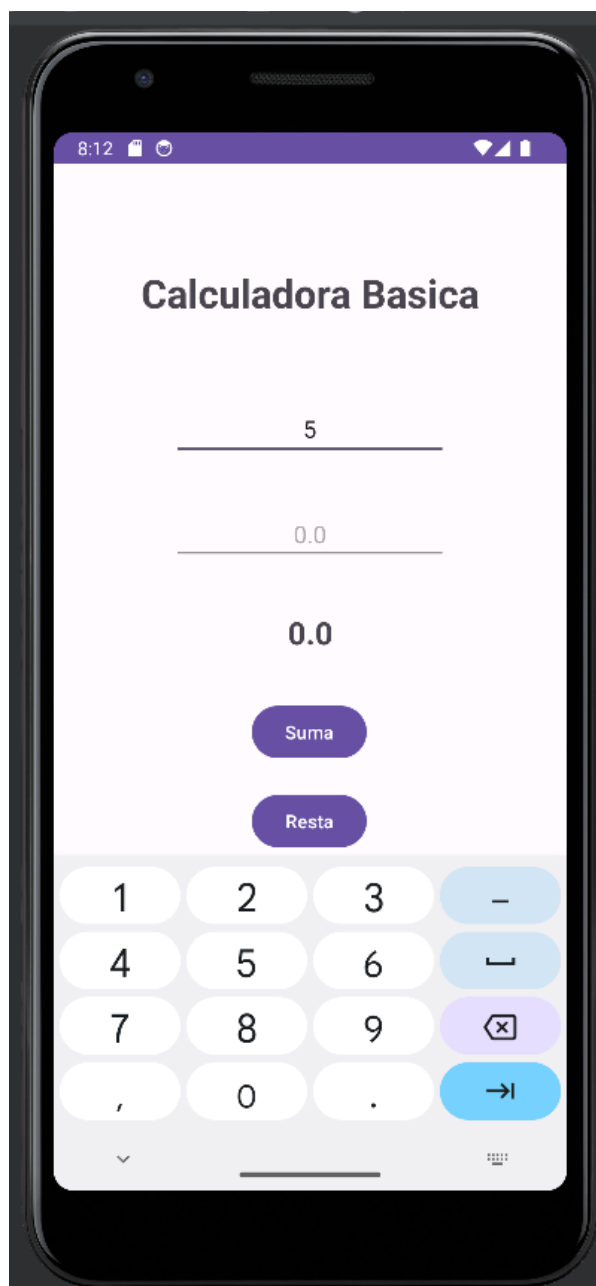
    // 2.1.-En caso de estar vacios mostrar un mensaje apropiado
    else
    {
        resultado.text= "Error de datos"
    }
}

```

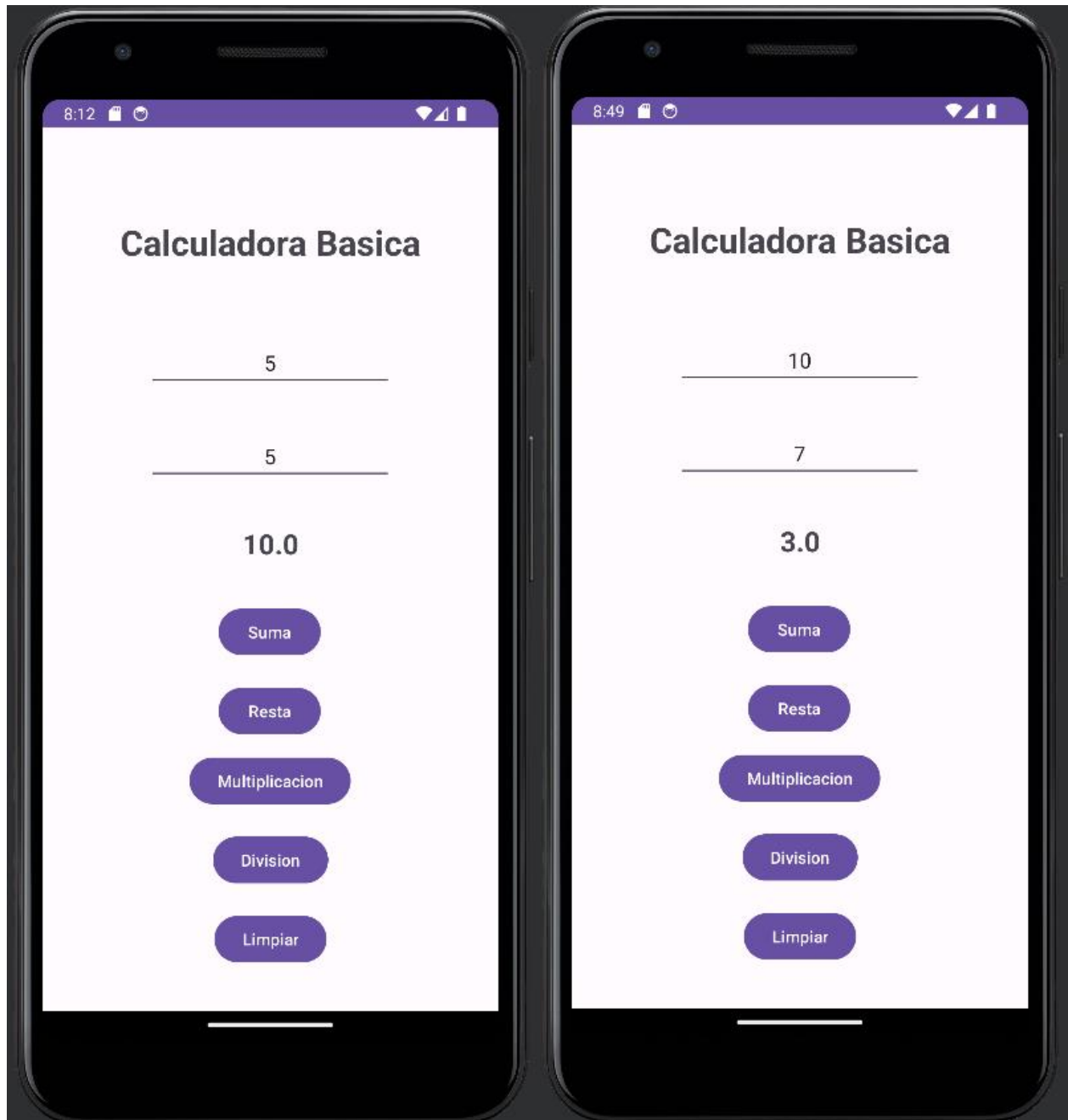
En el caso de la división es el mismo proceso solo que se compara antes de cualquier operación que el segundo número no sea un 0 ya que no se puede dividir entre 0, si se da el caso se arroja el mensaje “No se divide entre 0” esto para que se cambie la cantidad y pueda seguir usando la aplicación.



En esta vista se ve el resultado antes de usar cualquier componente empleado en la aplicación



En esta vista se muestra el como se ingresan los datos a las entradas 1 y 2 para poder realizar las operaciones.



En la primer imagen se usa el botón de suma donde $5 + 5 = 10$, mientras que en la segunda imagen se usa la resta donde $10 - 7 = 3$. Aquí se demuestra el correcto funcionamiento de la aplicación.



En la primera vista se muestra el resultado de la multiplicación donde $5 * 5 = 25$ mientras que en la segunda se muestra como se resuelve la división donde $5 / 5 = 1$. Comprobando el uso de los dos botones.



En estas dos ultimas se muestra el funcionamiento del botón Limpiar donde como se había visto con anterioridad se tenia en la entrada1 un 5, entrada2 un 5 y en el resultado un 1, al darle al botón en automático todos los valores regresan a 0. En la segunda vista se muestra el caso donde se intenta dividir $10 / 0$ el cual no es posible por lo tanto arroja el mensaje de "No se divide entre 0"

En este diseño el principal reto fue al no tener conocimiento de como programarlo se complicaba al programarlo pero conforme se fue avanzando se entendió más y se volvió sencillo el programarlo, esta app lleva este diseño ya que se opto por un diseño limpio y de fácil uso para el usuario.