



**INSTITUTO POLITÉCNICO NACIONAL**  
**ESCUELA SUPERIOR DE CÓMPUTO**  
Ingeniería en Sistemas Computacionales



# **CREANDO MI PRIMER APLICACIÓN**

Práctica V

## **DESARROLLO DE APLICACIONES MÓVILES NATIVAS**

**Alumno:**

Saucedo Moreno César Enrique.

**Profesora:**

Morales Guitrón Sandra Luz.

**Grupo 7CV1**

**15 / Octubre / 2024**

**2025 ~ 1**

<b>INTRODUCCIÓN.....</b>	<b>1</b>
<b>DESARROLLO.....</b>	<b>2</b>
Creando un nuevo proyecto.....	2
Ejecución de la aplicación:.....	4
<b>CONCLUSIÓN.....</b>	<b>10</b>

# INTRODUCCIÓN

Bajo el contexto del desarrollo de aplicaciones móviles, comprender los fundamentos del desarrollo es fundamental para dominar los conceptos básicos que forman la estructura de cualquier aplicación. Al iniciar con un ejercicio sencillo, como lo es la creación de una calculadora, puedo no solo familiarizarme con la interfaz de usuario, sino también desarrollar habilidades esenciales en la manipulación de eventos, la lógica de programación y el manejo de datos. Este tipo de proyecto, aunque básico, me permite aplicar de manera práctica los conocimientos adquiridos, sentando las bases para enfrentar desafíos más avanzados.

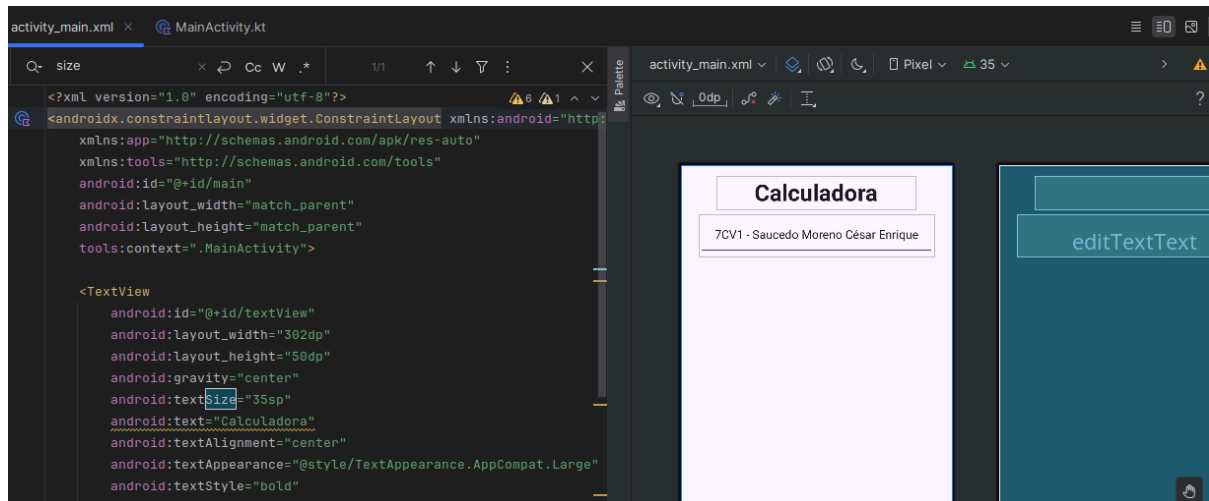
El crear una calculadora me da la oportunidad de trabajar con las herramientas específicas de desarrollo móvil, como Android Studio, la cual proporciona herramientas para el diseño de interfaces, comunicación con la propia aplicación, entre otras. A través de este proceso, puedo entender mejor el flujo de trabajo en el desarrollo de aplicaciones nativas, desde el diseño de la interfaz hasta la implementación de funcionalidades, logrando una comprensión más profunda de todo el ecosistema móvil.

# DESARROLLO

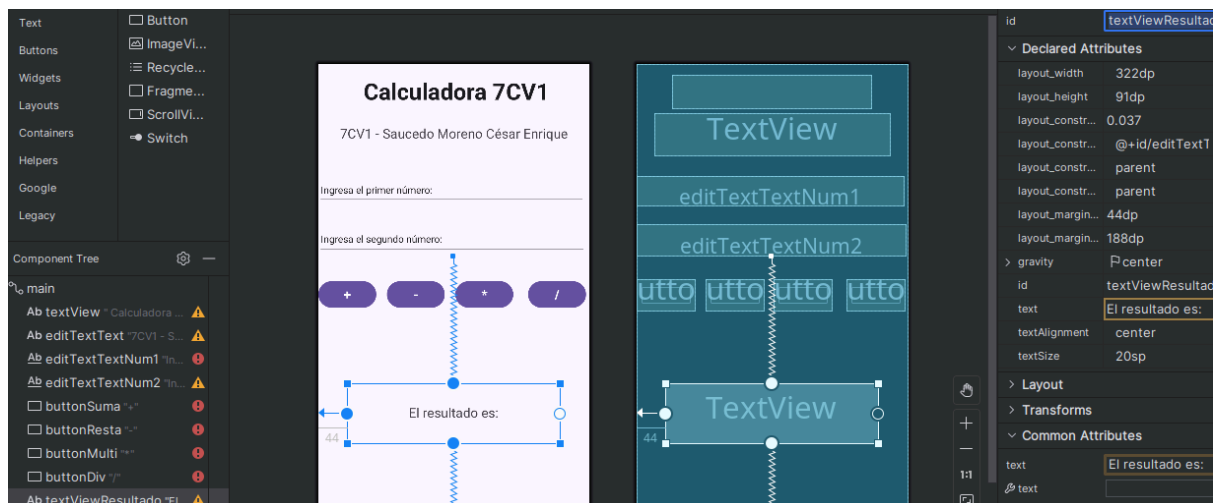
## Creando un nuevo proyecto.

Para el desarrollo de la calculadora cómo aplicación comenzamos creando un nuevo proyecto de tipo **Empty Activity**.

Desarrollando la interfaz de la aplicación mediante el XML:



El desarrollo completo de la interfaz de la aplicación es la siguiente, para ella, se hizo uso de TextViews y button para la aplicación.



Para la elaboración de mi aplicación, realizaremos el proceso inicial de la capturar las entradas, es decir, con un EditText recuperaremos los valores introducidos y se las asignaremos a las variables numero1 y número 2 para posteriormente realizar la función.

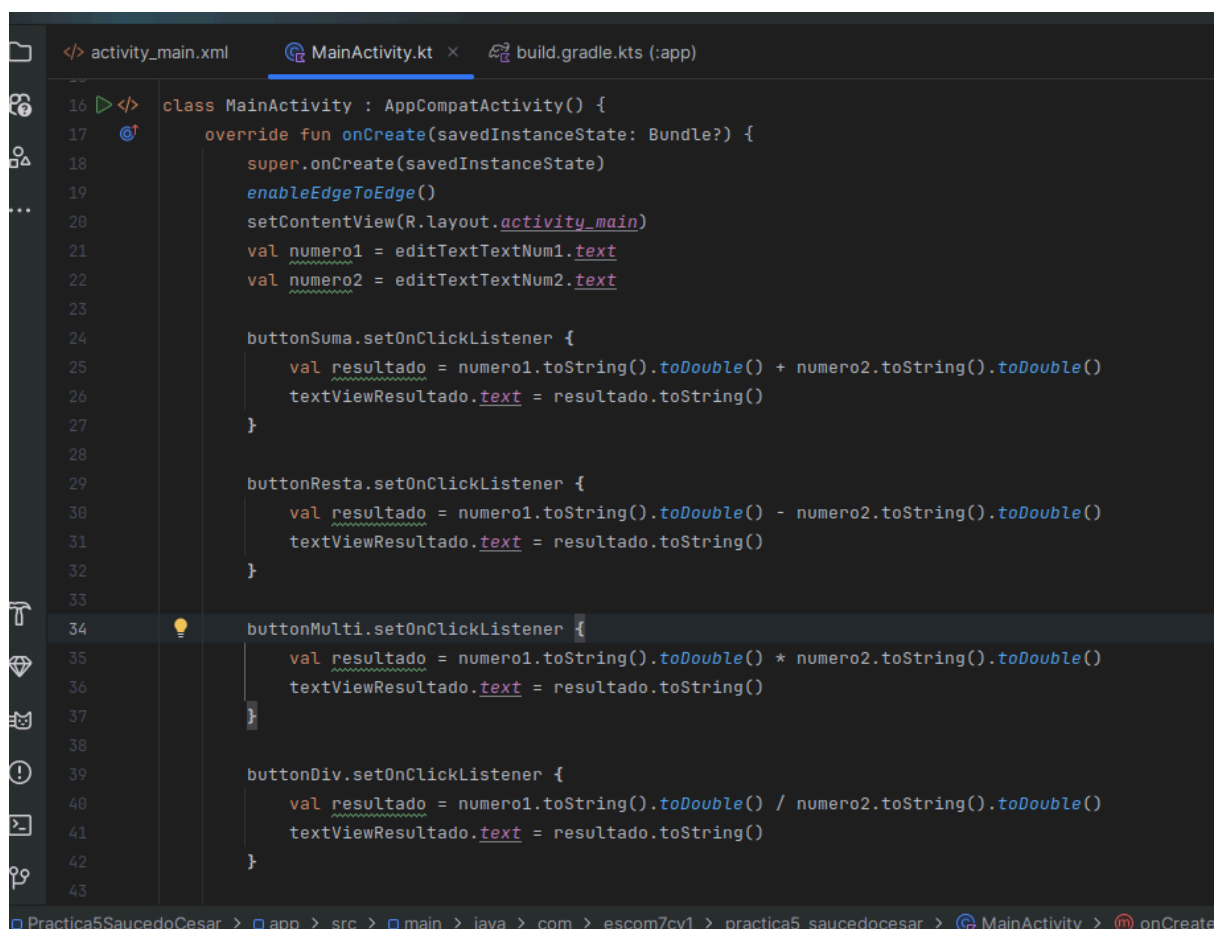
**Suma:** Al presionar el botón de suma (buttonSuma), se convierte el texto de los EditText en números utilizando el método toDouble(). Luego, se realiza la

operación de suma y el resultado es convertido nuevamente a texto para ser mostrado en el TextView.

**Resta:** El botón de resta (buttonResta) sigue el mismo proceso que el de suma, pero esta vez restando el segundo número al primero. El resultado también se muestra en el TextView.

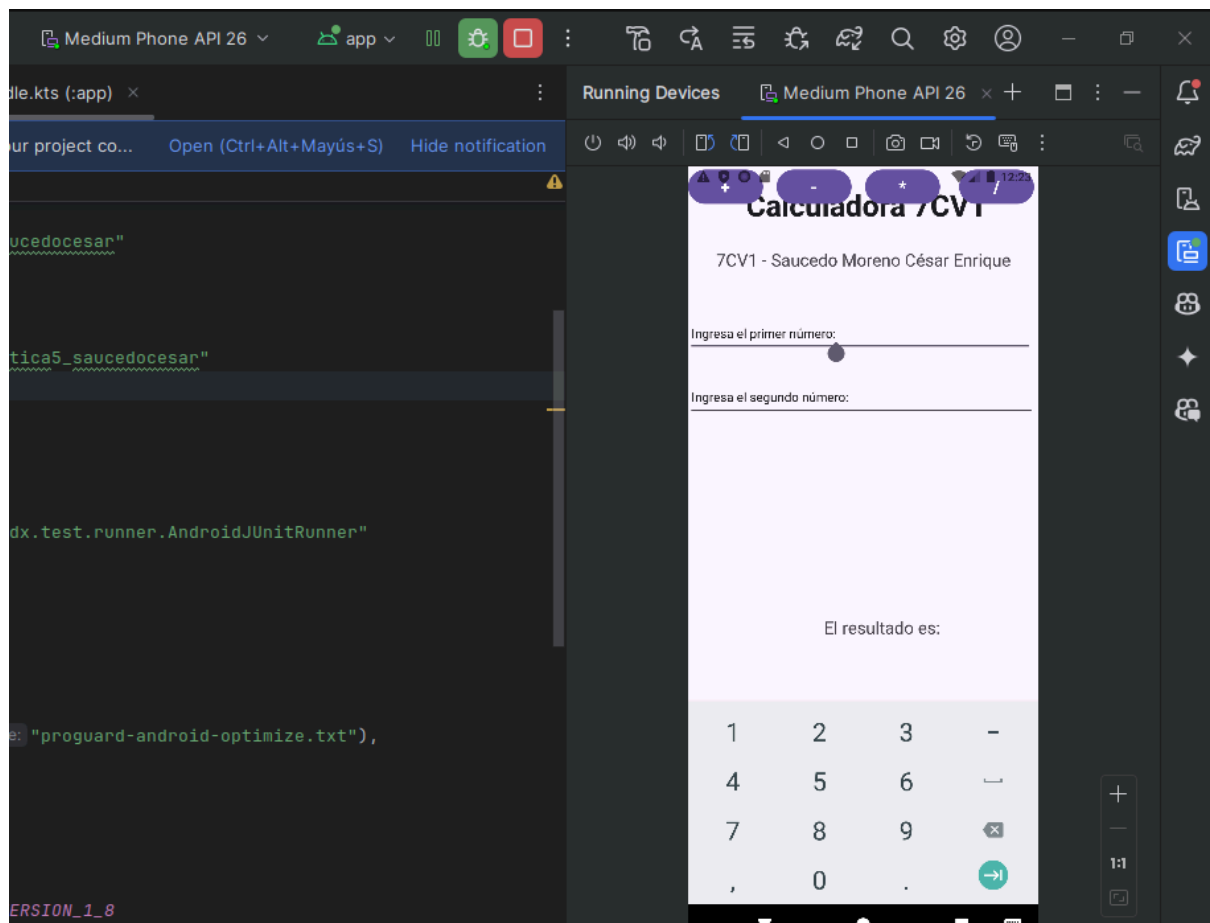
**Multipliación:** El botón de multiplicación (buttonMulti) multiplica los dos números ingresados por el usuario, siguiendo el mismo proceso de conversión y despliegue del resultado en el TextView.

**División:** Finalmente, el botón de división (buttonDiv) divide el primer número por el segundo, mostrando el resultado en el TextView. Esta operación es delicada, ya que requiere control sobre divisiones por cero para evitar errores en la aplicación.

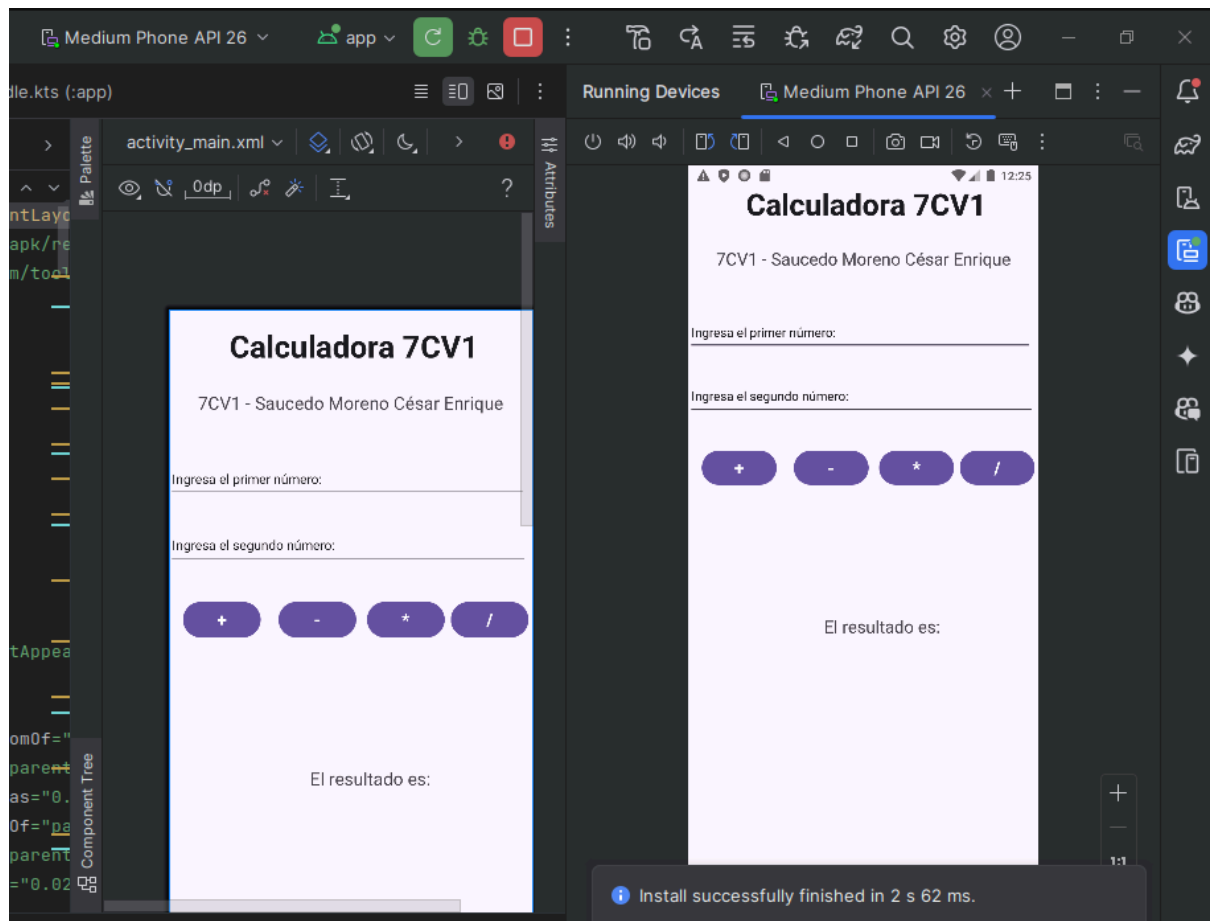


```
16 <> class MainActivity : AppCompatActivity() {
17     override fun onCreate(savedInstanceState: Bundle?) {
18         super.onCreate(savedInstanceState)
19         enableEdgeToEdge()
20         setContentView(R.layout.activity_main)
21         val numero1 = editTextTextNum1.text
22         val numero2 = editTextTextNum2.text
23
24         buttonSuma.setOnClickListener {
25             val resultado = numero1.toString().toDouble() + numero2.toString().toDouble()
26             textViewResultado.text = resultado.toString()
27         }
28
29         buttonResta.setOnClickListener {
30             val resultado = numero1.toString().toDouble() - numero2.toString().toDouble()
31             textViewResultado.text = resultado.toString()
32         }
33
34         buttonMulti.setOnClickListener {
35             val resultado = numero1.toString().toDouble() * numero2.toString().toDouble()
36             textViewResultado.text = resultado.toString()
37         }
38
39         buttonDiv.setOnClickListener {
40             val resultado = numero1.toString().toDouble() / numero2.toString().toDouble()
41             textViewResultado.text = resultado.toString()
42         }
43     }
}
```

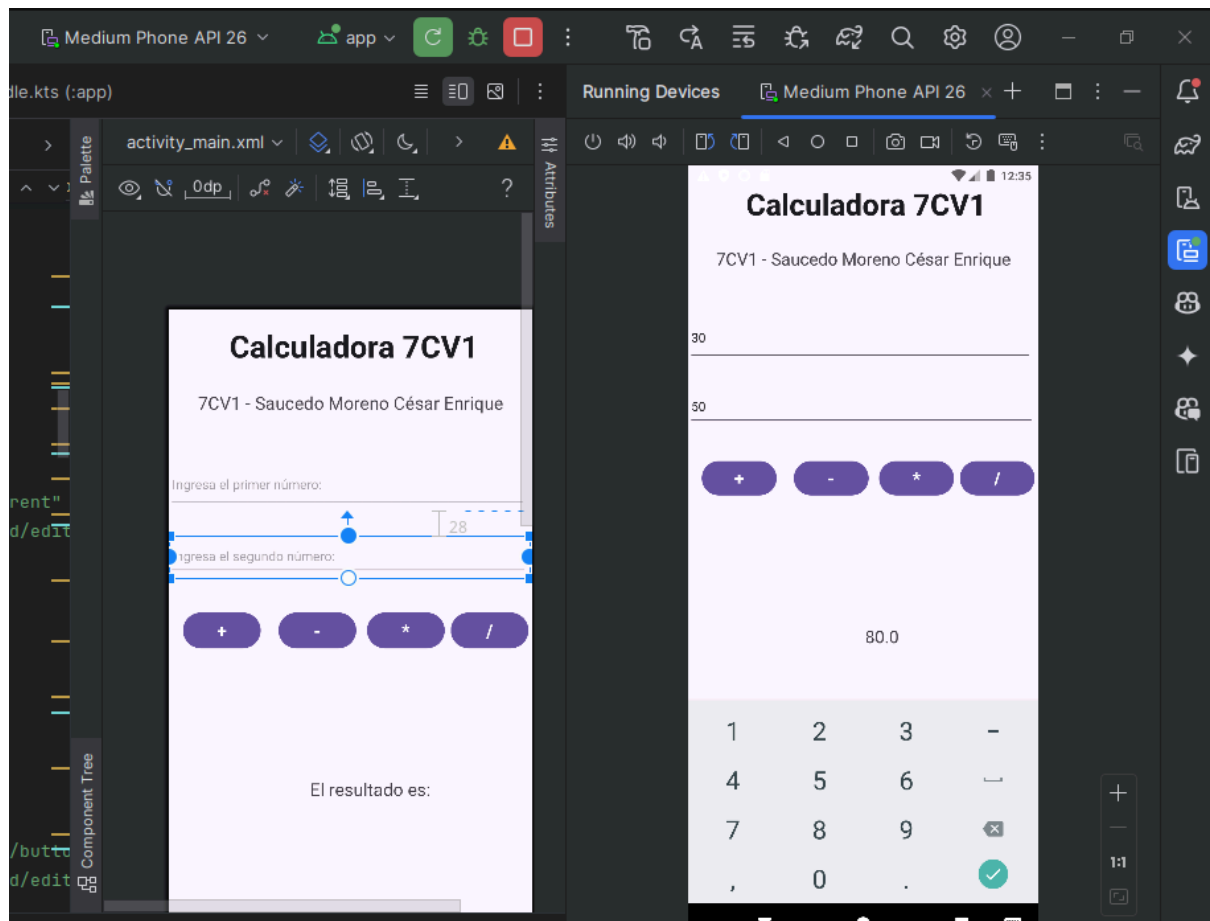
## Ejecución de la aplicación:



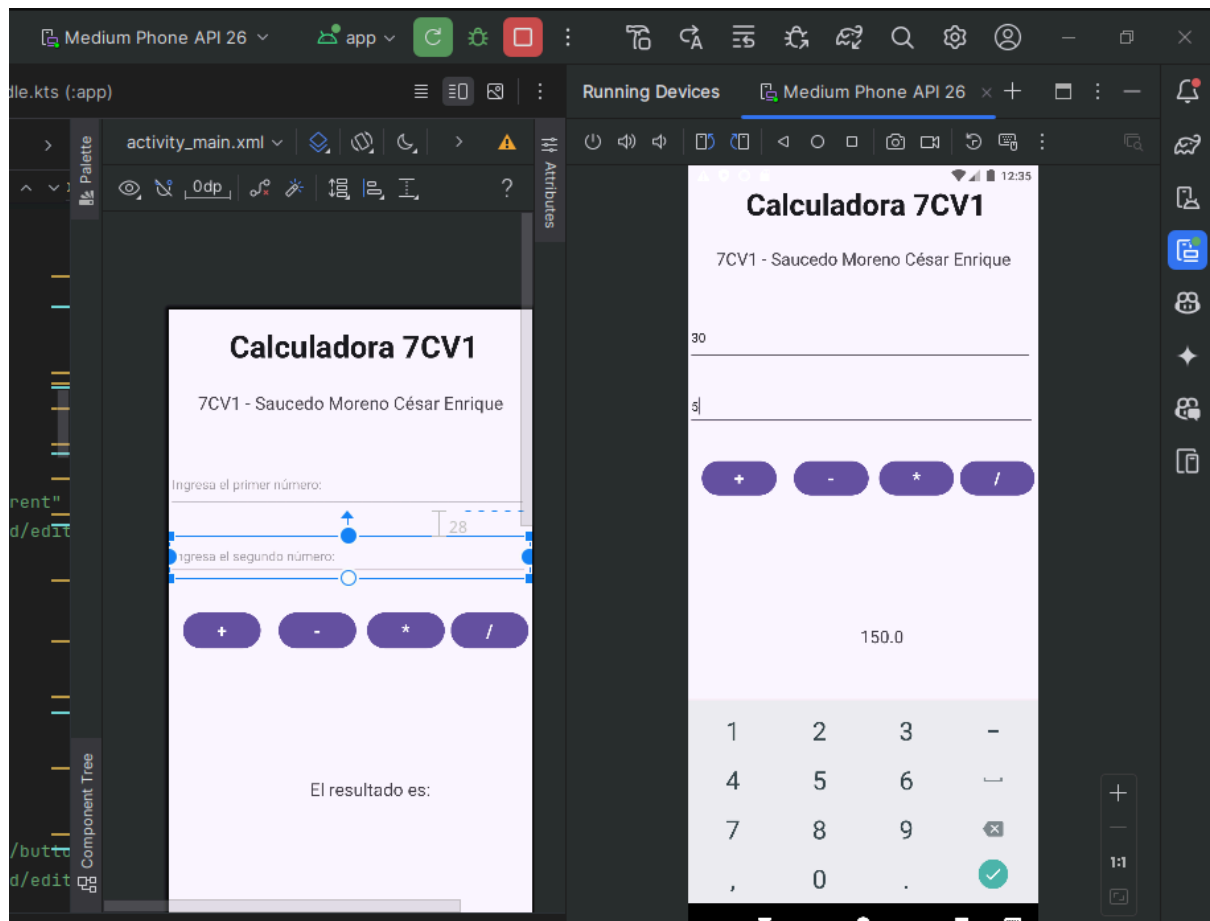
Corrigiendo la posición de los botones:

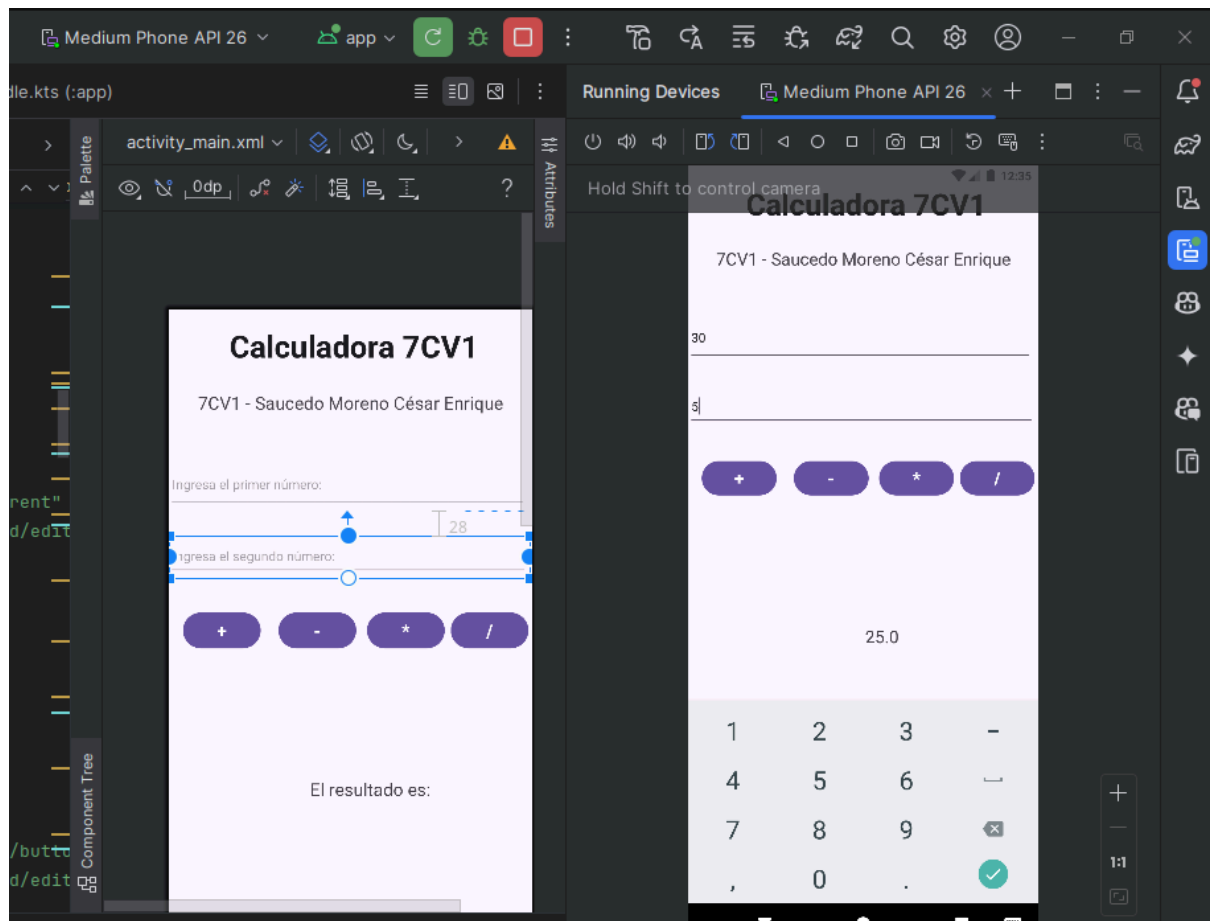


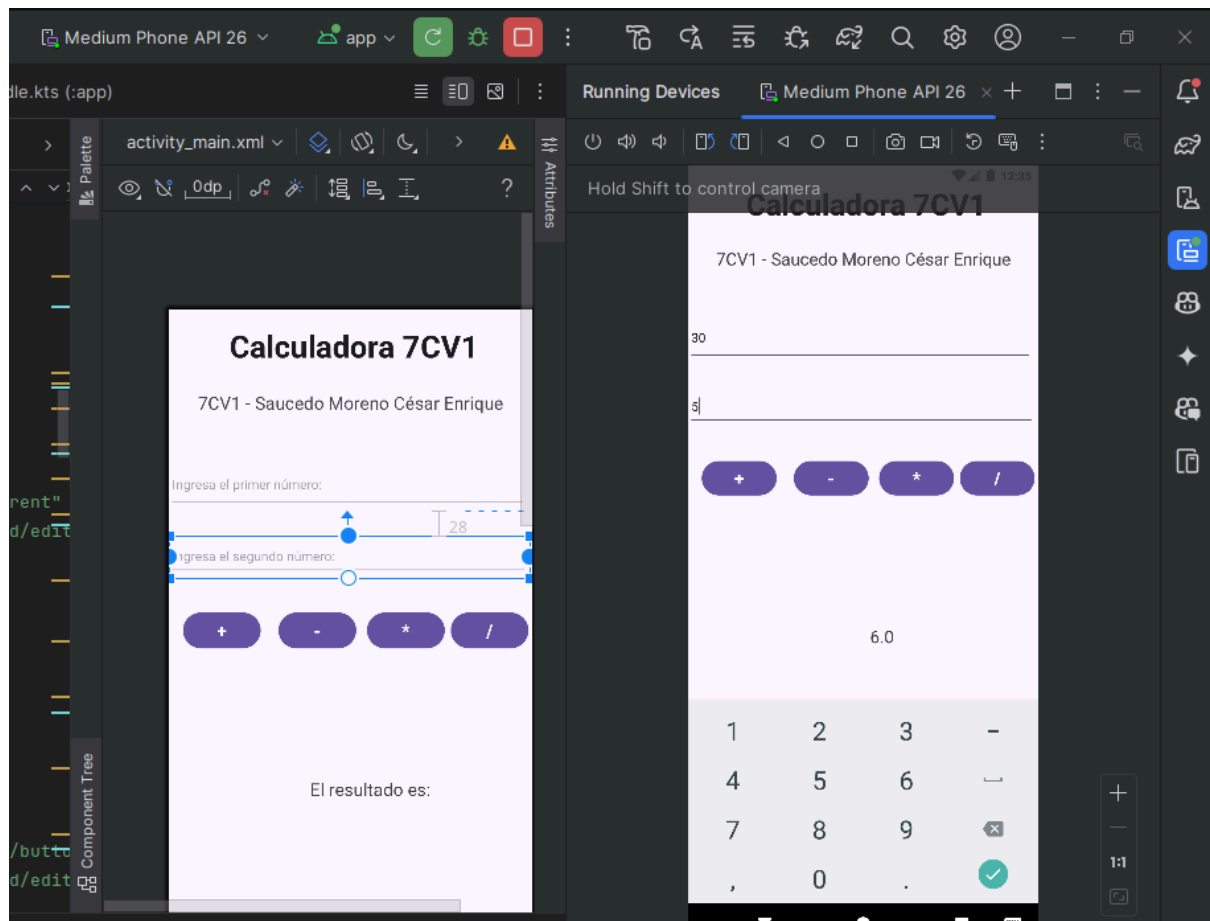
Realizando unas pruebas a la aplicación:











## **CONCLUSIÓN**

Durante el desarrollo de esta calculadora simple en Android Studio, recordé una práctica similar que realicé en la vocacional, en la cual también trabajé con Android Studio para desarrollar una aplicación móvil. Al igual que en aquella ocasión, noté lo exigente que puede ser este entorno de desarrollo, tanto en términos de recursos del sistema como al momento de ejecutar la simulación de la aplicación localmente.

El proyecto de la calculadora me permitió recordar conceptos como el manejo de los layouts, la interacción entre vistas y el uso de View Binding para enlazar los elementos de la interfaz de usuario con el código Kotlin.