



**Carrera:**

**Técnico en ingeniería en computación**

**Docente:**

**Ing. Manuel Alexander Jiménez García.**

**Materia:**

**Desarrollo de Software para Móviles DSM441 G03L**

**Integrantes:**

**Manuel Ezequiel Guerrero Granados GG241501**

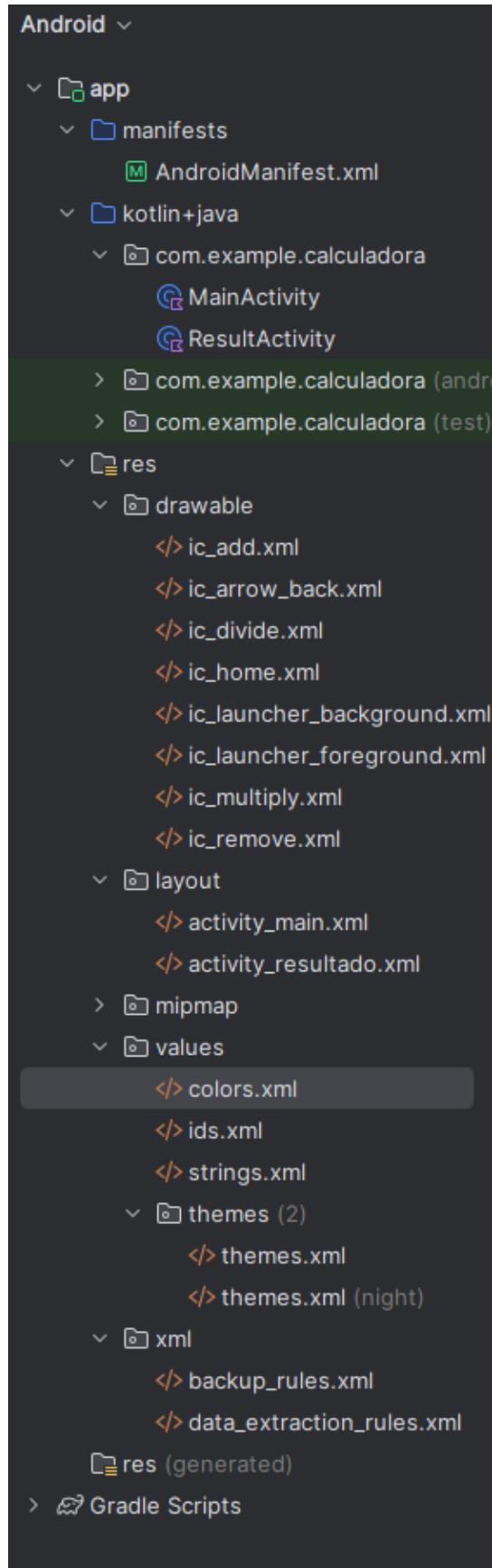
**Norman Emmanuel Quijano Amaya QA240640**

**Fecha:**

**26/07/2025**

**Enlace a GitHub:**

**<https://github.com/Manuelguerrero234/DSM441-G03L.git>**

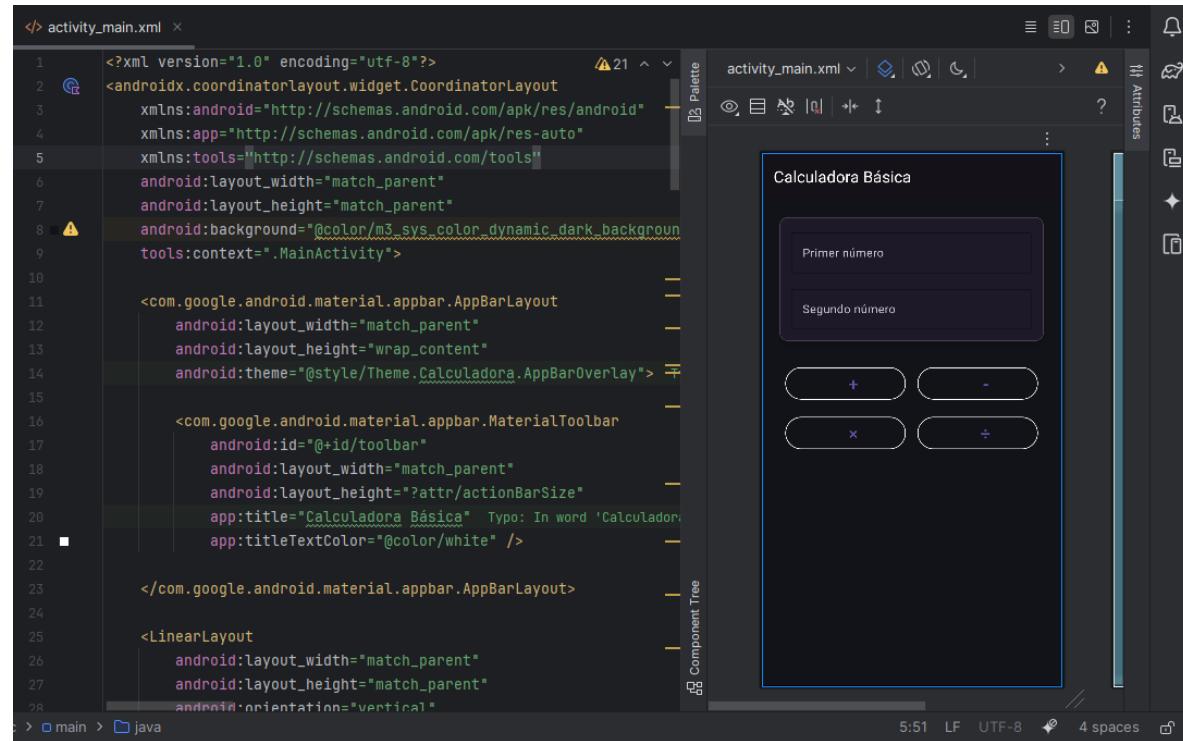


**Estructura del proyecto del desafío.**

## activity\_main.xml

-*Función:* Define la estructura visual de la pantalla principal de la calculadora.

-*Contenido clave:* Campos de entrada para los números, botones de operaciones y el diseño general usando componentes Material Design.



```
<?xml version="1.0" encoding="utf-8"?>
<androidx.coordinatorlayout.widget.CoordinatorLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="@color/m3_sys_color_dynamic_dark_background"
    tools:context=".MainActivity">

    <com.google.android.material.appbar.AppBarLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:theme="@style/Theme.Calculadora.AppBarOverlay">

        <com.google.android.material.appbar.MaterialToolbar
            android:id="@+id/toolbar"
            android:layout_width="match_parent"
            android:layout_height="?attr/actionBarSize"
            app:title="Calculadora Básica" Tipo: In word 'Calculadora'
            app:titleTextColor="@color/white" />

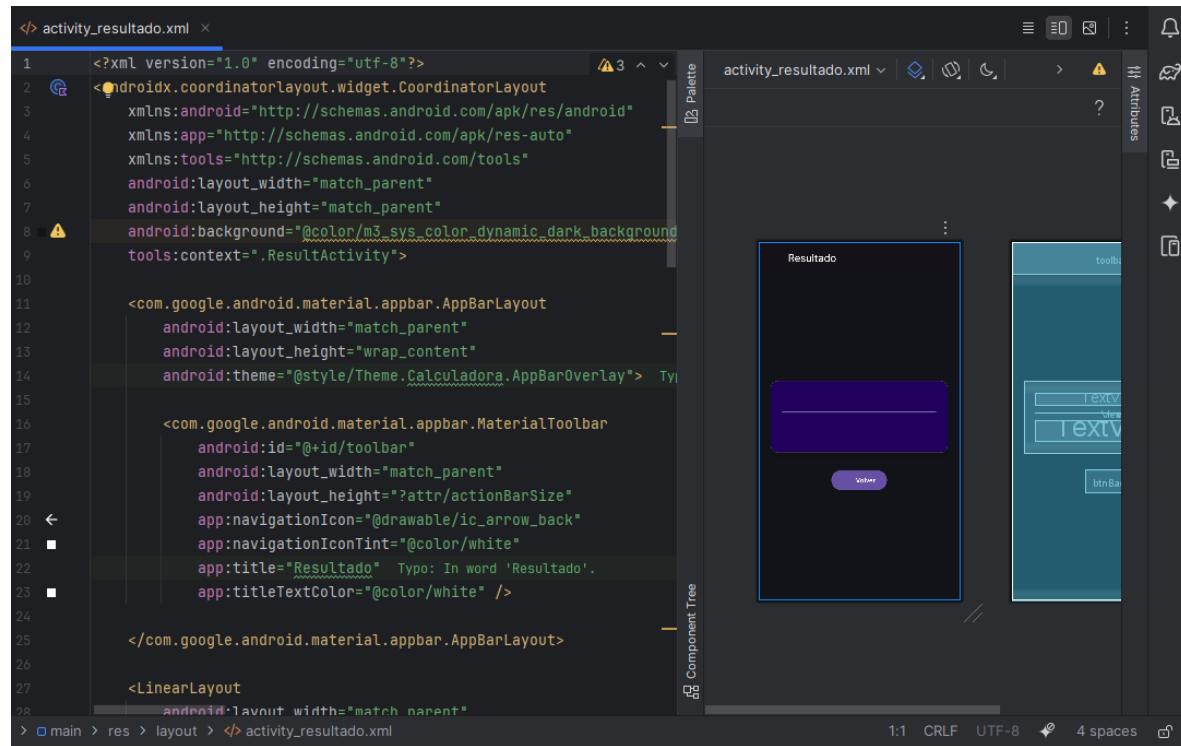
    </com.google.android.material.appbar.AppBarLayout>

    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:orientation="vertical">
```

## activity\_resultado.xml

-**Función:** Estructura la pantalla que muestra el resultado de las operaciones.

-**Contenido clave:** Área para visualizar la operación realizada y su resultado, con diseño de tarjeta (CardView) para mejor presentación.



The screenshot shows the Android Studio interface with the XML layout file `activity_resultado.xml` open. The code editor on the left contains the XML code for the layout, which includes a CoordinatorLayout, an AppBarLayout with a toolbar, and a CardView for displaying results. The preview window on the right shows a dark-themed mobile application interface with a toolbar at the top and a large purple CardView below it. The component tree on the bottom right lists the views used in the layout. The status bar at the bottom indicates the code is in CRLF format, UTF-8 encoding, and uses 4 spaces for indentation.

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.coordinatorlayout.widget.CoordinatorLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="@color/m3_svs_color_dynamic_dark_background"
    tools:context=".ResultActivity">

    <com.google.android.material.appbar.AppBarLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:theme="@style/Theme.Calculadora.AppBarOverlay" >
        <com.google.android.material.appbar.MaterialToolbar
            android:id="@+id/toolbar"
            android:layout_width="match_parent"
            android:layout_height="?attr/actionBarSize"
            app:navigationIcon="@drawable/ic_arrow_back"
            app:navigationIconTint="@color/white"
            app:title="Resultado"  Tipo: In word 'Resultado'.
            app:titleTextColor="@color/white" />
    </com.google.android.material.appbar.AppBarLayout>

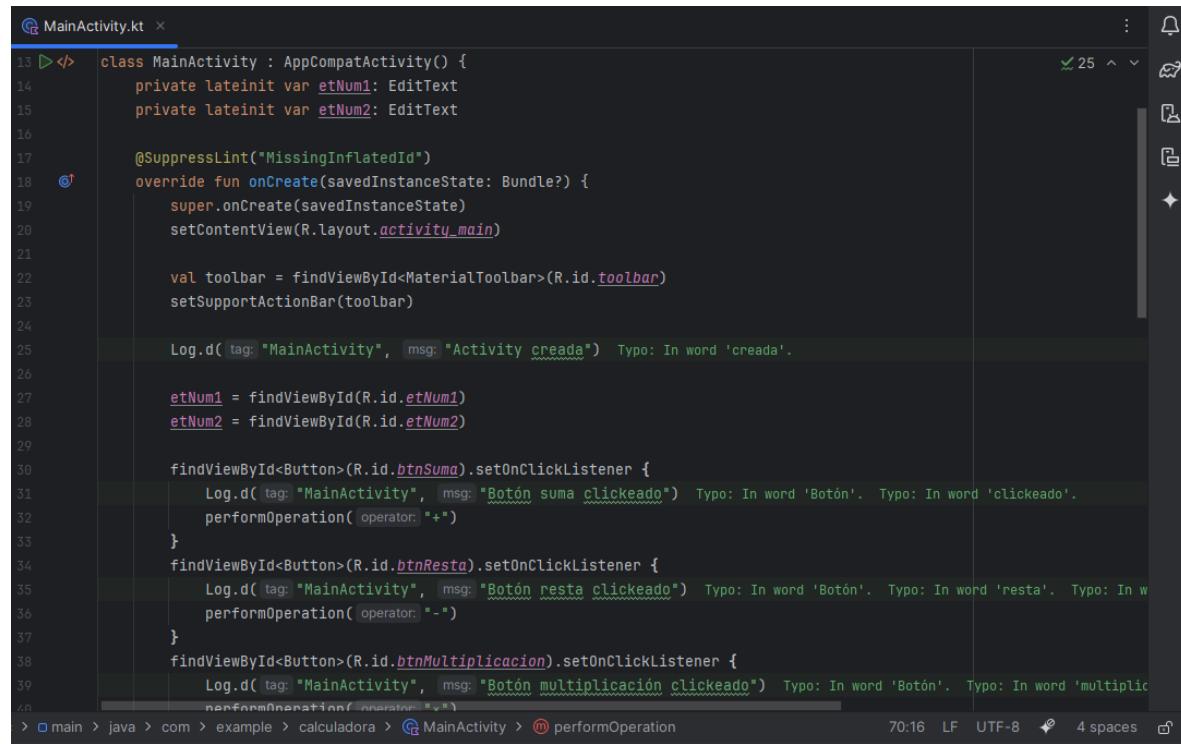
    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content">

```

## MainActivity.kt:

-*Función*: Controla la lógica de la pantalla principal.

-*Responsabilidades*: Gestiona los inputs del usuario, validaciones y navegación a la pantalla de resultados.



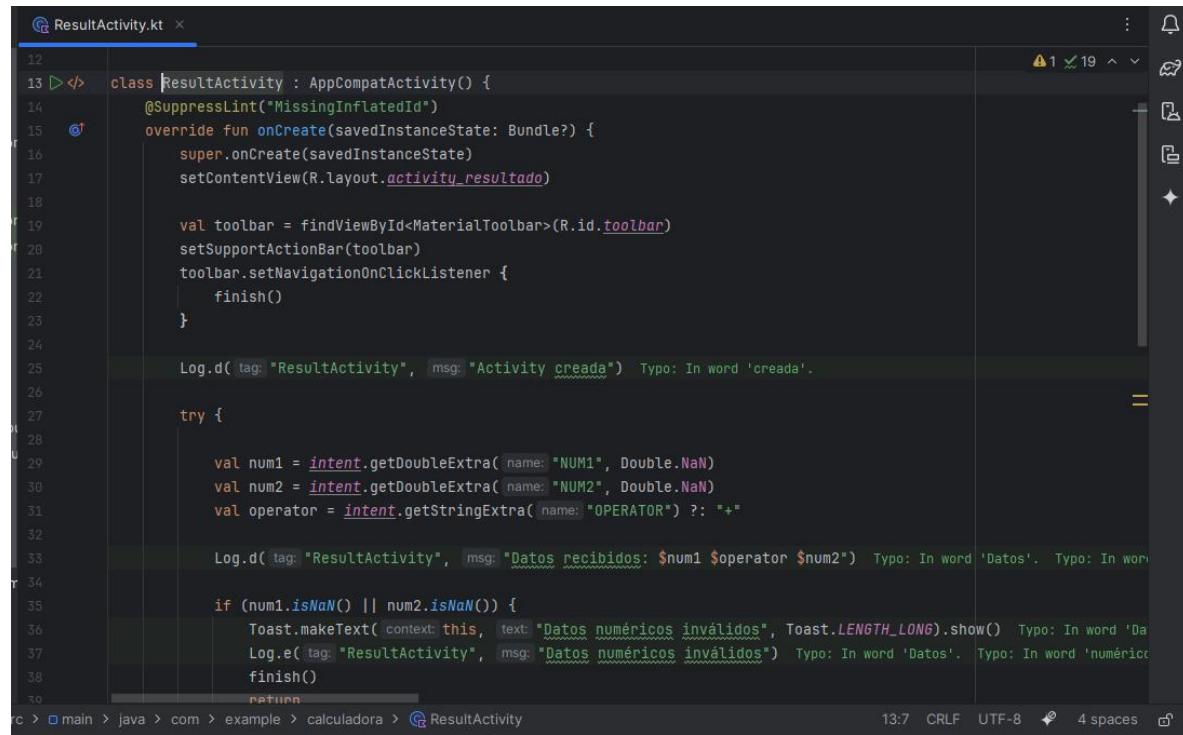
The screenshot shows the Android Studio code editor for MainActivity.kt. The code defines a class MainActivity that extends AppCompatActivity. It initializes two EditText fields, etNum1 and etNum2. In the onCreate method, it sets the content view to activity\_main and configures a toolbar. It then logs the creation of the activity and initializes two EditText fields. The code then sets up three button listeners: btnSuma, btnResta, and btnMultiplicacion. Each listener logs a message indicating the button was clicked and then calls the performOperation method with a specific operator (+, -, or \*) respectively. The code editor interface includes tabs for Java, XML, and Kotlin, and a status bar at the bottom showing the file path and encoding.

```
13 > MainActivity.kt ×
14 class MainActivity : AppCompatActivity() {
15     private lateinit var etNum1: EditText
16     private lateinit var etNum2: EditText
17
18     @SuppressLint("MissingInflatedId")
19     override fun onCreate(savedInstanceState: Bundle?) {
20         super.onCreate(savedInstanceState)
21         setContentView(R.layout.activity_main)
22
23         val toolbar = findViewById<MaterialToolbar>(R.id.toolbar)
24         setSupportActionBar(toolbar)
25
26         Log.d("MainActivity", "Activity creada")  Tipo: In word 'creada'.
27
28         etNum1 = findViewById(R.id.etNum1)
29         etNum2 = findViewById(R.id.etNum2)
30
31         findViewById<Button>(R.id.btnSuma).setOnClickListener {
32             Log.d("MainActivity", "Botón suma clickeado")  Tipo: In word 'Botón'.  Tipo: In word 'clickeado'.
33             performOperation(operator: "+")
34         }
35         findViewById<Button>(R.id.btnResta).setOnClickListener {
36             Log.d("MainActivity", "Botón resta clickeado")  Tipo: In word 'Botón'.  Tipo: In word 'resta'.  Tipo: In word 'clickeado'.
37             performOperation(operator: "-")
38         }
39         findViewById<Button>(R.id.btnMultiplicacion).setOnClickListener {
40             Log.d("MainActivity", "Botón multiplicación clickeado")  Tipo: In word 'Botón'.  Tipo: In word 'multiplicación'.  Tipo: In word 'clickeado'.
41             performOperation(operator: "*")
42     }
43
44     private fun performOperation(operator: String) {
45         // Implementation of the operation based on the operator
46     }
47 }
```

## ResultActivity.kt:

-*Función*: Maneja la pantalla de resultados.

-*Responsabilidades*: Recibe datos de la operación, calcula el resultado y lo muestra al usuario.



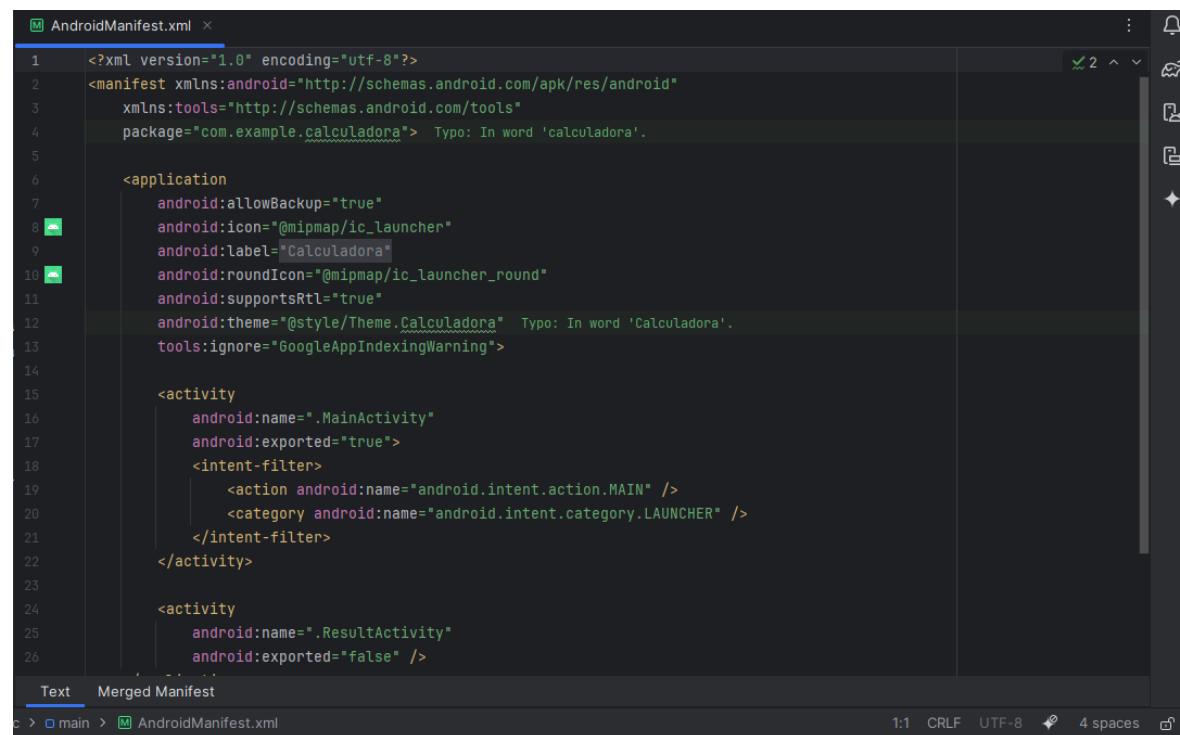
The screenshot shows the Android Studio code editor for the file `ResultActivity.kt`. The code is written in Kotlin and defines a class `ResultActivity` that extends `AppCompatActivity`. It overrides the `onCreate` method to set the content view and initialize a toolbar. It then receives data from an intent, logs it, and checks if the numbers are NaN before displaying a toast message. The code editor has a dark theme, and the code is well-formatted with color-coded syntax highlighting.

```
12
13 class ResultActivity : AppCompatActivity() {
14     @SuppressLint("MissingInflatedId")
15     override fun onCreate(savedInstanceState: Bundle?) {
16         super.onCreate(savedInstanceState)
17         setContentView(R.layout.activity_resultado)
18
19         val toolbar = findViewById<MaterialToolbar>(R.id.toolbar)
20         setSupportActionBar(toolbar)
21         toolbar.setNavigationOnClickListener {
22             finish()
23         }
24
25         Log.d(tag: "ResultActivity", msg: "Activity creada")  Tipo: In word 'creada'.
26
27     try {
28
29         val num1 = intent.getDoubleExtra(name: "NUM1", Double.NaN)
30         val num2 = intent.getDoubleExtra(name: "NUM2", Double.NaN)
31         val operator = intent.getStringExtra(name: "OPERATOR") ?: "+"
32
33         Log.d(tag: "ResultActivity", msg: "Datos recibidos: $num1 $operator $num2")  Tipo: In word 'Datos'.  Tipo: In word 'Datos'.
34
35         if (num1.isNaN() || num2.isNaN()) {
36             Toast.makeText(context: this, text: "Datos numéricos inválidos", Toast.LENGTH_LONG).show()  Tipo: In word 'Datos'.
37             Log.e(tag: "ResultActivity", msg: "Datos numéricos inválidos")  Tipo: In word 'Datos'.  Tipo: In word 'numéricos'.
38             finish()
39             return
40
41     
```

## AndroidManifest.xml:

-*Función*: Archivo de configuración global de la aplicación.

-*Contenido clave*: Declara las actividades, permisos y características de la app.



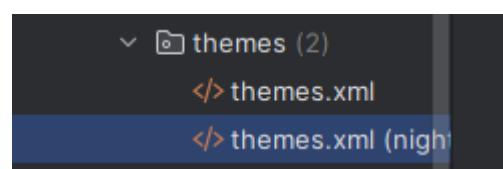
```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    package="com.example.calculadora">  Typo: In word 'calculadora'.
```

The screenshot shows the AndroidManifest.xml file in an IDE. The code is color-coded: green for XML tags, blue for attributes, and purple for values. The manifest defines a package named com.example.calculadora with an application tag containing various configuration options like theme and icons. It also lists two activities: MainActivity and ResultActivity. The MainActivity is exported, while ResultActivity is not. Intent filters are defined for the MainActivity to handle the MAIN intent and LAUNCHER category. The XML is displayed in 'Text' mode, with a 'Merged Manifest' tab visible below it. The bottom status bar shows file details like '1:1 CRLF' and 'UTF-8'.

## themes.xml:

-*Función*: Define los estilos visuales y colores base de la aplicación.

-*Importancia*: Establece la apariencia consistente en toda la app (colores, fuentes, etc.).



### colors.xml:

-*Función*: Centraliza la paleta de colores de la aplicación.

-*Ventaja*: Permite modificar fácilmente el esquema de colores en un solo lugar.

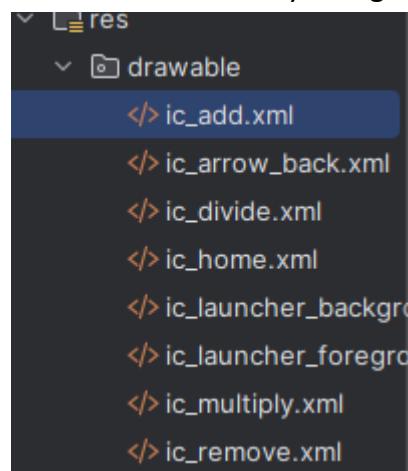
```
<?xml version="1.0" encoding="utf-8"?>
<resources>
    <color name="white">#FFFFFF</color>
    <color name="black">#FF000000</color>

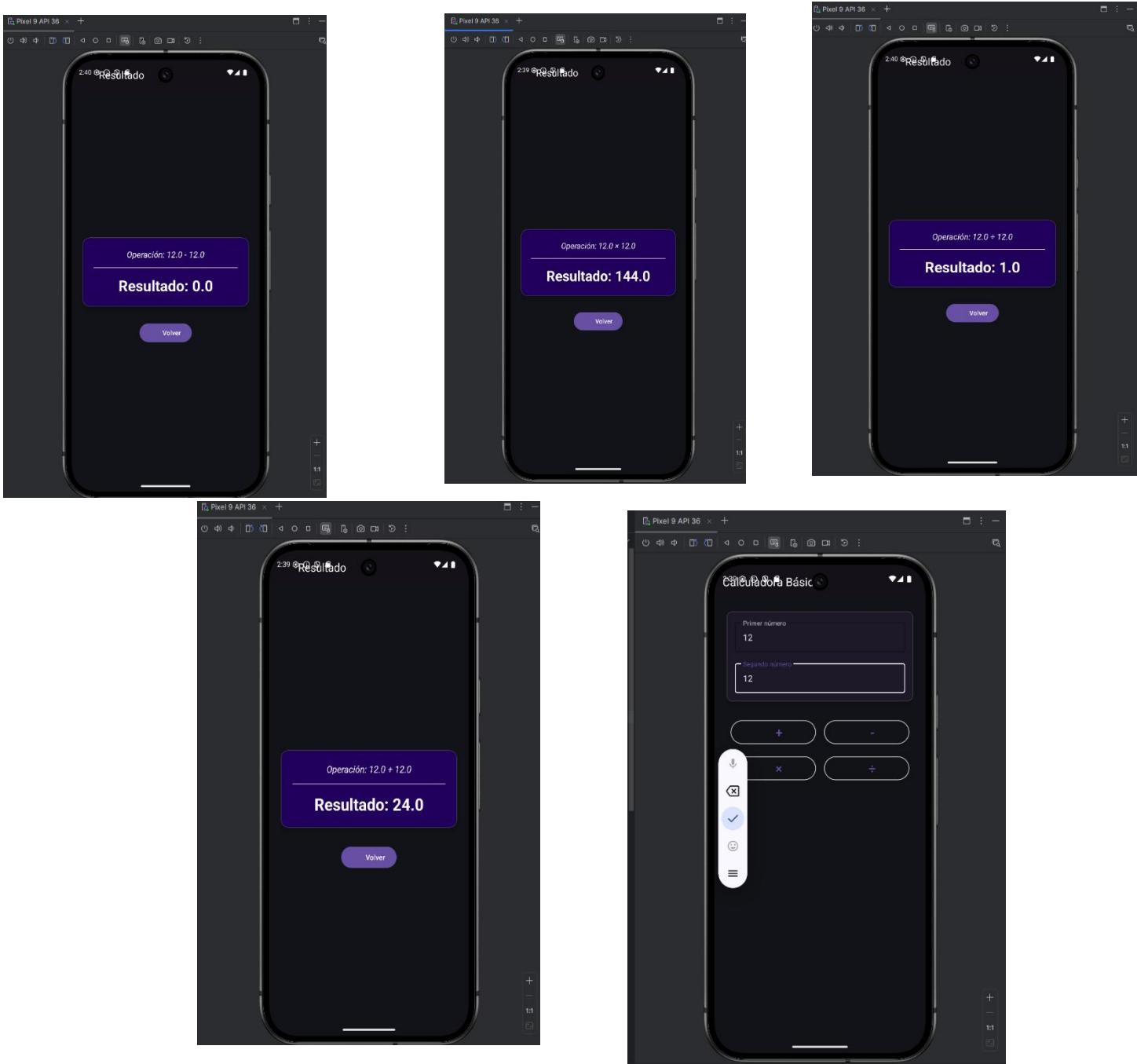
    <!-- Material 3 Color Palette -->
    <color name="m3_ref_palette_primary40">#FF6750A4</color> Overriding '@color/m3_ref_palette_primary40' which is marked as static
    <color name="m3_ref_palette_primary30">#FF4F378B</color> Overriding '@color/m3_ref_palette_primary30' which is marked as static
    <color name="m3_ref_palette_primary20">#FF21005D</color> Overriding '@color/m3_ref_palette_primary20' which is marked as static
    <color name="m3_ref_palette_secondary40">#FF625B71</color> Overriding '@color/m3_ref_palette_secondary40' which is marked as static
    <color name="m3_ref_palette_secondary30">#FF4A4458</color> Overriding '@color/m3_ref_palette_secondary30' which is marked as static
    <color name="m3_ref_palette_neutral80">#B3FFFFFF</color> Overriding '@color/m3_ref_palette_neutral80' which is marked as static
    <color name="m3_ref_palette_error20">#FFB3261E</color> Overriding '@color/m3_ref_palette_error20' which is marked as static
    <color name="m3_sys_color_dynamic_dark_background">#FF1C1B1F</color> Overriding '@color/m3_sys_color_dynamic_dark_background' which is marked as static
</resources>
```

### Archivos de iconos (ic\_\*.xml):

-*Función*: Contienen los vectores de los iconos usados en la interfaz.

-*Tipos*: Iconos para operaciones matemáticas y navegación.





### Interacción del usuario:

- **Ingreso de datos:**
  - El usuario escribe el primer número en el campo "Primer número" (un campo con borde blanco y fondo oscuro)
  - Luego escribe el segundo número en el campo "Segundo número"
- **Selección de operación:**
  - El usuario elige una de las 4 operaciones disponibles (+, -, x, ÷) haciendo clic en los botones con iconos:

- **Suma (+):** Botón con ícono  y texto "+"
- **Resta (-):** Botón con ícono  y texto "-"
- **Multiplicación (x):** Botón con ícono  y texto "x"
- **División (÷):** Botón con ícono  y texto "÷"

#### **Validaciones automáticas:**

- Si el usuario deja un campo vacío, aparece un mensaje rojo: "*Por favor ingrese ambos números*"
- Si intenta dividir por cero, aparece un mensaje: "*No se puede dividir por cero*" y bloquea la operación