

## Tarea UT03.1. Sumar números con botón programático.

Entrega una captura de tu aplicación con una suma recién completada y añade el contenido de los ficheros MainActivity.java, activity\_main.xml y colors.xml.

### Fichero MainActivity.java

```
package com.example.tarea3_1_suma;

import androidx.appcompat.app.AppCompatActivity;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.TextView;
import android.widget.Toast;

public class MainActivity extends AppCompatActivity {

    private EditText et1, et2;
    private Button btn;
    private TextView result;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        // Vinculación por ID
```

```
et1 = findViewById(R.id.etNumero1);
et2 = findViewById(R.id.etNumero2);
btn = findViewById(R.id.btnSumar);
result = findViewById(R.id.tvResultado);

// Captura de evento programática
btn.setOnClickListener(new View.OnClickListener() {

    @Override
    public void onClick(View v) {
        sumar();
    }
});

private void sumar() {
    try {
        // Obtenemos los datos para después mostrarlos
        int n1 = Integer.parseInt(et1.getText().toString());
        int n2 = Integer.parseInt(et2.getText().toString());

        int suma = n1 + n2;

        // Mostrar resultado
        result.setText(String.valueOf(suma));
    } catch (NumberFormatException e) {
        Toast.makeText(this, "Error: introduce números enteros",
        Toast.LENGTH_SHORT).show();
        result.setText("0");
    }
}
```

```
    }  
}  
}
```

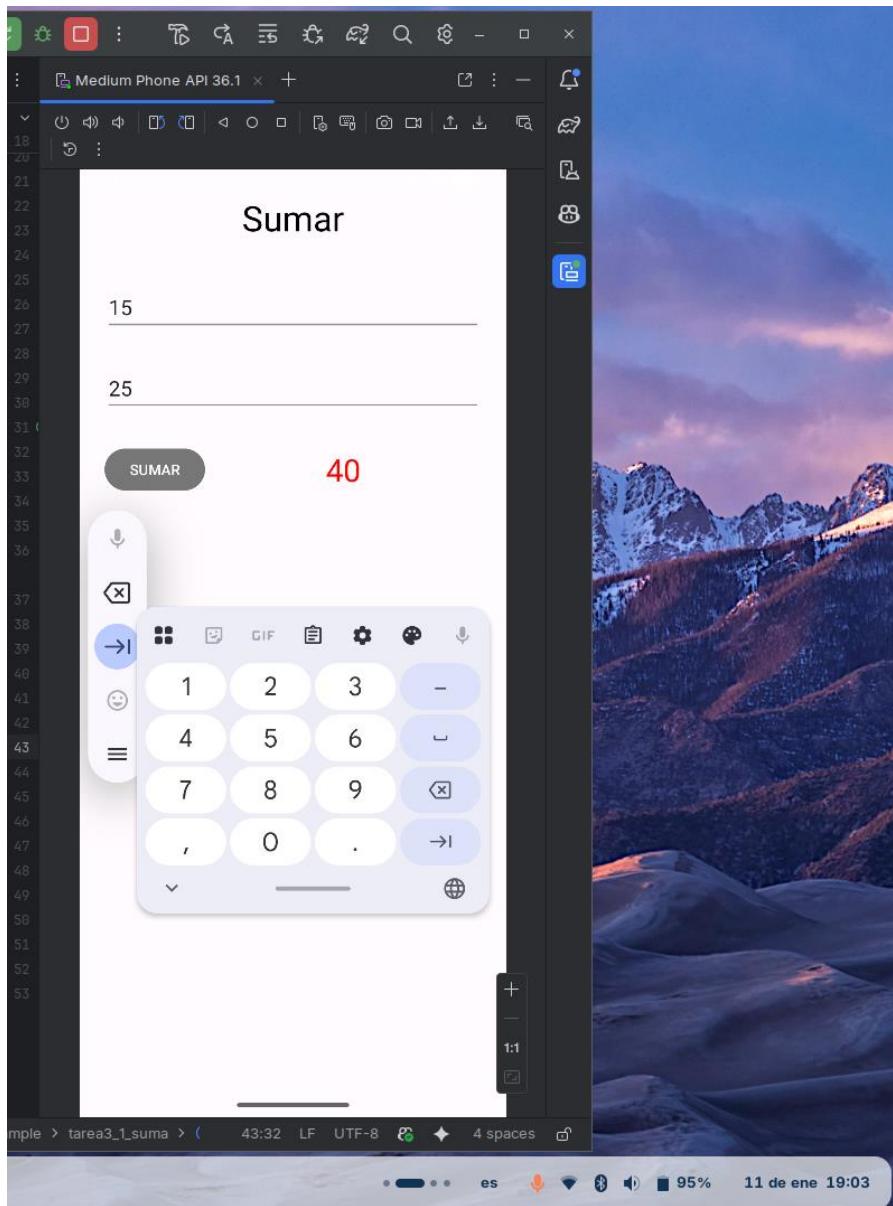
## Fichero activity\_main.xml

```
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"  
    android:layout_width="match_parent" android:layout_height="match_parent" android:orientation="vertical"  
    android:padding="24dp" android:gravity="top">  
    <TextView android:layout_width="match_parent" android:layout_height="wrap_content"  
        android:text="Sumar" android:textSize="34sp" android:gravity="center"  
        android:layout_marginBottom="40dp" android:textColor="@color/negro"/>  
    <EditText android:id="@+id/etNumero1" android:layout_width="match_parent"  
        android:layout_height="wrap_content" android:inputType="number" android:textSize="20sp"  
        android:layout_marginBottom="30dp" android:backgroundTint="#888888"/>  
    <EditText android:id="@+id/etNumero2" android:layout_width="match_parent"  
        android:layout_height="wrap_content" android:inputType="number" android:textSize="20sp"  
        android:layout_marginBottom="30dp" android:backgroundTint="#888888"/>  
    <LinearLayout android:layout_width="match_parent" android:layout_height="wrap_content"  
        android:orientation="horizontal" android:gravity="center_vertical">  
        <Button android:id="@+id/btnSumar" android:layout_width="wrap_content"  
            android:layout_height="wrap_content" android:text="SUMAR" android:backgroundTint="#777777"  
            android:textColor="@color/blanco"/>  
        <TextView android:id="@+id/tvResultado" android:layout_width="match_parent"  
            android:layout_height="wrap_content" android:text="0" android:textSize="30sp"  
            android:textColor="@color/rojo" android:gravity="center"/>  
    </LinearLayout>  
</LinearLayout>
```

## Fichero colors.xml

```
<resources>  
    <color name="negro">#FF000000</color>  
    <color name="blanco">#FFFFFF</color>  
    <color name="violeta">#FFBB86FC</color>  
    <color name="rojo">#FF0000</color>  
</resources>
```

## Captura de pantalla de la suma completada



## Tarea UT03.2. Calculadora sencilla utilizando RadioGroup y RadioButton sin botón.

Entrega una captura de tu aplicación dónde se vea el nombre de la aplicación con una operación recién completada y añade el contenido de los ficheros MainActivity.java, activity\_main.xml y colors.xml.

### Fichero MainActivity.java

```
package com.example.calculadora_radiob_luisrodrigomartineztabernerero;

import androidx.appcompat.app.AppCompatActivity;
import android.os.Bundle;
import android.widget.EditText;
import android.widget.RadioGroup;
import android.widget.TextView;
import android.widget.Toast;

public class MainActivity extends AppCompatActivity {

    private EditText et1, et2;
    private RadioGroup rgGrupo;
    private TextView tvRes;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        et1 = findViewById(R.id.etOperando1);
        et2 = findViewById(R.id.etOperando2);
        rgGrupo = findViewById(R.id.rgOperaciones);
        tvRes = findViewById(R.id.tvResultado);

        rgGrupo.setOnCheckedChangeListener(new
RadioGroup.OnCheckedChangeListener() {

    @Override
    public void onCheckedChanged(RadioGroup group, int checkedId) {
```

```
        calcular(checkedId);
    }
});
```

//Vamos a calcular cuando se seleccione una operación. Np es la mejor UX, pero es lo que se pide.

```
private void calcular(int idSeleccionado) {
    String txt1 = et1.getText().toString();
    String txt2 = et2.getText().toString();

    if (txt1.isEmpty() || txt2.isEmpty()) {
        Toast.makeText(this, "Faltan números",
        Toast.LENGTH_SHORT).show();
        rgGrupo.setOnCheckedChangeListener(null);
        rgGrupo.clearCheck();
        rgGrupo.setOnCheckedChangeListener(new
RadioGroup.OnCheckedChangeListener() {
            @Override
            public void onCheckedChanged(RadioGroup group, int checkedId) {
                calcular(checkedId);
            }
        });
        return;
    }

    try {
        double n1 = Double.parseDouble(txt1);
        double n2 = Double.parseDouble(txt2);
        double resultado = 0;
```

```

if (idSeleccionado == R.id.rbSuma) {
    resultado = n1 + n2;
} else if (idSeleccionado == R.id.rbResta) {
    resultado = n1 - n2;
} else if (idSeleccionado == R.id.rbMulti) {
    resultado = n1 * n2;
} else if (idSeleccionado == R.id.rbDiv) {
    if (n2 == 0) {
        tvRes.setText("Error: Div/0");
        Toast.makeText(this, "Cuidado no se puede dividir por 0. Cambie el operando 2", Toast.LENGTH_SHORT).show();
    }
    resultado = n1 / n2;
}

tvRes.setText(String.format("%.2f", resultado));

} catch (NumberFormatException e) {
    tvRes.setText("Error");
    Toast.makeText(this, "Formato numérico incorrecto",
    Toast.LENGTH_SHORT).show();
}
}

```

### Fichero activity\_main.xml

```

<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent" android:layout_height="match_parent" android:orientation="vertical"
    android:padding="20dp">

```

```

<Space android:layout_width="match_parent" android:layout_height="30dp"/>
<TextView android:layout_width="wrap_content" android:layout_height="wrap_content"
    android:text="Calculadora" android:textSize="34sp" android:layout_gravity="center"
    android:textColor="#444" android:layout_marginBottom="30dp"/>
<EditText android:id="@+id/etOperando1" android:layout_width="match_parent"
    android:layout_height="wrap_content" android:inputType="numberDecimal|numberSigned"
    android:text="40" android:textSize="20sp" android:layout_marginBottom="10dp"/>
<EditText android:id="@+id/etOperando2" android:layout_width="match_parent"
    android:layout_height="wrap_content" android:inputType="numberDecimal|numberSigned"
    android:text="0" android:textSize="20sp" android:layout_marginBottom="20dp"/>
<RadioGroup android:id="@+id/rgOperaciones" android:layout_width="match_parent"
    android:layout_height="wrap_content" android:orientation="vertical">
    <RadioButton android:id="@+id/rbSuma" android:layout_width="wrap_content"
        android:layout_height="wrap_content" android:text="@string/rb_sumar" android:textSize="16sp"/>
    <RadioButton android:id="@+id/rbResta" android:layout_width="wrap_content"
        android:layout_height="wrap_content" android:text="@string/rb_restar" android:textSize="16sp"/>
    <RadioButton android:id="@+id/rbMulti" android:layout_width="wrap_content"
        android:layout_height="wrap_content" android:text="@string/rb_multiplicar" android:textSize="16sp"/>
    <RadioButton android:id="@+id/rbDiv" android:layout_width="wrap_content"
        android:layout_height="wrap_content" android:text="@string/rb_dividir" android:textSize="16sp"/>
</RadioGroup>
<TextView android:id="@+id/tvResultado" android:layout_width="match_parent"
    android:layout_height="wrap_content" android:text="0.0" android:textSize="40sp"
    android:textColor="@color/rojo" android:gravity="center" android:layout_marginTop="30dp"/>
</LinearLayout>

```

## Fichero colors.xml

```

<resources>
    <color name="black">#FF000000</color>
    <color name="white">#FFFFFF</color>
    <color name="violeta">#FF6200EE</color>
    <color name="rojo">#FF0000</color>
</resources>

```

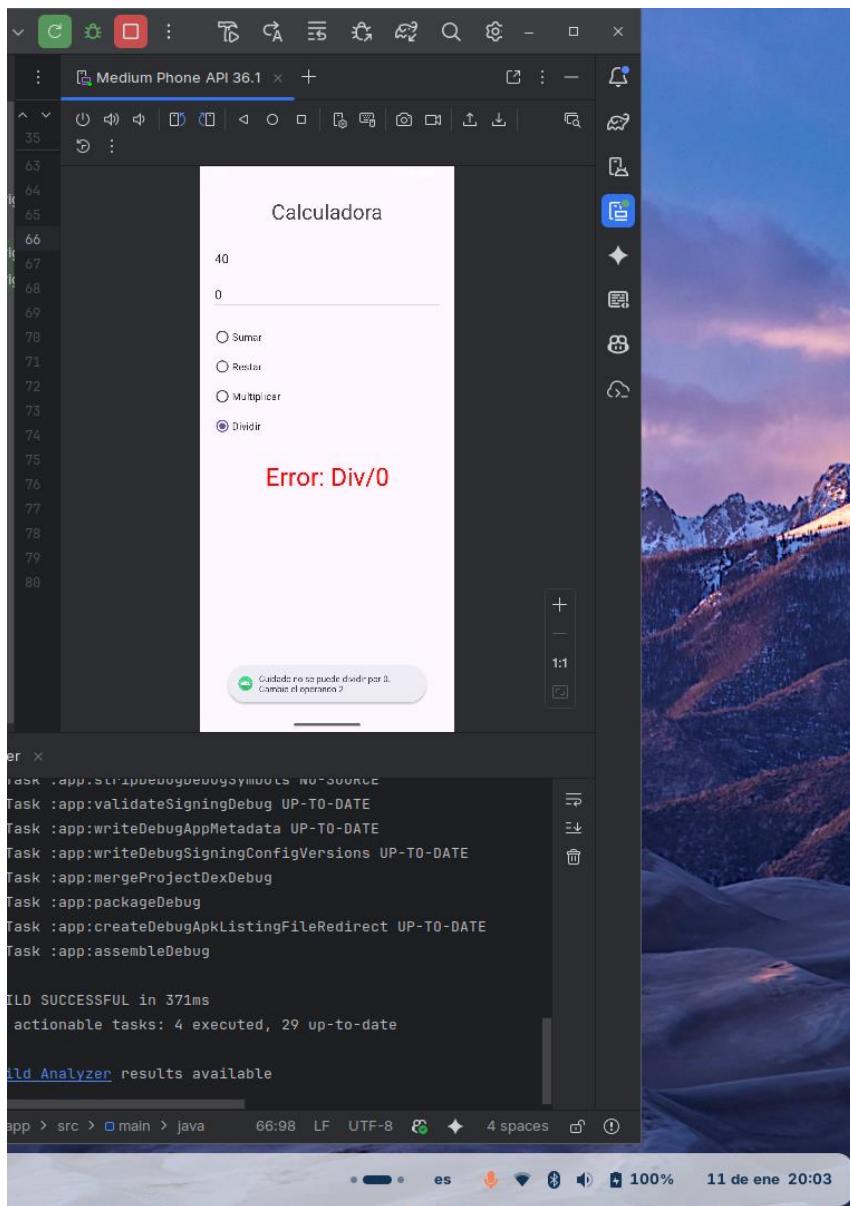
## Fichero strings.xml

```

<resources>
    <string name="app_name">Calculadora RadioB - Luis Rodrigo Martínez Tabernerero</string>
    <string name="rb_sumar">Sumar</string>
    <string name="rb_restar">Restar</string>
    <string name="rb_multiplicar">Multiplicar</string>
    <string name="rb_dividir">Dividir</string>
</resources>

```

## Captura de pantalla



## Tarea UT03.3. Sumar números con botón programático (v. kotlin + compose)

Entrega una captura de tu aplicación dónde se vea el nombre de la aplicación con una operación recién completada y añade el contenido del fichero `MainActivity.kt` y una explicación de la función Composable desarrollada. Sin esta explicación no será puntuado el ejercicio.

### Fichero `MainActivity.kt`

```
package com.example.tareasumakotlin

import android.os.Bundle
import androidx.activity.ComponentActivity
import androidx.activity.compose.setContent
import androidx.compose.foundation.layout.*
import androidx.compose.foundation.text.KeyboardOptions
import androidx.compose.material3.*
import androidx.compose.runtime.*
import androidx.compose.ui.Alignment
import androidx.compose.ui.Modifier
import androidx.compose.ui.graphics.Color
import androidx.compose.ui.text.input.KeyboardType
import androidx.compose.ui.tooling.preview.Preview
import androidx.compose.ui.unit.dp
import androidx.compose.ui.unit.sp

class MainActivity : ComponentActivity() {
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContent {
            MaterialTheme {
                Surface(
                    modifier = Modifier.fillMaxSize(),
                    color = MaterialTheme.colorScheme.background
                ) {
                    PantallaSuma()
                }
            }
        }
    }
}
```

```
        }

    }

}

@OptIn(ExperimentalMaterial3Api::class)
@Composable
fun PantallaSuma() {

    var sumando1 by remember { mutableStateOf("") }
    var sumando2 by remember { mutableStateOf("") }
    var resultadoTexto by remember { mutableStateOf("") }
```

```
Column(
    modifier = Modifier
        .fillMaxSize()
        .padding(16.dp),
    horizontalAlignment = Alignment.CenterHorizontally,
    verticalArrangement = Arrangement.Center
) {
```

```
// Campo de texto 1
TextField(
    value = sumando1,
    onValueChange = { sumando1 = it },
    label = { Text("Sumando 1") },
    keyboardOptions = KeyboardOptions(keyboardType =
KeyboardType.Number),
    modifier = Modifier.fillMaxWidth(),
    colors = TextFieldDefaults.colors(
        focusedContainerColor = Color(0xFFEFEFEF),
```

```
        unfocusedContainerColor = Color(0xFFEFEFEF),  
        disabledContainerColor = Color(0xFFEFEFEF)  
    )  
)
```

**Spacer(modifier = Modifier.height(16.dp))**

```
// Campo de texto 2  
  
TextField(  
    value = sumando2,  
    onValueChange = { sumando2 = it },  
    label = { Text("Sumando 2") },  
    keyboardOptions = KeyboardOptions(keyboardType =  
KeyboardType.Number),  
    modifier = Modifier.fillMaxWidth(),  
    colors = TextFieldDefaults.colors(  
        focusedContainerColor = Color(0xFFEFEFEF),  
        unfocusedContainerColor = Color(0xFFEFEFEF),  
        disabledContainerColor = Color(0xFFEFEFEF)  
    )  
)
```

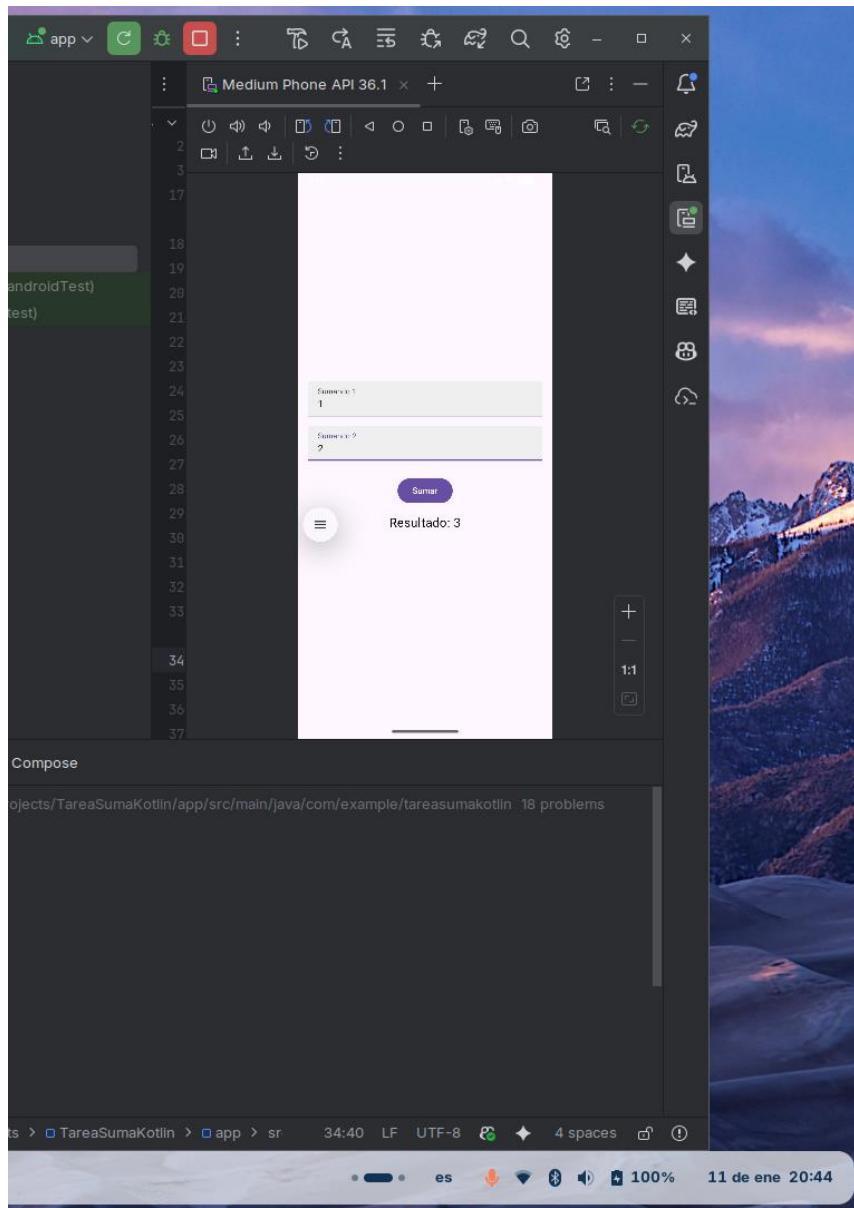
**Spacer(modifier = Modifier.height(24.dp))**

```
// Botón "Sumar"  
  
Button(  
    onClick = {  
        val n1 = sumando1.toIntOrNull() ?: 0  
        val n2 = sumando2.toIntOrNull() ?: 0  
    }
```

```
    resultadoTexto = "Resultado: ${n1 + n2}"  
},  
colors = ButtonDefaults.buttonColors(  
    containerColor = Color(0xFF6750A4)  
,  
    modifier = Modifier.wrapContentWidth()  
) {  
    Text("Sumar")  
}  
  
Spacer(modifier = Modifier.height(16.dp))  
  
// Texto del Resultado (al principio lo he escondido)  
if (resultadoTexto.isNotEmpty()) {  
    Text(  
        text = resultadoTexto,  
        fontSize = 20.sp,  
        color = Color.Black  
    )  
}  
}  
}  
}
```

```
//Composable para previsualizar la pantalla de suma  
@Preview(showBackground = true)  
@Composable  
fun PreviewSuma() {  
    PantallaSuma()  
}
```

## Captura de pantalla de la aplicación



## Explicación de la función Componible

La función Pantalla Suma la hemos hecho una función Componible. Esto quiere decir que gracias a usar Jetpack Compose podemos definir la interfaz de usuario sin necesitar usar XML, de una forma declarativa en vez de imperativa.

Esto tiene varias ventajas.

La principal es el manejo del estado mediante remember, para que los datos no se borren al redibujar los valores y mutableStateOf() para crear variables.

Podemos cambiar el estado de las variables y recomponer la interfaz con las funciones Text(), Button(), TextField()