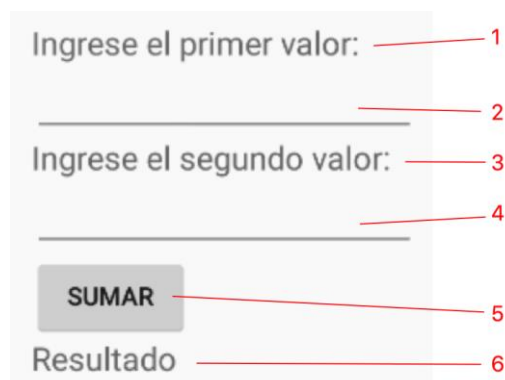


## Práctica 6: Dándole funcionalidad a los botones.

En esta práctica, agregarás a **MainActivity** un fragmento de código que permitirá responder al usuario cuando éste haga clic en un botón.

Construiremos una aplicación que permita la captura de dos números en controles de tipo **EditText**, acompañados de mensajes **TextView** que soliciten la carga de dichos números. Agregaremos un control **Button** que al hacer clic en él sume los números introducidos y presente el resultado en un tercer **TextView**. La siguiente figura muestra la interfaz que queremos crear:



- 1.- Guiándose en las prácticas anteriores, construya una nueva app de nombre **Practica6**.
- 2.- Antes de iniciar el diseño construya las siguientes strings:

Key	Default value
app_name	Sumador
text_primer_valor	Ingrese el primer valor:
text_segundo_valor	Ingrese el segundo valor:
text_resultado	Resultado
button_sumar	Sumar

- 3.- Diseñe la interfaz. Guíese en la tabla siguiente:

No.	Vista	Propiedad	Valor
1	TextView	text	@string/text_primer_valor
2	EditText	ID inputType	etNum1 number
3	TextView	text	@string/text_segundo_valor
4	EditText	ID inputType	etNum2 number
5	Button	ID text	btSumar @string/button_sumar
6	TextView	ID text	tvResultado @string/text_resultado

4.- *Declaración de variables a nivel de clase:* Abre el archivo **MainActivity.java** y declara las siguientes variables a nivel de clase:

```
package com.example.practica6;

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

public class MainActivity extends AppCompatActivity {

    private EditText etNum1, etNum2;
    private TextView tvResultado;
    private Button btSumar;

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

El código `public class MainActivity extends AppCompatActivity` indica que la clase **MainActivity** hereda de la clase **AppCompatActivity**, lo que la convierte en una actividad o “pantalla”.

En las líneas `private EditText etNum1, etNum2; private TextView tvResultado; private Button btSumar;` definimos variables en java para almacenar las referencias a las vistas definidas en el archivo XML. Declaramos cuatro variables, dos de tipo **EditText**, una de tipo **TextView** y una de tipo **Button**. Los nombres de las variables declaradas coinciden con los ID de las vistas definidas en el archivo XML, esto no es obligatorio, pero es aconsejable. Observe que no definimos variables para los otros dos "TextView" de los mensajes “Ingresa el primer valor:” e “Ingresa el segundo valor:” ya que no necesitaremos interactuar con ellos desde el código.

5.- *Enlazando las variables con los controles:* Para poder manipular los controles construidos en el **activity\_main.xml**, debemos enlazar las variables recientemente declaradas con dichos controles. Para hacer esto, modifique el método **onCreate** de la siguiente manera:

```
@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);
    etNum1 = (EditText)findViewById(R.id.etNum1);
    etNum2 = (EditText)findViewById(R.id.etNum2);
    tvResultado = (TextView)findViewById(R.id.tvResultado);
    btSumar = (Button)findViewById(R.id.btSumar);
}
```

El método **onCreate** se ejecutará al iniciarse la aplicación.

El método **onCreate** contiene la línea de código `super.onCreate(savedInstanceState);` en la cual se invoca al constructor de la clase padre **AppCompatActivity**.

El código `setContentView(R.layout.activity_main);` carga (infla) el diseño de la actividad implementado en el archivo **activity\_main.xml**.

Para facilitar el manejo de nuestra interfaz desde el código java, Android Studio crea una serie de clases autogeneradas. La clase **R** será la principal clase de este tipo y contendrá un conjunto de constantes para acceder a los recursos XML de la aplicación. Por lo tanto, con la instrucción **setContentView(R.layout.activity\_main)** estamos cargando la vista diseñada en el archivo **activity\_main.xml**.

En las líneas marcadas con amarillo podemos observar que al método **findViewById** debemos pasarle la constante creada en la clase **R**. El nombre de la constante debe ser igual al nombre de la propiedad **ID** de la vista creada en el archivo XML. Como el método **findViewById** retorna una clase de tipo **View**, entonces debemos utilizar el operador cast adecuado (es decir le antecedemos entre paréntesis el nombre de la clase).

6.- **Responder al botón Sumar:** Escriba el método **realizarSuma** una línea antes de la llave de cierre de la clase **MainActivity**:

```
public void realizarSuma(View view) {  
    int num1 = Integer.parseInt(etNum1.getText().toString());  
    int num2 = Integer.parseInt(etNum2.getText().toString());  
    int resultado = num1 + num2;  
    tvResultado.setText(String.valueOf(resultado));  
}
```

En las dos primeras líneas del método **realizarSuma**, extraemos el texto de los dos controles **EditText**, los convertimos a enteros y los almacenamos en las variables **num1** y **num2** de tipo **int**. Enseguida realizamos la suma y la almacenamos en la variable **resultado** de tipo **int**. Por último, convertimos **resultado** en **String** colocándolo en **tvResultado** para que muestre el resultado de la suma.

Toma nota de los detalles de este método que se requieren para que el sistema lo reconozca como compatible con el atributo **android:onClick**. En concreto, el método debe declarar lo siguiente:

- acceso público;
- un valor de retorno vacío;
- una **View** como único parámetro (es el objeto **View** en el que se hizo clic).

7.- Regresa al archivo **activity\_main.xml** para llamar a este método desde el botón:

- a) Haz clic para seleccionar el botón en el editor de diseño.
- b) En la ventana **Atributes**, encuentra la propiedad **onClick** y selecciona **realizarSuma** en la lista desplegable.
- c) Ahora, cuando se presione el botón, el sistema llamará al método **realizarSuma()**.

8.- Graba las modificaciones y ejecuta la app:

