

# JButton

El tercer control visual de uso muy común es el que provee la clase JButton. Este control visual muestra un botón.

El proceso para añadir botones a un control JFrame es similar a añadir controles de tipo JLabel.

Ahora veremos la captura de eventos con los controles visuales. Uno de los eventos más comunes es cuando hacemos clic sobre un botón.

Java implementa el concepto de interfaces para poder llamar a métodos de una clase existente a una clase desarrollada por nosotros.

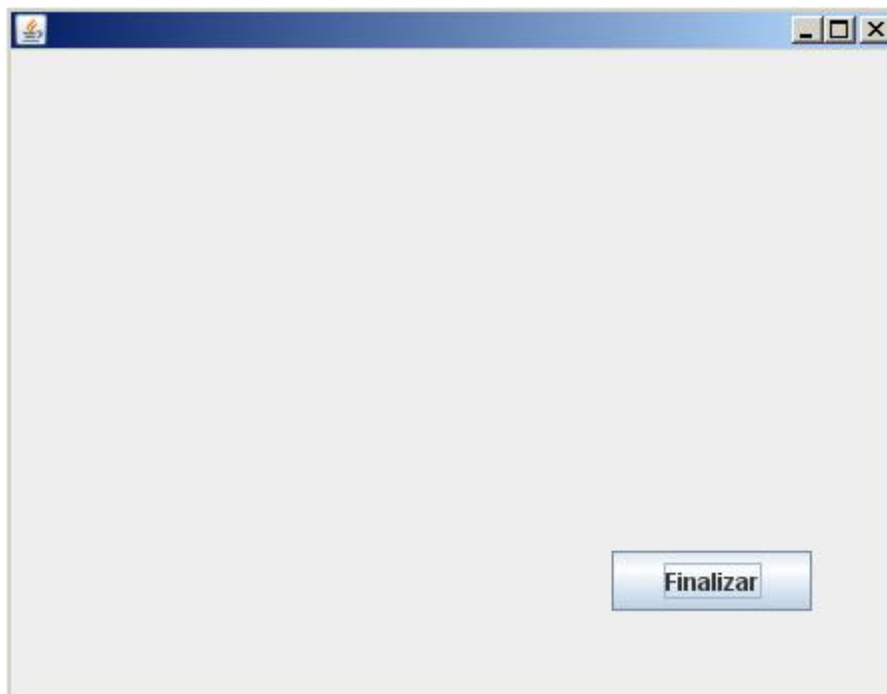
**Programa 1: Confeccionar una ventana que muestre un botón. Cuando se presione finalizar la ejecución del programa Java.**

```
import javax.swing.*;
import java.awt.event.*;
public class Formulario extends JFrame implements ActionListener {
    JButton boton1;
    public Formulario() {
        setLayout(null);
        boton1=new JButton("Finalizar");
        boton1.setBounds(300,250,100,30);
        add(boton1);
        boton1.addActionListener(this);
    }

    public void actionPerformed(ActionEvent e) {
        if (e.getSource()==boton1) {
            System.exit(0);
        }
    }

    public static void main(String[] ar) {
        Formulario formulario1=new Formulario();
        formulario1.setBounds(0,0,450,350);
        formulario1.setVisible(true);
        formulario1.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    }
}
```

**Salida:**



## Bibliografía:

- Extraído de [<<0 - Swing - JButton>>](#). Consultado el 28 de noviembre del 2022