

# Métodos Avanzados de Programación Científica y Computación

M<sup>a</sup> Luisa Díez Platas

## Laboratorio. Interfaces de usuario

# Buenas prácticas para crear una ventana

- Si es un `JFrame`.
  - Se recomienda implementar una clase que se derive de `JFrame`.
  - Si se van a programar eventos, que implemente también la interfaz `ActionListener`.
- Se debe implementar :
  - Un constructor
  - Una función para definir el tamaño y otras características de la Ventana
  - Opcional pero interesante un método para inserción de componentes. Si no están determinados se pueden buscar más opciones.
    - Plantearse el uso de paneles para agrupar elementos
    - Usar administradores de disposición

# FlowLayout

```
import java.awt.*;

import javax.swing.*;

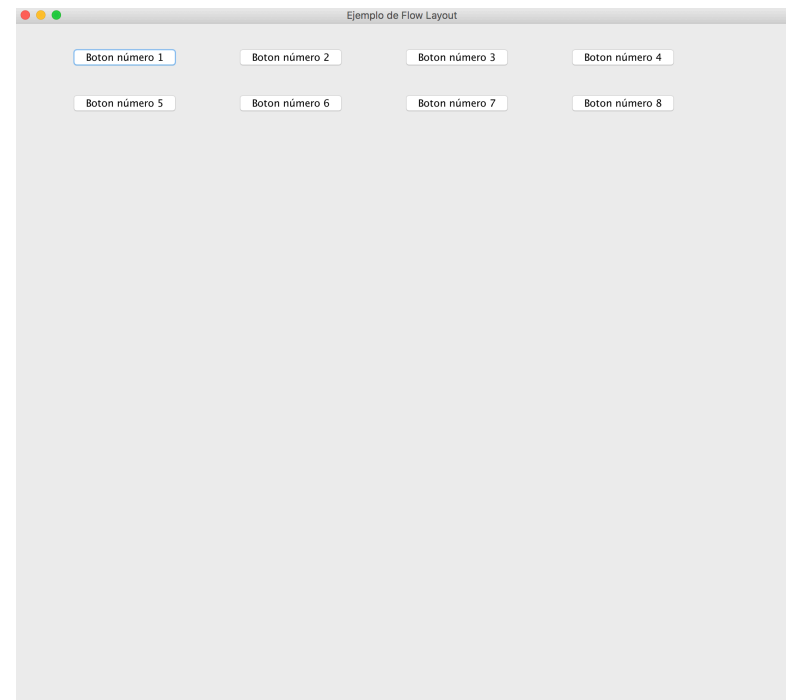
/**
 *
 * @author mluisadiez
 */
public class FlowLayoutProject extends JFrame{

    /**
     * @param args the command line arguments
     */
    public FlowLayoutProject() {
        super("Ejemplo de Flow Layout");

        setLayout(new FlowLayout(FlowLayout.LEFT,70,30));
        for (int i=1; i<=8; i++){
            add(new JButton("Boton número "+i));
        }

        setSize(1000,1000); //pack();
        setVisible(true);
        setDefaultCloseOperation(EXIT_ON_CLOSE);
    }

    public static void main(String[] args){
        FlowLayoutProject bl=new FlowLayoutProject();
    }
}
```



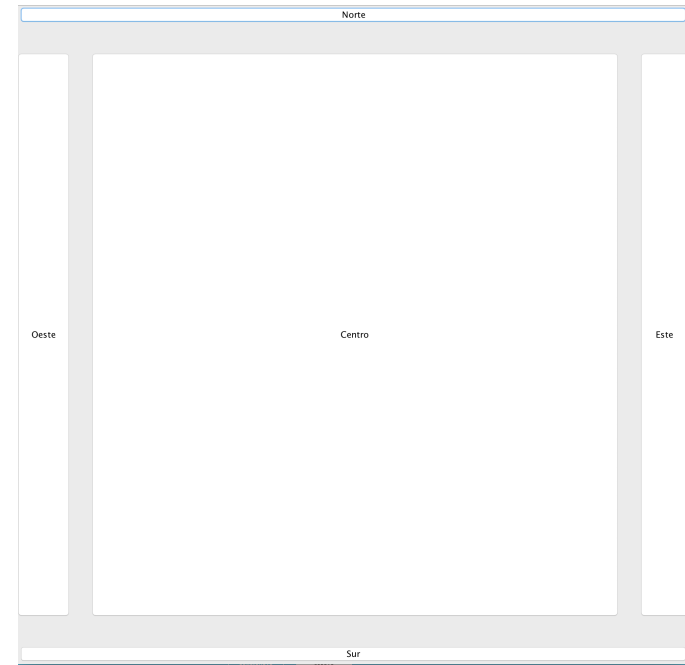
# BorderLayout

```
import javax.swing.*;
import java.awt.*;
public class BorderLayoutProject extends JFrame{

    public BorderLayoutProject() {
        super("Ejemplo de Border Layout");

        setLayout(new BorderLayout(30, 40));
        add(new JButton("Este"), BorderLayout.EAST);
        add(new JButton("Sur"), BorderLayout.SOUTH);
        add(new JButton("Oeste"), BorderLayout.WEST);
        add(new JButton("Norte"), BorderLayout.NORTH);
        add(new JButton("Centro"), BorderLayout.CENTER);
        setSize(1000,1000); //pack();
        setVisible(true);
        setDefaultCloseOperation(EXIT_ON_CLOSE);
    }

    public static void main(String[] args){
        BorderLayoutProject bl=new BorderLayoutProject();
    }
}
```



# GridLayout

```
import java.awt.*;

import javax.swing.*;
import static javax.swing.JFrame.EXIT_ON_CLOSE;

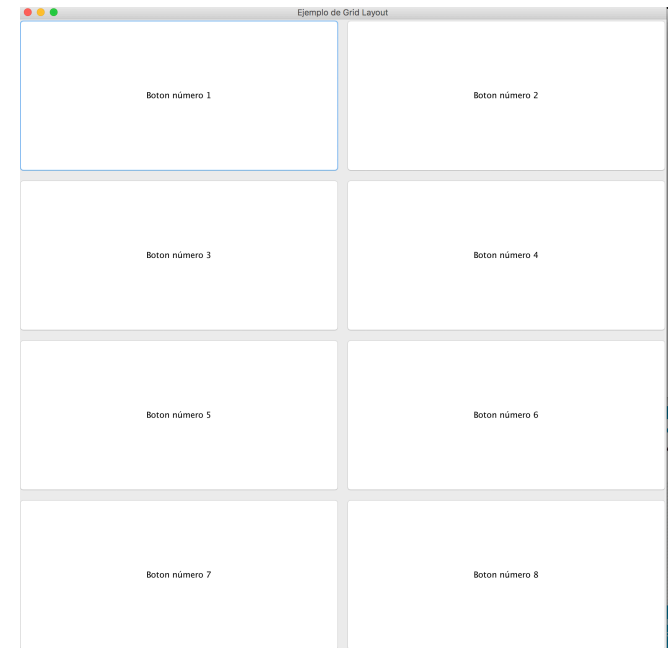
/**
 *
 * @author mluisadiez
 */
public class GridLayoutProject extends JFrame{

    /**
     * @param args the command line arguments
     */
    public GridLayoutProject() {
        super("Ejemplo de Grid Layout");

        setLayout(new GridLayout(4,3,10,10));
        for (int i=1; i<=8; i++){
            add(new JButton("Boton número "+i));
        }

        setSize(1000,1000); //pack();
        setVisible(true);
        setDefaultCloseOperation(EXIT_ON_CLOSE);
    }

    public static void main(String[] args){
        GridLayoutProject bl=new GridLayoutProject();
    }
}
```



# Uso de Paneles como contenedores

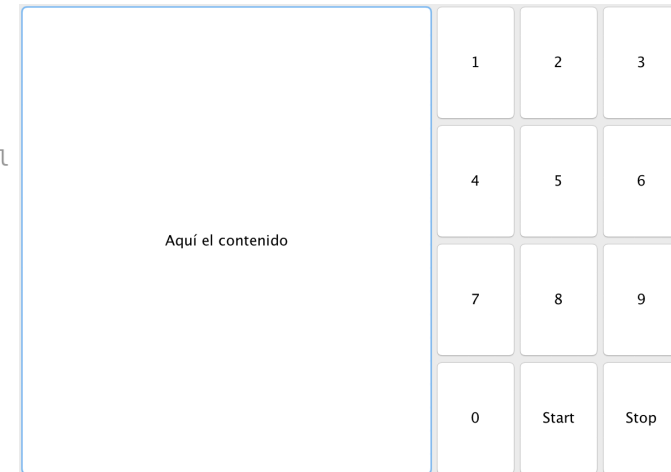
```
package panelcontenedor;
import java.awt.*;
import javax.swing.*;
import static javax.swing.JFrame.EXIT_ON_CLOSE;

/**
 *
 * A partir de Daniel Lang, 2001. "Introduction to Java Programming" Prentice Hall
 *
 */
public class PanelContenedor extends JFrame {

    public PanelContenedor(){
        setTitle("Prueba contenedor ");
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        setLayout(new BorderLayout());
        //Se crea un panel
        JPanel panel1=new JPanel();
        panel1.setLayout(new GridLayout(4,3));
        //se añaden los botones
        for (int i=1;i<=9;i++){
            panel1.add(new JButton(""+i));
        }
        panel1.add(new JButton(""+0));
        panel1.add(new JButton("Start"));
        panel1.add(new JButton("Stop"));
        // se crea otro panel
        JPanel panel2=new JPanel();
        panel2.setLayout(new BorderLayout());

        //se añaden los botones
        panel2.add(panel1,BorderLayout.CENTER);
        add(panel2,BorderLayout.EAST);
        add(new JButton("Aquí el contenido"),BorderLayout.CENTER);
        setSize(600,450);
        setVisible(true);
    }

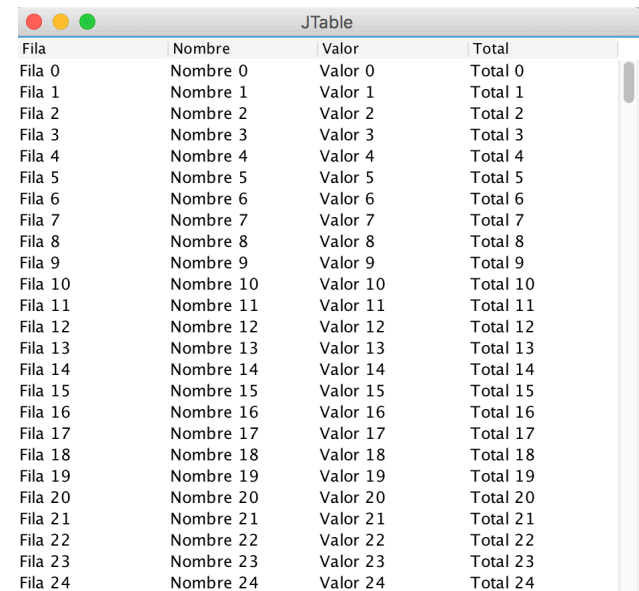
    public static void main(String[] args) {
        // TODO code application logic here
        PanelContenedor panelc=new PanelContenedor();
    }
}
```



# Tablas

- Las tablas ayudan a presentar los datos organizados

```
public class Tabla extends JFrame {  
    private JTable tabla = null;  
    DefaultTableModel modelo = null;  
    JScrollPane scroll = null;  
  
    public Tabla() {  
        String[] columnas = {"Fila", "Nombre", "Valor", "Total"};  
        tabla = new JTable();  
        modelo = new DefaultTableModel();  
        scroll = new JScrollPane(tabla);  
  
        // Parametros de la ventana  
        this.setTitle("JTable");  
        this.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);  
        this.setLayout(new BorderLayout());  
  
        // Modelo de la tabla  
        modelo.setColumnIdentifiers(columnas);  
  
        // Barras de desplazamiento  
        scroll.setHorizontalScrollBarPolicy(JScrollPane.HORIZONTAL_SCROLLBAR_AS_NEEDED);  
        scroll.setVerticalScrollBarPolicy(JScrollPane.VERTICAL_SCROLLBAR_AS_NEEDED);  
  
        // Propiedades de la tabla  
        tabla.setAutoResizeMode(JTable.AUTO_RESIZE_ALL_COLUMNS);  
        tabla.setFillsViewportHeight(true);  
  
        tabla.setModel(modelo);  
  
        // Se rellena la tabla  
        this.insertarDatos(modelo);  
        this.getContentPane().add(scroll, BorderLayout.NORTH);  
        this.pack();  
    }  
}
```



Fila	Nombre	Valor	Total
Fila 0	Nombre 0	Valor 0	Total 0
Fila 1	Nombre 1	Valor 1	Total 1
Fila 2	Nombre 2	Valor 2	Total 2
Fila 3	Nombre 3	Valor 3	Total 3
Fila 4	Nombre 4	Valor 4	Total 4
Fila 5	Nombre 5	Valor 5	Total 5
Fila 6	Nombre 6	Valor 6	Total 6
Fila 7	Nombre 7	Valor 7	Total 7
Fila 8	Nombre 8	Valor 8	Total 8
Fila 9	Nombre 9	Valor 9	Total 9
Fila 10	Nombre 10	Valor 10	Total 10
Fila 11	Nombre 11	Valor 11	Total 11
Fila 12	Nombre 12	Valor 12	Total 12
Fila 13	Nombre 13	Valor 13	Total 13
Fila 14	Nombre 14	Valor 14	Total 14
Fila 15	Nombre 15	Valor 15	Total 15
Fila 16	Nombre 16	Valor 16	Total 16
Fila 17	Nombre 17	Valor 17	Total 17
Fila 18	Nombre 18	Valor 18	Total 18
Fila 19	Nombre 19	Valor 19	Total 19
Fila 20	Nombre 20	Valor 20	Total 20
Fila 21	Nombre 21	Valor 21	Total 21
Fila 22	Nombre 22	Valor 22	Total 22
Fila 23	Nombre 23	Valor 23	Total 23
Fila 24	Nombre 24	Valor 24	Total 24

# Eventos

```
public class PruebaFlowLayout implements ActionListener{

    /**
     * @param args the command line arguments
     */
    private JPanel panelFlow;
    private JButton boton1, boton2, boton3, botonPanelFlow;

    public PruebaFlowLayout(){

        JFrame frame = new JFrame("Uso de JPanel, JButton y eventos en Java");
        frame.setLayout(null);

        flowJPanel(); //invocamos los metodos que contienen los paneles

        botonPanelFlow= new JButton("Botón para mostrar u ocultar un panel");

        panelFlow.setBounds(10, 10, 200, 200);

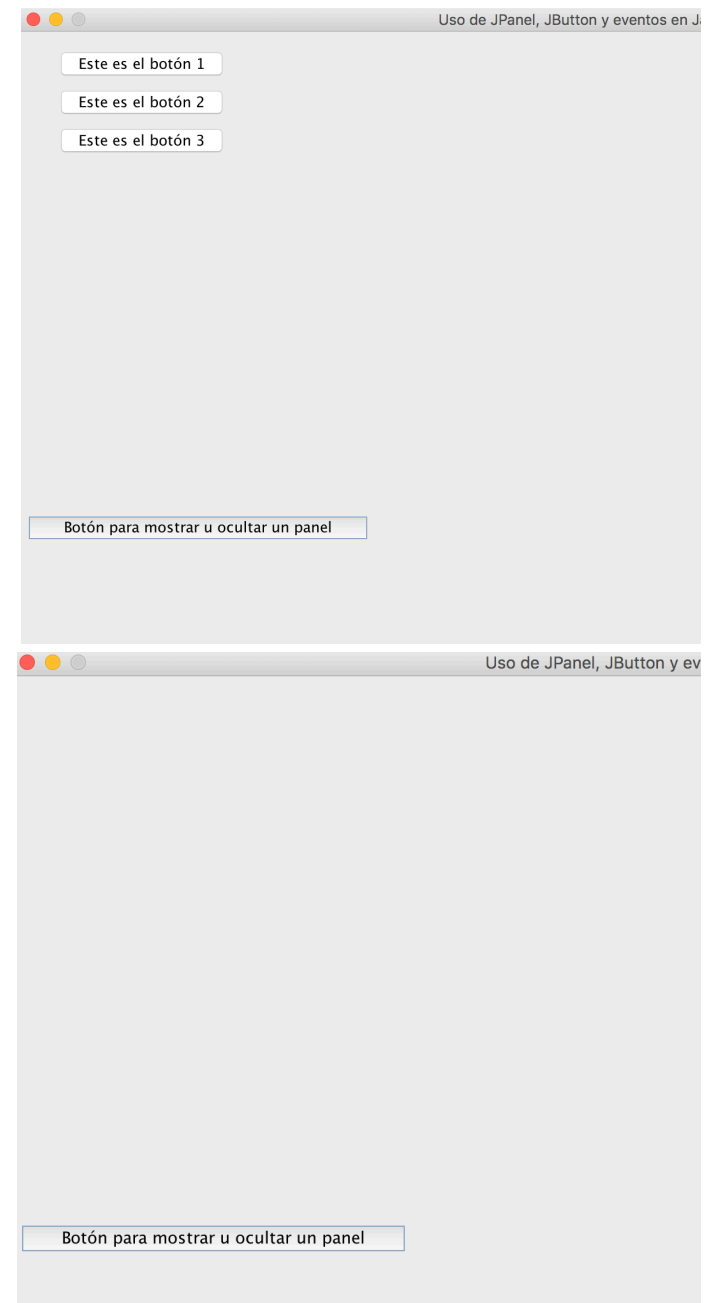
        botonPanelFlow.setBounds(10, 430, 300, 20);

        frame.add(panelFlow);
        frame.add(botonPanelFlow);

        botonPanelFlow.addActionListener(this);

        frame.setLocation(100, 50);
        frame.setResizable(false);
        frame.setVisible(true);
        frame.setSize(1000, 600);
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

    }
}
```





# Eventos

```
public void flowJPanel(){
    panelFlow= new JPanel(new FlowLayout());

    boton1= new JButton("Este es el botón 1"); boton2= new JButton("Este es el botón 2"); boton3= new JButton("Este es el botón 3")

    panelFlow.add(boton1); panelFlow.add(boton2); panelFlow.add(boton3); //añadimos los objetos al jpanel

    boton1.addActionListener(this); boton2.addActionListener(this); boton3.addActionListener(this);

    panelFlow.setVisible(true);
}

@Override
public void actionPerformed(ActionEvent e) { //sobreescribimos el metodo del listener

    if(e.getSource() == botonPanelFlow){
        if(panelFlow.isVisible()){
            panelFlow.setVisible(false);
        }else
            panelFlow.setVisible(true);
    }else{
        JOptionPane.showMessageDialog(null, e.getActionCommand());
    }

}

public static void main(String[] args) {
    // TODO code application logic here
    PruebaFlowLayout gj = new PruebaFlowLayout(); //uso de constructor para la ventana
}
```

UNIVERSIDAD  
INTERNACIONAL  
DE LA RIOJA

**unir**

[www.unir.net](http://www.unir.net)