



Sistemas Multimedia

SWING: Interfaces de usuario

Índice

- ▶ Jerarquía
- ▶ Componentes
- ▶ Contenedores
- ▶ Manejadores de disposición
- ▶ Menú

1000



Ejemplo base

Aplicación y frame



```
import java.awt.*;

public class MiAplicacion {

    public static void main (String args[]){
        JFrame ventana = new JFrame();
        ventana.setSize(500,500);
        ventana.setVisible(true);
    }
}
```

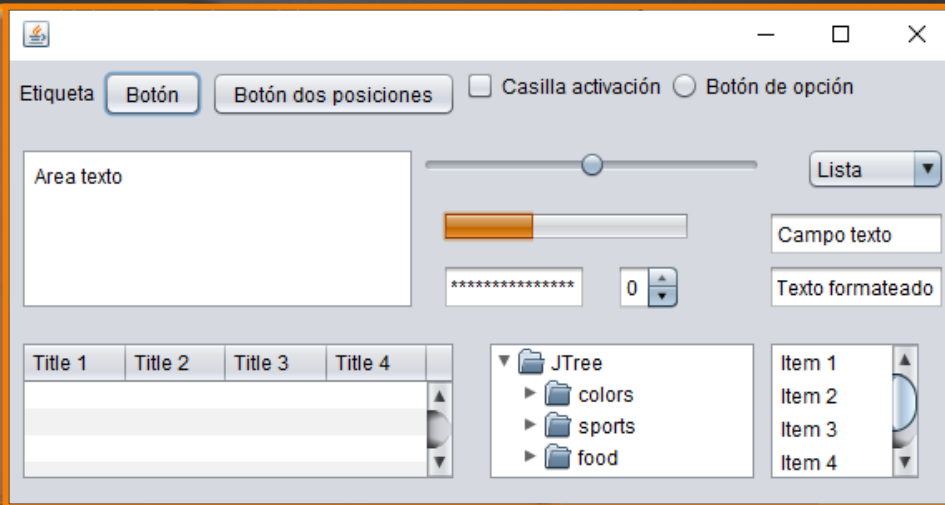
```
class MiFrame extends JFrame {
    public MiFrame() {
        this.setSize(500,500);
    }
}

public class MiAplicacion {
    public static void main (String args[]) {
        MiFrame ventana = new MiFrame();
        ventana.setVisible(true);
    }
}
```


Componentes

- ▶ Objetos asociados a elementos “visuales” de la aplicación.

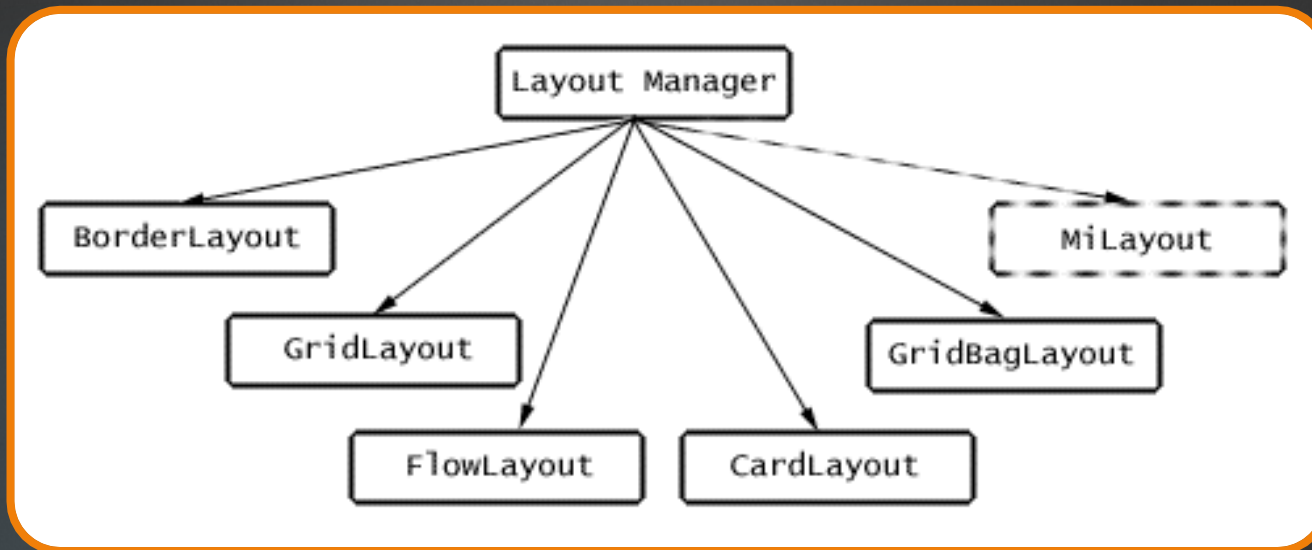
- Etiquetas
- Botones
- Casillas activación
- Listas desplegables
- Campos de texto
- Áreas de texto
-



```
Button b = new Button("Botón");  
this.add(b);
```

Manejadores disposición

- Determinan en qué posición y con qué tamaño se van a distribuir los componentes dentro del contenedor

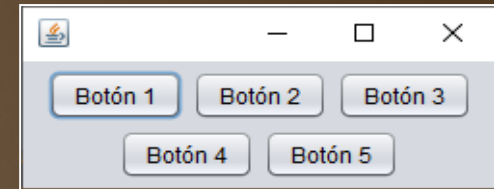


```
FlowLayout layout = new FlowLayout();  
this.setLayout(layout);
```

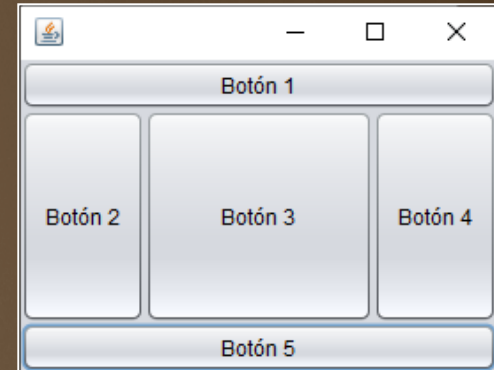
Manejadores disposición

Ejemplos

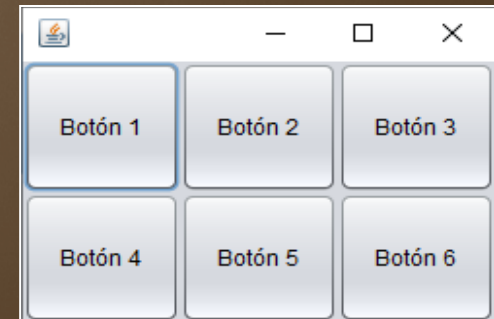
FlowLayout. Distribuye los componentes de izquierda a derecha horizontalmente, mientras quepan en la línea, antes de pasar a la línea siguiente. El tamaño del componente será el predeterminado.



BorderLayout. Distribuye los componentes en cinco zonas distintas: norte, sur, este, oeste y centro. En cada zona solo habrá un componente (visible) y este ocupará toda la zona.



GridLayout. Distribuye los componentes en una matriz MxN en la que todas las celdas tienen el mismo tamaño. El componente ocupará toda la celda.



Manejadores disposición

Y si lo anulamos...

- ▶ Es posible anular el efecto del `LayoutManager` e indicar directamente la posición y tamaño de los componentes. Para ello estableceremos:

```
setLayout (null) ;
```

y para cada objeto que queramos añadir indicaremos su situación y tamaño mediante una llamada al método

```
setBounds (x,y,ancho,alto)
```


Contenedores

- ▶ Un contenedor es un objeto que puede contener a otros componentes. Un contenedor es a su vez un componente, por lo que puede estar contenido en otro contenedor
- ▶ Ejemplos de contenedores:
 - Marco (Frame)
 - Ventana (Window)
 - Dialogo
 - Panel
- ▶ Asociado a un contenedor hay un manejador de disposición

```
JPanel panel = new JPanel();  
panel.setLayout(...);  
panel.add(...);
```

Menús

