

• Administradores de Disposición – Layouts

Los componentes se agregan al contenedor con el método `add()`.

```
JButton unBoton = new JButton("Texto del botón");  
panel.add(unBoton);
```

El efecto de `add()` depende del esquema de colocación o disposición (**layout**) del contenedor que se use.

Existen diversos esquemas de disposición: `FlowLayout`, `BorderLayout`, `GridLayout`, ...

Los objetos contenedores se apoyan en objetos `LayoutManager` (administradores de disposición).

Clases más usadas que implementa la interfaz `LayoutManager`:

FlowLayout: un componente tras otro de izquierda a derecha.

BorderLayout: 5 regiones en el contenedor (**North**, **South**, ...).

GridLayout: contenedor en **filas y columnas**.

Para organizar el contenedor se utiliza el método `setLayout()`:

```
public void setLayout(LayoutManager mgr)
```

↑
Crea una disposición para el contenedor actual

↑
Tipo de disposición (`BorderLayout`, ...)

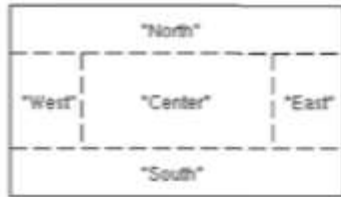
```
setLayout(new BorderLayout());
```

```
setLayout(new FlowLayout());
```

```
setLayout(new GridLayout(3,4));
```

El **layout manager** elige la mejor posición y tamaño de cada componente de acuerdo al espacio disponible.

BorderLayout organiza el contenedor en 5 zonas:



FlowLayout coloca los componentes de izquierda a derecha y de arriba hacia abajo:



Para distribuciones más complejas podemos insertar paneles (JPanel) en los contenedores y obtener el tamaño de un componente con el método `getSize()`.

• FlowLayout

Organiza los componentes en un flujo de izquierda a derecha.

Constructores:

- **FlowLayout()** Constructor básico sin parámetros, con valores predeterminados de alineación centrada y 5 unidades de espacio horizontal y vertical.
- **FlowLayout(int align)** Construye un nuevo FlowLayout con la alineación "align" especificada y un valor predeterminado de 5 unidades de espacio horizontal y vertical.
- **FlowLayout(int align, int hgap, int vgap)** Construye un nuevo FlowLayout con la alineación indicada y los espacios horizontales y verticales indicados.

```
import javax.swing.*;
import java.awt.*;

public class Gui02 extends JFrame {

    public Gui02() {
        super("Ejemplo de Layout");
        // Configurar componentes ;
        // Configurar layout ;
        setLayout(new FlowLayout(FlowLayout.LEFT, 10, 20));
        for(int i = 1; i <= 10; i++)
            add(new JButton("Componente " + i));
        setSize(200,200); //pack();
        setVisible(true);
        setDefaultCloseOperation(EXIT_ON_CLOSE);
    }

    public static void main(String args[]) {
        Gui02 aplicacion = new Gui02();
    }
}
```



Cambiando el tamaño se redistribuyen los componentes

• BorderLayout

Divide el contenedor en 5 partes

Constructores

- **BorderLayout()** Construye un gestor BorderLayout y sin espacios entre los componentes.
- **BorderLayout(int hgap, int vgap)** Construye un diseño BorderLayout con los espacios entre los componentes especificados. La distancia horizontal se especifica mediante hgap y la distancia vertical se especifica por vgap.

```
import javax.swing.*;
import java.awt.*;

public class Gui03 extends JFrame {

    public Gui03() {
        super("Ejemplo de Layout");
        // BorderLayout
        setLayout(new BorderLayout(5, 10));
        add(new JButton("1"), BorderLayout.EAST);
        add(new JButton("2"), BorderLayout.SOUTH);
        add(new JButton("3"), BorderLayout.WEST);
        add(new JButton("4"), BorderLayout.NORTH);
        add(new JButton("5"), BorderLayout.CENTER);
        setSize(200,200); //pack();
        setVisible(true);
        setDefaultCloseOperation(EXIT_ON_CLOSE);
    }
}
```



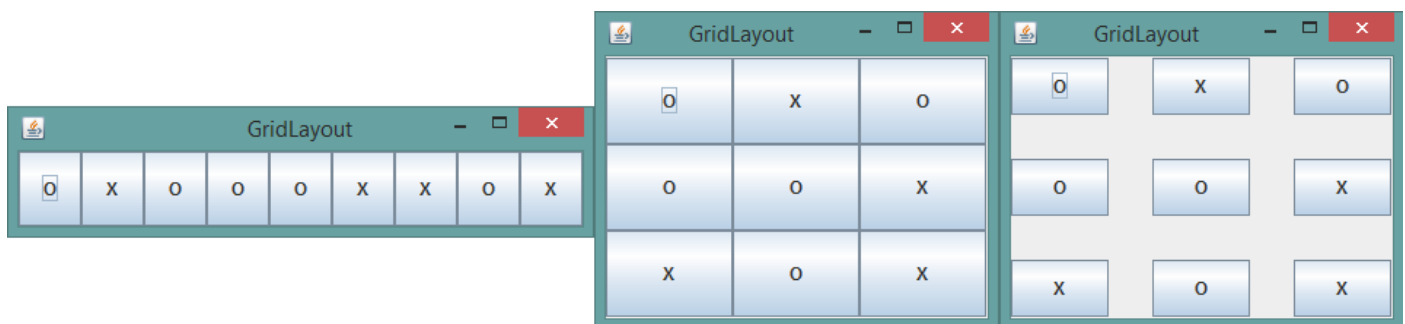
• GridLayout

Establece los componentes de un contenedor en una **rejilla rectangular**. El recipiente se divide en rectángulos de igual tamaño, y un componente se va colocando en cada rectángulo.

Constructores:

- **GridLayout()** Crea un diseño de cuadrícula por defecto de una columna por componente, en una sola fila.
- **GridLayout(int rows, int cols)** Crea una rejilla con el numero de filas(rows) y el numero de columnas(cols) que indiquemos.
- **GridLayout()(int rows, int cols, int hgap, int vgap)** Idem anterior con la salvedad que podemos indicar la distancia en pixels entre los componente de manera horizontal (hgap) y de manera vertical (vgap).

Podemos ver la diferencia entre los 3 constructores, el 1º pone todos los componentes en una sola fila y comluna, el 2º normal, y el 3º toma distancia entre los componentes de 30px tanto horizontal como verticalmente.



```
setLayout(new GridLayout(filas, columnas))
```

Crea una zona de **filas** x **columnas** componentes y éstos se van acomodando de izquierda a derecha y de arriba a abajo.

GridLayout tiene otro constructor que permite establecer la separación (en pixels) ente los componentes, que es cero con el primer constructor.

Así, por ejemplo:

```
new GridLayout(3, 4, 2, 2)
```

crea una organización de 3 filas y 4 columnas donde los componentes quedan a dos pixels de separación.

Ejemplo:

```
setLayout(new GridLayout(3, 4, 2, 2);  
for(int i = 0; i < 3 * 4; i++) {  
    add(new JButton(Integer.toString(i + 1)));  
}
```

```
import javax.swing.*;
import java.awt.*;

public class Gui03b extends JFrame {

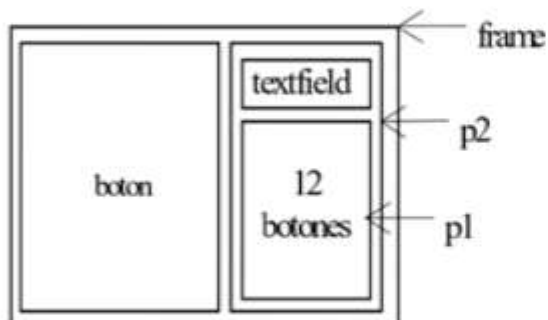
    public Gui03b() {
        super("Ejemplo de Layout");

        setLayout(new GridLayout(4, 3, 5, 5));
        for(int i = 1; i <= 10; i++)
            add(new JButton(Integer.toString(i)));
        setSize(200,200);
        setVisible(true);
        setDefaultCloseOperation(EXIT_ON_CLOSE);
    }
}
```



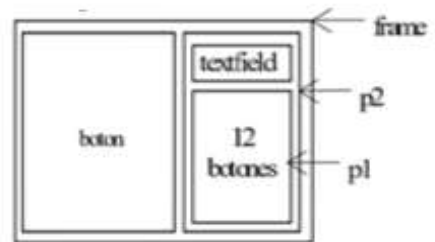
• Paneles como contenedores

Los paneles actúan como pequeños contenedores para agrupar componentes. Colocamos los componentes en paneles y los paneles en el **frame** o incluso en otros paneles.




```
import java.awt.*;
import javax.swing.*;
```

```
public class Gui04 extends JFrame {
    public Gui04() {
        setTitle("Horno microondas");
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        setLayout(new BorderLayout());
        // Create panel p1 for the buttons and set GridLayout
        JPanel p1 = new JPanel();
        p1.setLayout(new GridLayout(4, 3));
        // Add buttons to the panel
        for(int i = 1; i <= 9; i++) {
            p1.add(new JButton("" + i));
        }
        p1.add(new JButton("" + 0));
        p1.add(new JButton("Start"));
        p1.add(new JButton("Stop"));
        ...
    }
}
```



```
        // Create panel p2 to hold a text field and p1
        JPanel p2 = new JPanel();
        p2.setLayout(new BorderLayout());
        p2.add(new JTextField("Aquí el tiempo"),
            BorderLayout.NORTH);
        p2.add(p1, BorderLayout.CENTER);
        // Add p2 and a button to the frame
        add(p2, BorderLayout.EAST);
        add(new JButton("Aquí la comida"),
            BorderLayout.CENTER);

        setSize(400, 250);
        setVisible(true);
    }

    /** Main method */
    public static void main(String[] args) {
        Gui04 frame = new Gui04();
    }
}
```