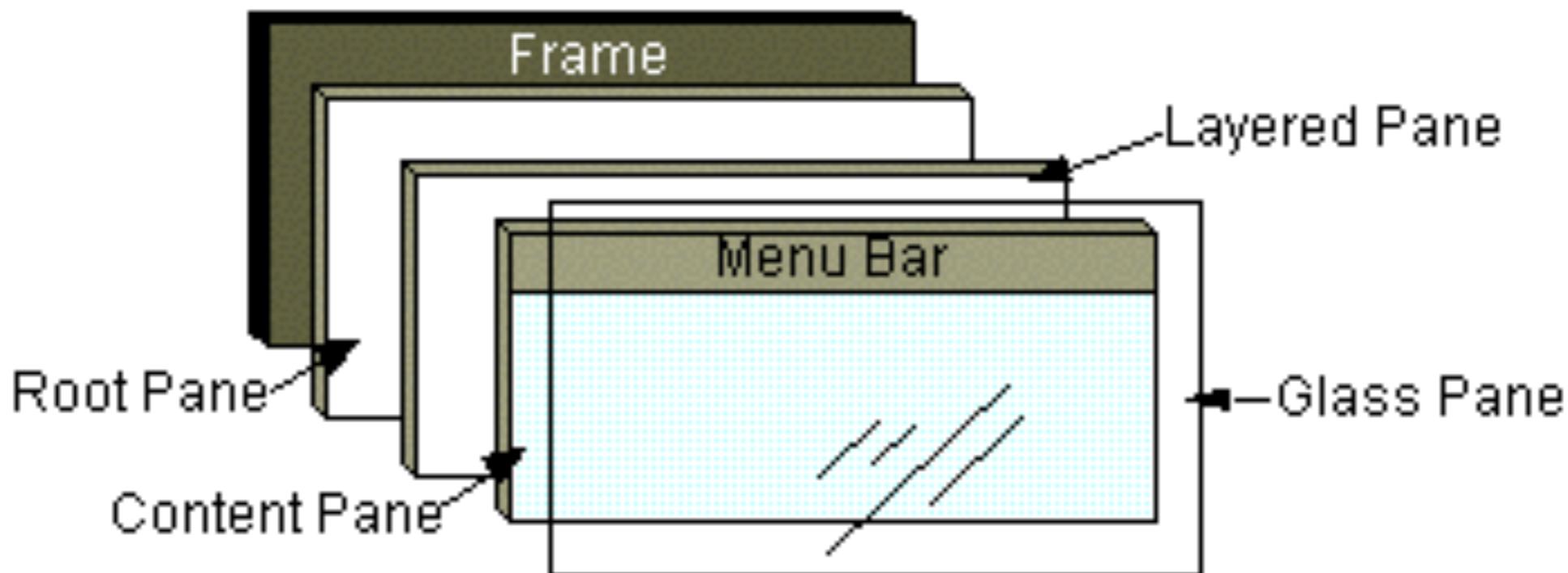


“Frame” en Swing: JFrame

La clase para crear un “frame” en Swing se llama **JFrame**. Representa una ventana como:



El “Content Pane” es donde colocamos los “widgets”, las otras partes (Panes) no son importantes de momento.

Tu Primer JFrame!

```
import javax.swing.*;  
  
public class FrameExample {  
  
    public static void main(String[] args) {  
        SwingUtilities.invokeLater(new Runnable() {  
            public void run() {  
                createAndShowGUI();  
            }  
        });  
    }  
  
    private static void createAndShowGUI() {  
  
        JFrame frame = new JFrame("Hello World!");  
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);  
  
        frame.setSize(400, 100);  
        frame.setVisible(true);  
    }  
}
```

Tu Primer JFrame!

```
import javax.swing.*; ← importar las clases de swing

public class FrameExample {

    public static void main(String[] args) {
        SwingUtilities.invokeLater(new Runnable() {
            public void run() {
                createAndShowGUI();
            }
        });
    }

    private static void createAndShowGUI() {

        JFrame frame = new JFrame("Hello World!");
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

        frame.setSize(400, 100);
        frame.setVisible(true);
    }
}
```

Tu Primer JFrame!

```
import javax.swing.*; ← importar las clases de swing

public class FrameExample {

    public static void main(String[] args) {
        SwingUtilities.invokeLater(new Runnable() {
            public void run() {
                createAndShowGUI();
            }
        });
    }

    private static void createAndShowGUI() {
        JFrame frame = new JFrame("Hello World!");
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

        frame.setSize(400, 100);
        frame.setVisible(true);
    }
}
```

importar las clases de swing

crear el JFrame

Tu Primer JFrame!

```
import javax.swing.*;
```

importar las clases de swing

```
public class FrameExample {
```

```
    public static void main(String[] args) {
        SwingUtilities.invokeLater(new Runnable() {
            public void run() {
                createAndShowGUI();
            }
        });
    }
```

```
    private static void createAndShowGUI() {
```

```
        JFrame frame = new JFrame("Hello World!");
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
```

```
        frame.setSize(400, 100);
        frame.setVisible(true);
    }
```

crear el JFrame

Que hacer cuando creamos la ventana. Si no usas EXIT la ventana se cierra pero Swing sigue ejecutando ...

Tu Primer JFrame!

```
import javax.swing.*;
```

importar las clases de swing

```
public class FrameExample {  
  
    public static void main(String[] args) {  
        SwingUtilities.invokeLater(new Runnable() {  
            public void run() {  
                createAndShowGUI();  
            }  
        });  
    }  
}
```

```
private static void createAndShowGUI() {
```

```
    JFrame frame = new JFrame("Hello World!");  
    frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
```

```
    frame.setSize(400, 100);  
    frame.setVisible(true);
```

```
}
```

Fijar el tamaño
(pixels)

crear el JFrame

Que hacer cuando creamos la ventana. Si no usas EXIT la ventana se cierra pero Swing sigue ejecutando ...

Tu Primer JFrame!

```
import javax.swing.*;
```

importar las clases de swing

```
public class FrameExample {
```

```
    public static void main(String[] args) {
        SwingUtilities.invokeLater(new Runnable() {
            public void run() {
                createAndShowGUI();
            }
        });
    }
```

```
    private static void createAndShowGUI() {
```

```
        JFrame frame = new JFrame("Hello World!");
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
```

```
        frame.setSize(400, 100);
        frame.setVisible(true);
    }
```

Hacerlo visible

Fijar el tamaño
(xpixels)

crear el JFrame

Que hacer cuando creamos la ventana. Si no usas EXIT la ventana se cierra pero Swing sigue ejecutando ...

Tu Primer JFrame!

```
import javax.swing.*;
```

importar las clases de swing

```
public class FrameExample {
```

```
    public static void main(String[] args) {
```

```
        SwingUtilities.invokeLater(new Runnable() {
```

```
            public void run() {  
                createAndShowGUI();
```

```
            }  
        });
```

```
}
```

programar una tarea para ejecutar createAndShowGUI en la hebra de Swing

```
    private static void createAndShowGUI() {
```

```
        JFrame frame = new JFrame("Hello World!");
```

```
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
```

```
        frame.setSize(400, 100);
```

```
        frame.setVisible(true);
```

```
}
```

crear el JFrame

Hacerlo visible

Fijar el tamaño (pixels)

Que hacer cuando creamos la ventana. Si no usas EXIT la ventana se cierra pero Swing sigue ejecutando ...

Es común/mejor Extender JFrame

```
import javax.swing.*;  
  
public class FrameExample_1 extends JFrame {  
  
    public FrameExample_1() {  
        super("[=] Hello World [=]");  
        initGUI();  
    }  
  
    private void initGUI() {  
        this.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);  
        this.setSize(400, 100);  
        this.setVisible(true);  
    }  
  
    public static void main(String[] args) {  
        SwingUtilities.invokeLater(new Runnable() {  
            public void run() {  
                new FrameExample_1();  
            }  
        });  
    }  
}
```

see: example.swing.misc.FrameExample_1

Es común/mejor Extender JFrame

```
import javax.swing.*;  
  
public class FrameExample_1 extends JFrame {  
  
    public FrameExample_1() {  
        super("[=] Hello World [=]");  
        initGUI();  
    }  
  
    private void initGUI() {  
        this.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);  
        this.setSize(400, 100);  
        this.setVisible(true);  
    }  
  
    public static void main(String[] args) {  
        SwingUtilities.invokeLater(new Runnable() {  
            public void run() {  
                new FrameExample_1();  
            }  
        });  
    }  
}
```

extender la clase JFrame

Es común/mejor Extender JFrame

```
import javax.swing.*;  
  
public class FrameExample_1 extends JFrame {  
  
    public FrameExample_1() {  
        super("[=] Hello World [=]");  
        initGUI();  
    }  
  
    private void initGUI() {  
        this.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);  
        this.setSize(400, 100);  
        this.setVisible(true);  
    }  
  
    public static void main(String[] args) {  
        SwingUtilities.invokeLater(new Runnable() {  
            public void run() {  
                new FrameExample_1();  
            }  
        });  
    }  
}
```

extender la clase JFrame

en la constructora, primero llamar a
la constructora de JFrame, después
construir el resto ...

Es común/mejor Extender JFrame

```
import javax.swing.*;  
  
public class FrameExample_1 extends JFrame {  
  
    public FrameExample_1() {  
        super("[=] Hello World [=]");  
        initGUI();  
    }  
  
    private void initGUI() {  
        this.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);  
        this.setSize(400, 100);  
        this.setVisible(true);  
    }  
  
    public static void main(String[] args) {  
        SwingUtilities.invokeLater(new Runnable() {  
            public void run() {  
                new FrameExample_1();  
            }  
        });  
    }  
}
```

extender la clase JFrame

en la constructora, primero llamar a la constructora de JFrame, después construir el resto ...

Mejor tener un método que inicializa la interfaz gráfica y llamarle desde la constructora, etc.

Es común/mejor Extender JFrame

```
import javax.swing.*;  
  
public class FrameExample_1 extends JFrame {  
  
    public FrameExample_1() {  
        super("[=] Hello World [=]");  
        initGUI();  
    }  
  
    private void initGUI() {  
        this.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);  
        this.setSize(400, 100);  
        this.setVisible(true);  
    }  
  
    public static void main(String[] args) {  
        SwingUtilities.invokeLater(new Runnable() {  
            public void run() {  
                new FrameExample_1();  
            }  
        });  
    }  
}
```

extender la clase JFrame

en la constructora, primero llamar a la constructora de JFrame, después construir el resto ...

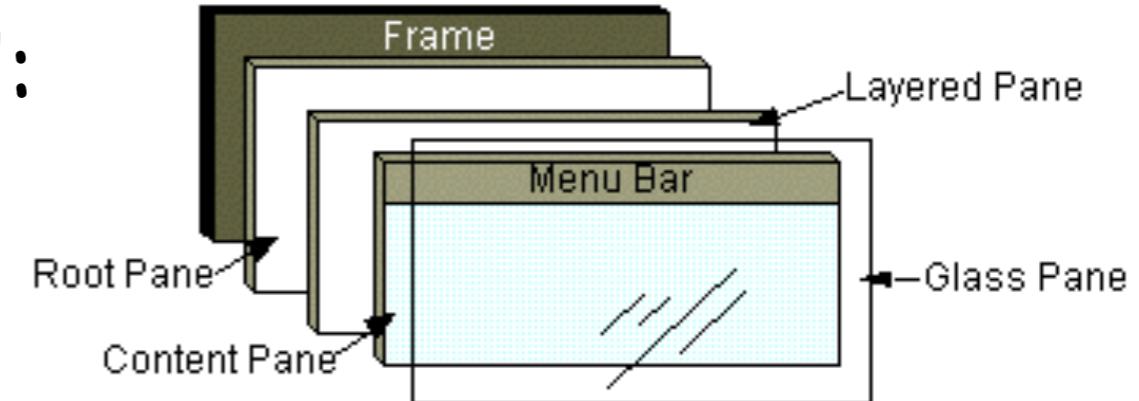
Mejor tener un método que inicializa la interfaz gráfica y llamarle desde la constructora, etc.

programar una tarea para crear una instancia de la clase desde la hebra de swing ...

Usa JPanel para Organizar Widgets

- ◆ Imagina la GUI como un árbol:

Si A está colocado sobre B
A es hijo de B



- ◆ Se puede colocar los widgets de manera absoluta, respecto a la esquina superior izquierda (0,0) del padre ...
- ◆ Poder agrupar widgets es fundamental, nos permite ...
 1. manejar conjunto de components
 2. mover varios a la vez
 3. definir propiedades comunes
- ◆ JPanel nos permite agrupar componentes

Usa JPanel para Colocar Widgets

```
public class PanelExample extends JFrame {  
  
    public PanelExample() {  
        super("[=] There's a JPanel in here! [=]");  
        initGUI();  
    }  
  
    private void initGUI() {  
        JPanel mainPanel = new JPanel();  
        mainPanel.setOpaque(true);  
  
        this.setContentPane(mainPanel);  
  
        this.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);  
        this.setSize(290, 100);  
        this.setVisible(true);  
    }  
  
    ...  
}
```

see: example.swing.misc.PanelExample

Usa JPanel para Colocar Widgets

```
public class PanelExample extends JFrame {  
  
    public PanelExample() {  
        super("[=] There's a JPanel in here! [=]");  
        initGUI();  
    }  
  
    private void initGUI() {  
        JPanel mainPanel = new JPanel();  
        mainPanel.setOpaque(true);  
  
        this.setContentPane(mainPanel);  
  
        this.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);  
        this.setSize(290, 100);  
        this.setVisible(true);  
    }  
  
    ...  
}
```

crear una instancia de JPanel

Usa JPanel para Colocar Widgets

```
public class PanelExample extends JFrame {
```

```
    public PanelExample() {  
        super("[=] There's a JPanel in here! [=]");  
        initGUI();  
    }
```

```
    private void initGUI() {  
        JPanel mainPanel = new JPanel();  
        mainPanel.setOpaque(true);
```

```
        this.setContentPane(mainPanel);
```

```
        this.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);  
        this.setSize(290, 100);  
        this.setVisible(true);
```

```
}
```

```
{...}
```

see: example.swing.misc.PanelExample

crear una instancia de JPanel

hacerlo opaco (no transparente)
No hace falta siempre...

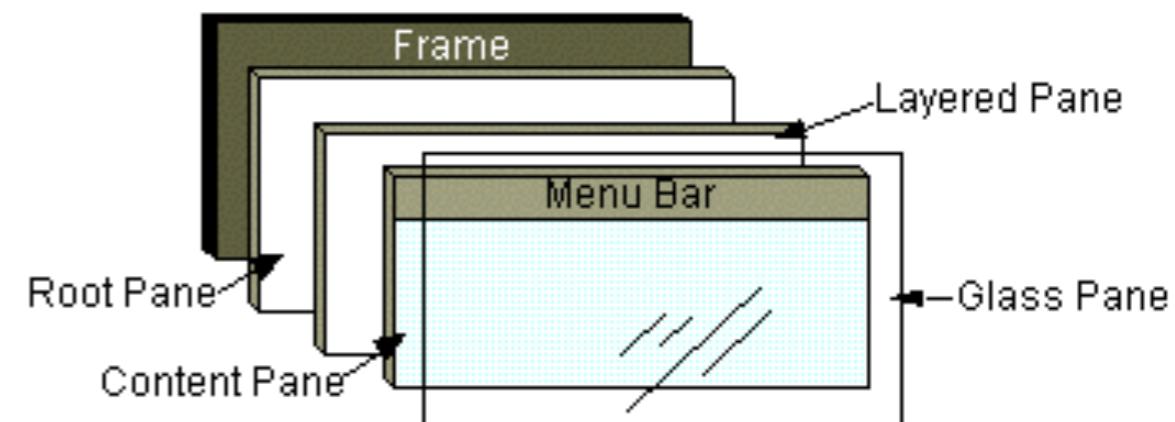
Usa JPanel para Colocar Widgets

```
public class PanelExample extends JFrame {  
  
    public PanelExample() {  
        super("[=] There's a JPanel in here! [=]");  
        initGUI();  
    }  
  
    private void initGUI() {  
        JPanel mainPanel = new JPanel();  
        mainPanel.setOpaque(true);  
  
        this.setContentPane(mainPanel);  
  
        this.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);  
        this.setSize(290, 100);  
        this.setVisible(true);  
    }  
  
    ...  
}
```

crear una instancia de JPanel

hacerlo opaco (no transparente)
No hace falta siempre...

colocar el JPanel en el
Content Pane del JFrame



see: example.swing.misc.PanelExample

JPanels Anidados

```
private void initGUI() {  
  
    JPanel mainPanel = new JPanel();  
    mainPanel.setLayout(null);  
  
    JPanel redPanel = new JPanel();  
    redPanel.setBackground(Color.red);  
    redPanel.setLocation(60, 10);  
    redPanel.setSize(50, 50);  
    mainPanel.add(redPanel);  
  
    JPanel bluePanel = new JPanel();  
    bluePanel.setBackground(Color.blue);  
    bluePanel.setLocation(200, 10);  
    bluePanel.setSize(50, 50);  
    mainPanel.add(bluePanel);  
  
    this.setContentPane(mainPanel);  
    this.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);  
    this.setSize(290, 100);  
    this.setVisible(true);  
}
```

see: example.swing.misc.PanelExample_1

JPanels Anidados

```
private void initGUI() {  
    JPanel mainPanel = new JPanel();  
    mainPanel.setLayout(null);  
  
    JPanel redPanel = new JPanel();  
    redPanel.setBackground(Color.red);  
    redPanel.setLocation(60, 10);  
    redPanel.setSize(50, 50);  
    mainPanel.add(redPanel);  
  
    JPanel bluePanel = new JPanel();  
    bluePanel.setBackground(Color.blue);  
    bluePanel.setLocation(200, 10);  
    bluePanel.setSize(50, 50);  
    mainPanel.add(bluePanel);  
  
    this.setContentPane(mainPanel);  
    this.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);  
    this.setSize(290, 100);  
    this.setVisible(true);  
}
```

crear una instancia de JPanel

JPanels Anidados

```
private void initGUI() {  
    JPanel mainPanel = new JPanel();  
    mainPanel.setLayout(null); ←  
    JPanel redPanel = new JPanel();  
    redPanel.setBackground(Color.red);  
    redPanel.setLocation(60, 10);  
    redPanel.setSize(50, 50);  
    mainPanel.add(redPanel);  
  
    JPanel bluePanel = new JPanel();  
    bluePanel.setBackground(Color.blue);  
    bluePanel.setLocation(200, 10);  
    bluePanel.setSize(50, 50);  
    mainPanel.add(bluePanel);  
  
    this.setContentPane(mainPanel);  
    this.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);  
    this.setSize(290, 100);  
    this.setVisible(true);  
}
```

crear una instancia de JPanel

usa “absolute positioning”
para colocar widgets

JPanels Anidados

```
private void initGUI() {  
  
    JPanel mainPanel = new JPanel();  
    mainPanel.setLayout(null); ← crear una instancia de JPanel  
  
    JPanel redPanel = new JPanel(); ← usa "absolute positioning"  
    para colocar widgets  
    redPanel.setBackground(Color.red);  
    redPanel.setLocation(60, 10);  
    redPanel.setSize(50, 50);  
    mainPanel.add(redPanel);  
  
    JPanel bluePanel = new JPanel();  
    bluePanel.setBackground(Color.blue);  
    bluePanel.setLocation(200, 10);  
    bluePanel.setSize(50, 50);  
    mainPanel.add(bluePanel);  
  
    this.setContentPane(mainPanel);  
    this.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);  
    this.setSize(290, 100);  
    this.setVisible(true);  
}
```

- crear una instancia de JPanel
- usa “absolute positioning” para colocar widgets
- crear otro JPanel

JPanels Anidados

```
private void initGUI() {
```

```
    JPanel mainPanel = new JPanel();
```

```
    mainPanel.setLayout(null);
```

```
    JPanel redPanel = new JPanel();
```

```
    redPanel.setBackground(Color.red);
```

```
    redPanel.setLocation(60, 10);
```

```
    redPanel.setSize(50, 50);
```

```
    mainPanel.add(redPanel);
```

```
    JPanel bluePanel = new JPanel();
```

```
    bluePanel.setBackground(Color.blue);
```

```
    bluePanel.setLocation(200, 10);
```

```
    bluePanel.setSize(50, 50);
```

```
    mainPanel.add(bluePanel);
```

```
    this.setContentPane(mainPanel);
```

```
    this.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
```

```
    this.setSize(290, 100);
```

```
    this.setVisible(true);
```

```
}
```

crear una instancia de JPanel

usa “absolute positioning”
para colocar widgets

crear otro JPanel

color, posición, tamaño, ...

JPanels Anidados

```
private void initGUI() {
```

```
    JPanel mainPanel = new JPanel();  
    mainPanel.setLayout(null);
```

```
    JPanel redPanel = new JPanel();  
    redPanel.setBackground(Color.red);  
    redPanel.setLocation(60, 10);  
    redPanel.setSize(50, 50);  
    mainPanel.add(redPanel);
```

```
    JPanel bluePanel = new JPanel();  
    bluePanel.setBackground(Color.blue);  
    bluePanel.setLocation(200, 10);  
    bluePanel.setSize(50, 50);  
    mainPanel.add(bluePanel);
```

```
    this.setContentPane(mainPanel);  
    this.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);  
    this.setSize(290, 100);  
    this.setVisible(true);
```

```
}
```

crear una instancia de JPanel

usa “absolute positioning”
para colocar widgets

crear otro JPanel

color, posición, tamaño, ...

añadirlo al mainPanel

JPanels Anidados

```
private void initGUI() {
```

```
    JPanel mainPanel = new JPanel();  
    mainPanel.setLayout(null);
```

```
    JPanel redPanel = new JPanel();  
    redPanel.setBackground(Color.red);  
    redPanel.setLocation(60, 10);  
    redPanel.setSize(50, 50);  
    mainPanel.add(redPanel);
```

```
    JPanel bluePanel = new JPanel();  
    bluePanel.setBackground(Color.blue);  
    bluePanel.setLocation(200, 10);  
    bluePanel.setSize(50, 50);  
    mainPanel.add(bluePanel);
```

```
    this.setContentPane(mainPanel);  
    this.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);  
    this.setSize(290, 100);  
    this.setVisible(true);
```

```
}
```

crear una instancia de JPanel

usa “absolute positioning”
para colocar widgets

crear otro JPanel

color, posición, tamaño, ...

añadirlo al mainPanel

otro JPanel azul

JPanels Anidados

```
private void initGUI() {
```

```
    JPanel mainPanel = new JPanel();  
    mainPanel.setLayout(null);
```

```
    JPanel redPanel = new JPanel();  
    redPanel.setBackground(Color.red);  
    redPanel.setLocation(60, 10);  
    redPanel.setSize(50, 50);  
    mainPanel.add(redPanel);
```

```
    JPanel bluePanel = new JPanel();  
    bluePanel.setBackground(Color.blue);  
    bluePanel.setLocation(200, 10);  
    bluePanel.setSize(50, 50);  
    mainPanel.add(bluePanel);
```

```
    this.setContentPane(mainPanel);  
    this.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);  
    this.setSize(290, 100);  
    this.setVisible(true);
```

```
}
```

crear una instancia de JPanel

usa “absolute positioning” para colocar widgets

crear otro JPanel

color, posición, tamaño, ...

añadirlo al mainPanel

otro JPanel azul

colocar mainPanel en el ContentPane del JFrame

JPanels Anidados

```
private void initGUI() {
```

```
    JPanel mainPanel = new JPanel();  
    mainPanel.setLayout(null);
```

```
    JPanel redPanel = new JPanel();  
    redPanel.setBackground(Color.red);  
    redPanel.setLocation(60, 10);  
    redPanel.setSize(50, 50);  
    mainPanel.add(redPanel);
```

```
    JPanel bluePanel = new JPanel();  
    bluePanel.setBackground(Color.blue);  
    bluePanel.setLocation(200, 10);  
    bluePanel.setSize(50, 50);  
    mainPanel.add(bluePanel);
```

```
    this.setContentPane(mainPanel);  
    this.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);  
    this.setSize(290, 100);  
    this.setVisible(true);
```

crear una instancia de JPanel

usa “absolute positioning” para colocar widgets

crear otro JPanel

color, posición, tamaño, ...

añadirlo al mainPanel

otro JPanel azul

colocar mainPanel en el ContentPane del JFrame

Se puede hacer cada componente visible o invisible ...

FlowLayout

```
private void initGUI() {  
    JPanel mainPanel = new JPanel( new FlowLayout( FlowLayout.LEFT, 10, 20 ) );
```

```
    mainPanel.add( createPanel(Color.red, 50, 50) );  
    mainPanel.add( createPanel(Color.blue, 50, 50) );  
    mainPanel.add( createPanel(Color.green, 50, 50) );  
    mainPanel.add( createPanel(Color.orange, 50, 50) );  
    mainPanel.add( createPanel(Color.yellow, 50, 50) );
```

```
    mainPanel.setOpaque(true);
```

```
    ...
```

```
}
```

```
private JPanel createPanel(Color color, int x, int y) {
```

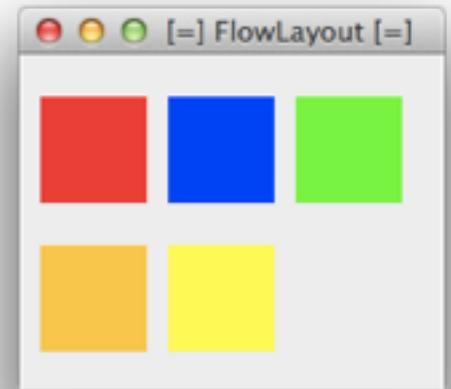
```
    JPanel panel;  
    panel = new JPanel();  
    panel.setBackground(color);  
    panel.setPreferredSize(new Dimension(x, y));  
    return panel;
```

```
}
```

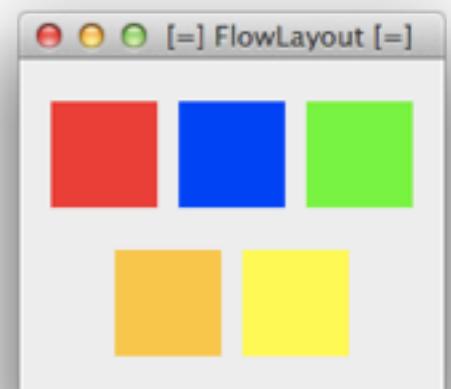
Los componentes se añaden de izquierda-derecha
de arriba-abajo ...

see: examples.swing.miscFlowLayout

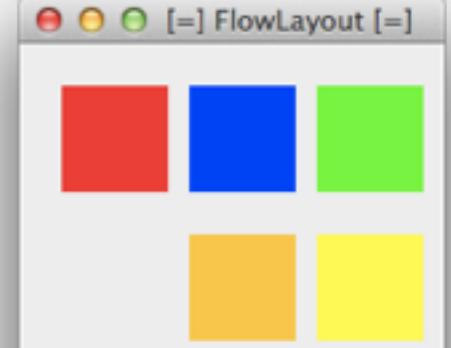
FlowLayout.LEFT



FlowLayout.CENTER



FlowLayout.RIGHT



FlowLayout

```
private void initGUI() {  
    JPanel mainPanel = new JPanel( new FlowLayout( FlowLayout.LEFT, 10, 20 ) );
```

```
    mainPanel.add( createPanel(Color.red, 50, 50) );  
    mainPanel.add( createPanel(Color.blue, 50, 50) );  
    mainPanel.add( createPanel(Color.green, 50, 50) );  
    mainPanel.add( createPanel(Color.orange, 50, 50) );  
    mainPanel.add( createPanel(Color.yellow, 50, 50) );
```

```
    mainPanel.setOpaque(true);
```

```
    ...
```

```
}
```

```
private JPanel createPanel(Color color, int x, int y) {
```

```
    JPanel panel;  
    panel = new JPanel();  
    panel.setBackground(color);  
    panel.setPreferredSize(new Dimension(x, y));  
    return panel;
```

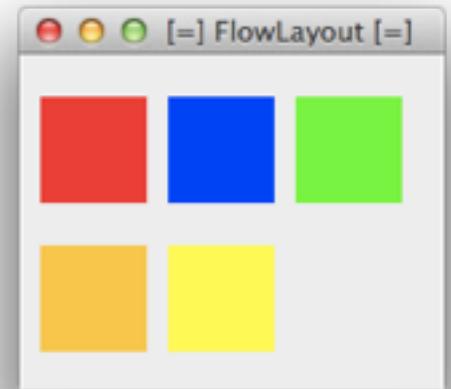
reorganiza cuando el tamaño
del panel cambia.

```
}
```

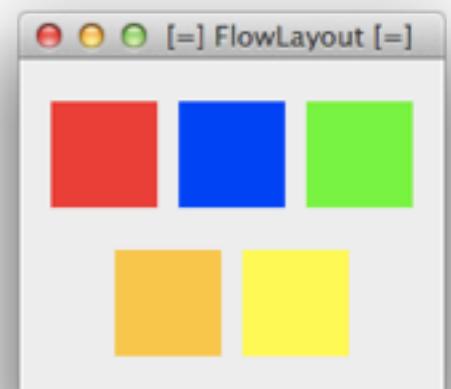
Los componentes se añaden de izquierda-derecha
de arriba-abajo ...

see: examples.swing.miscFlowLayout

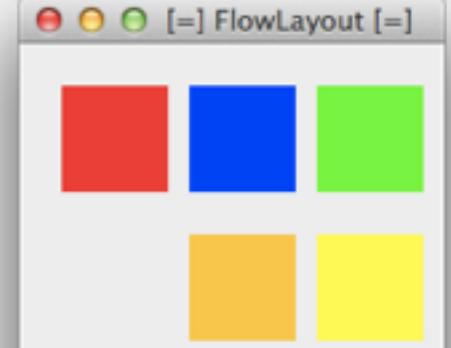
FlowLayout.LEFT



FlowLayout.CENTER



FlowLayout.RIGHT



FlowLayout

```
private void initGUI() {  
    JPanel mainPanel = new JPanel( new FlowLayout( FlowLayout.LEFT, 10, 20 ) );
```

```
    mainPanel.add( createPanel(Color.red, 50, 50) );  
    mainPanel.add( createPanel(Color.blue, 50, 50) );  
    mainPanel.add( createPanel(Color.green, 50, 50) );  
    mainPanel.add( createPanel(Color.orange, 50, 50) );  
    mainPanel.add( createPanel(Color.yellow, 50, 50) );
```

```
    mainPanel.setOpaque(true);  
    ...  
}
```

```
private JPanel createPanel(Color color, int x, int y) {
```

```
    JPanel panel;  
    panel = new JPanel();  
    panel.setBackground(color);  
    panel.setPreferredSize(new Dimension(x, y));  
    return panel;
```

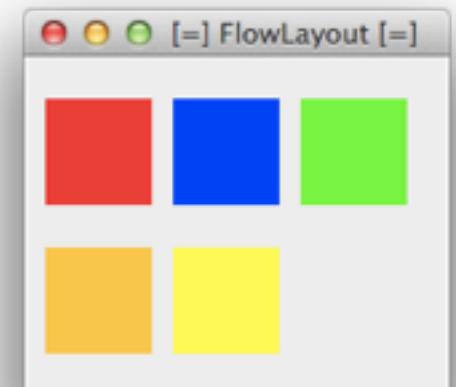
```
}
```

reorganiza cuando el tamaño
del panel cambia.

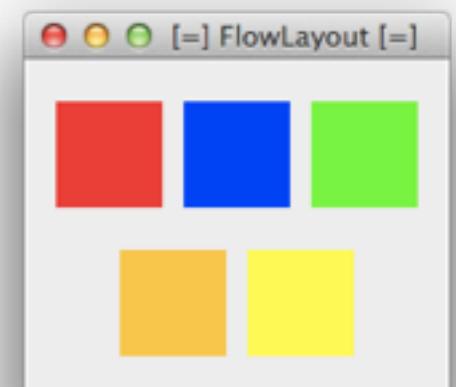
Los componentes se añaden de izquierda-derecha
de arriba-abajo ...

see: examples.swing.miscFlowLayout

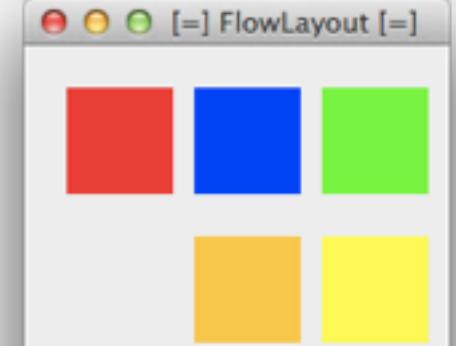
FlowLayout.LEFT



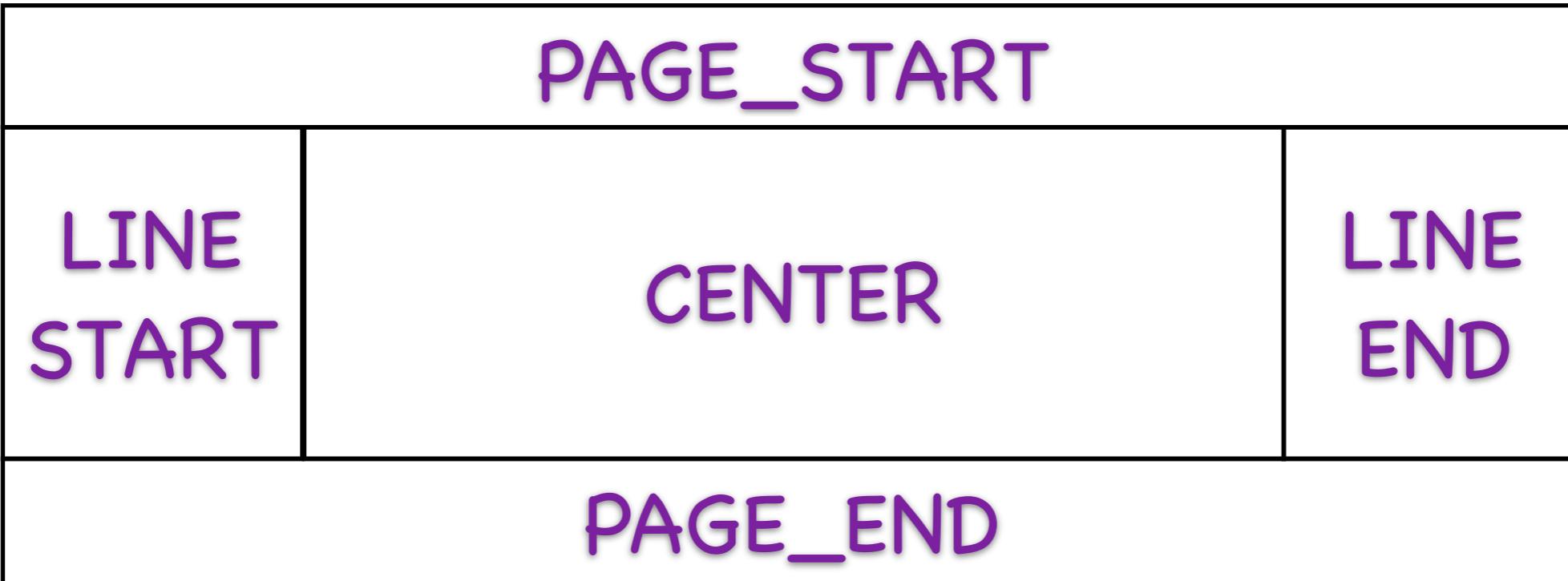
FlowLayout.CENTER



FlowLayout.RIGHT



BorderLayout

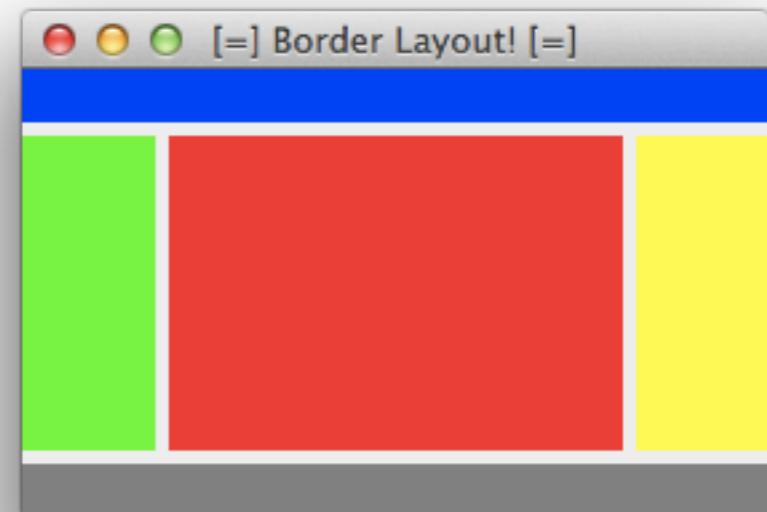


- ◆ Divide el panel en 5 partes, en cada parte podemos añadir sólo un componente (pero, podemos dejarla vacía también)
- ◆ Al redimensionar el panel
 1. Todos las partes cambian de tamaño para ajustarse al nuevo tamaño
 2. PAGE_START/END no cambia de altura
 3. LINE_START/END no cambia de anchura
 4. Cada parte “pide” al componente que está dentro de ajustar su tamaño al nuevo tamaño, ...

Border Layout - Cómo Usarlo

```
public JPanel initGUI() {  
    JPanel mainPanel = new JPanel( new BorderLayout(5,5) );  
  
    centerPanel = createPanel(Color.red,50,50);  
    mainPanel.add(centerPanel, BorderLayout.CENTER);  
  
    leftPanel = createPanel(Color.green,50,50);  
    mainPanel.add(leftPanel, BorderLayout.LINE_START);  
  
    rightPanel = createPanel(Color.yellow,50,50);  
    mainPanel.add(rightPanel, BorderLayout.LINE_END);  
  
    topPanel = createPanel(Color.blue,20,20);  
    mainPanel.add(topPanel, BorderLayout.PAGE_START);  
  
    botPanel = createPanel(Color.gray,20,20);  
    mainPanel.add(botPanel, BorderLayout.PAGE_END);  
  
    ...  
}  
  
private JPanel createPanel(Color color, int x, int y) {  
    JPanel panel;  
    panel = new JPanel();  
    panel.setBackground(color);  
    panel.setPreferredSize(new Dimension(x, y));  
    return panel;  
}
```

see: examples.swing.misc.BorderLayoutExample

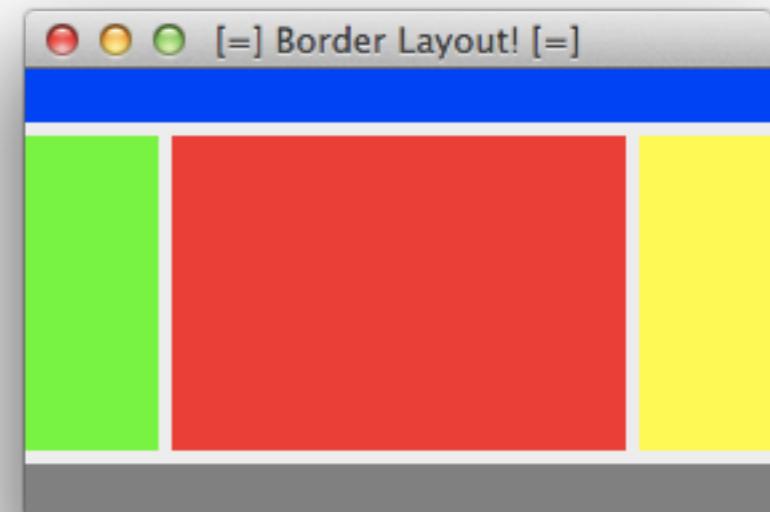


Border Layout - Cómo Usarlo

```
public JPanel initGUI() {  
    JPanel mainPanel = new JPanel( new BorderLayout(5,5) );  
  
    centerPanel = createPanel(Color.red,50,50);  
    mainPanel.add(centerPanel, BorderLayout.CENTER);  
  
    leftPanel = createPanel(Color.green,50,50);  
    mainPanel.add(leftPanel, BorderLayout.LINE_START);  
  
    rightPanel = createPanel(Color.yellow,50,50);  
    mainPanel.add(rightPanel, BorderLayout.LINE_END);  
  
    topPanel = createPanel(Color.blue,20,20);  
    mainPanel.add(topPanel, BorderLayout.PAGE_START);  
  
    botPanel = createPanel(Color.gray,20,20);  
    mainPanel.add(botPanel, BorderLayout.PAGE_END);  
  
    ...  
}  
  
private JPanel createPanel(Color color, int x, int y) {  
    JPanel panel;  
    panel = new JPanel();  
    panel.setBackground(color);  
    panel.setPreferredSize(new Dimension(x, y));  
    return panel;  
}
```

see: examples.swing.misc.BorderLayoutExample

Los parámetros son el espacio horizontal y vertical entre los partes



Border Layout - Cómo Usarlo

```
public JPanel initGUI() {  
    JPanel mainPanel = new JPanel( new BorderLayout(5,5) );  
  
    centerPanel = createPanel(Color.red,50,50);  
    mainPanel.add(centerPanel, BorderLayout.CENTER);  
  
    leftPanel = createPanel(Color.green,50,50);  
    mainPanel.add(leftPanel, BorderLayout.LINE_START);  
  
    rightPanel = createPanel(Color.yellow,50,50);  
    mainPanel.add(rightPanel, BorderLayout.LINE_END);  
  
    topPanel = createPanel(Color.blue,20,20);  
    mainPanel.add(topPanel, BorderLayout.PAGE_START);  
  
    botPanel = createPanel(Color.gray,20,20);  
    mainPanel.add(botPanel, BorderLayout.PAGE_END);  
  
    ...  
}  
  
private JPanel createPanel(Color color, int x, int y) {  
    JPanel panel;  
    panel = new JPanel();  
    panel.setBackground(color);  
    panel.setPreferredSize(new Dimension(x, y));  
    return panel;  
}
```

see: examples.swing.misc.BorderLayoutExample

Los parámetros son el espacio horizontal y vertical entre los partes

```
JPanel mainPanel = new JPanel();  
mgr = new BorderLayout(5,5);  
panel.setLayout(mgr);
```



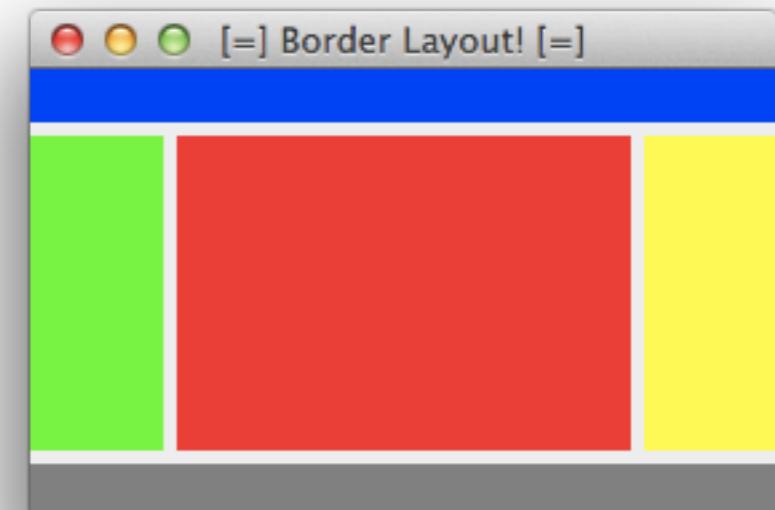
Border Layout - Cómo Usarlo

```
public JPanel initGUI() {  
    JPanel mainPanel = new JPanel( new BorderLayout(5,5) );  
  
    centerPanel = createPanel(Color.red,50,50);  
    mainPanel.add(centerPanel, BorderLayout.CENTER);  
  
    leftPanel = createPanel(Color.green,50,50);  
    mainPanel.add(leftPanel, BorderLayout.LINE_START);  
  
    rightPanel = createPanel(Color.yellow,50,50);  
    mainPanel.add(rightPanel, BorderLayout.LINE_END);  
  
    topPanel = createPanel(Color.blue,20,20);  
    mainPanel.add(topPanel, BorderLayout.PAGE_START);  
  
    botPanel = createPanel(Color.gray,20,20);  
    mainPanel.add(botPanel, BorderLayout.PAGE_END);  
  
    ...  
}  
  
private JPanel createPanel(Color color, int x, int y) {  
    JPanel panel;  
    panel = new JPanel();  
    panel.setBackground(color);  
    panel.setPreferredSize(new Dimension(x, y));  
    return panel;  
}
```

see: examples.swing.misc.BorderLayoutExample

Los parámetros son el espacio horizontal y vertical entre los partes

```
JPanel mainPanel = new JPanel();  
mgr = new BorderLayout(5,5);  
panel.setLayout(mgr);
```



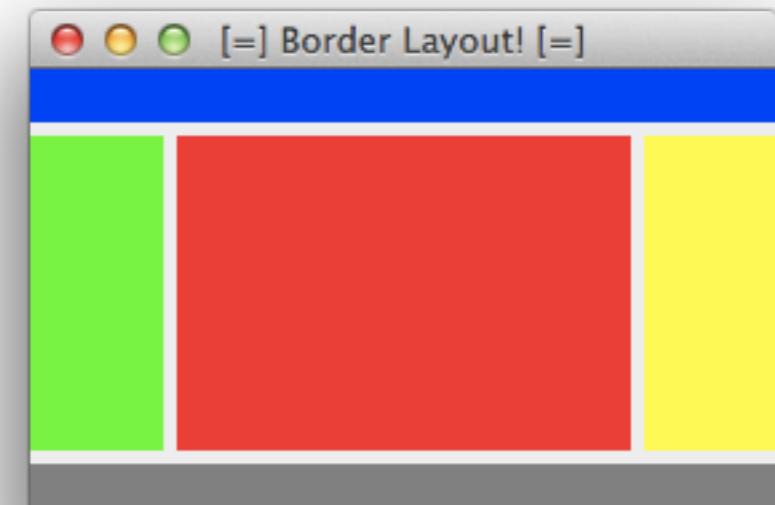
Border Layout - Cómo Usarlo

```
public JPanel initGUI() {  
    JPanel mainPanel = new JPanel( new BorderLayout(5,5) );  
  
    centerPanel = createPanel(Color.red,50,50);  
    mainPanel.add(centerPanel, BorderLayout.CENTER);  
  
    leftPanel = createPanel(Color.green,50,50);  
    mainPanel.add(leftPanel, BorderLayout.LINE_START);  
  
    rightPanel = createPanel(Color.yellow,50,50);  
    mainPanel.add(rightPanel, BorderLayout.LINE_END);  
  
    topPanel = createPanel(Color.blue,20,20);  
    mainPanel.add(topPanel, BorderLayout.PAGE_START);  
  
    botPanel = createPanel(Color.gray,20,20);  
    mainPanel.add(botPanel, BorderLayout.PAGE_END);  
  
    ...  
}  
  
private JPanel createPanel(Color color, int x, int y) {  
    JPanel panel;  
    panel = new JPanel();  
    panel.setBackground(color);  
    panel.setPreferredSize(new Dimension(x, y));  
    return panel;  
}
```

see: examples.swing.misc.BorderLayoutExample

Los parámetros son el espacio horizontal y vertical entre los partes

```
JPanel mainPanel = new JPanel();  
mgr = new BorderLayout(5,5);  
panel.setLayout(mgr);
```



Border Layout - Cómo Usarlo

```
public JPanel initGUI() {  
    JPanel mainPanel = new JPanel( new BorderLayout(5,5) );  
  
    centerPanel = createPanel(Color.red,50,50);  
    mainPanel.add(centerPanel, BorderLayout.CENTER);  
  
    leftPanel = createPanel(Color.green,50,50);  
    mainPanel.add(leftPanel, BorderLayout.LINE_START);  
  
    rightPanel = createPanel(Color.yellow,50,50);  
    mainPanel.add(rightPanel, BorderLayout.LINE_END);  
  
    topPanel = createPanel(Color.blue,20,20);  
    mainPanel.add(topPanel, BorderLayout.PAGE_START);  
  
    botPanel = createPanel(Color.gray,20,20);  
    mainPanel.add(botPanel, BorderLayout.PAGE_END);  
  
    ...  
}  
  
private JPanel createPanel(Color color, int x, int y) {  
    JPanel panel;  
    panel = new JPanel();  
    panel.setBackground(color);  
    panel.setPreferredSize(new Dimension(x, y));  
    return panel;  
}
```

see: examples.swing.misc.BorderLayoutExample

Los parámetros son el espacio horizontal y vertical entre los partes

```
JPanel mainPanel = new JPanel();  
mgr = new BorderLayout(5,5);  
panel.setLayout(mgr);
```

mainPanel.add(p) añade a BorderLayout.CENTER



Border Layout - Cómo Usarlo

```
public JPanel initGUI() {  
    JPanel mainPanel = new JPanel( new BorderLayout(5,5) );  
  
    centerPanel = createPanel(Color.red,50,50);  
    mainPanel.add(centerPanel, BorderLayout.CENTER);  
  
    leftPanel = createPanel(Color.green,50,50);  
    mainPanel.add(leftPanel, BorderLayout.LINE_START);  
  
    rightPanel = createPanel(Color.yellow,50,50);  
    mainPanel.add(rightPanel, BorderLayout.LINE_END);  
  
    topPanel = createPanel(Color.blue,20,20);  
    mainPanel.add(topPanel, BorderLayout.PAGE_START);  
  
    botPanel = createPanel(Color.gray,20,20);  
    mainPanel.add(botPanel, BorderLayout.PAGE_END);  
    ...  
}  
  
private JPanel createPanel(Color color, int x, int y) {  
    JPanel panel;  
    panel = new JPanel();  
    panel.setBackground(color);  
    panel.setPreferredSize(new Dimension(x, y));  
    return panel;  
}
```

see: examples.swing.misc.BorderLayoutExample

Los parámetros son el espacio horizontal y vertical entre los partes

```
JPanel mainPanel = new JPanel();  
mgr = new BorderLayout(5,5);  
panel.setLayout(mgr);
```

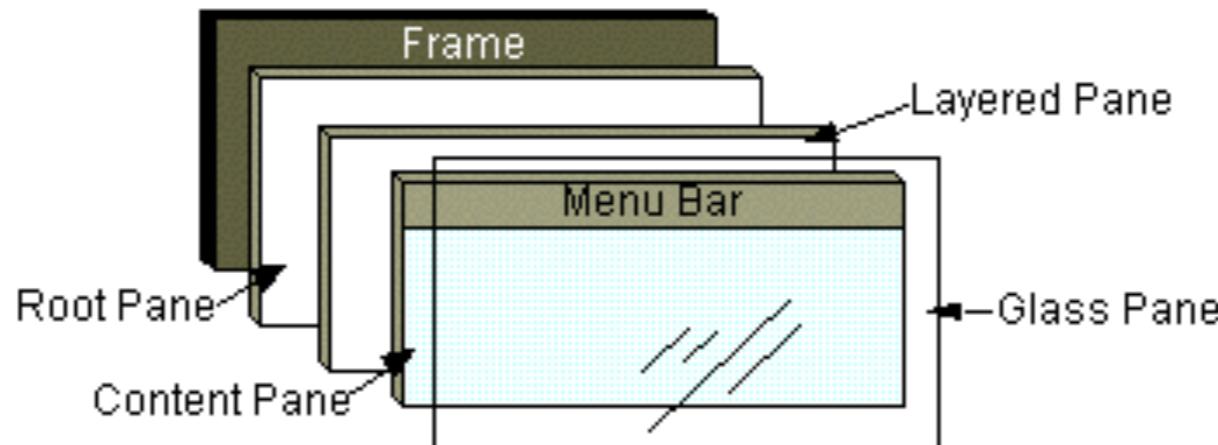
mainPanel.add(p) añade a BorderLayout.CENTER

¿Porqué mainPanel cambia de tamaño cuando cambia el tamaño del "frame"?



BorderLayout por Defecto ...

ContentPane y JFrame usan BorderLayout por defecto

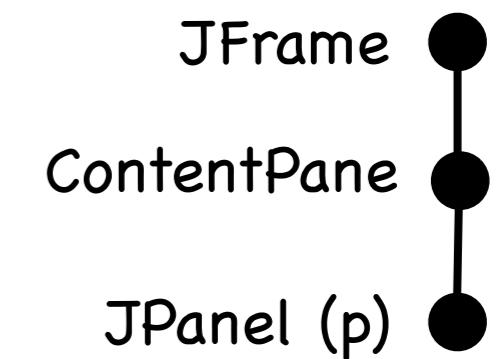
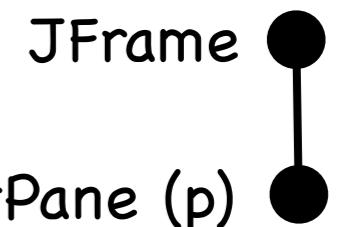


Cambia el content pane al JPanel p

```
JFrame frame = new JFrame("...");  
JPanel p = ...  
frame.setContentPane(p);
```

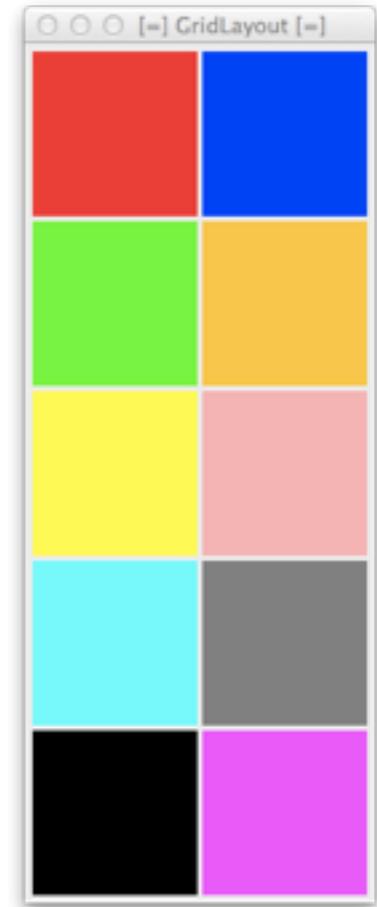
Añade p a la parte central del content pane

```
JFrame frame = new JFrame("...");  
JPanel p = ...  
frame.getContentPane().add(p);
```



GridLayout

```
private void initGUI() {  
    JPanel mainPanel = new JPanel(new GridLayout(5, 2, 3, 3));  
  
    JPanel red = createSquareJPanel(Color.red, 100);  
    JPanel blue = createSquareJPanel(Color.blue, 100);  
    JPanel green = createSquareJPanel(Color.green, 100);  
    JPanel orange = createSquareJPanel(Color.orange, 100);  
    JPanel yellow = createSquareJPanel(Color.yellow, 50);  
    JPanel pink = createSquareJPanel(Color.pink, 50);  
    JPanel cyan = createSquareJPanel(Color.cyan, 50);  
    JPanel gray = createSquareJPanel(Color.gray, 50);  
    JPanel black = createSquareJPanel(Color.black, 50);  
    JPanel magenta = createSquareJPanel(Color.magenta, 50);  
  
    mainPanel.add(red);  
    mainPanel.add(blue);  
    ...  
}
```



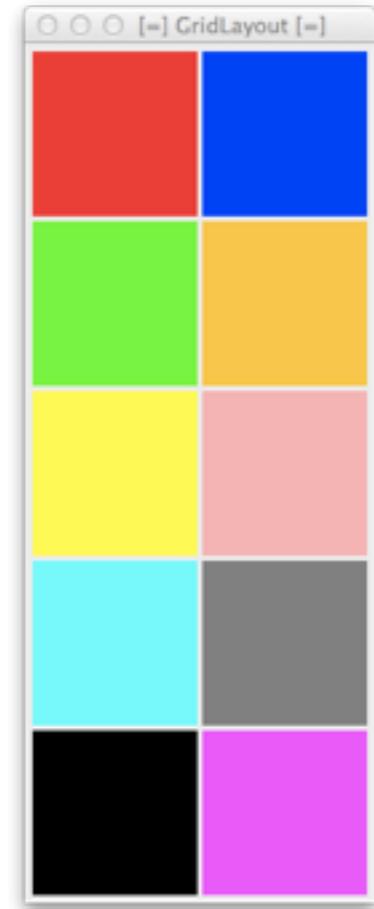
see: examples.swing.misc.GridLayout, examples.swing.misc.GridLayout_1

GridLayout

```
private void initGUI() {  
    JPanel mainPanel = new JPanel(new GridLayout(5, 2, 3, 3));
```

rows x cols spacing

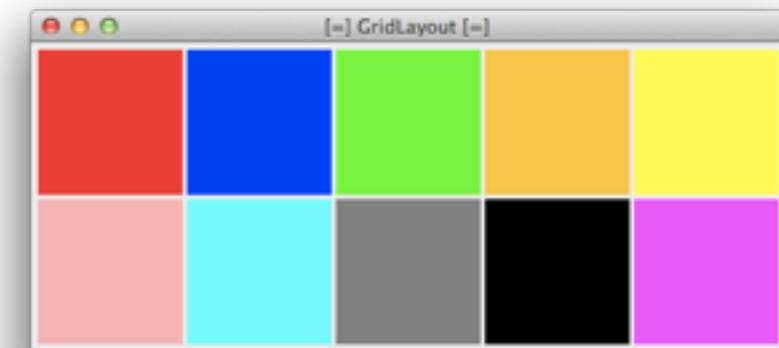
```
JPanel red = createSquareJPanel(Color.red, 100);  
JPanel blue = createSquareJPanel(Color.blue, 100);  
JPanel green = createSquareJPanel(Color.green, 100);  
JPanel orange = createSquareJPanel(Color.orange, 100);  
JPanel yellow = createSquareJPanel(Color.yellow, 50);  
JPanel pink = createSquareJPanel(Color.pink, 50);  
JPanel cyan = createSquareJPanel(Color.cyan, 50);  
JPanel gray = createSquareJPanel(Color.gray, 50);  
JPanel black = createSquareJPanel(Color.black, 50);  
JPanel magenta = createSquareJPanel(Color.magenta, 50);
```



```
mainPanel.add(red);  
mainPanel.add(blue);  
...  
}
```

- ◆ Los componentes se añaden de izq.-der. de arriba-abajo.
- ◆ Al redimensionar, el espacio extra se divide entre la casillas.
- ◆ El tamaño de un componente se ajusta al de su casilla

GridLayout(2,5)



see: examples.swing.misc.GridLayout, examples.swing.misc.GridLayout_1