



Java Avanzado

Java Foundation Classes (Swing)

Copyright

- Copyright (c) 2004
José M. Ordax
- Este documento puede ser distribuido solo bajo los términos y condiciones de la Licencia de Documentación de javaHispano v1.0 o posterior.
- La última versión se encuentra en
<http://www.javahispano.org/licencias/>

Java Foundation Classes

- Se trata de un conjunto de clases para mejorar el soporte al desarrollo de GUIs.
- Se creó como una extensión de la AWT añadiendo las siguientes características:
 - Componentes Swing.
 - Soporte de Look & Feel.
 - API de accesibilidad.
 - Java 2D API.
 - Soporte de Drag & Drop.

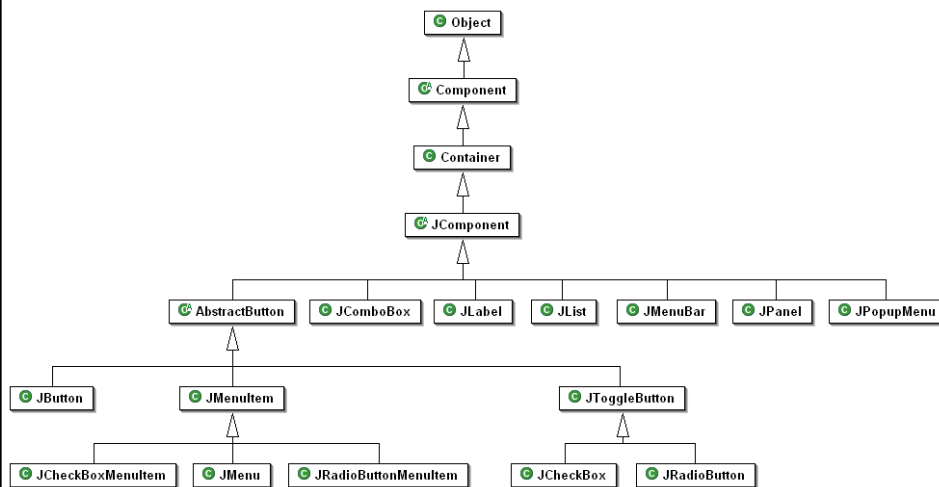
Swing vs. AWT

- Swing es el conjunto de nuevos componentes visuales.
- Habitualmente también se usa como término genérico para referirse a las JFC.
- Su diferencia más importante con la AWT es que los componentes son “lightweight”.
- Para diferenciar los componentes Swing de los AWT, sus nombres están precedidos por una ‘J’.
- Todas las clases Swing se encuentran en el paquete `javax.swing`.*

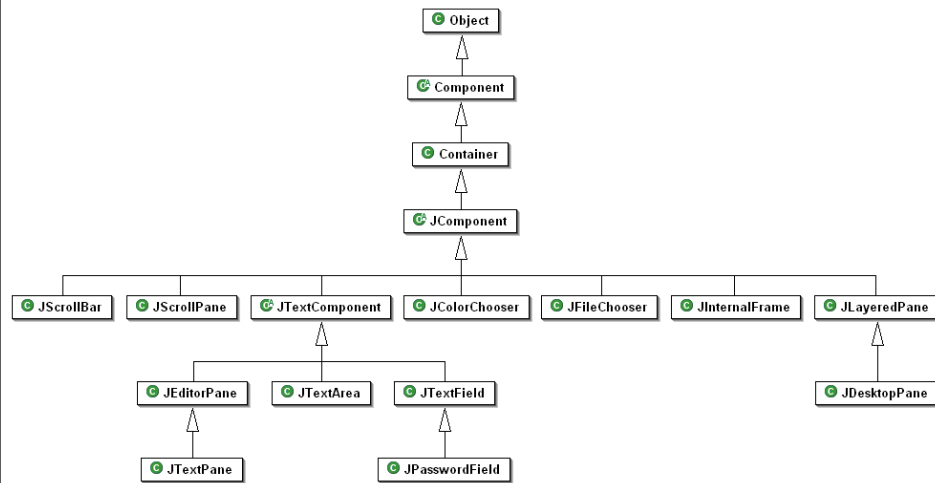
Swing vs. AWT (cont.)

- Nunca debemos mezclar componentes Swing con componentes AWT en una misma aplicación: “Lightweight” vs. “Heavyweight”.
- Swing sigue trabajando con los conceptos de la AWT:
 - Contenedores.
 - Componentes.
 - LayoutManagers.
 - Eventos.

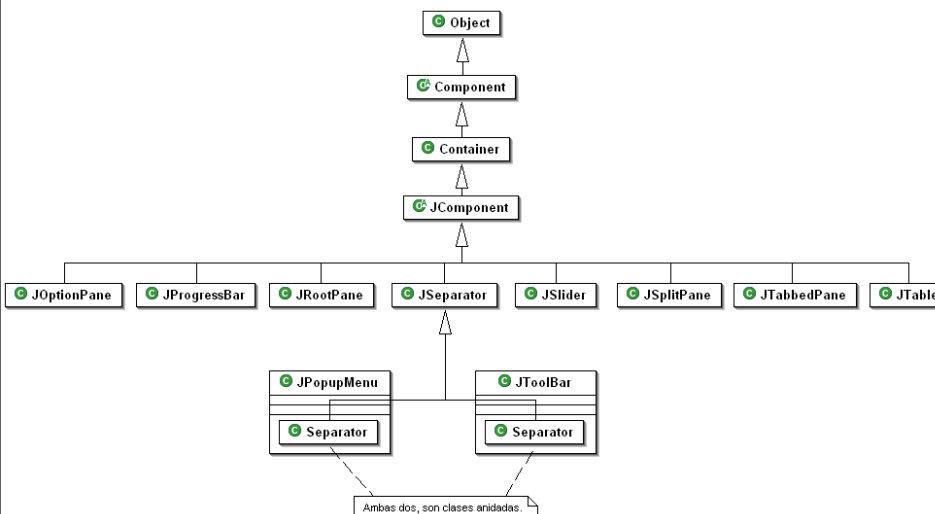
Jerarquía de clases



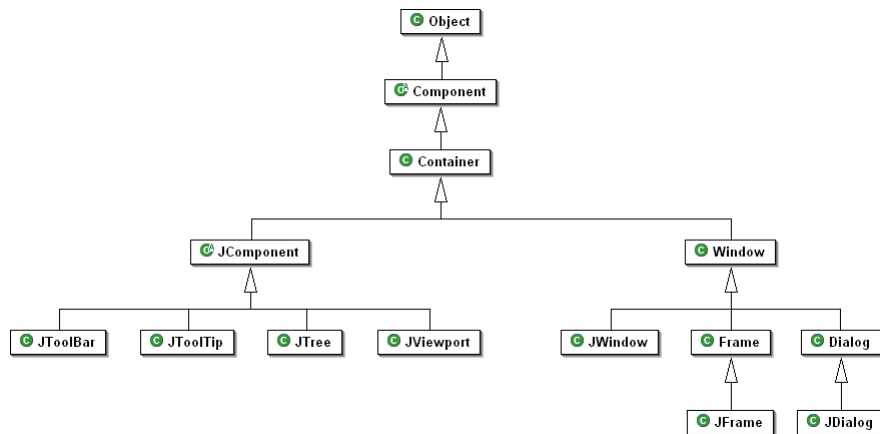
Jerarquía de clases (cont.)



Jerarquía de clases (cont.)



Jerarquía de clases (cont.)



javax.swing.JComponent

- Hereda de la clase `java.awt.Container`.
- Se trata de una clase abstracta que implementa toda la funcionalidad básica de las clases visuales.
- Ayudas emergentes.
- Bordes.
- Gestión del Look & Feel.
- Gestión de la accesibilidad.
- Gestión de teclas asociadas.
- Soporte de Drag & Drop.

Migrando de AWT a Swing

- Eliminar todos los import de paquetes java.awt.*
- Importar el paquete javax.swing.*
- Cambiar cada componente AWT por el Swing más parecido:
 - Frame -> JFrame, Button -> JButton, etc...
- ¡Ojo! No se pueden añadir componentes o establecer LayoutManagers directamente sobre JWindow, JFrame, JDialog o JApplet.
- Hay que hacerlo sobre el Container que devuelve el método: **public** Container getContentPane();



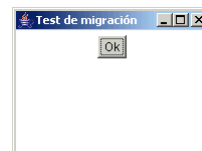
A partir de J2SE 5.0
ya no es necesario.

Migrando de AWT a Swing

AWT:

```
import java.awt.Button;
import java.awt.FlowLayout;
import java.awt.Frame;

public class Test
{
    public static void main(String[] args)
    {
        Frame f = new Frame();
        f.setTitle("Test de migración");
        f.setSize(200,150);
        f.setLayout(new FlowLayout());
        Button b = new Button("Ok");
        f.add(b);
        f.setVisible(true);
    }
}
```

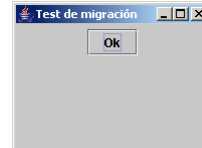


Migrando de AWT a Swing

Swing:

```
import java.awt.FlowLayout;
import javax.swing.JButton;
import javax.swing.JFrame;
```

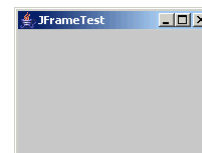
```
public class Test
{
    public static void main(String[] args)
    {
        JFrame f = new JFrame();
        f.setTitle("Test de migración");
        f.setSize(200,150);
        f.getContentPane().setLayout(new FlowLayout());
        JButton b = new JButton("Ok");
        f.getContentPane().add(b);
        f.setVisible(true);
    }
}
```



javax.swing.JFrame

```
import javax.swing.JFrame;
```

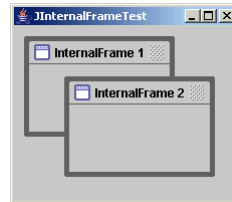
```
public class JFrameTest
{
    public static void main(String[] args)
    {
        JFrame f = new JFrame();
        f.setTitle("JFrameTest");
        f.setSize(200,150);
        f.setVisible(true);
    }
}
```



javax.swing.JInternalFrame

```
import javax.swing.*;

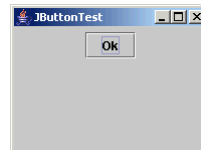
public class JInternalFrameTest
{
    public static void main(String[] args)
    {
        JFrame f = new JFrame();
        f.setTitle("JInternalFrameTest");
        f.getContentPane().setLayout(null);
        f.setSize(230,200);
        JInternalFrame f1 = new JInternalFrame("InternalFrame 1");
        f1.setBounds(10,10,150,100);
        f1.setVisible(true);
        JInternalFrame f2 = new JInternalFrame("InternalFrame 2");
        f2.setBounds(50,50,150,100);
        f2.setVisible(true);
        f.getContentPane().add(f2);
        f.getContentPane().add(f1);
        f.setVisible(true);
    }
}
```



javax.swing.JButton

```
import java.awt.FlowLayout;
import javax.swing.JButton;
import javax.swing.JFrame;

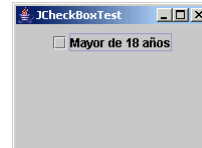
public class JButtonTest
{
    public static void main(String[] args)
    {
        JFrame f = new JFrame();
        f.setTitle("JButtonTest");
        f.setSize(200,150);
        f.getContentPane().setLayout(new FlowLayout());
        JButton b = new JButton("Ok");
        f.getContentPane().add(b);
        f.setVisible(true);
    }
}
```



javax.swing.JCheckBox

```
import java.awt.FlowLayout;
import javax.swing.JCheckBox;
import javax.swing.JFrame;

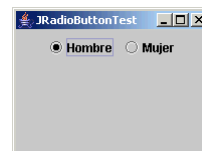
public class JCheckboxTest
{
    public static void main(String[] args)
    {
        JFrame f = new JFrame();
        f.setTitle("JCheckBoxTest");
        f.setSize(200,150);
        f.getContentPane().setLayout(new FlowLayout());
        JCheckBox c = new JCheckBox("Mayor de 18 años");
        f.getContentPane().add(c);
        f.setVisible(true);
    }
}
```



javax.swing.JRadioButton

```
import java.awt.FlowLayout;
import javax.swing.*;
import javax.swing.JFrame;
import javax.swing.JRadioButton;

public class JRadioButtonTest
{
    public static void main(String[] args)
    {
        JFrame f = new JFrame();
        f.setTitle("JRadioButtonTest");
        f.setSize(200,150);
        f.getContentPane().setLayout(new FlowLayout());
        ButtonGroup bg = new ButtonGroup();
        JRadioButton c1 = new JRadioButton("Hombre",true);
        bg.add(c1);
        JRadioButton c2 = new JRadioButton("Mujer",false);
        bg.add(c2);
        f.getContentPane().add(c1);
        f.getContentPane().add(c2);
        f.setVisible(true);
    }
}
```



javax.swing.JToggleButton

```
import java.awt.FlowLayout;
import javax.swing.ButtonGroup;
import javax.swing.JFrame;
import javax.swing.JToggleButton;

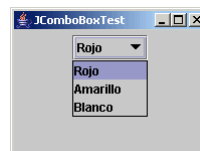
public class JToggleButtonTest
{
    public static void main(String[] args)
    {
        JFrame f = new JFrame();
        f.setTitle("JToggleButtonTest");
        f.setSize(200,150);
        f.getContentPane().setLayout(new FlowLayout());
        ButtonGroup bg = new ButtonGroup();
        JToggleButton b1 = new JToggleButton("Hombre",true);
        bg.add(b1);
        JToggleButton b2 = new JToggleButton("Mujer",false);
        bg.add(b2);
        f.getContentPane().add(b1);
        f.getContentPane().add(b2);
        f.setVisible(true);
    }
}
```



javax.swing.JComboBox

```
import java.awt.FlowLayout;
import javax.swing.JComboBox;
import javax.swing.JFrame;

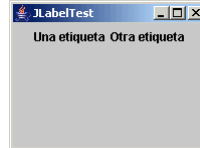
public class JComboBoxTest
{
    public static void main(String[] args)
    {
        JFrame f = new JFrame();
        f.setTitle("JComboBoxTest");
        f.setSize(200,150);
        f.getContentPane().setLayout(new FlowLayout());
        String[] list = {"Rojo","Amarillo","Blanco"};
        JComboBox c = new JComboBox(list);
        f.getContentPane().add(c);
        f.setVisible(true);
    }
}
```



javax.swing.JLabel

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

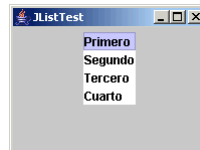
public class JLabelTest
{
    public static void main(String[] args)
    {
        JFrame f = new JFrame();
        f.setTitle("JLabelTest");
        f.setSize(200,150);
        f.getContentPane().setLayout(new FlowLayout());
        JLabel l1 = new JLabel("Una etiqueta");
        JLabel l2 = new JLabel();
        l2.setText("Otra etiqueta");
        f.getContentPane().add(l1);
        f.getContentPane().add(l2);
        f.setVisible(true);
    }
}
```



javax.swing.JList

```
import java.awt.FlowLayout;
import javax.swing.JFrame;
import javax.swing.JList;

public class JListTest
{
    public static void main(String[] args)
    {
        JFrame f = new JFrame();
        f.setTitle("JListTest");
        f.setSize(200,150);
        f.getContentPane().setLayout(new FlowLayout());
        String[] list = {"Primero", "Segundo", "Tercero", "Cuarto"};
        JList l = new JList(list);
        f.getContentPane().add(l);
        f.setVisible(true);
    }
}
```

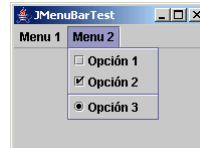


javax.swing.JMenuBar

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

```
public class JMenuBarTest
```

```
{  
    public static void main(String[] args)  
    {  
        JFrame f = new JFrame("JMenuBarTest");  
        f.setSize(200, 150);  
        JMenuBar mb = new JMenuBar();  
        JMenu m1 = new JMenu("Menu 1");  
        m1.add(new JMenuItem("Opción 1"));  
        m1.add(new JMenuItem("Opción 2"));  
        JMenu m2 = new JMenu("Menu 2");  
        m2.add(new JCheckBoxMenuItem("Opción 1"));  
        m2.add(new JCheckBoxMenuItem("Opción 2", true));  
        m2.addSeparator();  
        m2.add(new JRadioButtonMenuItem("Opción 3", true));  
        mb.add(m1);  
        mb.add(m2);  
        f.setJMenuBar(mb);  
        f.setVisible(true);  
    }  
}
```

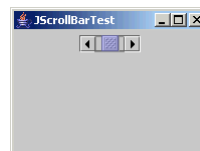


javax.swing.JScrollBar

```
import java.awt.FlowLayout;  
import javax.swing.JFrame;  
import javax.swing.JScrollBar;
```

```
public class JScrollBarTest
```

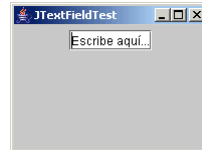
```
{  
    public static void main(String[] args)  
    {  
        JFrame f = new JFrame();  
        f.setTitle("JScrollBarTest");  
        f.setSize(200, 150);  
        f.getContentPane().setLayout(new FlowLayout());  
        JScrollBar sb = new JScrollBar(JScrollBar.HORIZONTAL, 0, 5, -100, 100);  
        f.getContentPane().add(sb);  
        f.setVisible(true);  
    }  
}
```



javax.swing.JTextField

```
import java.awt.FlowLayout;
import javax.swing.JFrame;
import javax.swing.JTextField;
```

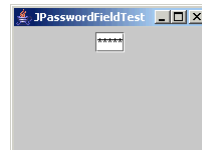
```
public class JTextFieldTest
{
    public static void main(String[] args)
    {
        JFrame f = new JFrame();
        f.setTitle("JTextFieldTest");
        f.setSize(200,150);
        f.getContentPane().setLayout(new FlowLayout());
        JTextField tf = new JTextField("Escribe aquí...");
        f.getContentPane().add(tf);
        f.setVisible(true);
    }
}
```



javax.swing.JPasswordField

```
import java.awt.FlowLayout;
import javax.swing.JFrame;
import javax.swing.JPasswordField;
```

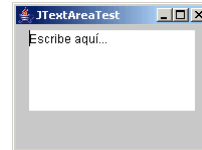
```
public class JPasswordFieldTest
{
    public static void main(String[] args)
    {
        JFrame f = new JFrame();
        f.setTitle("JPasswordFieldTest");
        f.setSize(200,150);
        f.getContentPane().setLayout(new FlowLayout());
        JPasswordField pf = new JPasswordField("chemi");
        f.getContentPane().add(pf);
        f.setVisible(true);
    }
}
```



javax.swing.JTextArea

```
import java.awt.FlowLayout;
import javax.swing.JFrame;
import javax.swing.JTextArea;
```

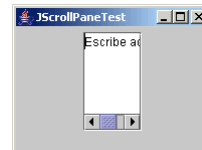
```
public class JTextAreaTest
{
    public static void main(String[] args)
    {
        JFrame f = new JFrame();
        f.setTitle("JTextAreaTest");
        f.setSize(200,150);
        f.getContentPane().setLayout(new FlowLayout());
        JTextArea ta = new JTextArea("Escribe aquí...",5,15);
        f.getContentPane().add(ta);
        f.setVisible(true);
    }
}
```



javax.swing.JScrollPane

```
import java.awt.FlowLayout;
import javax.swing.JFrame;
import javax.swing.JScrollPane;
import javax.swing.JTextArea;
```

```
public class JScrollPaneTest
{
    public static void main(String[] args)
    {
        JFrame f = new JFrame();
        f.setTitle("JScrollPaneTest");
        f.setSize(200,150);
        f.getContentPane().setLayout(new FlowLayout());
        JTextArea ta = new JTextArea("Escribe aquí...",5,5);
        JScrollPane p = new JScrollPane(ta);
        f.getContentPane().add(p);
        f.setVisible(true);
    }
}
```

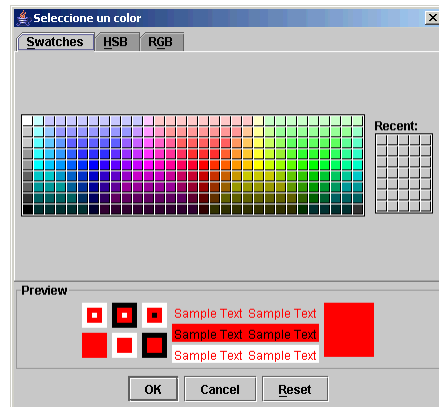


javax.swing.JColorChooser

```
import java.awt.Color;
import javax.swing.JColorChooser;
import javax.swing.JFrame;

public class JColorChooserTest
{
    public static void main(String[] args)
    {
        JFrame f = new JFrame();
        f.setTitle("JColorChooserTest");
        f.setSize(200,150);
        f.setVisible(true);

        Color c = JColorChooser.showDialog(f, "Seleccione un color", Color.RED);
        System.out.println("El color seleccionado es: " + c);
    }
}
```

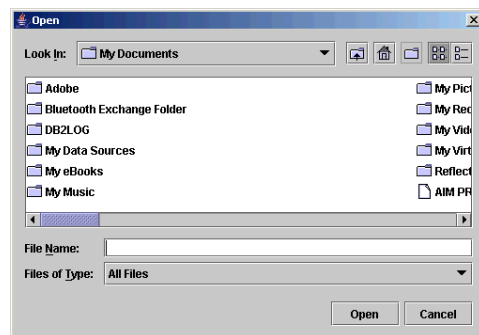


javax.swing.JFileChooser

```
import javax.swing.JFileChooser;
import javax.swing.JFrame;

public class Test
{
    public static void main(String[] args)
    {
        JFrame f = new JFrame();
        f.setTitle("JFileChooserTest");
        f.setSize(200,150);
        f.setVisible(true);

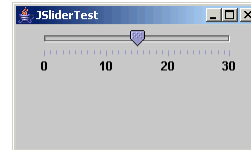
        JFileChooser fc = new JFileChooser();
        int op = fc.showOpenDialog(f);
        if(op == JFileChooser.APPROVE_OPTION)
            System.out.println(fc.getSelectedFile());
    }
}
```



javax.swing.JSlider

```
import java.awt.FlowLayout;
import javax.swing.JFrame;
import javax.swing.JSlider;

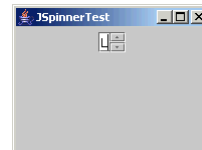
public class JSliderTest
{
    public static void main(String[] args)
    {
        JFrame f = new JFrame();
        f.setTitle("JSliderTest");
        f.setSize(200,150);
        f.getContentPane().setLayout(new FlowLayout());
        JSlider s = new JSlider(JSlider.HORIZONTAL,0,30,15);
        s.setMajorTickSpacing(10);
        s.setMinorTickSpacing(1);
        s.setPaintTicks(true);
        s.setPaintLabels(true);
        f.getContentPane().add(s);
        f.setVisible(true);
    }
}
```



javax.swing.JSpinner

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

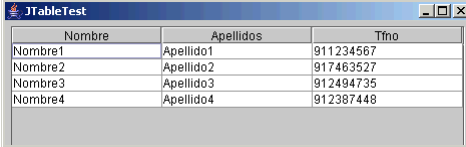
public class JSpinnerTest
{
    public static void main(String[] args)
    {
        JFrame f = new JFrame();
        f.setTitle("JSpinnerTest");
        f.setSize(200,150);
        f.getContentPane().setLayout(new FlowLayout());
        String[] dias = { "L", "M", "X", "J", "V", "S", "D" };
        SpinnerListModel modelo = new SpinnerListModel(dias);
        JSpinner s = new JSpinner(modelo);
        f.getContentPane().add(s);
        f.setVisible(true);
    }
}
```



javax.swing.JTable

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

public class JTableTest
{
    public static void main(String[] args)
    {
        JFrame f = new JFrame();
        f.setTitle("JTableTest");
        f.setSize(200,150);
        f.getContentPane().setLayout(new FlowLayout());
        Object[][] datos =
        {
            { "Nombre1", "Apellido1", new Integer(911234567) },
            { "Nombre2", "Apellido2", new Integer(917463527) },
            { "Nombre3", "Apellido3", new Integer(912494735) },
            { "Nombre4", "Apellido4", new Integer(912387448) },
        };
        String[] columnas = {"Nombre", "Apellidos", "Tfno"};
        JTable t = new JTable(datos, columnas);
        JScrollPane sp = new JScrollPane(t);
        f.getContentPane().add(sp);
        f.setVisible(true);
    }
}
```

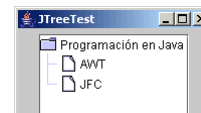


Nombre	Apellidos	Tfno
Nombre1	Apellido1	911234567
Nombre2	Apellido2	917463527
Nombre3	Apellido3	912494735
Nombre4	Apellido4	912387448

javax.swing.JTree

```
import java.awt.FlowLayout;
import javax.swing.*;
import javax.swing.tree.DefaultMutableTreeNode;

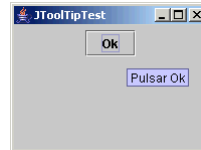
public class JTreeTest
{
    public static void main(String[] args)
    {
        JFrame f = new JFrame();
        f.setTitle("JTreeTest");
        f.setSize(200,150);
        f.getContentPane().setLayout(new FlowLayout());
        DefaultMutableTreeNode titulo = new DefaultMutableTreeNode("Programación en Java");
        DefaultMutableTreeNode capitulo = new DefaultMutableTreeNode("AWT");
        titulo.add(capitulo);
        capitulo = new DefaultMutableTreeNode("JFC");
        titulo.add(capitulo);
        JTree tree = new JTree(titulo);
        JScrollPane sp = new JScrollPane(tree);
        f.getContentPane().add(sp);
        f.setVisible(true);
    }
}
```



java.swing.JToolTip

```
import java.awt.FlowLayout;
import javax.swing.JButton;
import javax.swing.JFrame;

public class JToolTipTest
{
    public static void main(String[] args)
    {
        JFrame f = new JFrame();
        f.setTitle("JToolTipTest");
        f.setSize(200,150);
        f.getContentPane().setLayout(new FlowLayout());
        JButton b = new JButton("Ok");
        b.setToolTipText("Pulsar Ok");
        f.getContentPane().add(b);
        f.setVisible(true);
    }
}
```



javax.swing.JDialog

```
import javax.swing.JDialog;
import javax.swing.JFrame;

public class JDialogTest
{
    public static void main(String[] args)
    {
        JFrame f = new JFrame();
        f.setTitle("JFrameTest");
        f.setSize(200,150);
        f.setVisible(true);

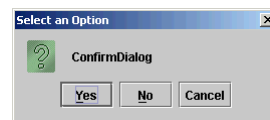
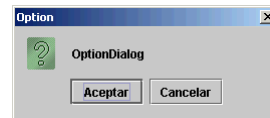
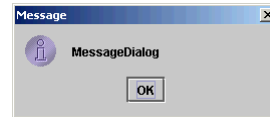
        JDialog d = new JDialog(f);
        d.setTitle("JDialogTest");
        d.setBounds(50,50,70,50);
        d.setVisible(true);
    }
}
```



javax.swing.JOptionPane

```
import javax.swing.*;

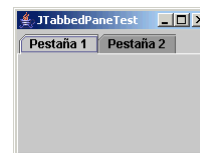
public class JOptionPaneTest
{
    public static void main(String[] args)
    {
        JFrame f = new JFrame();
        f.setTitle("JOptionPaneTest");
        f.setSize(200,150);
        f.setVisible(true);
        JOptionPane.showMessageDialog(f, "MessageDialog");
        Object[] opciones = { "Aceptar", "Cancelar" };
        int i = JOptionPane.showOptionDialog(f,"OptionDialog",
        "Option",JOptionPane.YES_NO_OPTION,
        JOptionPane.QUESTION_MESSAGE,null,
        opciones,opciones[0]);
        i = JOptionPane.showConfirmDialog(f,"ConfirmDialog");
        String s = JOptionPane.showInputDialog(f,"InputDialog");
    }
}
```



javax.swing.JTabbedPane

```
import javax.swing.*;
import javax.swing.JPanel;
import javax.swing.JTabbedPane;

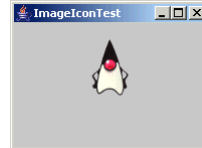
public class JTabbedPaneTest
{
    public static void main(String[] args)
    {
        JFrame f = new JFrame();
        f.setTitle("JTabbedPaneTest");
        f.setSize(200,150);
        JTabbedPane tabbedPane = new JTabbedPane();
        JPanel panel1 = new JPanel();
        tabbedPane.addTab("Pestaña 1", panel1);
        JPanel panel2 = new JPanel();
        tabbedPane.addTab("Pestaña 2", panel2);
        f.getContentPane().add(tabbedPane);
        f.setVisible(true);
    }
}
```



javax.swing.ImageIcon

```
import java.awt.FlowLayout;
import javax.swing.ImageIcon;
import javax.swing.JFrame;
import javax.swing.JLabel;

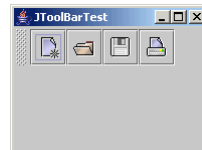
public class ImageIconTest
{
    public static void main(String[] args)
    {
        JFrame f = new JFrame();
        f.setTitle("ImageIconTest");
        f.setSize(200,150);
        f.getContentPane().setLayout(new FlowLayout());
        JLabel l = new JLabel();
        l.setIcon(new ImageIcon("duke.gif")); // Soporta formatos GIF, JPG y PNG.
        f.getContentPane().add(l);
        f.setVisible(true);
    }
}
```



javax.swing.JToolBar

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

public class JToolBarTest
{
    public static void main(String[] args)
    {
        JFrame f = new JFrame();
        f.setTitle("JToolBarTest");
        f.setSize(200,150);
        JToolBar tb = new JToolBar();
        JButton b = new JButton(new ImageIcon("New24.gif"));
        tb.add(b);
        b = new JButton(new ImageIcon("Open24.gif"));
        tb.add(b);
        b = new JButton(new ImageIcon("Save24.gif"));
        tb.add(b);
        b = new JButton(new ImageIcon("Print24.gif"));
        tb.add(b);
        f.getContentPane().add(tb,BorderLayout.NORTH);
        f.setVisible(true);
    }
}
```



Nota:
Galería de iconos para el Java Look & Feel "Metal":
<http://java.sun.com/developer/techDocs/hi/repository/>

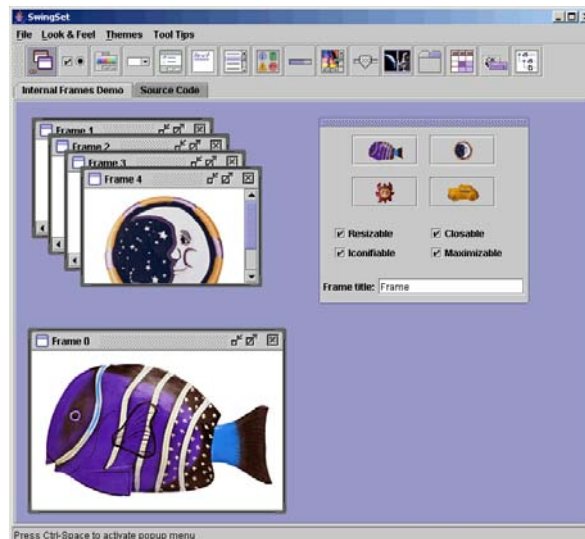
javax.swing.JSplitPane

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

public class JSplitPaneTest
{
    public static void main(String[] args)
    {
        JFrame f = new JFrame();
        f.setTitle("JSplitPaneTest");
        f.setSize(275,252);
        JLabel l1 = new JLabel(new ImageIcon("argh.jpg"));
        l1.setMinimumSize(new Dimension(20, 20));
        JLabel l2 = new JLabel(new ImageIcon("comic.jpg"));
        l2.setMinimumSize(new Dimension(20, 20));
        JSplitPane sp = new JSplitPane(JSplitPane.HORIZONTAL_SPLIT,l1,l2);
        sp.setContinuousLayout(true);
        sp.setOneTouchExpandable(true);
        sp.setDividerLocation(100);
        f.getContentPane().add(sp);
        f.setVisible(true);
    }
}
```



Aplicación SwingSet2



Incluida en las demos del SDK. Arrancar mediante: `java -jar SwingSet2.jar`

Layout Managers

- Todos los contenedores Swing tienen asociado un `LayoutManager` para coordinar el tamaño y la situación de sus componentes.
- JPanel -> `FlowLayout`
- JFrame -> `BorderLayout`
- Cada Layout se caracteriza por el estilo que emplea para situar los componentes en su interior:
 - Alineación de izquierda a derecha.
 - Alineación en rejilla.
 - Alineación del frente a atrás.

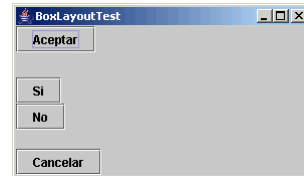
Nuevos Layout Managers

- Implementan el interface `java.awt.LayoutManager`.
- `BoxLayout`: sitúa los componentes en línea vertical u horizontal. Respeta sus tamaños.
La clase `javax.swing.Box` tiene métodos para crear zonas con espacio como `createVerticalStrut(int)` y zonas que absorban los espacios como `createVerticalGlue(int)`.
- `SpringLayout`: permite definir la relación (distancia) entre los límites de los distintos controles.
- `ScrollPaneLayout`, `ViewportLayout`: utilizados internamente por Swing para algunos de los componentes como el `ScrollPane`.

javax.swing.BoxLayout

```
import javax.swing.Box;
import javax.swing.BoxLayout;
import javax.swing.JButton;
import javax.swing.JFrame;

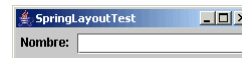
public class BoxLayoutTest
{
    public static void main(String[] args)
    {
        JFrame f = new JFrame();
        f.setTitle("BoxLayoutTest");
        f.setSize(300,150);
        f.getContentPane().setLayout(new BoxLayout(f.getContentPane(),BoxLayout.Y_AXIS));
        f.getContentPane().add(new JButton("Aceptar"));
        f.getContentPane().add(Box.createVerticalStrut(25));
        f.getContentPane().add(new JButton("Si"));
        f.getContentPane().add(new JButton("No"));
        f.getContentPane().add(Box.createVerticalGlue());
        f.getContentPane().add(new JButton("Cancelar"));
        f.setVisible(true);
    }
}
```



javax.swing.SpringLayout

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

public class SpringLayoutTest
{
    public static void main(String[] args)
    {
        JFrame f = new JFrame();
        f.setTitle("SpringLayoutTest");
        Container container = f.getContentPane();
        SpringLayout layout = new SpringLayout();
        container.setLayout(layout);
        JLabel label = new JLabel("Nombre: ");
        JTextField text = new JTextField(15);
        f.getContentPane().add(label);
        f.getContentPane().add(text);
        layout.putConstraint(SpringLayout.WEST, label, 5, SpringLayout.WEST, container);
        layout.putConstraint(SpringLayout.NORTH, label, 5, SpringLayout.NORTH, container);
        layout.putConstraint(SpringLayout.WEST, text, 5, SpringLayout.EAST, label);
        layout.putConstraint(SpringLayout.NORTH, text, 5, SpringLayout.NORTH, container);
        layout.putConstraint(SpringLayout.EAST, container, 5, SpringLayout.EAST, text);
        layout.putConstraint(SpringLayout.SOUTH, container, 5, SpringLayout.SOUTH, text);
        f.pack();
        f.setVisible(true);
    }
}
```



Nuevos tipos de eventos

AncestorEvent	Un padre se ha añadido, movido o eliminado.
CaretEvent	El cursor en un texto ha cambiado.
ChangeEvent	El estado ha cambiado.
DocumentEvent	Los atributos de un Document han cambiado, se ha insertado o se ha eliminado contenido.
HyperlinkEvent	Se ha activado, entrado o salido de un hyperlink.
InternalFrameEvent	Se ha activado, cerrado, desactivado, minimizado, maximizado o abierto un internal frame.
ListDataEvent	Ha cambiado el contenido.

Nuevos tipos de eventos

ListSelectionEvent	Ha cambiado la selección en una lista.
MenuDragMouseEvent	El ratón se ha arrastrado, entrado, salido, soltado en un menú.
MenuEvent	Se ha seleccionado o deseleccionado un menú.
MenuKeyEvent	Se ha pulsado, soltado o tecleado sobre un menú.
PopupMenuEvent	Se ha mostrado, ocultado o seleccionado un menú emergente.
TableColumnModelEvent	Se ha añadido, eliminado, movido, redimensionada o seleccionada una columna.
TableModelEvent	El modelo de la tabla ha cambiado.

Nuevos tipos de eventos

TreeExpansionEvent	Se ha abierto o cerrado el árbol.
TreeModelEvent	Se ha cambiado, añadido o eliminado un elemento del árbol.
TreeSelectionEvent	Ha cambiado la selección en el árbol.
UndoableEditEvent	Se ha realizado una operación que no se puede deshacer.

Origen de eventos

Componente Swing	Listener
AbstractButton DefaultButtonModel JComboBox JFileChooser JTextField Timer	ActionListener
JScrollBar	AdjustmentListener
JComponent	AncestorListener
JTextComponent	CaretListener
DefaultCellEditor DefaultTreeCellEditor	CellEditorListener

Origen de eventos

Componente Swing	Listener
AbstractButton DefaultBoundedRangeModel DefaultButtonModel DefaultCaret DefaultColorSelectionModel DefaultSingleSelectionModel JProgressBar JSlider JTabbedPane JViewport MenuSelectionManager StyleContext StyleContext.NamedStyle	ChangeListener
DefaultTableColumnModel	ColumnModelListener

Origen de eventos

Componente Swing	Listener
AbstractDocument DefaultStyledDocument	DocumentListener
JEditorPane	HyperlinkListener
JInternalFrame	InternalFrameListener
AbstractButton DefaultButtonModel JComboBox	ItemListener
AbstractListModel	ListDataListener
DefaultListSelectionModel JList	ListSelectionListener
JMenuItem	MenuDragMouseListener
JMenuItem	MenuKeyListener

Origen de eventos

Componente Swing	Listener
JMenu	MenuListener
JPopupMenu	PopupMenuListener
AbstractAction DefaultTreeSelectionModel JComponent SwingPropertyChangeSupport TableColumn UIDefaults UIManager	PropertyChangeListener
AbstractTableModel	TableModelListener
JTree	TreeExpansionListener
DefaultTreeModel	TreeModelListener
DefaultTreeSelectionModel JTree	TreeSelectionListener

Origen de eventos

Componente Swing	Listener
AbstractDocument UndoableEditSupport	UndoableEditListener
JComponent	VetoableChangeListener

Métodos de los interfaces

Listener interface	Métodos
AncestorListener	ancestorAdded ancestorMoved ancestorRemoved
CaretListener	caretUpdate
CellEditorListener	editingCanceled editingStopped
ChangeListener	stateChanged
DocumentListener	changedUpdate insertUpdate removeUpdate
HyperlinkListener	hyperlinkUpdate

Métodos de los interfaces

Listener interface	Métodos
InternalFrameListener	internalFrameActivated internalFrameClosed internalFrameClosing internalFrameDeactivated internalFrameDeiconified internalFrameIconified internalFrameOpened
ListDataListener	contentsChanged intervalAdded intervalRemoved
ListSelectionListener	valueChanged

Métodos de los interfaces

Listener interface	Métodos
MenuDragMouseListener	menuDragMouseDragged menuDragMouseEntered menuDragMouseExited menuDragMouseReleased
MenuKeyListener	menuKeyPressed menuKeyReleased menuKeyTyped
MenuListener	menuCanceled menuDeselected menuSelected

Métodos de los interfaces

Listener interface	Métodos
MouseListener	mouseClicked mouseDragged mouseEntered mouseExited mouseMoved mousePressed mouseReleased
PopupMenuListener	popupMenuCanceled popupMenuWillBecomeInvisible popupMenuWillBecomeVisible

Métodos de los interfaces

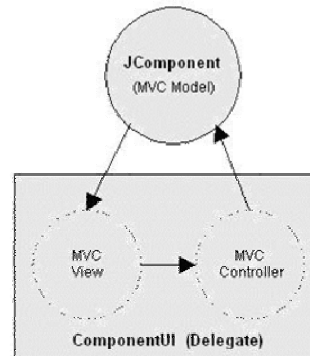
Listener interface	Métodos
TableColumnModelListener	columnAdded columnMarginChanged columnMoved columnRemoved columnSelectionChanged
TableModelListener	tableChanged
TreeExpansionListener	treeCollapsed treeExpanded
TreeModelListener	treeNodesChanged treeNodesInserted treeNodesRemoved treeStructureChanged

Métodos de los interfaces

Listener interface	Métodos
TreeSelectionListener	valueChanged
UndoableEditListener	undoableEditHappened

Look & Feel

AWT: Heavyweight Swing: Lightweight



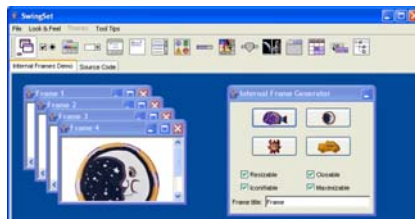
Look & Feel

Windows Look & Feel. Funciona solo en plataforma Microsoft Windows.



`com.sun.java.swing.plaf.windows.WindowsLookAndFeel`

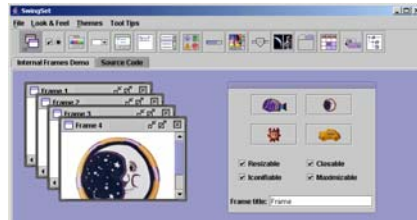
En WinXP, el Look & Feel de Windows es diferente.



`com.sun.java.swing.plaf.windows.WindowsLookAndFeel`

Look & Feel

- Java Look & Feel (“Metal”). Multiplataforma.



`javax.swing.plaf.metal.MetalLookAndFeel`

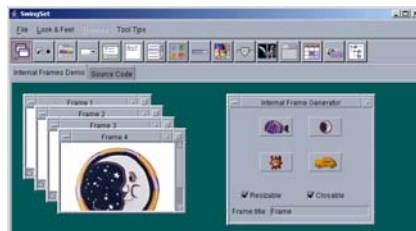
- Mac Look & Feel (“Aqua”). Funciona solo en plataforma Macintosh.



`com.sun.java.swing.plaf.mac.MacLookAndFeel`

Look & Feel

- Motif Look & Feel. Multiplataforma.



`com.sun.java.swing.plaf.motif.MotifLookAndFeel`

- GTK Look & Feel. Multiplataforma.



`com.sun.java.swing.plaf.gtk.GTKLookAndFeel`

Look & Feel



Existen distintas alternativas para seleccionar el Look & Feel de una aplicación Java:



Por línea de comando:

```
java -Dswing.defaultlaf=com.sun.java.swing.plaf.motif.MotifLookAndFeel MiApp
```



Mediante el fichero swing.properties (localizado en el directorio \lib del JRE):

```
# Swing properties
swing.defaultlaf=com.sun.java.swing.plaf.gtk.GTKLookAndFeel
```



Por código de forma estática (al inicio del programa):

```
try
{
    UIManager.setLookAndFeel("javax.swing.plaf.metal.MetalLookAndFeel");
}
catch (Exception ex) { }
```

Look & Feel




Por código de forma dinámica:

```
try
{
    UIManager.setLookAndFeel("javax.swing.plaf.metal.MetalLookAndFeel");
    SwingUtilities.updateComponentTreeUI(frame);
    frame.pack();
}
catch (Exception ex)
{
}
```

Nota: siendo frame, el contenedor raíz de la aplicación.



En J2SE 5.0, el Java Look & Feel “Metal” tiene una configuración distinta (“theme Ocean”). 

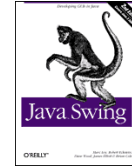


Bibliografía



Java Swing (2nd edition).

Marc Loy, Robert Eckstein, Dave Wood, James Elliot y Brian Cole.
O'Reilly



Graphic Java 2, Volume 2: Swing (3rd edition)

David M. Geary.
Prentice Hall.



The Swing Tutorial (2nd edition)

Kathy Walrath y Mary Campione.
Addison-Wesley.



The Swing tutorial (on-line)

<http://java.sun.com/docs/books/tutorial/uiswing/>