Programación 2: Java Swing
UNERG - AIS

Importar paquetes

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

Contenedores

JFrame: Ventana de la aplicación, contenedor principal
JDialog: Ventana de tipo diálogo, también puede ser un contenedor principal.
JPanel: Permite la creación de paneles independientes donde se almacenan otros componentes.
JScrollPane: permite la vinculación de barras de desplazamiento en un contenedor.
JSplitPane: permite la creación de un contenedor dividido en 2 secciones.
JTabbedPane: Permite la creación de pestañas, cada pestaña representa un contenedor independiente.
JDesktopPane: Permite crear ventanas dentro de una ventana principal
JToolBar: Permite introducir una Barra de herramientas

Crear Ventana

```
JFrame ventana = new JFrame();
ventana.setSize(400, 300);
ventana.setLocationRelativeTo(null);
ventana.setDefaultCloseOperation(
      JFrame.EXIT_ON_CLOSE);
ventana.setTitle("Titulo de la Ventana");
ventana.setResizable(false);
// …
ventana.setVisible(true);
```

Administradores de diseño

FlowLayout coloca los componentes horizontalmente.
BorderLayout coloca los componentes en cinco secciones: Norte, Sur, Este, Oeste y Centro.
BoxLayout Parecido a FlowLayout, coloca los componentes, tanto horizontal como vertical
CardLayout Coloca los componentes superpuestos en la misma zona
GridLayout se crea una cuadrícula (filas, columnas), y los componentes se colocan en cada celda de esa cuadrícula.
GridBagLayout Igual al *GridLayout*, con la diferencia que los Componentes no necesitan tener el mismo tamaño.

Agregar panel a la ventana

```
JPanel panel = new JPanel(new BorderLayout());
ventana.add(panel);

JPanel panel2 = new JPanel(new FlowLayout());
ventana.add(panel2);
```

Agregar nuevos paneles al diseño

```
JPanel top = new JPanel(new FlowLayout());
panel.add(top, BorderLayout.NORTH);

JPanel bottom = new JPanel(
      new FlowLayout(FlowLayout.RIGHT));
panel.add(bottom, BorderLayout.SOUTH);
```

Componentes Atómicos

Los componentes atómicos son los elementos que no pueden almacenar otros objetos o componentes gráficos.
JLabel Permite Vincular Etiquetas, tanto de texto como de imágenes
JButton Permite vincular Botones simples.
JCheckBox Son Casilla de verificación, ideal para selección múltiples.
JRadioButton Permite presentar opciones de selección similares a las checkbox, solo que el enfoque de estas es de única selección.
JToggleButton Botón que al oprimirlo se quedará presionado hasta que se ejecute otro evento.
JComboBox Permite mostrar una lista de elementos como un combo de selección.
JScrollBar Permite mostrar una barra de desplazamiento, regularmente usada en áreas de texto o paneles donde el contenido es mayor que el tamaño del componente.
JSeparator Permite separar opciones, es una barra simple.
JSlider Permite vincular un Deslizador en nuestra ventana.
JSpinner permite vincular una caja de texto con botones integrados para seleccionar algún valor.
JProgressBar Establece una barra de progreso.

Ejemplo Componentes Atómicos

```
JButton aceptar = new JButton("Aceptar");
JButton cancelar = new JButton("Cancelar");
bottom.add(aceptar);
bottom.add(cancelar);
```

Componentes de Texto

Son todos aquellos que nos permiten procesar cadenas de texto, sea como entrada o salida de información.
JTextField Permite introducir un campo de texto simple.
JFormattedTextField: Permite introducir un campo de texto con formato, (si definimos que solo recibe números no permitirá letras...)
JPasswordField Campo de texto que oculta los caracteres ingresados.

Prof. Alberto G. Rodríguez U.
arodriguez@unerg.edu.ve

Componentes de Texto (Cont.)

JTextArea Permite vincular un área de texto donde el usuario ingresará información o simplemente para presentar cadenas de texto.
JEditorPane Permite vincular un área de texto con propiedades de formato.
JTextPane Similar al anterior, permitiendo otras opciones de formato, colores, iconos entre otros.

Ejemplo Componentes de Texto

```
JTextField text = new JTextField("texto");
bottom.add(text);
JTextArea area = new JTextArea(rows, columns);
bottom.add(area);
```

Componentes de Menús

Estos componentes permiten vincular opciones de menú en nuestras ventanas, tipo menú principal, como por ejemplo el conocido Inicio, Archivo, Edición etc.
JMenuBar Permite vincular una barra de menús.
JMenu Permite vincular botones o enlaces que al ser pulsados despliegan un menú principal.
JMenuItem Botón u opción que se encuentra en un menú.
JCheckBoxMenuItem Elemento del menú como opciones de checkbox.
JRadioButtonMenuItem Elemento del menú como botón de selección.
JPopupMenu Opciones de menú emergentes.

Componentes Complejos

Estos son componentes un poco más avanzados, cumplen con funciones más enfocadas a procesos específicos y complejos, como por ejemplo obtener gran cantidad de información de una base de datos, trabajo con nodos, colores entre otros.
JTable Permite vincular una tabla de datos con sus respectivas filas y columnas.
JTree Carga un árbol donde se establece cierta jerarquía visual, tipo directorio.
JList Carga una lista de elementos, dependiendo de las propiedades puede tenerse una lista de selección múltiple.
JFileChooser Es un componente que permite la búsqueda y selección de ficheros entre otras.
JColorChooser Componente que permite cargar un panel selector de color
JOptionPane No es algo complejo sino más un componente independiente que permite mostrar un cuadro de diálogo personalizable.

Administrador de eventos

Los eventos son las acciones que puede realizar el usuario, al realizar un evento se produce una serie de acciones
ActionListener Administra la acción del evento del componente que se encuentra suscrito.

Otros Administradores de eventos

ComponentListener	*WindowListener*
ItemListener	*AdjustmentListener*
KeyListener	*ContainerListener*
MouseListener	*MouseMotionListener*
TextListener	*FocusListener*

Ejemplo de Ventana con ActionListener

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

public class View implements ActionListener{
  private JButton btn;
  private JFrame ventana;

  public View() {
    ventana = new JFrame();
    ventana.setTitle("Ejemplo ActionListener");
    ventana.setSize(400, 300);
    ventana.setLayout(new BorderLayout());

    btn = new JButton("Click aquí");
    btn.addActionListener(this);

    ventana.add(btn, BorderLayout.SOUTH);
    ventana.setVisible(true);
  }

  @Override
  public void actionPerformed(ActionEvent arg0){
    if(arg0.getSource().equals(this.btn)){
      JOptionPane.showMessageDialog(ventana,
          "Ha hecho click en el botón");
    }
  }

  public static void main(String[] args){
    new View();
  }
}
```

Prof. Alberto G. Rodríguez U.
arodriguez@unerg.edu.ve