



Confección de Interfaces de Usuario

Tema 1. Diseño de Interfaces

Objetivos

- Librerías de programación de interfaces
- Introducción a Swing
- Gestión de eventos



Librerías de diseño para Java

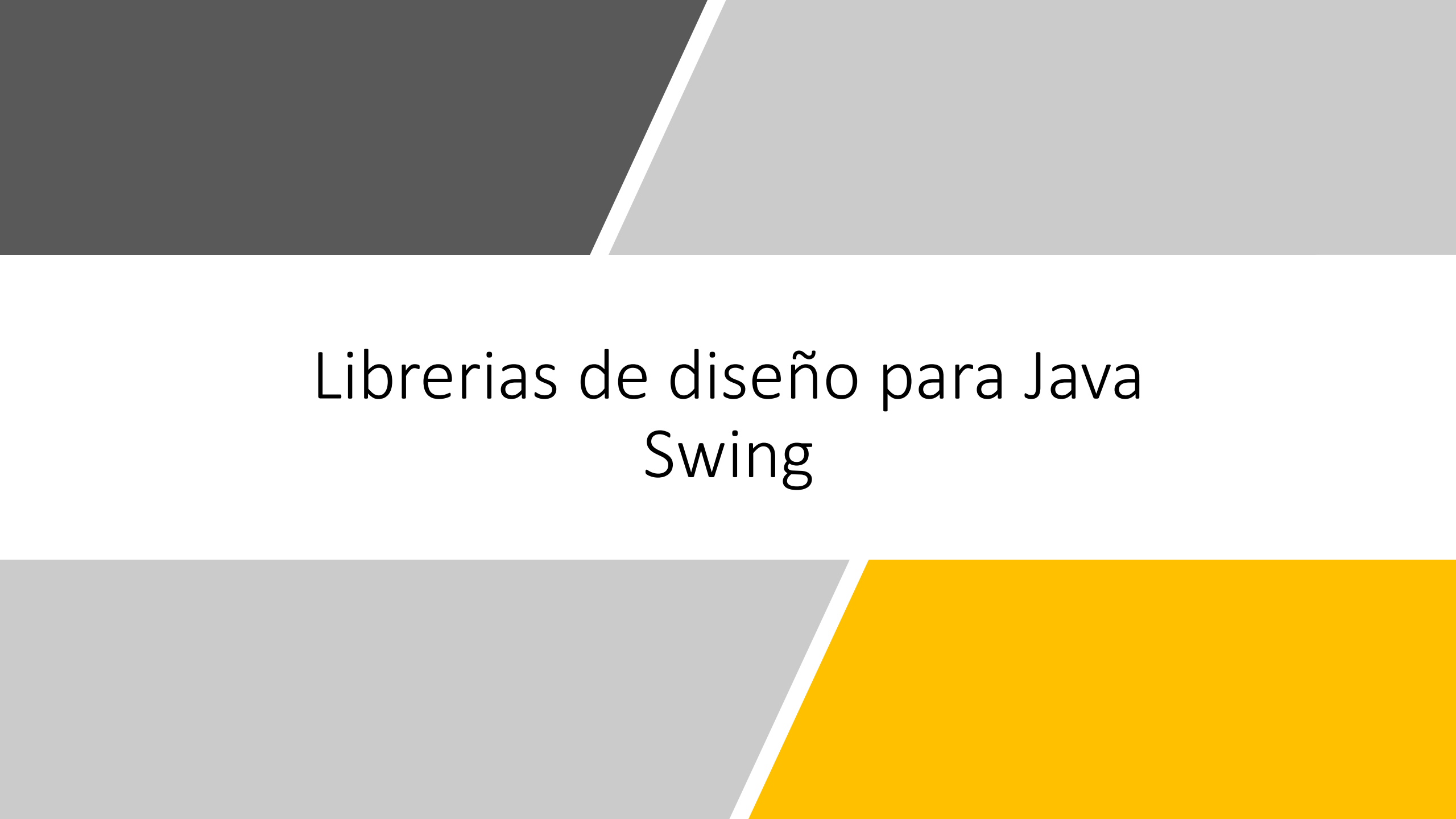
Abstract Widget Toolkit (AWT)

AWT

- Es la librería más antigua y está integrada en el SDK de Java.
- Compatible con todos los sistemas
- Solo usa los elementos del sistema.

<https://docs.oracle.com/javase/8/docs/api/java/awt/package-summary.html>

```
var button=new  
Button("Hola mundo");
```

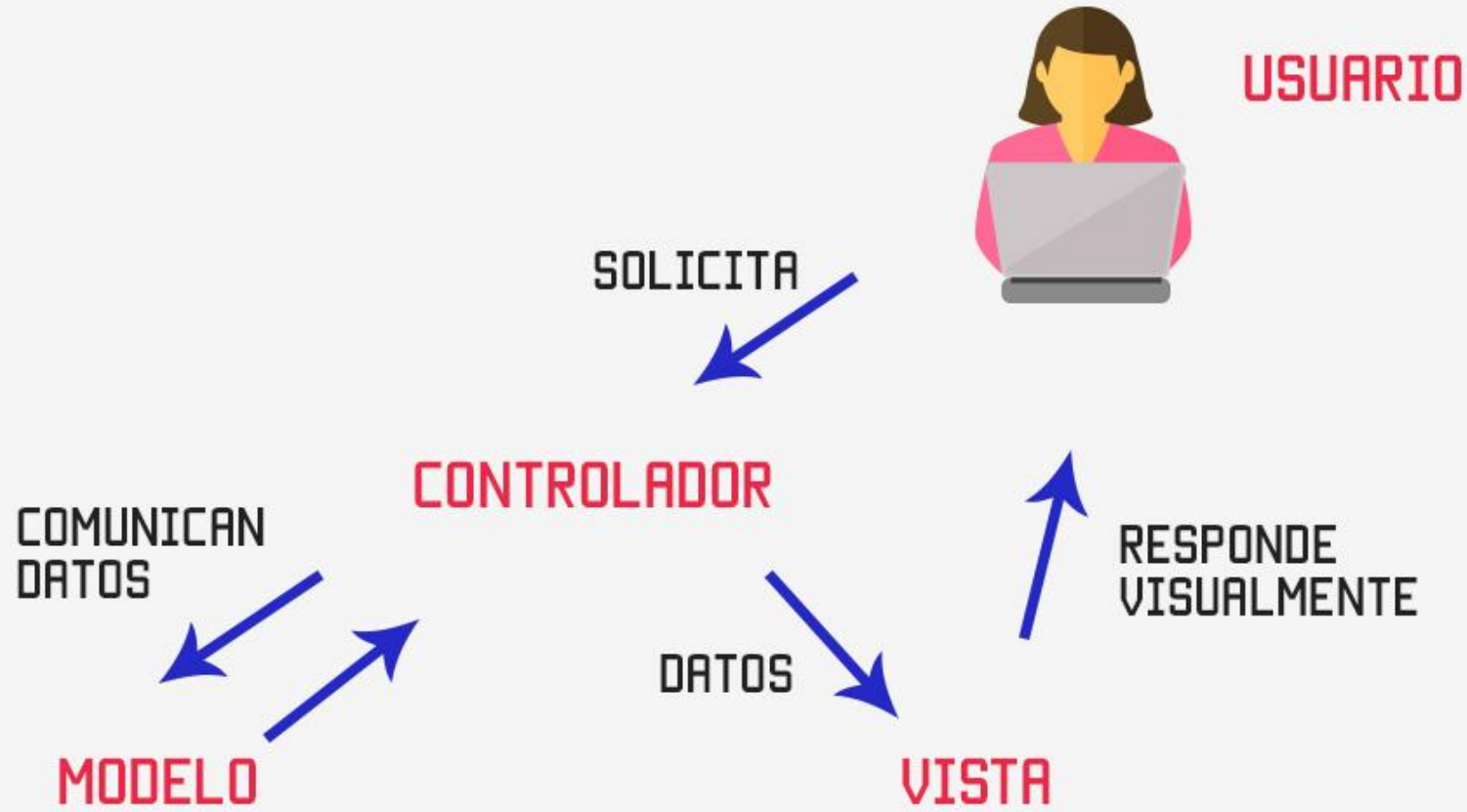


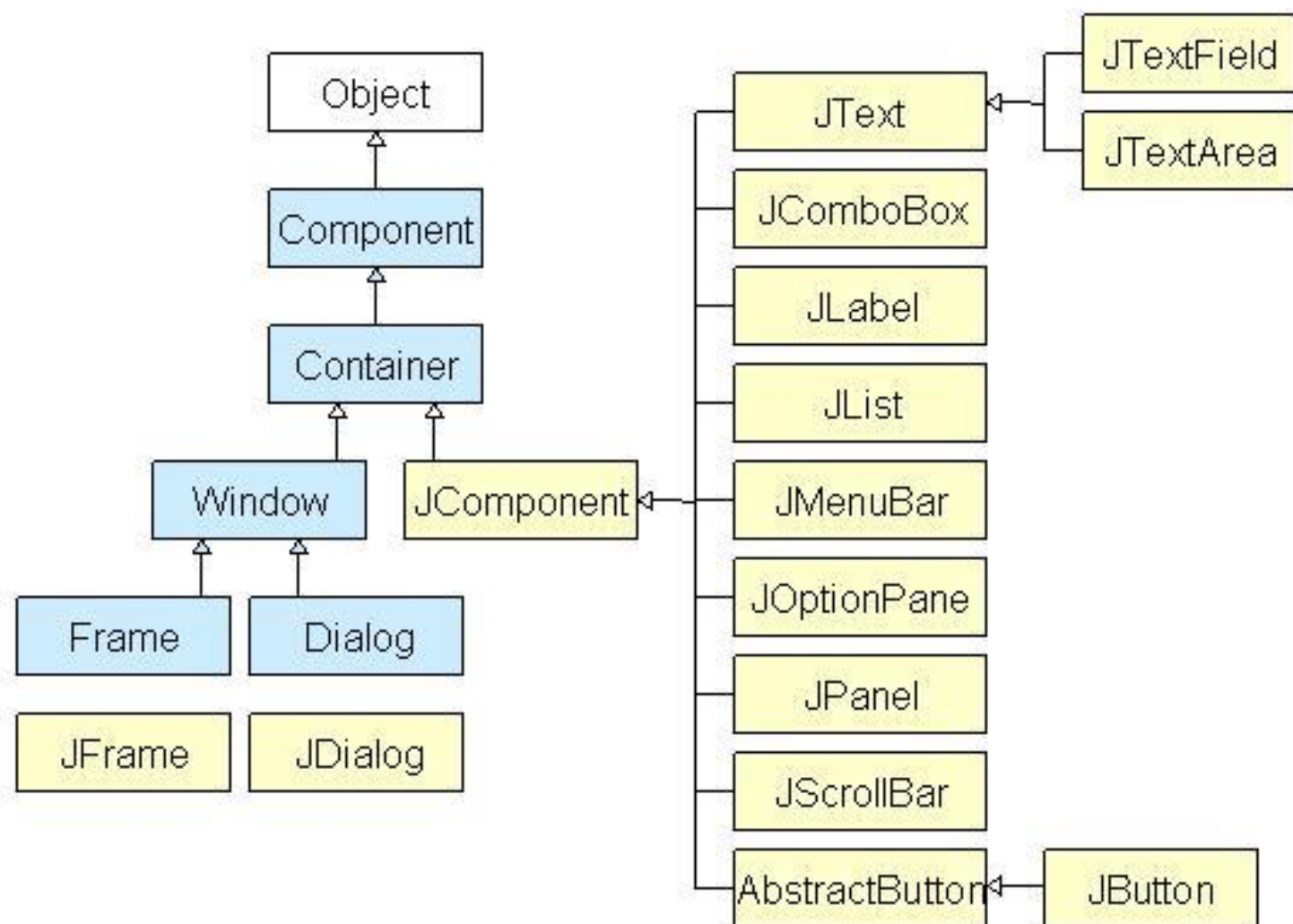
Librerías de diseño para Java Swing

Swing

- Independiente de la plataforma.
- Extensible
- Personalizable
- Sigue el modelo MVC

<https://docs.oracle.com/javase/8/docs/api/javax/swing/package-summary.html>





Clases principales de Swing

JFrame

Establece la ventana donde mostrar los elementos.

Métodos principales

- setSize()
- setVisible()
- setDefaultCloseOperation()

Clases principales de Swing

JPanel

Establece un panel que agrupe a los elementos. Toda ventana tiene al menos un JPanel

Métodos principales

- `getContentPane()`
- `setLayout()`
- `add()`

Clases principales de Swing

JOptionPane

Muestra una ventana modal.

Métodos principales

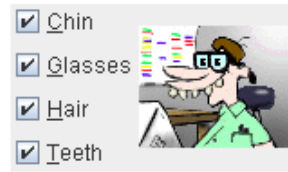
- `showOptionDialog()`
- `showInputDialog()`
- `showMessageDialog()`
- `showConfirmDialog()`

```
 JButton button =  
new JButton("Hello World");
```

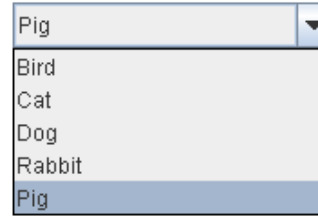
Elementos de interfaz



[JButton](#)



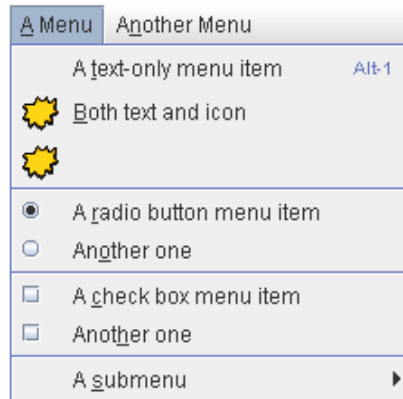
[JCheckBox](#)



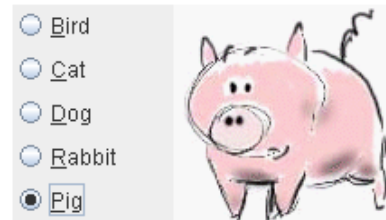
[JComboBox](#)



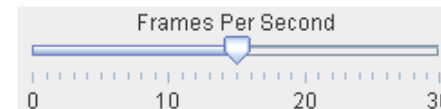
[JList](#)



[JMenu](#)



[JRadioButton](#)



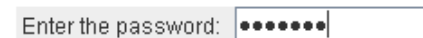
[JSlider](#)



[JSpinner](#)



[JTextField](#)



[JPasswordField](#)

Eventos


- Cada elemento tiene un conjunto de eventos, tanto propios como heredados.
- Cada vez que el usuario interactúa con la aplicación se lanza un evento, que recibe un "escuchador" (Listener) con, al menos, un método determinado que se ejecutará.

Clases anónimas

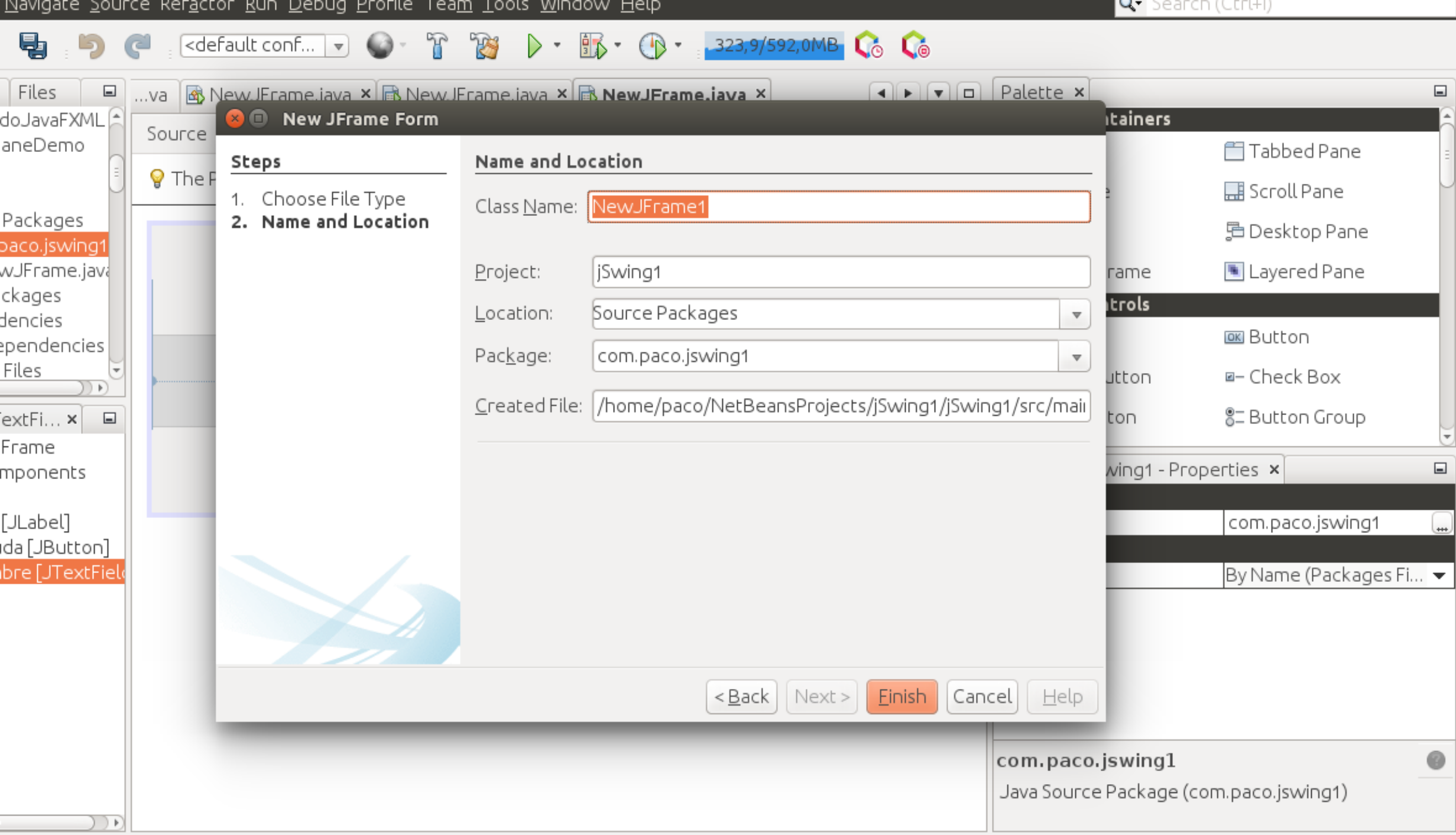
- Las Clases anónimas en JAVA son una solución rápida para implementar una clase que se va utilizar una vez y de forma inmediata.
- Se realizan en el mismo momento de la creación del objeto y se basan siempre en un Interfaz o Clase original, de la cual "copian" tanto métodos como atributos.

Añadir un JTextField a la ventana para que el usuario introduzca el nombre y usarlo en el saludo

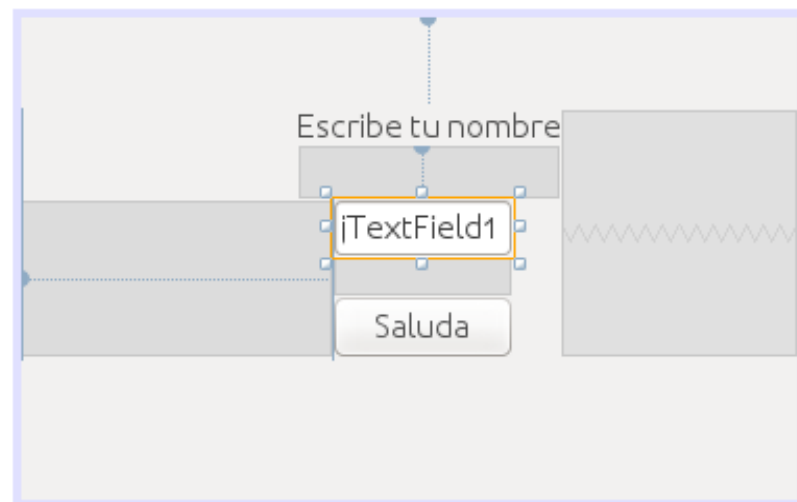




Editor gráfico de NetBeans



The Preview Design button (in the toolbar) enables you to test the design of the form. x



Swing Containers

- Panel
- Split Pane
- Tool Bar
- Internal Frame
- Tabbed Pane
- Scroll Pane
- Desktop Pane
- Layered Pane

Swing Controls

- Label
- Toggle Button
- Radio Button
- Button
- Check Box
- Button Group

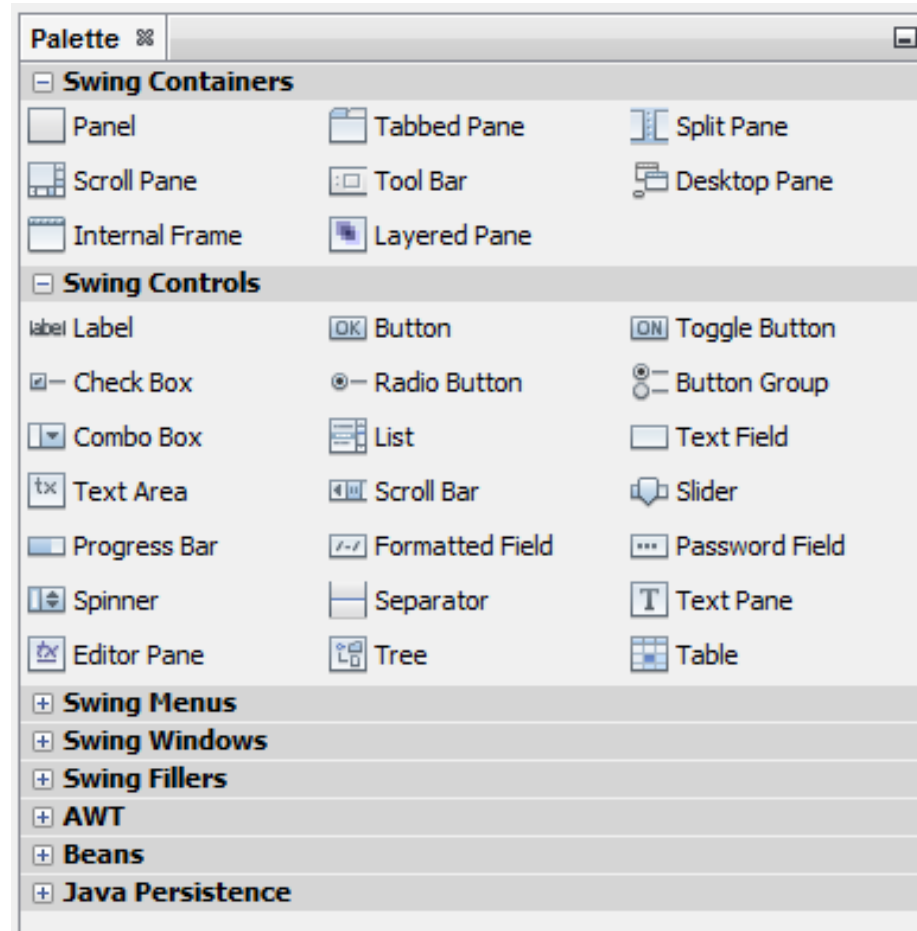
Properties

| | | |
|---------------------|----------------------------------------|--|
| editable | <input checked="" type="checkbox"/> | |
| background | <input type="checkbox"/> [242,241,240] | |
| columns | 0 | |
| document | <default> | |
| font | Ubuntu 15 Plain | |
| foreground | <input type="checkbox"/> [60,60,60] | |
| horizontalAlignment | LEADING | |

txtNombre [JTextField]

Hacer el saludador con el GUI de Netbeans

Paleta de componentes



Tipos de Layout básicos

- FlowLayout
- GridLayout
- GridBagLayout
- BorderLayout
- BoxLayout



Implementar una aplicación que repita todos los eventos que el usuario realiza.



jToggleButton1

jTextField1

50

0

↕

☐ jCheckBox1

Item 1

Item 2

Item 3

Item 4

Item 5

☐ jRadioButton1

☐ jRadioButton2

☐ jRadioButton3

☐ jRadioButton4

jToggleButton2

jTextField1

50

0

↕

☐ jCheckBox1

Item 1

Item 2

Item 3

Item 4

Item 5

☐ jRadioButton1

☐ jRadioButton2

☐ jRadioButton3

☐ jRadioButton4

Implementar una mini calculadora



resultado

Operador1

0














Operador2

0

+



=

- ▼  Form NewJFrame
 - Other Components**
 - ▼  [JFrame]
 -  BoxLayout
 - ▼  jPanel1 [JPanel]
 -  CardLayout
 - label lbResultado [JLabel]
 - ▼  jPanel2 [JPanel]
 -  GridLayout
 - ▼  jPanel9 [JPanel]
 -  BoxLayout
 - label jLabel1 [JLabel]
 - txtOp1 [JTextField]
 - ▼  jPanel8 [JPanel]
 -  BoxLayout
 - label jLabel2 [JLabel]
 - txtOp2 [JTextField]
 - ▼  jPanel5 [JPanel]
 -  GridLayout
 - comboOperacion [JComboBox]
 - btnIgual [JButton]

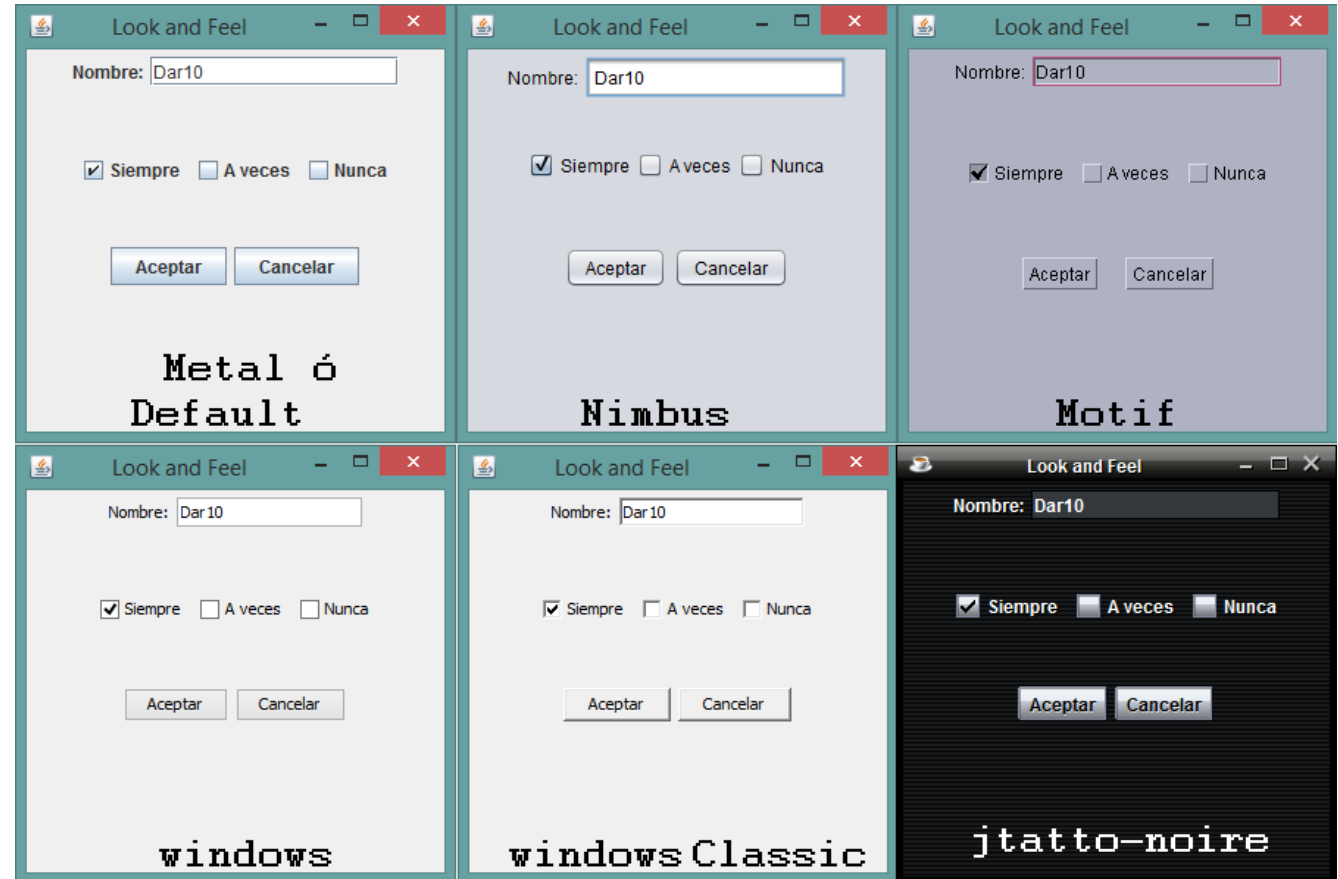
Radio Buttons

```
//Creamos una instancia de ButtonGroup
ButtonGroup btg=new ButtonGroup();

//Añadimos los botones radiobutton
//Si no lo hacemos, los botones seran independientes
btg.add( rdbWindows);
btg.add( rdbLinux);
btg.add( rdbMac);
```

Look&Feel

- Metal
- Nimbus
- Motif
- Windows



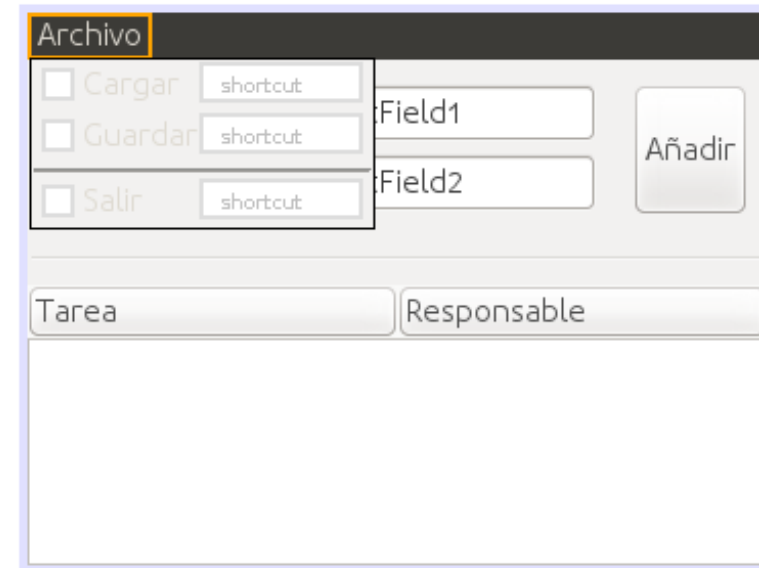
Look&Feel

```
/**
 * @param args the command line arguments
 */
public static void main(String args[]) {
    /* Set the Nimbus look and feel */
    //<editor-fold defaultstate="collapsed" desc=" Look and feel setting code (optional) ">
    /* If Nimbus (introduced in Java SE 6) is not available, stay with the default look and feel.
     * For details see http://download.oracle.com/javase/tutorial/uiswing/lookandfeel/plaf.html
     */
    try {
        for (javax.swing.UIManager.LookAndFeelInfo info : javax.swing.UIManager.getInstalledLookAndFeels()) {
            if ("Nimbus".equals(info.getName())) {
                javax.swing.UIManager.setLookAndFeel(info.getClassName());
                break;
            }
        }
    } catch (ClassNotFoundException ex) {
        java.util.logging.Logger.getLogger(NewJFrame.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
    } catch (InstantiationException ex) {
        java.util.logging.Logger.getLogger(NewJFrame.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
    } catch (IllegalAccessException ex) {
        java.util.logging.Logger.getLogger(NewJFrame.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
    } catch (javax.swing.UnsupportedLookAndFeelException ex) {
        java.util.logging.Logger.getLogger(NewJFrame.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
    }
    //</editor-fold>
}
```

Menus

Swing Menus

- Menu Item
- Menu Bar
- Menu
- Menu Item / CheckBox
- Menu Item / RadioButton
- Popup Menu
- Separator



```
206  
207 private void jMenuItem2ActionPerformed(java.awt.event.ActionEvent evt) {  
208     // TODO add your handling code here:  
209 }  
210
```


Cuadros de diálogo avanzados

- Clase JFileChooser

```
152  
153     int returnVal = seleccionArchivo.showOpenDialog(this);  
154  
155     if (returnVal == JFileChooser.APPROVE_OPTION) {  
156         archivo = seleccionArchivo.getSelectedFile();  
157         /* TODO */  
158     }  
159
```

Temporizar tareas

- Clases Timer y TimerTask

```
201  
202  
203  
204  
205  
206  
207  
208  
209  
210  
211
```

```
Timer timer = new Timer();  
TimerTask tarea = new TimerTask(){  
    @Override  
    public void run() {  
        /* TODO */  
    }  
};  
  
timer.scheduleAtFixedRate(tarea, 0, 1000);
```

Modelo MVC

- Todos los elementos tienen asociado un modelo de datos que permite gestionar el contenido.
- Permite acceder a la información del elemento de forma controlada y sin necesidad de actuar directamente sobre el elemento gráfico.
- Se usa especialmente con las listas y tablas.

Añadir elementos a una lista

Item 1

Item 2

Item 3

Item 4

Item 5

textField1

Añadir

Modelo MVC

```
public ListModelDemo() {  
    initComponents();  
    contenido = new DefaultListModel();  
  
    contenido.addElement("Hola");  
    contenido.addElement("Mundo");  
  
    ArrayList<String> s = new ArrayList<String>();  
    s.add("Francisco");  
    s.add("Romero");  
  
    contenido.addAll(0, s);  
  
    jList1.setModel(contenido);  
}
```

Gestionar tablas

| | | |
|-------------|------------------------------------------|---------------------------------------|
| Tarea | <input type="text" value="jTextField1"/> | <input type="button" value="Añadir"/> |
| Responsable | <input type="text" value="jTextField2"/> | |

| | |
|------------------------------------|------------------------------------------|
| <input type="text" value="Tarea"/> | <input type="text" value="Responsable"/> |
|------------------------------------|------------------------------------------|

Modelo MVC

```
/**
 *
 * @author paco
 */
public class Tareas extends javax.swing.JFrame {

    DefaultTableModel modelo;

    /**
     * Creates new form Tareas
     */
    public Tareas(java.awt.Frame parent, boolean modal) {
        initComponents();
        modelo = (DefaultTableModel) tabla.getModel();
    }

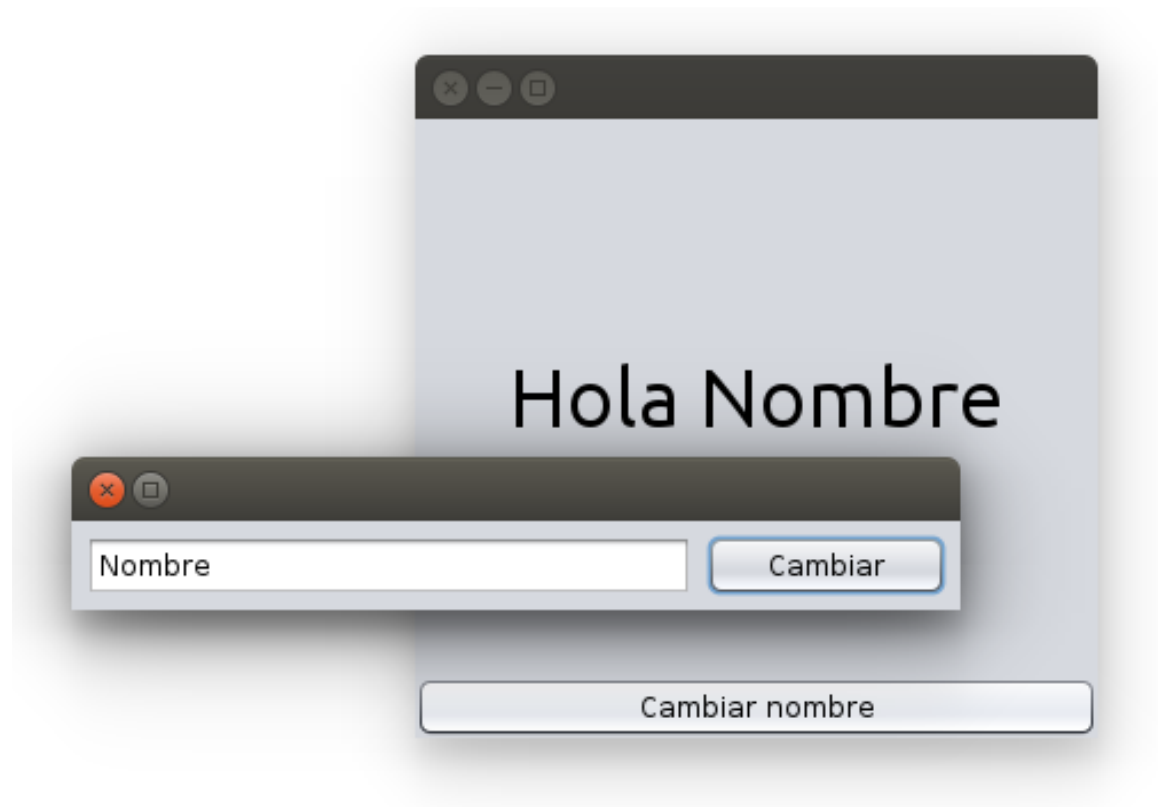
    /**
     * This method is called from within the constructor to initialize the form.
     * WARNING: Do NOT modify this code. The content of this method is always
     * regenerated by the Form Editor.
     */
    @SuppressWarnings("unchecked")
```

Modelo MVC

```
- /**
 * This method is called from within the constructor to initialize the form.
 * WARNING: Do NOT modify this code. The content of this method is always
 * regenerated by the Form Editor.
 */
@SuppressWarnings("unchecked")
Generated Code

- private void btnAniadirActionPerformed(java.awt.event.ActionEvent evt) {
    var s = new String [] { txtTarea.getText(), txtResponsable.getText() };
    modelo.addRow(s);
}
```

Abrir ventanas de dialogo



Compartir datos entre ventanas

```
public String nombre;  
public MyDialog dialogo;  
  
public String getNombre() {  
    return nombre;  
}  
  
public void setNombre(String nombre) {  
    this.nombre = nombre;  
    lbl.setText("Hola "+nombre);  
}  
  
public MyFrame() {  
    initComponents();  
    nombre="";  
    dialogo = new MyDialog(this,true);  
}
```

```
public MyDialog(java.awt.Frame parent, boolean modal) {  
    super(parent, modal);  
    initComponents();  
    this.parent=(MyFrame) parent;  
}  
  
this.parent.setVisible(true);  
this.parent.setNombre(txt.getText());  
this.setVisible(false);
```

Crear una calculadora básica en Swing que permita las siguientes operaciones: suma, resta, multiplicación, división, raíz cuadrada, exponencial, y logaritmo.



Crear un editor de texto similar al notepad de Windows.

Además de poder cargar y guardar archivos de texto deberá tener funcionalidades extra.



Ejercicio práctico

- Mostrar información del documento: ubicación y tamaño.
- Mostrar hora y fecha actual en tiempo real en el footer.
- Una ventana de "acerca de...".
- Un menú de formato desde el que se pueda:
 - Cambiar el tamaño de la fuente
 - Cambiar el color de la fuente

