



INSTITUTO NACIONAL DE APOPA

BACHILLERATO TÉCNICO EN DESARROLLO DE SOFTWARE TERCER AÑO DE BACHILLERATO

GUIA PRACTICA	3	MÓDULO 3.2	USO DE SWING Y USO DE COMPONENTES BÁSICOS DE UNA GUI EN JAVA
ELABORADA POR	Profe. Ever Estrada		

I. OBJETIVOS

Desarrollar programas con ventanas interactivas, Aprender a utilizar elementos gráficos como campos de texto, casillas de verificación, radios, combos y listas, combinándolos con estructura if y case

II. INTRODUCCIÓN

Para crear GUI en Java, es necesario usar librerías que permitan manipular y generar las GUI estas son:

Swing (Incluida en Java SE)

✚ Es la más usada para aplicaciones de escritorio porque es parte de Java Standard Edition (Java SE).

AWT (Abstract Window Toolkit)

✚ Es la primera librería gráfica de Java, pero es más limitada que Swing.

JavaFX (Recomendada para proyectos modernos)

SWT (Standard Widget Toolkit)

✚ Es una alternativa a Swing desarrollada por Eclipse.

Swing es una biblioteca gráfica para crear interfaces de usuario en Java. Forma parte de **Java Foundation Classes (JFC)** y permite desarrollar aplicaciones con ventanas, botones, cuadros de texto, menús y más.

✚ **Características principales de Swing:**

- Es parte de **Java SE** y no requiere instalaciones adicionales.
- Ofrece componentes avanzados como JButton, JLabel, JTextField, JTable, etc.
- Es independiente de la plataforma porque se basa en **Java Virtual Machine (JVM)**.

- Soporta personalización mediante el sistema de **Look and Feel**.
- Funciona sobre **AWT** (Abstract Window Toolkit), pero ofrece más flexibilidad.

En **Swing**, que es la biblioteca de Java para crear interfaces gráficas (GUIs), puedes utilizar una variedad de **componentes** para construir aplicaciones interactivas. A continuación te doy una lista de los **componentes más comunes** de **Swing** que puedes usar para crear una GUI:

1 Contenedores de Ventanas

Estos son los componentes principales que representan la ventana en la que se colocan los demás elementos.

- **JFrame**
Es la ventana principal de la aplicación.
Usada para contener otros componentes.
 - **JDialog**
 - Se usa para crear ventanas emergentes o de diálogo, como formularios o mensajes.
 - **JWindow**
 - Se utiliza para crear ventanas sin barra de título ni bordes, generalmente usadas para pequeños mensajes flotantes.
-

2 Contenedores para Organización de Componentes

Sirven para organizar otros componentes dentro de la ventana.

- **JPanel**
 - Un contenedor flexible que puede contener otros componentes, utilizado para agrupar elementos de la interfaz gráfica.
 - **JScrollPane**
 - Contenedor que permite agregar barras de desplazamiento a componentes, como **JTextArea** o **JList**, cuando el contenido excede el tamaño de la ventana.
 - **JTabbedPane**
 - Crea una serie de pestañas (tabs) para mostrar diferentes paneles de contenido.
-

3 Componentes para Mostrar Información

- **JLabel**
 - Se usa para mostrar texto estático o imágenes.
- **TextField**
Campo de texto de una sola línea para que el usuario ingrese datos.
- **JTextArea**
Área de texto de varias líneas para ingresar o mostrar información más extensa.
- **JList**
Muestra una lista de elementos que el usuario puede seleccionar.
- **JComboBox**
Un menú desplegable que permite elegir una opción entre varias.

4 Componentes para Entrada de Usuario

- **JButton**
Botón que ejecuta una acción cuando se hace clic en él.
 - **JCheckBox**
Casilla de verificación que permite seleccionar o deseleccionar una opción.
 - **JRadioButton**
Botón de opción que permite seleccionar solo una opción de un grupo de opciones mutuamente excluyentes.
 - **JPasswordField**
Campo de texto que oculta lo que el usuario escribe, útil para ingresar contraseñas.
 - **JSpinner**
Permite al usuario seleccionar un valor numérico con incrementos, como la cantidad de elementos.
-

5 Componentes para Control de Disposición

- **JScrollBar**
Barra de desplazamiento que puede agregarse para desplazarse por un área que no cabe en la ventana.
 - **JProgressBar**
Muestra el progreso de una tarea en curso, como una barra de carga.
 - **JSplitPane**
Divide el área de la ventana en dos paneles ajustables, uno encima del otro o uno al lado del otro.
-

6 Componentes para Menús y Barras de Herramientas

- **JMenuBar**
Barra de menús que contiene uno o más menús.
 - **JMenu**
 - Un menú dentro de la barra de menús, que puede contener varios elementos de menú (**JMenuItem**).
 - **JMenuItem**
 - Elemento dentro de un **JMenu**, al cual el usuario puede hacer clic para ejecutar una acción.
 - **JToolBar**
 - Barra de herramientas que puede contener botones e iconos para ejecutar acciones rápidas.
-

7 Otros Componentes Comunes

- **JToolTip**
 - Muestra un texto emergente cuando el usuario pasa el ratón sobre un componente.
- **JTree**
 - Un componente que representa una estructura jerárquica, como un árbol de directorios.

- **JTable**
 - Muestra datos en una tabla, útil para presentar información tabulada.
- **JColorChooser**
 - Muestra un cuadro de diálogo para que el usuario elija un color.

PROCEDIMIENTO

EJERCICIO 1 CREAR EL SIGUIENTE EJERCICIO, USO DE SELECT, RADIO Y COMBOBOX

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

public class SelectRadioExample extends JFrame {
    private JComboBox<String> comboBox;
    private JRadioButton radio1, radio2, radio3;
    private JLabel resultLabel;

    public SelectRadioExample() {
        setTitle("Ejemplo de JComboBox y JRadioButton");
        setSize(400, 200);
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        setLayout(new FlowLayout());

        // Crear JComboBox (Select)
        String[] opciones = {"Opción 1", "Opción 2", "Opción 3"};
        comboBox = new JComboBox<>(opciones);

        // Crear JRadioButton y agruparlos
        radio1 = new JRadioButton("Radio 1");
        radio2 = new JRadioButton("Radio 2");
        radio3 = new JRadioButton("Radio 3");

        ButtonGroup grupoRadios = new ButtonGroup();
        grupoRadios.add(radio1);
        grupoRadios.add(radio2);
        grupoRadios.add(radio3);

        // Crear botón de acción
        JButton btnMostrar = new JButton("Mostrar selección");

        // Etiqueta para mostrar resultado
```

```

resultLabel = new JLabel("Selecciona una opción");

// Agregar ActionListener
btnMostrar.addActionListener(new ActionListener() {
    @Override
    public void actionPerformed(ActionEvent e) {
        String seleccionado = "ComboBox: " + comboBox.getSelectedItems();

        if (radio1.isSelected()) {
            seleccionado += ", Radio: " + radio1.getText();
        } else if (radio2.isSelected()) {
            seleccionado += ", Radio: " + radio2.getText();
        } else if (radio3.isSelected()) {
            seleccionado += ", Radio: " + radio3.getText();
        } else {
            seleccionado += ", No se seleccionó ningún radio";
        }

        resultLabel.setText(seleccionado);
    }
});

// Agregar componentes al JFrame
add(comboBox);
add(radio1);
add(radio2);
add(radio3);
add(btnMostrar);
add(resultLabel);

setVisible(true);
}

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

```

EJERCICIO 2

SUMANDO DOS NÚMEROS CON INTERFAZ GRÁFICA MEJORADA

```

import javax.swing.*; // Importar Swing
import java.awt.*;    // Importar clases para el diseño de componentes
import java.awt.event.*; // Importar eventos para manejar acciones

public class SumarDosNumerosMejorado {
    public static void main(String[] args) {
        // Crear la ventana principal con un título y tamaño más grande
        JFrame ventana = new JFrame("Suma de Dos Números");
    }
}

```

```

ventana.setSize(350, 250); // Tamaño de la ventana
ventana.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
ventana.setLocationRelativeTo(null); // Centrar la ventana en la pantalla

// Configurar el layout de la ventana (FlowLayout con alineación centrada)
ventana.setLayout(new FlowLayout(FlowLayout.CENTER, 10, 20)); // Espaciado
personalizado

// Crear los componentes de la interfaz
JLabel etiqueta1 = new JLabel("Número 1:");
JTextField campoNumero1 = new JTextField(15); // Campo de texto para el primer
número
JLabel etiqueta2 = new JLabel("Número 2:");
JTextField campoNumero2 = new JTextField(15); // Campo de texto para el segundo
número

JButton botonSumar = new JButton("Sumar");
JLabel etiquetaResultado = new JLabel("Resultado: "); // Etiqueta para mostrar el
resultado

// Estilizar los componentes
etiqueta1.setFont(new Font("Arial", Font.BOLD, 14)); // Fuente para etiquetas
etiqueta2.setFont(new Font("Arial", Font.BOLD, 14));
etiquetaResultado.setFont(new Font("Arial", Font.PLAIN, 14));

campoNumero1.setFont(new Font("Arial", Font.PLAIN, 14)); // Fuente para campos
de texto
campoNumero2.setFont(new Font("Arial", Font.PLAIN, 14));

botonSumar.setBackground(new Color(60, 179, 113)); // Fondo del botón
botonSumar.setForeground(Color.WHITE); // Color del texto del botón
botonSumar.setFont(new Font("Arial", Font.BOLD, 14)); // Fuente para el botón
botonSumar.setFocusPainted(false); // Eliminar el borde al hacer clic en el botón

// Estilo general de la ventana (fondo de la ventana)
ventana.getContentPane().setBackground(new Color(240, 248, 255)); // Fondo azul
claro

// Configurar la acción del botón (realizar la suma al hacer clic)
botonSumar.addActionListener(new ActionListener() {
    @Override
    public void actionPerformed(ActionEvent e) {
        try {
            // Obtener los números ingresados por el usuario
            double num1 = Double.parseDouble(campoNumero1.getText());
            double num2 = Double.parseDouble(campoNumero2.getText());

            // Realizar la suma

```

```

        double resultado = num1 + num2;

        // Mostrar el resultado en la etiqueta
        etiquetaResultado.setText("Resultado: " + resultado);
    } catch (NumberFormatException ex) {
        // Si ocurre un error al convertir los números, mostrar un mensaje de error
        JOptionPane.showMessageDialog(ventana, "Por favor, ingresa números
válidos.", "Error", JOptionPane.ERROR_MESSAGE);
    }
}
});

// Agregar los componentes a la ventana
ventana.add(etiqueta1);
ventana.add(campoNumero1);
ventana.add(etiqueta2);
ventana.add(campoNumero2);
ventana.add(botonSumar);
ventana.add(etiquetaResultado);

// Hacer visible la ventana
ventana.setVisible(true);
}
}

```

EJERCICIO 3 - USO DE CASILLAS DE VERIFICACIÓN APLICANDO SENTENCIA IF CON SWING

```

import javax.swing.*.*;
import java.awt.*.*;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;

public class CasillasVerificacion {
    public static void main(String[] args) {
        // Crear ventana
        JFrame frame = new JFrame("Ejemplo de JCheckBox");
        frame.setSize(300, 200);
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        frame.setLayout(new FlowLayout());

        // Crear casillas de verificación
        JCheckBox chkOpcion1 = new JCheckBox("Opción 1");
        JCheckBox chkOpcion2 = new JCheckBox("Opción 2");
        JCheckBox chkOpcion3 = new JCheckBox("Opción 3");
    }
}

```

```

// Crear botón para mostrar selección
JButton btnMostrar = new JButton("Mostrar Selección");
JLabel lblResultado = new JLabel("Seleccionado: Ninguno");

// Agregar evento al botón
btnMostrar.addActionListener(new ActionListener() {
    @Override
    public void actionPerformed(ActionEvent e) {
        String seleccion = "Seleccionado: ";
        if (chkOpcion1.isSelected()) seleccion += "Opción 1 ";
        if (chkOpcion2.isSelected()) seleccion += "Opción 2 ";
        if (chkOpcion3.isSelected()) seleccion += "Opción 3 ";
        if (!chkOpcion1.isSelected() && !chkOpcion2.isSelected() &&
!chkOpcion3.isSelected()) {
            seleccion += "Ninguno";
        }
        lblResultado.setText(seleccion);
    }
});

// Agregar componentes a la ventana
frame.add(chkOpcion1);
frame.add(chkOpcion2);
frame.add(chkOpcion3);
frame.add(btnMostrar);
frame.add(lblResultado);

// Hacer visible la ventana
frame.setVisible(true);
}
}

```

TAREA:

En la introducción teórica, se muestra la lista de los componentes más utilizados en la creación de GUI's con Swing. Se pide presentar exposición en donde se explique el uso y funcionamiento de componente asignado por el docente y demostrar su aplicación en un programa.

CASOS DE DESAFÍO

Tomando en cuenta cada uno de los ejercicios anteriores, debes de crear las modificaciones necesarias en donde recicles el código pero aplicado en contexto diferente. (Crear 3 ejercicios diferentes).