

# Java Swing

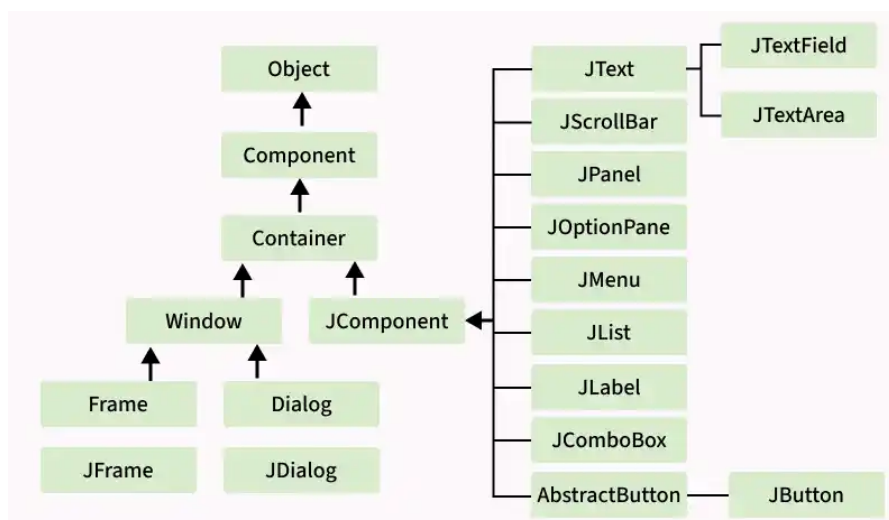
## 1. Introducción a Java Swing

**Java Swing** es un conjunto de herramientas de **interfaz gráfica de usuario (GUI)** que forma parte de las **Java Foundation Classes (JFC)**. Proporciona un amplio conjunto de componentes *ligeros* para crear aplicaciones de escritorio con una interfaz más flexible y completa que AWT.

- Ofrece una gran variedad de componentes GUI, como botones, etiquetas, tablas y árboles.
- Admite una apariencia personalizable, lo que permite a las aplicaciones tener un aspecto único.
- Es independiente de la plataforma, lo que permite que las aplicaciones se ejecuten de forma consistente en diferentes sistemas operativos.

## 2. Jerarquía de clases de Swing

Este diagrama ilustra la jerarquía de herencia de las clases de Swing, mostrando cómo los componentes GUI principales, como `JFrame`, `JButton`, `JLabel`, `JCheckBox`, `JList` y `JProgressBar`, se derivan de clases base como `Container` y `JComponent` dentro del framework Java Swing.



## 3. AWT (Abstract Window Toolkit)

AWT es la biblioteca GUI original de Java, utilizada para crear aplicaciones basadas en ventanas. Proporciona componentes básicos como botones, etiquetas, campos de texto y menús, y se basa en los componentes de interfaz gráfica de usuario (GUI) nativos del sistema operativo.

- Forma parte del paquete `java.awt`.
- Utiliza componentes nativos de la plataforma, lo que hace que su apariencia dependa de la plataforma.
- Proporciona controles básicos de GUI y mecanismos de gestión de eventos.
- Sirve como base sobre la que se construye Swing.

## 4. ¿Qué es JFC?

**JFC** significa **Java Foundation Classes** (*Clase Fundamental de Java*). JFC es el conjunto de componentes de GUI que simplifican las aplicaciones de escritorio. Muchos programadores creen que JFC y Swing son lo mismo, pero no es así. JFC contiene Swing (un paquete de componentes de interfaz de usuario) y muchas otras funciones:

- Cortar y pegar: Compatibilidad con el portapapeles.
- Funciones de accesibilidad: Diseñadas para desarrollar interfaces gráficas de usuario para personas con discapacidad.
- Las funciones de colores de escritorio se introdujeron por primera vez en Java 1.1.
- Java 2D: Ofrece compatibilidad mejorada con colores, imágenes y texto.

## 5. Características de la clase Swing

- **Independencia de la plataforma:** Los componentes Swing proporcionan una interfaz gráfica de usuario (GUI) consistente en diferentes sistemas operativos sin depender de los controles nativos del sistema operativo.
- **Componentes ligeros:** Los componentes Swing están escritos completamente en Java y no dependen de los recursos del sistema específicos de la plataforma.
- **Aspecto personalizable:** Swing permite cambiar la apariencia de los componentes dinámicamente sin modificar el código de la aplicación.
- **Arquitectura MVC:** Swing sigue el patrón Modelo-Vista-Controlador (MVC), separando los datos, la interfaz de usuario y la lógica de control.
- **Altamente personalizable:** Los componentes Swing se pueden personalizar fácilmente para adaptarse a los requisitos específicos de la aplicación.
- **Amplio conjunto de controles:** Swing proporciona componentes GUI avanzados como `JTable`, `JTree`, `JTabbedPane` y `JScrollPane`.
- **Gestión avanzada de eventos:** Swing ofrece un sólido mecanismo de gestión de eventos para responder eficientemente a las acciones del usuario.

## 6. Ventajas de Swing sobre AWT

Los componentes Swing ofrecen mayor flexibilidad y funciones avanzadas en comparación con los componentes AWT.

- **Compatibilidad con imágenes:** Los componentes Swing, como botones y etiquetas, pueden mostrar imágenes junto con texto o en lugar de él. Bordes personalizados: Los bordes de los componentes Swing se pueden personalizar fácilmente para mejorar la apariencia de la interfaz de usuario.
- **Componentes no rectangulares:** Swing permite que componentes como los botones tengan formas personalizadas, incluyendo botones redondos.
- **Compatibilidad con la accesibilidad:** Swing se integra perfectamente con tecnologías de asistencia como los lectores de pantalla, lo que hace que las aplicaciones sean más accesibles para usuarios con discapacidades.

## 7. Ejemplos

**Ejemplo 1:** Desarrollar un programa usando etiquetas (Swing) para mostrar el mensaje *Clic en el sitio web de GFG*.

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

// Main class
class GFG {

    // Main driver method
    public static void main(String[] args)
    {
        // Creando una instancia de JFrame
        JFrame frame = new JFrame();

        // Creando una instancia de JButton
        JButton button = new JButton("Haga clic en el sitio web de GFG");

        // x axis, y axis, width, height
        button.setBounds(150, 200, 220, 50);

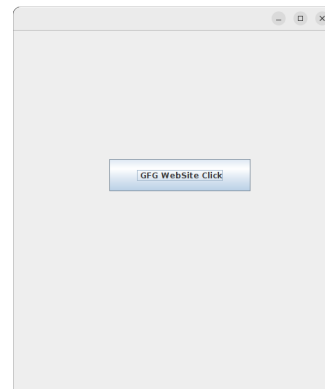
        // Agregar botón en JFrame
        frame.add(button);

        // 500 de ancho y 600 de alto
        frame.setSize(500, 600);

        // No usar gestores de diseño
        frame.setLayout(null);

        // Haciendo visible el marco
        frame.setVisible(true);
    }
}
```

Salida:



**Ejemplo 2:** Escribe un programa para crear tres botones con las etiquetas Aceptar, Enviar y Cancelar.

```
import java.awt.*;

class button {
    button()
    {
        Frame f = new Frame();

        // Botón 1 creado
        // OK button
        Button b1 = new Button("OK");
        b1.setBounds(100, 50, 50, 50);
        f.add(b1);

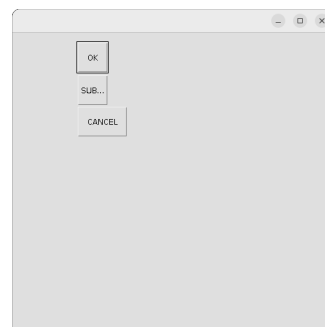
        // Botón 2 creado
        // Submit button
        Button b2 = new Button("ENTREGAR");
        b2.setBounds(100, 101, 50, 50);
        f.add(b2);

        // Botón 3 creado
        // Cancel button
        Button b3 = new Button("CANCELAR");
        b3.setBounds(100, 150, 80, 50);
        f.add(b3);

        f.setSize(500, 500);
        f.setLayout(null);
        f.setVisible(true);
    }

    public static void main(String a[]) { new button(); }
}
```

Salida:



**Ejemplo 3:** Programa para agregar una casilla de verificación en el marco.

```
import java.awt.*;

// Driver Class
class Lan {
    // Main Function
    Lan()
    {
        // Marco creado
        Frame f = new Frame();

        Label l1 = new Label("Seleccione los idiomas que conoce");

        l1.setBounds(100, 50, 120, 80);
        f.add(l1);

        // Se ha creado una casilla de verificación
        Checkbox c2 = new Checkbox("Castellano");
        c2.setBounds(100, 150, 50, 50);
        f.add(c2);

        // Se ha creado una casilla de verificación
        Checkbox c3 = new Checkbox("Inglés");
        c3.setBounds(100, 200, 80, 50);
        f.add(c3);

        // Se ha creado una casilla de verificación
        Checkbox c4 = new Checkbox("Portugues");
        c4.setBounds(100, 250, 80, 50);
    }
}
```

```

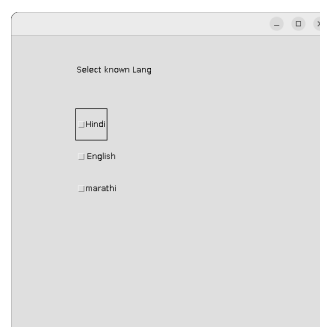
        f.add(c4);

        f.setSize(500, 500);
        f.setLayout(null);
        f.setVisible(true);
    }

    public static void main(String ar[]) { new Lan(); }
}

```

Salida:



## 8. Componentes de la clase Swing

| Clase          | Descripción                                                                                                                                                                                                                                                                  |
|----------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Component      | Un componente es la clase base abstracta para los controles de interfaz de usuario que no son menús en Java SWING. Los componentes representan un objeto con una representación gráfica.                                                                                     |
| Container      | Un contenedor es un componente que puede contener componentes Java SWING.                                                                                                                                                                                                    |
| JComponent     | Un JComponent es una clase base para todos los componentes de interfaz de usuario de Swing. Para utilizar un componente de Swing que hereda de JComponent, el componente debe estar en una jerarquía de contención cuya raíz sea un contenedor Java Swing de nivel superior. |
| JLabel         | Un JLabel es un componente de objeto para colocar texto en un contenedor.                                                                                                                                                                                                    |
| JButton        | Esta clase crea un botón con etiqueta.                                                                                                                                                                                                                                       |
| JColorChooser  | Un JColorChooser proporciona un panel de controles diseñado para permitir al usuario manipular y seleccionar un color.                                                                                                                                                       |
| JCheckBox      | Un JCheckBox es un componente gráfico (GUI) que puede estar en estado activado (verdadero) o desactivado (falso).                                                                                                                                                            |
| JRadioButton   | La clase JRadioButton es un componente gráfico (GUI) que puede estar en estado activado (verdadero) o desactivado (falso) en el grupo                                                                                                                                        |
| JList          | Un componente JList representa al usuario con una lista desplazable de elementos de texto.                                                                                                                                                                                   |
| JComboBox      | Un componente JComboBox presenta al usuario un menú desplegable con opciones.                                                                                                                                                                                                |
| JTextField     | Un objeto JTextField es un componente de texto que permite editar una sola línea de texto.                                                                                                                                                                                   |
| JPasswordField | El objeto JPasswordField es un componente de texto especializado para la introducción de contraseñas.                                                                                                                                                                        |
| JTextArea      | Un objeto JTextArea es un componente de texto que permite editar varias líneas de texto.                                                                                                                                                                                     |
| ImageIcon      | Un control ImageIcon es una implementación de la interfaz Icon que dibuja iconos a partir de imágenes.                                                                                                                                                                       |
| JScrollbar     | Un control JScrollbar representa un componente de barra de desplazamiento para permitir a los usuarios seleccionar valores dentro de un rango.                                                                                                                               |
| JOptionPane    | JOptionPane proporciona un conjunto de cuadros de diálogo estándar que solicitan a los usuarios un valor o algo.                                                                                                                                                             |
| JFileChooser   | Un JFileChooser que controla representa una ventana de diálogo desde la cual el usuario puede seleccionar un archivo.                                                                                                                                                        |
| JProgressBar   | A medida que la tarea avanza hacia su finalización, la barra de progreso muestra el porcentaje de la tarea completada.                                                                                                                                                       |
| JSlider        | Esta clase JSlider permite al usuario seleccionar gráficamente (GUI) un valor deslizando un control giratorio dentro de un intervalo delimitado.                                                                                                                             |
| JSpinner       | Un JSpinner de esta clase es un campo de entrada de una sola línea que permite al usuario seleccionar mediante un número o un valor de objeto de una secuencia ordenada.                                                                                                     |