

# Investigación de GUI

Semana: 9

**Nombre del estudiante:**

DAVID ANDRES OLIVA HERNANDEZ

**Número de cuenta:**

12311121

**Sede de estudio:**

Tegucigalpa M.D.C

**Docente:**

Ing. Claudia Cortés

**Sección:**

1555

**Fecha de entrega:**

Jueves, 7 de diciembre de 2023

# Introducción

Dentro del extenso campo de la programación en Java, la creación de Interfaces Gráficas de Usuario (GUI) emerge como un elemento esencial para proporcionar experiencias de usuario fluidas y comprensibles. Este informe se sumerge en el fascinante universo de las bibliotecas AWT (Abstract Window Toolkit) y Swing, dos fundamentos cruciales en el desarrollo de GUI en Java. Además, exploraremos detalladamente los componentes GUI, sus propiedades adaptables y la intrincada interacción de eventos y manejadores que impulsa la interactividad en estas interfaces. Únase a nosotros en este viaje a través del arte y la ciencia de crear interfaces visuales en el lenguaje de programación Java.

En el contexto de Java, GUI se refiere a "Graphical User Interface" o interfaz gráfica de usuario. Es un término utilizado para describir la parte de un programa de software que interactúa con el usuario a través de elementos visuales como ventanas, botones, menús y otros componentes gráficos. Las aplicaciones GUI proporcionan una forma más intuitiva y fácil de interactuar con un programa en comparación con las interfaces de

## **línea de Bibliotecas de Desarrollo de GUI en Java: AWT y Swing**

- **AWT (Abstract Window Toolkit):** AWT, venerada como la biblioteca de GUI pionera en Java, ofrece una amalgama de componentes de nivel bajo para la creación de interfaces visuales. Desde ventanas hasta botones y cuadros de texto, AWT proporciona una base sólida. La creación de una interfaz con AWT involucra la instanciación de objetos de componentes GUI, meticulosamente dispuestos en una ventana. La personalización de propiedades permite adaptar la apariencia y el comportamiento de estos componentes según las necesidades específicas.
- **Swing:** Swing, la evolución moderna de las GUI en Java, trasciende las limitaciones de AWT. Dotada de una interfaz de usuario más rica y adaptable, Swing se convierte en la elección contemporánea. La creación de interfaces en Swing sigue un proceso análogo al de AWT, pero con un enfoque más robusto y estilos avanzados. comandos.

## **Componentes GUI: Fundamentos de la Interfaz Gráfica**

- **Definición:** Los componentes GUI, pilares fundamentales de una interfaz gráfica de usuario, comprenden elementos distintivos como botones, cuadros de texto y menús.

- **Propiedades Configurables:** Estos componentes GUI se distinguen por sus propiedades configurables, que permiten ajustar su tamaño, posición, color, estilo y funcionalidad. La versatilidad de estas propiedades posibilita la personalización meticulosa de la interfaz, adaptándola a las exigencias particulares del programa.

## Eventos de los Componentes GUI y Manejadores de Eventos

- **Eventos:** Los componentes GUI, en su interacción con el usuario, generan eventos significativos como clics del ratón, pulsaciones de teclas y movimientos del mouse.
- **Manejadores de Eventos:** Los manejadores de eventos, ingeniosas funciones ejecutadas en respuesta a eventos específicos son la piedra angular para garantizar una interactividad sin fisuras y adaptativa. Su función trascendental radica en la respuesta hábil a las acciones del usuario.

## Ejemplo de una GUI simple

El siguiente código muestra un ejemplo de una GUI simple en Java:

```
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import javax.swing.JButton;
import javax.swing.JFrame;
public class JavaApplication35 {
    public static void main(String[] args) {

        // Crea una ventana
        JFrame frame = new JFrame( title: "Hola mundo" );

        // Crea un botón
        JButton button = new JButton( text: "Saludar" );

        // Agrega el botón a la ventana
        frame.add( comp: button );

        // Define el manejador de eventos para el botón
        button.addActionListener( new ActionListener() {
            @Override
            public void actionPerformed( ActionEvent e ) {
                System.out.println( x: "Hola mundo" );
            }
        } );

        // Muestra la ventana
        frame.setVisible( b: true );
    }
}
```

## **Conclusión**

En Conclusión, el desarrollo de Interfaces Gráficas de Usuario (GUI) en Java implica el hábil uso de bibliotecas como AWT y Swing. La personalización de componentes GUI, con propiedades adaptables, es esencial para crear interfaces atractivas y funcionales. La gestión de eventos y manejadores añade sofisticación, mejorando la interactividad. En última instancia, el éxito en este campo combina habilidades técnicas con una apreciación por el diseño intuitivo, destacando la importancia de la excelencia tanto en la ciencia como en el arte de la programación.