Instituto Tecnológico Nacional de México

Instituto Tecnológico de Orizaba

Tópicos Avanzados de Programación

Ingeniería en Sistemas Computacionales

Tema 1. Interfaz gráfica de usuario.

Maciel Villa Valeria 21010985

Mávil Barojas Luis Enrique 21010994

Vargas Puertos Luis Gerardo 21011056

Grupo 4g2A

17/febrero/2023

Introducción:

Durante el desarrollo de un interfaz Grafico de Usuario (GUI) existe una gran variedad de métodos con diversas funciones para facilitar al usuario introducir su información y datos.

Competencia especifica:

Desarrolla programas para interactuar con el usuario de una manera amigable, utilizando GUI (Interfaz Grafico de Usuario) manipuladas a través de eventos.

Marco teórico:

Las interfaces de usuario son un elemento clave en la experiencia de usuario y en la eficacia de las aplicaciones de software. Es por ello que es fundamental entender los elementos que conforman este aspecto crucial de la programación y del diseño de aplicaciones.

Las interfaces de usuario consisten en un conjunto de elementos que permiten la interacción entre los usuarios y el software. Entre estos elementos, se incluyen la disposición de los botones, la tipografía, los colores, las formas, y la organización del contenido.

La experiencia de usuario es fundamental en las interfaces de usuario.

Material y equipo:

- → Computadora
- → Apache NetBeans IDE 17
- → Material de apoyo proporcionados por la maestra (PFD)

Desarrollo de la practica:

Clase	Método	Descripción
JFrame	Frame()	Constructor de la clase.

	void setLocationRelativeTo	Estable la ubicación de la ventana en la relación con el componente especificado.
	void setDefaultCloseOperation	Usado para especificar una de las siguientes opciones del botón de cierre: EXIT_ON_CLOSE.
	void setResizable (boolean resizable)	Para evitar que se cambie el tamaño de la ventana
	void setVisible	Muestra u oculta la ventana según el valor del parámetro.
	void setTitle (String título)	Establece el título de la ventana con el String especificado
	void setSize(int x, int y)	Cambia el tamaño del componente para que tenga una anchura x y una altura y
Component	void addActionListener(this)	Añade un oyente de eventos al componente actual.
	void setBounds(int x, int y, int ancho, int alto)	Mueve y cambia el tamaño del componente.
Container	Container getContentPane()	Retorna un objeto ContentPane de la ventana.
	void setLayout	Estable el layout de la ventana.
	Component add(Component comp)	Añade el componente especificado al final del contenedor.
ActionListener	ActionListener	Interface que debe ser implementada para gestionar eventos.
	void actionPerformed(actionEvente)	Se invoca cuando ocurre un evento.
JLabel	JLabel()	Añade el componente especificado al final del contenedor.
	Void setText (String text)	Define una línea de texto que mostrará el componente.
JTextField	JTextField()	Constructor de la clase JTextField.
	String getText()	Retorna el texto contenido en el componente de texto.
JButton	JButton()	Constructor de la clase JButton

	void setText(String text)	Define una línea de texto que mostrará este componente.
JList	JList()	Constructor de la clase JList
	void setText(String text)	Define una línea de texto que mostrará este componente.
	void setSelectionMode(int modoSelección)	Establece el modo de selección de la lista
	void setModel(ListModel<E> model)	Establece el modelo que representa el contenido de la lista.
	int getSelectedIndex()	Devuelve el índice seleccionado.
JScrollPane	JScrollPane()	Constructor de la clase JScrollPane.
	void setViewportView(Component view)	Crea una ventana gráfica si es necesario y luego establece su vista.
Event	Object getSource()	El objeto sobre el cual el evento inicialmente ha ocurrido

GUI 1

La indicación es hacer una ventana, un frame, pero con herencia

```
1  package GUI;
2
3  import javax.swing.JFrame;
4
5  public class GUI1 extends JFrame{
6      public GUI1(){
7      super("Ventana usando herencia");
8      setSize(370,450);
9      setLocationRelativeTo(null);
10     setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
11     setResizable(false);
12     setVisible(true);
13
14     }
15
16
17 }
18
```

El código empieza en la línea 1 con la declaración del paquete, después en la línea 3 importamos la clase JFrame, un JFrame es una clase para generar ventanas en las cuales se les puedes poner otros componentes con las que el usuario puede o no interactuar.

En la línea 5 es la declaración de la clase “public class” seguido del nombre y después la palabra reservada “extends” esta palabra se puede referir como “extensión de” y nos sirve para heredar los métodos y atributos de una clase (llamada clase padre) a otra(s) (clases hijas)

En la siguiente línea tenemos el constructor de la clase, sirve para inicializar el objeto y establecer sus propiedades y valores predeterminados.

La palabra super hace referencia al constructor de la clase padre y pondrá como titulo la cadena que pongas como parámetro, podemos acceder a los métodos “setSize ()”, “setLocationRelativeTo ()”, etc., accedemos sin necesidad de crear un objeto gracias a la herencia

La clase main arranca la aplicación, creamos un objeto de tipo GUI1 para acceder a la clase y basta con llamar al constructor mostrar la ventada

Dado que están en diferentes paquetes es necesario importar el paquete seguido de un punto y después la clase a importar, el “*” hace referencia a que se importan todas las clases de ese paquete

```
1
2   package topicostema1;
3   import GUI.*;
4
5
6
7   public class Main {
8
9       /**
10       * @param args the command line arguments
11       */
12      public static void main(String[] args) {
13          GUI1 A= new GUI1();
14
15
16          // TODO code application logic here
17      }
```

GUI 2

La indicación para esta práctica es crear una ventana sin usar herencia

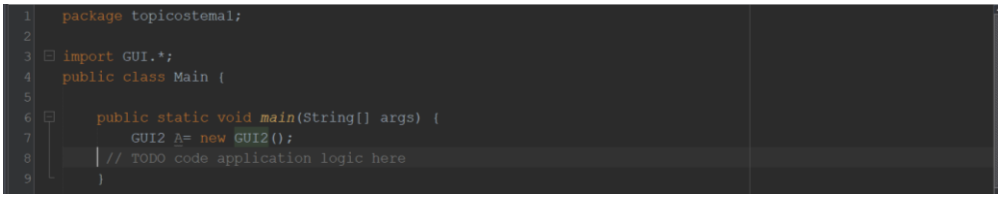
Cuando no se puede o no se debe heredar de una clase la forma correcta de hacerlo es creando un objeto de esa clase y accediendo a sus métodos y atributos de esa clase por medio de aquel objeto, sin embargo, aún se debe importar esa clase, la

forma de acceder a esos métodos es poniendo el nombre del objeto seguido de un punto “.” Y después el nombre del método, si el método necesita parámetros, pasar como parámetros las variables según el tipo de dato solicitado

```
package GUI;

import javax.swing.JFrame;

public class GUI2 {
    public GUI2(){
        JFrame frame=new JFrame("Sin herencia");
        frame.setSize(370,450);
        frame.setLocationRelativeTo(null);
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        frame.setResizable(true);
        frame.setVisible(true);

    }


}
```

La clase main arranca la aplicación, creamos un objeto de tipo GUI1 para acceder a la clase y basta con llamar al constructor mostrar la ventada

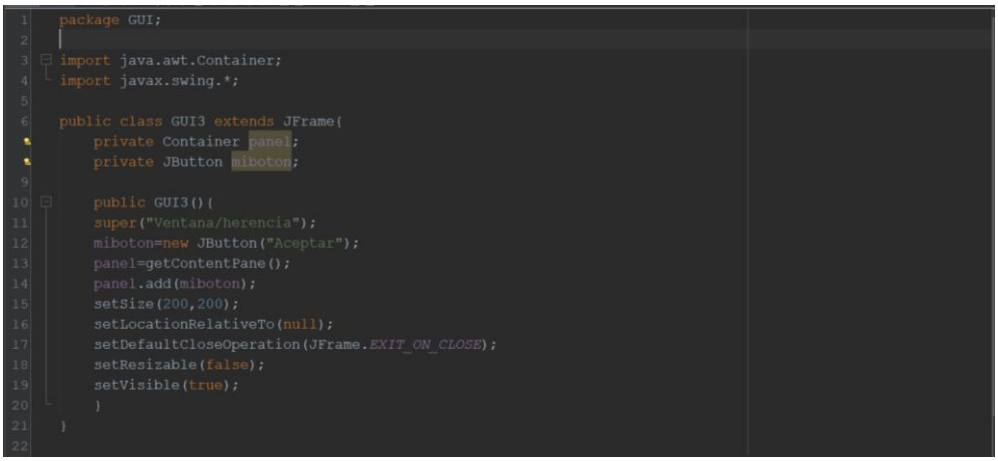
Dado que están en diferentes paquetes es necesario importar el paquete seguido de un punto y después la clase a importar, el “*” hace referencia a que se importan todas las clases de ese paquete



```
package topicostema1;

import GUI.*;
public class Main {

    public static void main(String[] args) {
        GUI2 A= new GUI2();
    // TODO code application logic here
    }
```

GUI 3

Crea una ventana que contenga un contener el cual a su vez contenga un botón



```
1  package GUI;
2
3  import java.awt.Container;
4  import javax.swing.*;
5
6  public class GUI3 extends JFrame{
7      private Container panel;
8      private JButton miboton;
9
10     public GUI3(){
11     super("Ventana/herencia");
12     miboton=new JButton("Aceptar");
13     panel=getContentPane();
14     panel.add(miboton);
15     setSize(200,200);
16     setLocationRelativeTo(null);
17     setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
18     setResizable(false);
19     setVisible(true);
20     }
21 }
22
```

La clase empieza en la línea 1 declarando del paquete, en las líneas 3 y 4 se importan las clases de los objetos que van a utilizarse, en esta ocasión haremos la ventana haciendo uso de la herencia, en las líneas 7 y 8 declaramos dos objetos privados, uno de tipo Container de nombre panel y otro de tipo “JButton” de nombre “miBoton”

Creamos el constructor de la clase e instanciamos el objeto “miboton” con la palabra reservada “new” y el constructor de la clase pasando como parámetro una cadena, la cual será el texto que tendrá el botón, el método “getContentPane ()” nos instancia y devuelve un panel JFrame, el método “add ()” lo que hará es agregar al componente que le pases como parámetro.

La clase main de nuevo arranca el programa es suficiente con declarar e instanciar un objeto de la clase GUI3 para poder ver la ventana

```
1  package topicostema1;
2
3  import GUI.*;
4  public class Main {
5
6      public static void main(String[] args) {
7          GUI3 A= new GUI3();
8       // TODO code application logic here
9      }
```

GUI 5

Crear un frame en el cual pondrás una casilla de selección, una etiqueta, un campo de texto y un botón alineados de manera horizontal

La clase empieza en la línea uno con la declaración del paquete, en las líneas 3 y 4 se importan las clases de los objetos que van a utilizarse, en esta ocasión haremos la ventana haciendo uso de la herencia, en la línea 6 se declara la clase y la herencia, en la línea 7 empieza el constructor de clase con la palabra super hace referencia al constructor de la clase padre y pondrá como título “Ventana con herencia ”, en la línea 8, el método “getContentPane()” retorna una instanciación del JFrame, se hace referencia a su propia ventana, usamos el método “setLayout()”, en las siguientes 4 lineas declaramos e instanciamos los 4 componentes que usaremos, no importa que se declaren dentro del constructor pues no se ocuparan fuera de.

Después mediante el método “add ()” empezamos a agregar los componentes uno por uno, utilizamos el método “setDefaultCloseOperation ()” le pasamos como parámetro “JFrame.EXIT_ON_CLOSE” es para finalizar por completo la aplicación una vez cerrando la ventana

El método “pack ()” ayuda acomodar los componentes, después utilizamos los métodos set

```
1  package GUI;
2
3  import java.awt.FlowLayout;
4  import javax.swing.*;
5
6  public class GUI5 extends JFrame{
7      public GUI5(){
8      super("Ventana con herencia");
9      getContentPane().setLayout(new FlowLayout());
10     JButton boton=new JButton("Aceptar");
11     JLabel eti=new JLabel("Dato");
12     JTextField campo=new JTextField(10);
13     JCheckBox ch=new JCheckBox("check box");
14     getContentPane().add(boton);
15     getContentPane().add(eti);
16     getContentPane().add(campo);
17     getContentPane().add(ch);
18     setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
19     pack();
20     setLocationRelativeTo(null);
21     setVisible(true);
22     }
23
24 }
```

La clase main de nuevo arranca el programa es suficiente con declarar e instanciar un objeto de la clase GUI5 para poder ver la ventana

```
package topicostema1;

import GUI.*;
public class Main {

    public static void main(String[] args) {
        GUI5 A= new GUI5();
     // TODO code application logic here
    }
```

GUI 6

En este GUI le asignamos al título el texto de “Ventana con herencia” utilizando la palabra super, después con el ciclo for haremos que haga botones del 1 hasta el límite que es 9, y hacemos que nuestra ventana este posicionada en el centro y que nuestro botón de salida nos permita salir de la aplicación, con el setVisible haremos que la ventana se alcance a ver.

```
package GUI;
import java.awt.GridLayout;
import javax.swing.JButton;
import javax.swing.JFrame;
public class Gui6 extends JFrame {
    public Gui6(){
        super("Ventana con Herencia"); //Estable el titutlo de la ventana
        getContentPane().setLayout(new GridLayout(4,3,5,5));
        for (int i=1; i<=10; i++)
            add(new JButton(Integer.toString(i)));
        setLocationRelativeTo(null); //La ventana se pocisiona en el centro de la pantalla
        //Establece que el boton de cerra que permitra salir de la aplicacion
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        pack();
        setVisible(true);
    }
}
```

GUI 7

En este GUI también le asignamos el título de la ventana llamado “Ventana con herencia” y le agregaremos botones con los números del 1 al 5 y después a cada uno de estos lo pondremos en una dirección diferente con las palabras EAST,

SOUTH, WEST, NORTH, CENTER, para que así cada uno tenga un espacio diferente en la ventana, lo posicionamos en el centro, que no se pueda cambiar el tamaño pero que esta sea del tamaño necesario para que todos los componentes quepan en ella, por ultimo hacemos que sea visible.

```java
package GUI;
import java.awt.BorderLayout;
import javax.swing.JButton;
import javax.swing.JFrame;

public class Gui7 extends JFrame{
    //constantes y componentes(objeto)
    public Gui7(){
        super("Ventana con Herencia"); //Estable el titutlo de la ventana
        setLayout(new BorderLayout(5,10));
        add(new JButton("1"),BorderLayout.EAST);
        add(new JButton("2"),BorderLayout.SOUTH);
        add(new JButton("3"),BorderLayout.WEST);
        add(new JButton("4"),BorderLayout.NORTH);
        add(new JButton("5"),BorderLayout.CENTER);

        setLocationRelativeTo(null); //La ventana se pocisiona en el centro de la pantalla
        //Establece que el boton de cerrar permita la salida de la aplicacion
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        setResizable(false); //Establece que el tamaño de la ventana no se puede cambiar
        pack(); //Presenta una ventana con el tamaño necesario para mostrar los componentes que hay en ella
        setVisible(true);
    }
}
```

GUI 8

En este GUI como en los anteriores hacemos los import, y vamos a declarar unas variables private, vamos asignarle el título a la ventana, le asignamos el tamaño y que no pueda ser cambiado, y vamos a poner una etiqueta con el JLabel para que se pueda ingresar texto, que en este caso pedimos el nombre y le asignamos la posición de la linea donde se ingresa el texto, pusimos un botón con el nombre saludar e hicimos que de acción al presionarlo nos salude y utilice el nombre que le proporcionamos.

```java
package GUI;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import javax.swing.JButton;
import javax.swing.JFrame;
import javax.swing.JLabel;
import javax.swing.JOptionPane;
import javax.swing.JTextField;
public class Gui8 extends JFrame implements ActionListener{
    private JTextField textfield1;
    private JLabel label1;
    private JButton boton1;

    public Gui8(){
        super("Ventana con Herencia");
        setLayout(null);
        setSize(200,150);
        setResizable(false);
        setDefaultCloseOperation(EXIT_ON_CLOSE);

        //Etiqueta sirve para escribir texto, adicionalmente puede mostrar una imagen
        label1 = new JLabel("Nombre; ");
        label1.setBounds(10, 10, 100, 30);
        add(label1);

        textfield1 = new JTextField();
        textfield1.setBounds(80, 10, 150, 30);
        add(textfield1);

        boton1 = new JButton("Saludar");
        boton1.setBounds(10,80,100,30);
        add(boton1);

        boton1.addActionListener(this);
        setVisible(true);
    }

        @Override
    public void actionPerformed(ActionEvent e) {
        if (e.getSource()==boton1) {
            JOptionPane.showInputDialog("Hola "+textfield1.getText()+" como estas?");
        }
    }
}
```

GUI 9

En este GUI hacemos los import, creamos las variables private, asignamos el título a la ventana, el tamaño, si se puede modificar o no, y la posición de donde

aparecerá, haremos botones de circulo para que el usuario pueda marcar uno, y después de estos viene un texto a seleccionar cada opción, le asignamos a cada uno un tamaño y posición específica para que no esté uno encima del otro.

```java
package GUI;
import java.awt.Color;
import javax.swing.*;
import javax.swing.event.ChangeEvent;
import javax.swing.event.ChangeListener;
public class Gui9 extends JFrame implements ChangeListener{
    private JRadioButton radio1, radio2, radio3;
    private ButtonGroup grupoBotones;

    public Gui9(){
    setLayout(null); //Layout absoluto
    setTitle("Ejemplo 11"); //Título del JFrame
    setSize(350, 230); //Dimensiones del JFrame
    setResizable(false); //No redimensionable
    setDefaultCloseOperation(JFrame.EXIT ON CLOSE);
    setVisible(true); //Mostrar JFrame

    grupoBotones = new ButtonGroup();
    radio1=new JRadioButton("Tamaño del Jframe 640*480");
    radio1.setBounds(10,20,200,30);
    radio1.addChangeListener((ChangeListener) this);
    add(radio1);
    grupoBotones.add(radio1);

    radio2=new JRadioButton("Tamaño del Jframe 640*480");
    radio2.setBounds(10,40,200,30);
    radio2.addChangeListener((ChangeListener) this);
    add(radio2);
    grupoBotones.add(radio2);

    radio3=new JRadioButton("Tamaño del Jframe 640*480");
    radio3.setBounds(10,60,200,30);
    radio3.addChangeListener((ChangeListener) this);
    add(radio3);
    grupoBotones.add(radio3);

```

```
37      }
38      public void stateChanged(ChangeEvent e) {
39          if (radio1.isSelected()) {
40          setSize(640,480);
41          setTitle("Ejemplo 11/640-480");
42          this.getContentPane().setBackground(Color.orange);
43          }
44          if (radio2.isSelected()) {
45              setSize(800,600);
46              setTitle("Ejemplo 11/800-600");
47              this.getContentPane().setBackground(Color.PINK);
48          }
49          if (radio3.isSelected()) {
50          setSize(1024,768);
51          setTitle("Ejemplo 11/1024-768");
52          this.getContentPane().setBackground(Color.darkGray);
53          }
54      }
55  }
```

GUI 11

En este GUI creamos un tipo diferente de manera para poder marcar opciones con el CheckBox que este sirve para poder seleccionar una u otra opcion, que son las que ingresamos al crear las variables private, como siempre asignamos el título a la ventana, tamaño y lugar, y dentro del texto para la primera checkbox le pondremos sistemas, a la segunda informática y a la última electrónica, de las cuales lo que haremos seleccionar una y en la parte de abajo sobre la linea nos aparece el texto de la opcion seleccionada.

```java
package GUI;
import java.awt.FlowLayout;
import java.awt.event.ItemEvent;
import java.awt.event.ItemListener;
import javax.swing.JCheckBox;
import javax.swing.JFrame;
import javax.swing.JTextArea;
public class Gui11 extends JFrame implements ItemListener{
private final JCheckBox sistemas;
private final JCheckBox informatica;
private final JCheckBox electronica;
private final JTextArea areaTexto;
    public Gui11() {
        setTitle("JCheckBox 11");
        setSize(300,100);
        setLocationRelativeTo(null);
        setResizable(false);
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        setLayout(new FlowLayout());

    sistemas=new JCheckBox("sistemas");
    add(sistemas);
    sistemas.addItemListener(this);
    informatica=new JCheckBox("informatica");
    add(informatica);
    informatica.addItemListener(this);
    electronica=new JCheckBox("electronica");
    add(electronica);
    electronica.addItemListener(this);
    areaTexto=new JTextArea(0,15);
    add(areaTexto);
    setVisible(true);

    }

    public void itemStateChanged(ItemEvent event){

    String nuevaLinea="\r\n";
    areaTexto.setText("");
        if (sistemas.isSelected()){
        areaTexto.append("Selecciono Ing. en Sistemas"+ nuevaLinea);}
        if (informatica.isSelected()){
        areaTexto.append("Selecciono Ing. Informatica"+ nuevaLinea);}
        if (electronica.isSelected()){
        areaTexto.append("Selecciono Ing. Electronica"+ nuevaLinea);}
    }
}

```

GUI 12

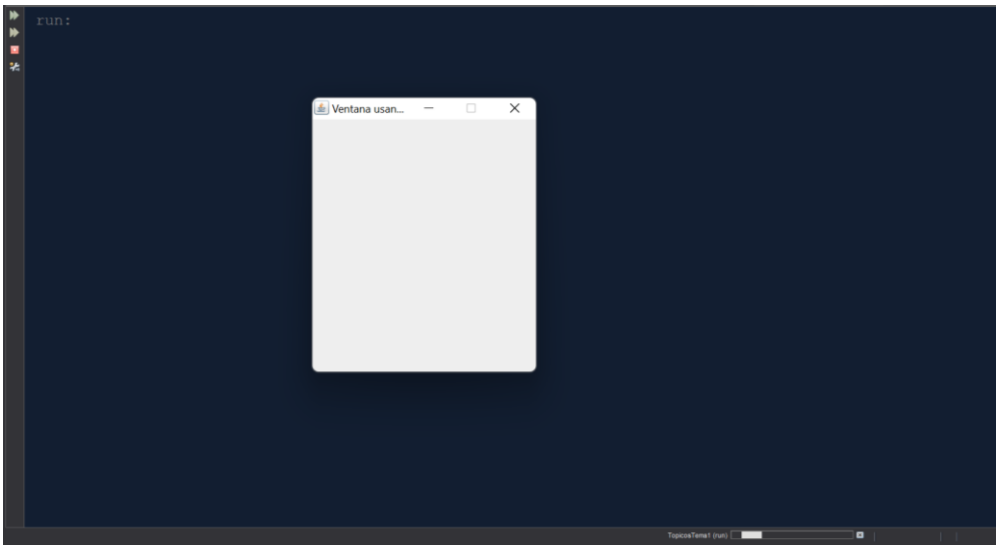
Hacemos todos los import correspondientes, creamos las variables para los contenedores, botones y textos, creamos el contenedor y le damos el título a la venta, su tamaño, su localización y no poder cambiarle el tamaño, creamos las líneas para poder ingresar texto con los JLabel así los usuarios podrán ingresar el texto de cada cosa que nosotros les pedimos que en este caso será su número de control, nombre, semestre y edad, cada uno de estos le asignaremos una posición específica para que no estén unos sobre otros, seguido a esto creamos los botones que serán, añadir, eliminar y borrar todos, estos serán para poder como su nombre lo dice añadir a una persona, eliminar de la lista, o borrar toda la lista, después de esto tendríamos que hacer las acciones de cada botón, pero en nuestro caso al ser uno de prueba solamente vimos lo superficial para la ventana y nos enfocamos en lo gráfico, por ende este GUI no tiene ninguna función en sus botones.

```java
package GUI;
import java.awt.Container;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import javax.swing.JButton;
import javax.swing.JFrame;
import javax.swing.JLabel;
import javax.swing.JList;
import javax.swing.JScrollPane;
import javax.swing.JTextField;
public class Gui12 extends JFrame implements ActionListener{
    private JScrollPane scrollLista;
    private JList listaNombres;
    private Container contenedor;
    private JTextField textoCtrl, textoNombre, textoSem, textoEdad;
    private JLabel numCtrl, Nombre, Semestre, Edad;
    private JButton añadir, eliminar, elimTodos;
    public Gui12(){
        contenedor =getContentPane();
        contenedor.setLayout(null);
        setTitle("Captura de Datos");
        setSize(330,350);
        setLocationRelativeTo(null);
        setResizable(false);
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        setLayout(null);

        numCtrl = new JLabel("Num. Control");
        numCtrl.setBounds(10, 10, 100, 30);
        add(numCtrl);
        textoCtrl = new JTextField();
        textoCtrl.setBounds(90, 15, 180, 20);
        add(textoCtrl);

       Nombre = new JLabel("Nombre");
       Nombre.setBounds(10, 35, 100, 30);
```

```
37              add(Nombre);
38              textoNombre = new JTextField();
39              textoNombre.setBounds(90, 40, 180, 20);
40              add(textoNombre);
41
42              Semestre= new JLabel("Semestre");
43              Semestre.setBounds(10, 60, 100, 30);
44              add(Semestre);
45              textoSem = new JTextField();
46              textoSem.setBounds(90, 65, 180, 20);
47              add(textoSem);
48
49              Edad = new JLabel("Edad");
50              Edad.setBounds(10, 85, 100, 30);
51              add(Edad);
52              textoEdad = new JTextField();
53              textoEdad.setBounds(90, 90, 180, 20);
54              add(textoEdad);
55
56              añadir = new JButton("Añadir");
57              añadir.setBounds(10,120,75,20);
58              add(añadir);
59              setVisible(true);
60              eliminar = new JButton("Eliminar");
61              eliminar.setBounds(90,120,80,20);
62              add(eliminar);
63              setVisible(true);
64              elimTodos = new JButton("Eliminar todos");
65              elimTodos.setBounds(175,120,135,20);
66              add(elimTodos);
                elimTodos.addActionListener(this);
68              setVisible(true);
69          }
70
            public void actionPerformed(ActionEvent evento) {
72              if (evento.getSource()==añadir) {

73              }
74          }
75      }
```

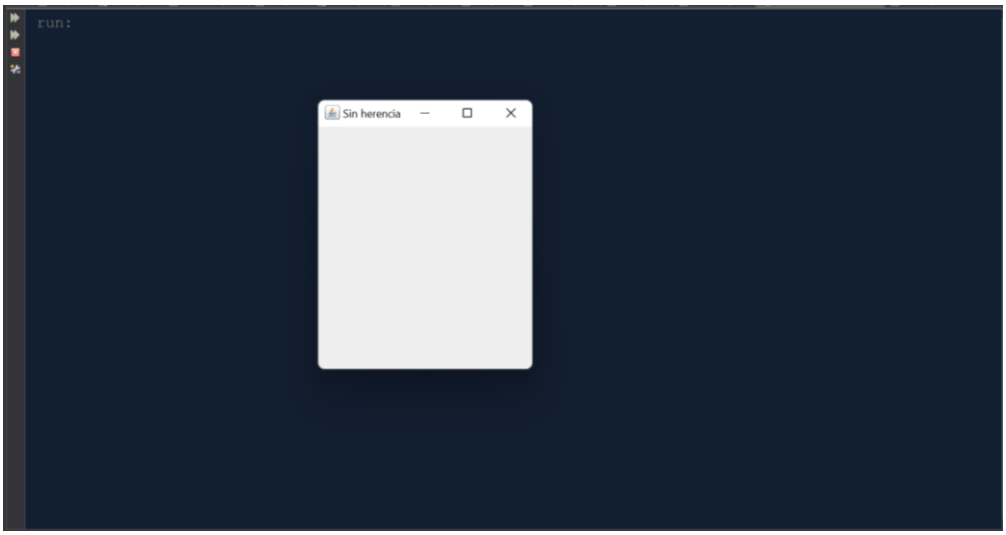
Resultados:

GUI 1



Aquí podemos ver la ventana con el título “Ventana usando herencia”

GUI 2

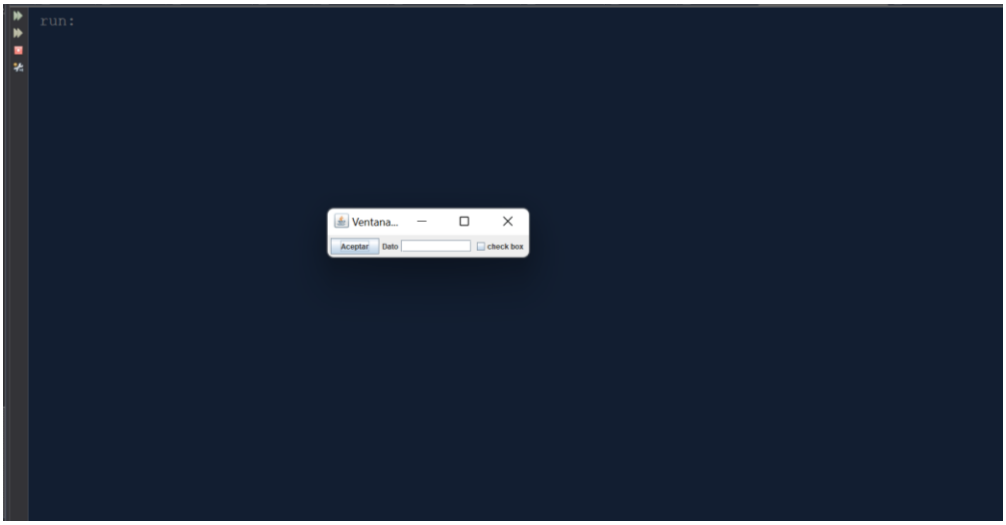


Como podemos ver el resultado es igual, pero es una manera diferente de hacerlo

GUI 3

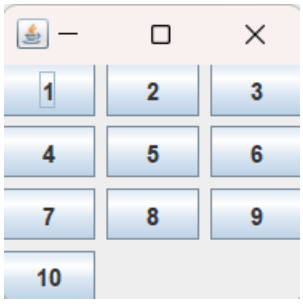
Como podemos ver dentro de la ventana hay un panel y dentro del panel un botón

GUI 5



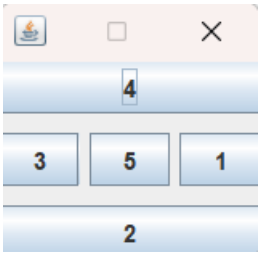
Podemos apreciar los diferentes componentes del JFrame

GUI 6



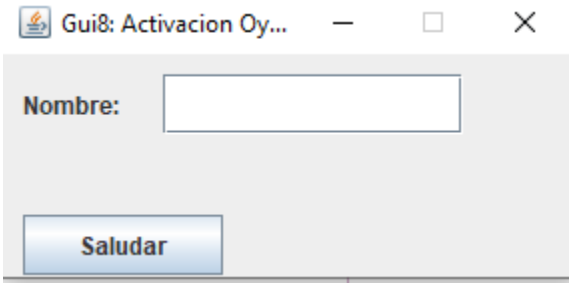
Podemos apreciar los botones están colocados en orden.

GUI 7



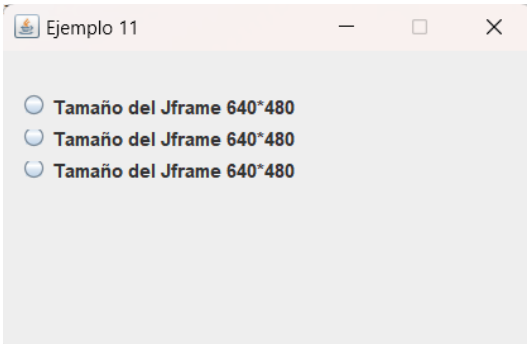
Se puede notar que los botones se encuentran en posición de los puntos cardinales como lo indica el programa.

GUI 8



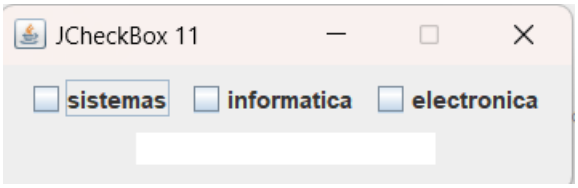
Nos imprime un contenedor para el nombre y un botón del lado izquierdo de nombre “saludar”

GUI 9



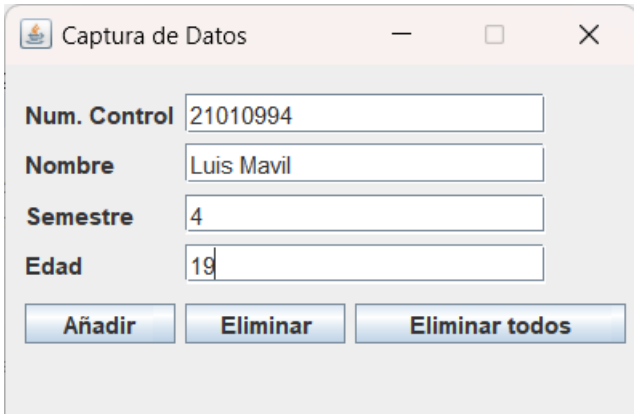
Nos imprime un listado y podemos seleccionar una

GUI 11



Nos da 3 casillas para seleccionar la que queramos que aparezca en el espacio de abajo, cuando se selecciona se palomea

GUI 12



Es una interfaz más elaborada, tenemos 4 espacios para ingresar nuestros datos y 3 botones para hacer lo que necesitemos.

Conclusión

La interfaz de usuario es la parte del programa que permite al usuario interaccionar con él. Las interfaces gráficas de usuario (GUI) ofrecen al usuario ventanas, cuadros de diálogo, barras de herramientas, botones, listas desplegables y muchos otros elementos con los que ya estamos muy acostumbrados a tratar. Las aplicaciones son conducidas por eventos y se desarrollan haciendo uso de las clases que para ello nos ofrece la API de Java. Las interfaces gráficas permiten desarrollar aplicaciones más complejas. Aumentan la interactividad y la productividad

Sin embargo, su desarrollo conlleva una serie de complicaciones añadidas:

- Los componentes de la interfaz deben ser programados
- Necesitamos mecanismos para disponerlos en la pantalla
- Necesitamos mecanismos de control de eventos.

Afortunadamente, hay bibliotecas que nos facilitan mucho las cosas

Bibliografías:

https://www.google.com/url?sa=t&source=web&rct=j&url=https://laurel.datsi.fi.upm.es/_media/docencia/cursos/java/2012/guis_en_java-1pp_2012_.pdf&ved=2ahUKEwjR2JTC07L-AhUgk2oFHZhTDOsQFnoECCgQAQ&usg=AOvVaw0j1jOn312cDFwtyLFyb4hn

https://www.google.com/url?sa=t&source=web&rct=j&url=https://www.fdi.ucm.es/profesor/jpavon/poo/tema6resumido.pdf&ved=2ahUKEwjR2JTC07L-AhUgk2oFHZhTDOsQFnoECCsQAQ&usg=AOvVaw1LHvgiFIV_Pfm4B2uJITrz