

	<p>UNIVERSIDAD NACIONAL DE SAN AGUSTIN FACULTAD DE INGENIERÍA DE PRODUCCIÓN Y SERVICIOS ESCUELA PROFESIONAL DE INGENIERÍA DE SISTEMA</p>	
<p><b>Formato:</b> Guía de Práctica de Laboratorio / Talleres / Centros de Simulación</p>		
<p>Aprobación: 2022/03/01</p>	<p>Código: GUIA-PRLE-001</p>	<p>Página: 1</p>

## INFORME DE LABORATORIO

### (formato estudiante)

INFORMACIÓN BÁSICA					
<b>ASIGNATURA:</b>	FUNDAMENTOS DE PROGRAMACION 2				
<b>TÍTULO DE LA PRÁCTICA:</b>	<i>Herencia y Poliformismo, Clases interfaces</i>				
<b>NÚMERO DE PRÁCTICA:</b>	<i>21</i>	<b>AÑO LECTIVO:</b>	<i>2024-A</i>	<b>NRO. SEMESTRE:</b>	<i>//</i>
<b>FECHA DE PRESENTACIÓN</b>	<i>11/07/2024</i>	<b>HORA DE PRESENTACIÓN</b>			
<b>INTEGRANTE (s):</b> Quiñonez Delgado Aarón Fernando				<b>NOTA:</b>	
<b>DOCENTE(s):</b> <i>ING. RONALD MANCINI TICONA</i>					

SOLUCIÓN Y RESULTADOS
<p><b>I. SOLUCIÓN DE EJERCICIOS/PROBLEMAS</b></p> <ul style="list-style-type: none"> <li><i>Ejercicio 1</i></li> </ul> <pre style="background-color: #2e3436; color: #eeeeec; padding: 10px;">import javax.swing.*; import java.awt.*; import java.awt.event.ActionEvent; import java.awt.event.ActionListener; import java.util.Stack;  public class Ejercicio1 extends JFrame {     private Stack&lt;Double&gt; numbers = new Stack&lt;&gt;();     private Stack&lt;Character&gt; operations = new Stack&lt;&gt;();     private boolean isOperatorClicked = false;</pre>

```
private JTextField operaciones;

private JPanel panel;

public Ejercicio1() {

    setTitle("Calculadora Avanzada");

    setSize(400, 500);

    setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

    setLayout(new BorderLayout());

    createContents();

    setVisible(true);

}

public void createContents() {

    operaciones = new JTextField();

    operaciones.setFont(new Font("Arial", Font.PLAIN, 18));

    operaciones.setHorizontalAlignment(JTextField.RIGHT);

    operaciones.setEditable(false);

    add(operaciones, BorderLayout.NORTH);

    panel = new JPanel();

    panel.setLayout(new GridLayout(5, 4));

    String[] buttonLabels = {

        "7", "8", "9", "/", "C",

        "4", "5", "6", "*", "√",

        "1", "2", "3", "-", "^",

        "0", ".", "=", "+"
    }
}
```

```
};

for (String label : buttonLabels) {

    JButton button = new JButton(label);

    button.setFont(new Font("Arial", Font.PLAIN, 18));

    button.addActionListener(new Listener());

    panel.add(button);

}

add(panel, BorderLayout.CENTER);

}

private class Listener implements ActionListener {

    public void actionPerformed(ActionEvent e) {

        String command = e.getActionCommand();

        if (command.matches("[0-9]") || command.equals(".")) {

            if (isOperatorClicked) {

                operaciones.setText("");

                isOperatorClicked = false;

            }

            operaciones.setText(operaciones.getText() + command);

        } else if (command.equals("C")) {

            operaciones.setText("");

            numbers.clear();

            operations.clear();

        } else if (command.equals("=")) {

            performOperation();

            isOperatorClicked = true;

        }

    }

}
```

```
    } else if (command.equals("√")) {  
        double value = Double.parseDouble(operaciones.getText());  
        operaciones.setText(String.valueOf(Math.sqrt(value)));  
        isOperatorClicked = true;  
    } else if (command.equals("^")) {  
        double base = Double.parseDouble(operaciones.getText());  
        operaciones.setText("");  
        isOperatorClicked = true;  
        operations.push('^');  
        numbers.push(base);  
    } else {  
        performOperation();  
        operations.push(command.charAt(0));  
        isOperatorClicked = true;  
    }  
}  
  
private void performOperation() {  
    if (operations.isEmpty()) {  
        numbers.push(Double.parseDouble(operaciones.getText()));  
        return;  
    }  
  
    double secondOperand =  
Double.parseDouble(operaciones.getText());  
  
    double firstOperand = numbers.pop();
```

```
char operation = operations.pop();

switch (operation) {

    case '+':

        numbers.push(firstOperand + secondOperand);

        break;

    case '-':

        numbers.push(firstOperand - secondOperand);

        break;

    case '*':

        numbers.push(firstOperand * secondOperand);

        break;

    case '/':

        if (secondOperand != 0) {

            numbers.push(firstOperand / secondOperand);

        } else {

            JOptionPane.showMessageDialog(Ejercicio1.this,
            "Error: División por cero", "Error", JOptionPane.ERROR_MESSAGE);

            numbers.push(firstOperand);

        }



        break;

    case '^':

        numbers.push(Math.pow(firstOperand, secondOperand));

        break;

}
```

	<p>UNIVERSIDAD NACIONAL DE SAN AGUSTIN FACULTAD DE INGENIERÍA DE PRODUCCIÓN Y SERVICIOS ESCUELA PROFESIONAL DE INGENIERÍA DE SISTEMA</p>	
<p>Formato: Guía de Práctica de Laboratorio / Talleres / Centros de Simulación</p>		
<p>Aprobación: 2022/03/01</p>	<p>Código: GUIA-PRLE-001</p>	<p>Página: 6</p>

```

        operaciones.setText(numbers.peek().toString());
    }
}

public static void main(String[] args){
    new Ejercicio1();
}
}

```

## ● Ejercicio 2

```

import javax.swing.*.*;
import javax.swing.filechooser.FileNameExtensionFilter;
import javax.swing.text.*;
import java.awt.*.*;
import java.io.*.*;

public class Ejercicio2 extends JFrame {
    private JTextPane archivoTexto;
    private JFileChooser archivos;

    public Ejercicio2() {
        setTitle("Editor de Texto");
        setSize(600, 400);
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        setLayout(new BorderLayout());
    }
}

```

```
archivoTexto = new JTextPane();

JScrollPane scrollPane = new JScrollPane(archivoTexto);

add(scrollPane, BorderLayout.CENTER);


archivos = new JFileChooser();

FileNameExtensionFilter filter = new
FileNameExtensionFilter("Archivos de texto", "txt");

archivos.setFileFilter(filter);


barraMenu();

createToolBar();


setVisible(true);
}


public void barraMenu() {

JMenuBar menuBar = new JMenuBar();


JMenu menu = new JMenu("Archivo");

JMenuItem archivoNuevo = new JMenuItem("Nuevo");

JMenuItem abrirArchivo = new JMenuItem("Abrir");

JMenuItem guardarArchivo = new JMenuItem("Guardar");


archivoNuevo.addActionListener(e -> archivoTexto.setText(""));

abrirArchivo.addActionListener(e -> abrir());

guardarArchivo.addActionListener(e -> guardar());
```

```
menu.add(archivoNuevo);

menu.add(abrirArchivo);

menu.add(guardarArchivo);


menuBar.add(menu);

setJMenuBar(menuBar);

}


public void createToolBar() {

    JToolBar barraHerramientas = new JToolBar();


    JButton negrita = new JButton("N");

    JButton cursiva = new JButton("I");

    JButton subrayado = new JButton("U");


    negrita.setFont(new Font("Arial", Font.BOLD, 12));

    cursiva.setFont(new Font("Arial", Font.ITALIC, 12));

    subrayado.setFont(new Font("Arial", Font.PLAIN, 12));


    negrita.addActionListener(e ->
aplicarEstilo(StyleConstants.Bold));

    cursiva.addActionListener(e ->
aplicarEstilo(StyleConstants.Italic));

    subrayado.addActionListener(e ->
aplicarEstilo(StyleConstants.Underline));
```



```
        barraHerramientas.add(negrita);

        barraHerramientas.add(cursiva);

        barraHerramientas.add(subrayado);

    add(barraHerramientas, BorderLayout.NORTH);
}

public void aplicarEstilo(Object bold) {
    StyledDocument doc = archivoTexto.getStyledDocument();
    int inicio = archivoTexto.getSelectionStart();
    int fin = archivoTexto.getSelectionEnd();
    Element element = doc.getCharacterElement(inicio);
    AttributeSet as = element.getAttributes();

    SimpleAttributeSet sas = new SimpleAttributeSet();
    if (bold == StyleConstants.Bold) {
        StyleConstants.setBold(sas, !StyleConstants.isBold(as));
    } else if (bold == StyleConstants.Italic) {
        StyleConstants.setItalic(sas, !StyleConstants.isItalic(as));
    } else if (bold == StyleConstants.Underline) {
        StyleConstants.setUnderline(sas,
!StyleConstants.isUnderline(as));
    }

    doc.setCharacterAttributes(inicio, fin - inicio, sas, false);
}
```

```
private void abrir() {  
  
    int returnValue = archivos.showOpenDialog(this);  
  
    if (returnValue == JFileChooser.APPROVE_OPTION) {  
  
        File file = archivos.getSelectedFile();  
  
        try (BufferedReader reader = new BufferedReader(new  
FileReader(file))) {  
  
            archivoTexto.read(reader, null);  
  
        } catch (IOException e) {  
  
            JOptionPane.showMessageDialog(this, "No se pudo abrir el  
archivo", "Error", JOptionPane.ERROR_MESSAGE);  
  
        }  
  
    }  
  
}  
  
private void guardar() {  
  
    int returnValue = archivos.showSaveDialog(this);  
  
    if (returnValue == JFileChooser.APPROVE_OPTION) {  
  
        File file = archivos.getSelectedFile();  
  
        try (BufferedWriter writer = new BufferedWriter(new  
FileWriter(file))) {  
  
            archivoTexto.write(writer);  
  
        } catch (IOException e) {  
  
            JOptionPane.showMessageDialog(this, "No se pudo guardar el  
archivo", "Error", JOptionPane.ERROR_MESSAGE);  
  
        }  
  
    }  
  
}
```

	<p>UNIVERSIDAD NACIONAL DE SAN AGUSTIN FACULTAD DE INGENIERÍA DE PRODUCCIÓN Y SERVICIOS ESCUELA PROFESIONAL DE INGENIERÍA DE SISTEMA</p>	
<p>Formato: Guía de Práctica de Laboratorio / Talleres / Centros de Simulación</p>		
<p>Aprobación: 2022/03/01</p>	<p>Código: GUIA-PRLE-001</p>	<p>Página: 11</p>

```

    public static void main(String[] args) {
        new Ejercicio2();
    }
}

```

### ● Ejercicio 3

```

import javax.swing.*.*;
import java.awt.*.*;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.io.File;
import java.util.ArrayList;

public class Ejercicio3 extends JFrame {
    private ArrayList<ImageIcon> imagenes;
    private JLabel etiquetaImagen;
    private int indiceActual;

    public Ejercicio3() {
        setTitle("Galería de Imágenes");
        setSize(1500, 800);
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        setLayout(new BorderLayout());
    }
}

```

```
imagenes = new ArrayList<>();

etiquetaImagen = new JLabel();

etiquetaImagen.setHorizontalAlignment(JLabel.CENTER);

add(etiquetaImagen, BorderLayout.CENTER);


JMenuBar menuBar = new JMenuBar();

JMenuItem itemCargar = new JMenuItem("Cargar Imágenes");

itemCargar.addActionListener(new ActionListener() {

    public void actionPerformed(ActionEvent e) {

        cargarImagen();

    }

});

menuBar.add(itemCargar);

setJMenuBar(menuBar);


JPanel panelBotones = new JPanel();

JButton anterior = new JButton("Anterior");

JButton siguiente = new JButton("Siguiente");

anterior.addActionListener(new ActionListener() {

    public void actionPerformed(ActionEvent e) {

        mostrarImagenAnterior();

    }

});

siguiente.addActionListener(new ActionListener() {
```

```
        public void actionPerformed(ActionEvent e) {

            mostrarImagenSiguiente();

        }

    });

    panelBotones.add(anterior);

    panelBotones.add(siguiente);

    add(panelBotones, BorderLayout.SOUTH);

    setVisible(true);

}

private void cargarImagen() {

    JFileChooser fileChooser = new JFileChooser();

    fileChooser.setMultiSelectionEnabled(true);

    int resultado = fileChooser.showOpenDialog(this);

    if (resultado == JFileChooser.APPROVE_OPTION) {

        File[] archivosSeleccionados = fileChooser.getSelectedFiles();

        for (File archivo : archivosSeleccionados) {

            imagenes.add(new ImageIcon(archivo.getAbsolutePath()));

        }

        if (!imagenes.isEmpty()) {

            indiceActual = 0;

            mostrarImagenActual();

        }

    }

}
```

```
    }

    }

    private void mostrarImagenActual() {

        if (!imagenes.isEmpty() && indiceActual >= 0 && indiceActual <
imagenes.size()) {

            etiquetaImagen.setIcon(imagenes.get(indiceActual));

        }

    }

    private void mostrarImagenAnterior() {

        if (indiceActual > 0) {

            indiceActual--;

            mostrarImagenActual();

        }

    }

    private void mostrarImagenSiguiente() {

        if (indiceActual < imagenes.size() - 1) {

            indiceActual++;

            mostrarImagenActual();

        }

    }

    public static void main(String[] args){

        new Ejercicio3();

    }

}
```

	<p style="text-align: center;">UNIVERSIDAD NACIONAL DE SAN AGUSTIN FACULTAD DE INGENIERÍA DE PRODUCCIÓN Y SERVICIOS ESCUELA PROFESIONAL DE INGENIERÍA DE SISTEMA</p>	
<p style="text-align: center;"><b>Formato:</b> Guía de Práctica de Laboratorio / Talleres / Centros de Simulación</p>		
<p><b>Aprobación:</b> 2022/03/01</p>	<p><b>Código:</b> GUIA-PRLE-001</p>	<p><b>Página:</b> 15</p>

--	--

## CUESTIONARIO

### 1. ¿Cómo se maneja el diseño responsivo en JavaFX?

Se maneja utilizando contenedores de diseño como HBox, VBox, BorderPane, y GridPane, que distribuyen los elementos de manera flexible y se adaptan automáticamente al tamaño de la ventana. Por ejemplo, BorderPane organiza los nodos en cinco regiones, y los contenedores HBox y VBox organizan los nodos en filas y columnas, respectivamente. Estas estructuras permiten que la interfaz gráfica se ajuste dinámicamente a los cambios de tamaño de la ventana, proporcionando un diseño adaptable y responsivo.

### 2. ¿Qué es FXML y cómo se utiliza en JavaFX?

Es un lenguaje basado en XML que se utiliza para definir la estructura de la interfaz de usuario en aplicaciones JavaFX. Permite separar el diseño de la interfaz de usuario de la lógica de la aplicación, facilitando el desarrollo y el mantenimiento del código. FXML se utiliza en conjunto con controladores (clases Java) que manejan la lógica de la interfaz. Para utilizar FXML en JavaFX, se crea un archivo .fxml que define la interfaz, se carga este archivo en la aplicación mediante FXMLLoader, y se asocia un controlador a la interfaz definida.

### 3. ¿Cómo se puede utilizar la clase WebView en JavaFX para mostrar contenido web?

La clase WebView en JavaFX permite mostrar contenido web dentro de una aplicación JavaFX mediante la incorporación de un motor de navegador web basado en WebKit. Para usar WebView, se crea una instancia de esta clase, se añade a la escena de la aplicación y se carga la URL o el contenido HTML deseado. WebView facilita la integración de capacidades de navegación web en aplicaciones JavaFX, proporcionando una manera sencilla de visualizar páginas web.

## REFERENCIAS Y BIBLIOGRAFÍA

--